НА ДОПОМОГУ СТУДЕНТУ УДФСУ

100

О. В. Гладченко, В. О. Ніжегородцев,
В. А. Одинець, О. В. Поденежко,
Т. В. Ратушняк

ЕКОНОМІЧНА ІНФОРМАТИКА: КУРС ЛЕКЦІЙ



ЕКОНОМІЧНА ІНФОРМАТИКА: КУРС ЛЕКЦІЙ

УНІВЕРСИТЕТ ДЕРЖАВНОЇ ФІСКАЛЬНОЇ СЛУЖБИ УКРАЇНИ

СЕРІЯ «НА ДОПОМОГУ СТУДЕНТУ УДФСУ»
Заснована 2016 року

О. В. Гладченко, В. О. Ніжегородцев,
В. А. Одинець, О. В. Поденежко,
Т. В. Ратушняк

ЕКОНОМІЧНА ІНФОРМАТИКА: КУРС ЛЕКЦІЙ

Навчальний посібник

Ірпінь
2021

УДК 004.9:33(075.8)
ББК 32.973я73
Е 45

Рекомендовано до друку Вченою радою
Університету ДФС України
(протокол № 15 від 24 грудня 2020 р.)

Рецензенти:

С.П. Ріппа, д.е.н., професор кафедри інформаційних систем в економіці Київського національного економічного університету ім. В. Гетьмана;

А.Ю. Горбовий, д.т.н., професор кафедри інтелектуальних управляючих та обчислювальних систем ННІ інформаційних систем УДФСУ.

Автори: **Гладченко О.В.,** к.п.н., доцент; **Ніжегородцев В.О.,** к.п.н., доцент; **Одинець В.А.,** к.е.н., доцент; **Поденежко О.В.,** к.ф.-м.н., доцент; **Ратушняк Т.В.,** к.ф-м.н., доцент

Е 45 Економічна інформатика: курс лекцій / О. В. Гладченко, В. О. Ніжегородцев, В. А. Одинець, О. В. Поденежко, Т. В. Ратушняк. – Ірпінь : Університет ДФС України, 2021. – 430 с. – (Серія «На допомогу студенту УДФСУ» ; т. 83).
ISBN 978-966-337-611-0

Навчальний посібник призначений для підготовки першого бакалаврського рівня вищої освіти галузі знань 05 «Соціальні та поведінкові науки» спеціальностей 051 «Економіка» та 07 «Управління та адміністрування» для спеціальностей: 071 «Облік і оподаткування», 072 «Фінанси, банківська справа та страхування», 076 «Підприємництво, торгівля та біржова діяльність» денної та заочної форми навчання з дисципліни «Економічна інформатика».

Відповідно до робочої програми викладено основи економічної інформатики та інформаційних процесів, технологій роботи із структурованими документами, сховищами та базами даних, можливості використання комп'ютерних технологій у фаховій діяльності, технології розробки програмних додатків та перспективи розвитку інформатики та інформаційних систем і технологій. Містить курс лекцій, контрольні запитання, тестові завдання для самоперевірки та глосарій.

Рекомендований студентам економічних спеціальностей та викладачам економічних закладів вищої освіти, спеціалістам-практикам, які займаються проблемами використання ІТ-технологій в економіці.

УДК 004.9:33(075.8)
ББК 32.973я73

© Гладченко О.В., Ніжегородцев В.О.,
Одинець В.А., Поденежко О.В.,
Ратушняк Т.В., 2021

ISBN 978-966-337-611-0

© Університет ДФС України, 2021

ЗМІСТ

ПЕРЕДМОВА 7

ЛЕКЦІЯ 1. ПРЕДМЕТ, МЕТОДИ І ЗАВДАННЯ ДИСЦИПЛІНИ. ТЕОРЕТИЧНІ ОСНОВИ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ 9
1.1. Предмет, методи і завдання дисципліни 9
1.2. Інформатизація суспільства. Цифрова економіка 11
1.3. Інформація та дані. Подання інформації в комп'ютері. Одиниці інформації 18
1.4. Інформаційні процеси в економіці. Економічна інформація та її особливості 24
1.5. Класифікація та кодування інформації. Єдина система класифікації техніко-економічної інформації 30
1.6. Формалізація, алгоритмізація та автоматизована обробка економічної інформації 35
Запитання для самоконтролю 42

ЛЕКЦІЯ 2. СИСТЕМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ В ЕКОНОМІЧНІЙ ДІЯЛЬНОСТІ 44
2.1. Апаратне забезпечення інформаційних процесів в економіці 44
2.2. Програмне забезпечення інформаційних процесів в економіці 51
2.3. Класифікація та структура операційних систем 58
2.4. Організація та робота з об'єктами файлової системи ОС MS Windows 66
2.5. Інформаційна безпека, основи захисту інформації 69
Запитання для самоконтролю 80

ЛЕКЦІЯ 3. ПРОГРАМНІ ЗАСОБИ РОБОТИ З БАЗАМИ ТА СХОВИЩАМИ ДАНИХ УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСІВ 82
3.1. Поняття бази даних та бази знань 82
3.2. Різновиди моделей даних. Типи зв'язків 87

3.3. Проєктування реляційної бази даних
економічної діяльності: метод нормальних форм;
метод суть-зв'язок (ER-діаграм);
засоби автоматизації проєктування 91
3.4. Програмні засоби роботи з базами даних.
СУБД в економіці. 102
3.5. Структура сховищ даних та програмні засоби
роботи зі сховищами даних 108
Запитання для самоконтролю 119

**ЛЕКЦІЯ 4. РОБОТА З БАЗОЮ ДАНИХ
У СЕРЕДОВИЩІ MS ACCESS 2013** 121
4.1. Технології конструювання бази даних та обробки
економічної інформації в СУБД Microsoft Access 2013 121
4.2. Модель бази даних *Особові рахунки*
для банківської установи 154
Запитання для самоконтролю 170

**ЛЕКЦІЯ 5. АНАЛІЗ ЧИСЛОВИХ ДАНИХ
У СЕРЕДОВИЩІ MS EXCEL 2013** 173
5.1. Технологія роботи із структурованими
документами у табличному процесорі MS Excel 2013 173
5.2. Побудова реляційної бази даних бізнес-процесів
в MS Excel 2013 181
5.3. Засоби роботи з базою даних у MS Excel 2013 183
5.4. Моделювання економічних процесів
засобами MS Excel 2013 201
Запитання для самоконтролю 216

**ЛЕКЦІЯ 6. РОБОТА З ТЕКСТАМИ
У СЕРЕДОВИЩІ MS WORD 2013** 217
6.1. Електронний документ. Електронний документообіг 217
6.2. Технологія роботи із структурованими
документами у текстовому процесорі MS Word 2013 229
Запитання для самоконтролю 244

ЛЕКЦІЯ 7. ОСНОВИ ОФІСНОГО ПРОГРАМУВАННЯ.....245
7.1. Призначення та основні поняття системи об'єктно-орієнтованого програмування VBA: редактор, процедури та функції, основні конструкції та оператори мови.....245
7.2. Створення макросів, функцій користувача, форм з елементами управління у додатках Microsoft Office 2013.....273
Запитання для самоконтролю.....289

ЛЕКЦІЯ 8. ПРОГРАМУВАННЯ ТИПОВИХ АЛГОРИТМІВ У СЕРЕДОВИЩІ MS OFFICE 2013.....291
8.1. Приклад розроблення форми засобами VBA *Анкета студента*.....291
8.2. Автоматизований розрахунок обміну валют.....294
Запитання для самоконтролю.....299

ЛЕКЦІЯ 9. ОСНОВИ ВЕБДИЗАЙНУ.....301
9.1. Етапи проєктування та розробки вебсайту.....301
9.2. Структура вебсторінки та її об'єкти.....305
9.3. Використання фреймів.....310
9.4. Створення списків, таблиць.....311
9.5. Оформлення тексту, створення заголовків. Графічні об'єкти.....317
9.6. Гіперпосилання на вебсторінці.....330
9.7. Динамічні ефекти та засоби їх створення.....334
9.8. Поняття про інтерактивні вебсторінки та засоби розробки сценаріїв.....335
Запитання для самоконтролю.....353

ЛЕКЦІЯ 10. ЗАСТОСУВАННЯ ІНТЕРНЕТ-ТЕХНОЛОГІЙ В ЕКОНОМІЦІ. ЕКСПЕРТНІ І НАВЧАЛЬНІ СИСТЕМИ 354
10.1. Телеконференції (групи новин) в економічній діяльності 354
10.2. Мережеві технології в економіці 363
10.3. Хмарні технології – як основа бізнес-середовища 367
10.4. Електронна комерція та біржові операції через інтернет 373
10.5. Віртуальна корпорація та віртуальний офіс 374
10.6. Експертні системи, їх будова та застосування в економіці 380
Запитання для самоконтролю 385

ТЕСТОВІ ЗАВДАННЯ ДЛЯ САМОПЕРЕВІРКИ 386

ГЛОСАРІЙ ОСНОВНИХ ТЕРМІНІВ 398

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ 424

ПЕРЕДМОВА

На сучасному етапі розвитку людства значна роль відведена інформатизації економічних процесів та впровадженню в практику управління новітніх інформаційних технологій. Інформація перетворилася на найважливіший ресурс суспільства. Величезні обсяги інформації і необхідність її оброблення з метою прийняття адекватних управлінських рішень потребує від фахівців використання автоматизованих інформаційних технологій. Інформатизація суспільства є одним з головних показників соціально-економічного розвитку держави.

Уряд України працює над створенням Національної економічної стратегії до 2030 року, в якій визначено основні вектори економічного розвитку на найближчі роки. Серед них – розвиток цифрової економіки, для реалізації якого потрібно активно працювати над удосконаленням цифрової інфраструктури в цілому та забезпеченням 95 % населення доступом до високошвидкісного інтернету; необхідно створювати комфортні умови для ведення електронного бізнесу в Україні; цифровізувати роботу державного апарату і державні послуги зокрема; підвищити цифрову грамотність населення. Ще одним важливим вектором Національної економічної стратегії є планування заходів щодо збільшення інвестиційної привабливості України; серед них – створення в Україні привабливого правового та податкового режимів для ІТ-компаній. Міністерство цифрової трансформації серед пріоритетів роботи визначило: забезепечення регуляторного зближення з правом ЄС у цифровій сфері, ліквідацію бар'єрів для цифрової торгівлі, прискорення цифровізації в Україні для розширення доступу до онлайн ринків та цифрових технологій ЄС. Такі кроки мають забепечити зростання ВВП в Україні до 5 %, покращити ефективність роботи бізнесу та зростання обсягу двосторонньої торгівлі товарами і послугами між Україною та ЄС.

Світові тенденції розвитку економіки та суспільства вимагають від закладів вищої освіти якісної сучасної підготовки фахівців з урахуванням цифрової трансформації економічних процесів. Здобувачі вищої освіти повинні отримати конкурентоспроможні знання, щоб відповідати вимогам провідних українських та міжнародних компаній.

Навчальна дисципліна «Економічна інформатика» є обов'язковою у підготовці здобувачів першого бакалаврського рівня вищої освіти галузей знань 05 «Соціальні та поведінкові науки», 07 «Управління та адміністрування» денної та заочної форми та необхідною для майбутніх фахівців Міністерства фінансів України.

Предмет дисципліни – засоби автоматизації інформаційних процесів обробки та використання економічної інформації. Метою дисципліни є формування знань про принципи побудови та функціонування обчислювальних машин, організацію обчислювальних процесів на персональних комп'ютерах та їх алгоритмізацію, програмне забезпечення персональних комп'ютерів і комп'ютерних мереж, а також ефективне використання сучасних інформаційно-комунікаційних технологій у професійній діяльності. Завдання дисципліни – вивчення теоретичних основ економічної інформатики і набуття навичок використання прикладних систем оброблення економічних даних та систем програмування для персональних комп'ютерів і локальних комп'ютерних мереж під час дослідження соціально-економічних систем та розв'язування завдань фахового спрямування.

Електронний документообіг, електронна торгівля, інтернет-конференції, оптимізація бізнес-процесів, конструювання і підтримка сайту підприємства тощо вимагають відповідних знань і навичок під час роботи з сучасними комп'ютерними засобами, які студенти отримують завдяки навчальній дисципліні «Економічна інформатика».

Навчальна дисципліна «Економічна інформатика» є необхідним та ефективним засобом підготовки спеціалістів галузей знань 05 «Соціальні та поведінкові науки» спеціальності 051 «Економіка» та 07 «Управління та адміністрування» для спеціальностей: 071 «Облік і оподаткування», 072 «Фінанси, банківська справа та страхування», 076 «Підприємництво, торгівля та біржова діяльність».

ЛЕКЦІЯ 1
ПРЕДМЕТ, МЕТОДИ І ЗАВДАННЯ ДИСЦИПЛІНИ. ТЕОРЕТИЧНІ ОСНОВИ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ

1.1. Предмет, методи і завдання дисципліни

Слово «інформатика» (informatique) походить від сполучення двох французьких слів: information (інформація) і automatique (автоматика), введене у Франції для визначення галузі діяльності, що займається автоматизованою обробкою інформації.

Існує багато визначень економічної інформатики. Економічна інформатика – це наука про інформацію, способи її збору, зберігання, опрацювання та передавання з використанням комп'ютерної техніки. Економічна інформатика – це прикладна дисципліна, що вивчає структуру і загальні властивості економічної інформації та економічних процесів. Економічна інформатика – це технічна наука, яка систематизує прийоми створення, зберігання, відтворення, обробки та передачі економічних даних засобами обчислювальної техніки, а також принципи функціонування цих засобів та методи управління ними.

З цього визначення можна зробити висновок, що економічна інформатика дуже близька до технології, тому її предмет нерідко називають *інформаційними технологіями*.

Економічна інформатика складається з трьох взаємозв'язаних складових: економічна інформатика як фундаментальна наука, як прикладна дисципліна і як галузь виробництва.

Основними об'єктами економічної інформатики виступають:

- економічна інформація;
- комп'ютери;
- інформаційні системи.

Загальні теоретичні основи економічної інформатики:

- інформація;
- системи числення;
- кодування;
- алгоритми.

Структура сучасної економічної інформатики:

1. Теоретична інформатика.
2. Обчислювальна техніка.
3. Програмування.
4. Інформаційні системи.
5. Штучний інтелект.

Предмет економічної інформатики становлять такі поняття:

- апаратне забезпечення джерел комп'ютерної техніки;
- програмне забезпечення джерел комп'ютерної техніки;
- засоби взаємодії апаратного та програмного забезпечення;
- засоби взаємодії людини з апаратними та програмними засобами.

Методи та засоби взаємодії людини з апаратними та програмними засобами в інформатиці мають назву інтерфейси користувача. Існують апаратні інтерфейси, програмні інтерфейси та апаратно-програмні інтерфейси.

Головним завданням економічної інформатики є систематизація прийомів та методів роботи з апаратними та програмними засобами комп'ютерної техніки. Мета систематизації полягає у виділенні, впровадженні та розвитку передових, найбільш ефективних технологій, в автоматизації етапів роботи з даними, а також у методичному забезпеченні нових досліджень.

Економічна інформатика – це також практична наука. Її досягнення повинні отримувати підтвердження практикою та прийматись у тих випадках, коли вони відповідають критерію підвищеної ефективності управління. У складі основного завдання економічної інформатики сьогодні виділені такі напрями для практичних додатків:

- архітектура обчислювальних систем (прийоми і методи побудови систем, призначених для автоматизації обробки даних);
- інтерфейси обчислювальних систем (прийоми і методи управління апаратним та програмним забезпеченням);
- програмування (прийоми, методи і засоби розробки комп'ютерних програм);
- перетворення даних (прийоми та методи перетворення структур даних);

• захист інформації (розробка методів та засобів захисту даних);

• автоматизація (функціонування програмно-апаратних засобів без участі людини);

• стандартизація (забезпечення суміщення між апаратними і програмними засобами, а також між форматами наведених даних).

Під у*правлінням* в економічній інформатиці розуміють насамперед інформаційний процес, що передбачає виконання функцій зі збору, передачі, зберігання, обробки і аналізу інформації, необхідних для вироблення відповідних управлінських рішень.

Всі процеси управління, що протікають в управлінській системі, відбуваються у вигляді інформаційних процесів, тобто процесів з опрацювання (обробки) інформації.

У сучасних умовах прийняття ефективних рішень у сфері управління вимагає опрацювання великих обсягів інформації. Якісна неоднорідність такої інформації і складність її опрацювання вимагають розділення функцій з отримання, передачі, зберігання і обробки інформації між людиною і технічними засобами, серед яких центральне місце відводиться комп'ютерам. Однак пріоритети на прийняття управлінського рішення і відповідальність за його наслідки залишаються за фахівцем.

1.2. Інформатизація суспільства. Цифрова економіка

Вступивши у XXI століття, людство відкрило новий етап розвитку – інформаційний з домінантною роллю інформаційних ресурсів. Зміни, що відбуваються у суспільстві у зв'язку із зміною визначального ресурсу, вже викликали відповідні радикальні зміни у політичних, економічних і правових системах, як приклад – зміна характеру економічних комерційних відносин (розвиток електронної комерції), зміни у економічних виробничих відносинах – впровадження гнучких автоматизованих систем проєктування і виробництва продукції. Комп'ютери перетворилися в невід'ємну частину сучасної культури і є рушійною силою економічного зростання в усьому світі. Сучасний період становлення

ринкової економіки характеризується переходом до нової економічної моделі, головне місце в якій займають інформаційні технології, засновані на комунікаційних засобах та засобах опрацювання, збереження і захисту інформації.

Інформація стала одним із найцінніших ресурсів суспільства, таких як природні копалини, електрична енергія тощо, тому процес обробки інформації закономірно сприймати як технологію. У сучасному інформаційному суспільстві інформація – це найважливіший стратегічний ресурс.

Інформаційні ресурси (ІР) – це інформація, яка має певну цінність та може бути використана людиною в продуктивній діяльності.

Інформаційні ресурси та документопотоки – основні складові інформаційних технологій та систем. ІР – це, зокрема, окремі документи або масиви документів у ІС (бібліотеках, архівах, фондах, банках даних). Вони використовуються в усіх сферах людської діяльності. Це поняття широко застосовується поряд із такими поняттями, як матеріальні, енергетичні і трудові ресурси. Можна розглядати ІР окремої особистості, підрозділу, підприємства, країни, корпорації.

Сучасне суспільство перетворюється з індустріального на інформаційне, яке характеризується не лише технологічною складовою, але й відповідним рівнем розвитку культури, знань, рівнем освіти, випереджаючими темпами розвитку науки та наукомістких технологій. В інформаційному суспільстві більша частина працездатного населення зайнята у сфері опрацювання інформації.

Про перехід до інформаційного суспільства свідчать такі реалії сьогодення:

- прозорість географічних і геополітичних кордонів в умовах міжнародного інформаційного обміну;
- глобалізація інформаційних потоків і формування способів та засобів інформаційного обміну, що слабко піддаються контролю державами;
- розширення можливостей безпосередніх контактів між різними соціальними і політичними групами;

• поява нових форм та видів діяльності в інформаційних мережах (робота, торгівля, відпочинок, творчість, розваги, освіта, медичні послуги тощо);

• розвиток нового мультимедійного середовища, яке надає можливість конструювати віртуальні світи і діяти у них.

Інформаційні впливи на держави, суспільства, людей сьогодні часом бувають ефективнішими за політичні, економічні чи військові.

Нові комп'ютерні технології вимагають організаційних змін у менеджменті, кадрах, системах документообігу. Значна частина комп'ютеризації об'єктів і процесів пов'язана зі систематизацією документопотоків та їх відображенням у комп'ютері в певному формалізованому вигляді. Ці дії регламентують механізми накопичення, зберігання та обробки даних. Результатом комп'ютеризації є розроблення інтелектуалізованих автоматизованих систем управління, що охоплюють по вертикалі та горизонталі всі рівні і ланки виробництва.

В інформаційному суспільстві акцент зміщується на виробництво та використання інтелекту, знань, що призводить до збільшення питомої ваги розумової праці. Матеріальною та технологічною базою інформаційного суспільства стають різноманітні системи на базі комп'ютерної техніки, телекомунікаційних мереж, ІТ.

Інформатизація економіки – це процес її інтенсифікації на базі сучасних ІТ та створення індустрії інформаційних послуг. Інформаційна послуга – діяльність, спрямована на задоволення інформаційних потреб користувача.

Інформатизація суспільства охоплює всі процеси в економіці держави. Наслідком цього має стати створення єдиного ринку фінансів, товарів, робочої сили, послуг, розвиток фінансово-економічних структур та механізмів в інформаційному суспільстві, підтримка електронного бізнесу та електронної комерції, систем електронних платежів.

Сьогодні впевнено можна говорити про те, що ІТ визначають життя всього людства і кожної людини зокрема. Саме ІТ впливають на розвиток інших виробництв, за їх допомогою створюють

нові ліки, знаходять корисні копалини, пишуть нові книги та газети, забезпечується управління підприємствами і державами, навчаються діти і створюються нові витвори мистецтва. Процесори керують пральними машинами і мікрохвильовими пічками. Дозвілля, розваги, відпочинок – усе це реалізується за допомогою інформаційних технологій. Тобто людина не тільки усвідомила, що її життя – це постійна робота з інформацією, але й навчилася її швидко та ефективно опрацьовувати. Як наслідок, якість життя, рівень комфорту стали значно вищими. Актуальним стає завдання кожного щодо досягнення високого рівня загальної культури, складовою якої є інформаційна культура, головними компонентами якої є:

1. Розуміння сутності інформації та інформаційних процесів, їх ролі в пізнанні навколишньої дійсності та творчої діяльності людини, в управлінні технічними і соціальними процесами, в забезпеченні зв'язку живого із зовнішнім оточенням.

2. Розуміння проблем подання, оцінки і вимірювання інформації, її сприймання і розуміння, сутності формалізації суджень, зв'язку між змістом та формою, ролі інформаційного моделювання в сучасній інформаційній технології.

3. Розуміння сутності неформалізованих, творчих компонент мислення.

4. Уміння добирати і формулювати мету, здійснювати постановку завдань, висувати гіпотези, будувати інформаційні моделі досліджуваних процесів і явищ, аналізувати їх за допомогою засобів нових інформаційних технологій та інтерпретувати отримані результати, систематизувати факти, осмислювати і формулювати висновки, узагальнювати спостереження, передбачати наслідки рішень, що приймаються, і дій, що виконуються, та вміти їх оцінювати.

5. Вміння визначати послідовність операцій і дій у діяльності, розробляти програму спостереження, досліду, експерименту.

6. Володіння інструментальним застосуванням комп'ютерів, системами опрацювання текстової, числової і графічної інформації, баз даних і знань, предметно-орієнтованими прикладними системами, системами телекомунікацій.

7. Розуміння сутності штучного інтелекту.

8. Уміння адекватно формалізувати наявні у людини знання і адекватно інтерпретувати формалізовані описи, дотримуватися належної рівноваги між формалізованою і неформалізованою складовими.

9. Володіння основами алгоритмізації.

10. Вміння передбачати наслідки власних дій, підпорядковувати свої інтереси тим нормам поведінки, яких необхідно дотримуватися в інтересах суспільства, свідомо приймати всі ті обмеження і заборони, які будуть вироблятися «колективним інтелектом», суспільством.

Цифрова економіка – це інноваційна динамічна економіка, що базується на активному впровадженні інновацій та інформаційно-комунікаційних технологій в усі види економічної діяльності та сфери життєдіяльності суспільства, що дозволяє підвищити ефективність та конкурентоспроможність окремих компаній, економіки та рівень життя населення.

Міжнародна організація OECD (Organisation for Economic Cooperation and Development), взявши до уваги дослідження Томаса Мезенбурга, виділяє три основні компоненти цифрової економіки:

- підтримуюча інфраструктура (апаратне та програмне забезпечення, телекомунікації, мережі та ін.);
- електронний бізнес або e-business (ведення господарської діяльності та будь-яких інших бізнес-процесів через комп'ютерні мережі);
- електронна комерція або e-commerce (дистрибуція товарів через інтернет).

Характерною особливістю цифрової економіки є її зв'язок з економікою на вимогу (on-demand economy), яка передбачає не тільки продаж товарів і послуг, а отримання доступу до них саме в той момент, коли це потрібно. Отримання замовлень відбувається онлайн, а їх виконання – офлайн.

Тому пріоритетним завданням держави є забезпечення для кожної людини доступу до інтернету. Зауважимо, що це є одним із цільових показників «Цілей сталого розвитку на період до 2030 року», затверджених на Саміті ООН зі сталого розвитку 25 вересня 2015 року.

Європейський Союз спрямовує зусилля на реалізацію таких завдань:

- створення до 2025 р. «гігабіт-суспільства» (гігабітний зв'язок для всіх основних соціально-економічних об'єктів, таких як школи, транспортні вузли, постачальники державних послуг та підприємства, які інтенсивно використовують цифрові технології);
- розгортання безперебійного 5G покриття для всіх міст та головних наземних транспортних шляхів;
- розширення можливостей безкоштовного доступу громадян до WI-FI;
- подальший розвиток конкуренції і захист прав суб'єктів цифрового ринку на основі нового Кодексу електронних комунікацій.

Велику увагу Європейський Союз приділяє впровадженню у промисловість таких цифрових технологій нового покоління, як:

- інтернет речей;
- хмарні обчислення;
- великі дані і аналітика даних;
- робототехніка та 3D-друк.

Ці технології відкривають нові горизонти для вироблення інноваційних продуктів та послуг.

Наразі запроваджено Індекс цифрової економіки та суспільства (DESI) – це складений індекс, який узагальнює відповідні цифрові показники Європи та відстежує розвиток країн-членів ЄС у сфері цифрової конкурентоспроможності. Індекс DESI дозволяє проводити аналіз за чотирма напрямами:

1. Загальна оцінка ефективності: для визначення загальної характеристики результативності окремих держав-членів ЄС шляхом отримання їх загального показника DESI та розмірів основних показників.

2. Детальний аналіз: визначити напрями, де ефективність держав-членів ЄС може бути покращено шляхом аналізу групових показників та індивідуальних показників.

3. Перспективний аналіз: оцінити, чи є прогрес у часі.

4. Порівняльний аналіз: об'єднати держави-члени відповідно до їх індексу, порівнюючи країни на схожих стадіях цифрового розвитку, щоб визначити політику їх вдосконалення у відповідних сферах.

Індекс DESI не визначається для України з причини відсутності відповідної інформації і звітності. Науковцями і практиками неодноразово порушувалося питання про вдосконалення інформаційного забезпечення, статистичної звітності з врахуванням змін, що відбуваються в умовах розвитку інформаційного суспільства. Відсутність на сьогодні такої звітності не тільки не дозволяє оцінити рівень цифровізації та інноваційного розвитку, але й ускладнює контроль ризиків за цими операціями та об'єктивну оцінку можливості стабільного розвитку та конкурентоспроможності.

У рамках співпраці Держагентства з питань електронного урядування України та Генерального директорату з інформатизації Єврокомісії Україна вперше увійшла в огляд Європейської комісії щодо цифрового врядування «Digital Government Factsheets 2019», ставши четвертою країною, що не входить до ЄС або Європейської асоціації вільної торгівлі, яку охоплює огляд.

Оскільки базовим показником для вимірювання рівня цифрової економіки є швидкість і доступність інтернету, розглянемо рівень України за цим показником. За даними американського вебсайту speedtest.net/global-index, який використовують для аналізу пропускної швидкості інтернету, станом на 2020 рік Україна має такий рейтинг (рис.1.1).



Рис. 1.1. Показники рівня розвитку цифрової економіки України

2018 рік став переломним в Україні для руху в напрямі цифрового суспільства, насамперед завдяки схваленню Кабінетом Міністрів України «Концепції розвитку цифрової економіки та суспільства України на 2018–2020 роки», запуску та поширенню 4G. До того ж вартість мобільного інтернету в Україні є однією з найдоступніших у світі. Прийнятий Закон України «Про електронні довірчі послуги» посилює захист і надійність електронних підписів, створює передумови для взаємного визнання українських та європейських сертифікатів відкритих ключів та електронних підписів. У свою чергу, Закон України «Про основні засади забезпечення кібербезпеки України» (2018 р.) враховує частину положень Директиви «NIS» (про заходи для високого спільного рівня безпеки мережевих та інформаційних систем) ЄС і має забезпечити захист інтересів суспільства у кіберпросторі.

Запроваджена низка електронних адміністративних послуг для фізичних і юридичних осіб та Єдиний державний портал адміністративних послуг, ведеться робота над Системою електронної взаємодії державних електронних інформаційних ресурсів «Трембіта». Для розширення надання електронних адміністративних послуг на початку 2019 року були прийняті Концепція розвитку системи електронних послуг та план заходів для її реалізації на 2019 – 2020 рр.

1.3. Інформація та дані. Подання інформації в комп'ютері. Одиниці інформації

Поняття: дані, інформація і знання належать до базових понять, які використовуються в економічній інформатиці. Ці поняття часто використовуються як синоніми, проте між цими поняттями існують принципові відмінності.

Інформація – це фундаментальне наукове поняття. Наукове визначення інформації дається достатньо просто, якщо припустити, що інформація – це динамічний об'єкт, який не існує у природі сам по собі, а утворений у процесі взаємодії даних та методів. Він існує рівно стільки, скільки триває ця взаємодія, а увесь інший час перебуває у вигляді даних.

Інформація – це продукт взаємодії даних та методів, розглянутий у контексті цієї взаємодії.

Термін «*дані*» походить від слова data – факт, а інформація (informatio) означає роз'яснення, виклад.

Дані – це інформація, подана у формі, сприятливій для формальної обробки персональним комп'ютером або користувачем.

Інформація – це результат перетворення, опрацювання і аналізу даних. Відмінність інформації від даних полягає в тому, що дані – це фіксовані відомості про події і явища, що зберігаються на певних носіях, а інформація з'являється в результаті опрацювання даних під час розв'язування конкретних задач.

Наприклад, в базах даних зберігаються різні дані, а за певним запитом система управління базою даних видає необхідну інформацію.

Існують і інші визначення інформації, наприклад, інформація – це відомості про об'єкти і явища навколишнього середовища, їх параметри, властивості і стан, які зменшують ступінь невизначеності про них, неповноту знань.

Знання – це перевірений практикою результат пізнання дійсності, її відображення в свідомості людини.

Знання – це вид інформації, що містить як дані, так і їх опис і зберігається в базі знань та відображає знання фахівця з конкретної предметної галузі.

На підставі отриманої інформації і наявних знань здійснюється прийняття рішень. Залежно від сфери використання інформація може бути різною: науковою, технічною, управляючою, економічною тощо.

Усі дії, які можна провадити з інформацією, називаються *інформаційними процесами*.

Під час інформаційного процесу дані перетворюються з одного виду в інший за допомогою методів обробки. Обробка даних включає в себе багато різних операцій.

У структурі можливих операцій з даними можна виділити такі основні операції:

- збір даних – накопичення з метою забезпечення достатньої інформації для прийняття рішень;

• формалізація даних – приведення даних, що надходять із різних джерел, до однакової форми, щоб зробити їх зіставними між собою;

• сортування даних – упорядкування даних за наданою ознакою з метою зручності використання (підвищує доступність інформації);

• групування даних – об'єднання даних за наданою ознакою з метою збільшення зручності використання;

• архівація даних – організація збереження даних у зручній та легкодоступній формі, слугує для зниження економічних витрат на зберігання та збільшує загальну надійність інформаційного процесу в цілому;

• захист даних – комплекс заходів, спрямованих на запобігання втрачання даних, відтворення та модифікацію даних;

• транспортування даних – прийняття та передача (доставляння і постачання) даних між віддаленими учасниками інформаційного процесу. При цьому джерело даних в економічній інформатиці прийнято називати сервером, а споживача – клієнтом.

Вони містять такі складові:

• отримання;

• зберігання;

• обробка;

• передача інформації.

При цьому інформація обов'язково повинна мати такі властивості:

• достовірність;

• зрозумілість;

• актуальність;

• корисність;

• повнота;

• однозначність;

інакше втрачається сенс інформаційного процесу.

Подання інформації в комп'ютері. Для опрацювання інформації на ПК її кодують і подають у цифровій формі, тобто всі повідомлення подають у вигляді цифрових кодів. Кодом називається

правило для перетворення одного набору знаків в інший набір знаків. Кодування повідомлень – це подання їх за допомогою деякої послідовності знаків, тобто процес перетворення одного набору знаків в інший набір знаків. Одне і те ж повідомлення можна кодувати по-різному.

Оскільки пристрої комп'ютера побудовані на елементах, що мають два стійких стани, то будь-яка інформація в пам'яті комп'ютера подається у вигляді двійкового коду. Двійковий код означає, що будь-яка інформація в пам'яті комп'ютера подається за допомогою лише двох символів: нуля і одиниці. В електронних елементах комп'ютера відбувається передавання і перетворення електричних сигналів. Двійкові символи розпізнають так: є сигнал – одиниця, немає сигналу – нуль. На магнітних носіях одиниці відповідає намагнічена ділянка поверхні, нулеві – ненамагнічена. Це дозволяє технічно реалізувати збереження і опрацювання інформації.

Одиниці інформації

Біт (від англ. Bit – двійкова одиниця) – найменша довжина двійкового коду (один двійковий розряд). Байт – це послідовність з 8 бітів. Загальна кількість різних комбінацій двійкових значень у байті дорівнює 2^8=256. Для кодування різних символів і для зберігання їх у запам'ятовуючих пристроях комп'ютера найчастіше використовують американський стандартний код для обміну інформацією – ASCII (Amecan Standart Code for Information Interchange), який являє собою стандартну таблицю кодування знаків.

Широко використовують також ще більші одиниці інформації:

1 Кілобайт (Кбайт) = 1024 байт = 2^{10} байт;

1 Мегабайт (Мбайт) = 1024 Кбайт = 2^{20} байт;

1 Гігабайт (Гбайт) = 1024 Мбайт = 2^{30} байт.

Останнім часом у зв'язку зі збільшенням обсягів опрацювання інформації застосовують такі більші одиниці, як:

1 Терабайт (Тбайт) = 1024 Гбайт = 2^{40} байт;

1 Петабайт (Пбайт) = 1024 Тбайт = 2^{50} байт;

1 Екзабайт (Ебайт) = 1024 Ебайт = 2^{60} байт.

Кодування чисел та тексту. Отже, інформація у ПК записується за допомогою елементів пам'яті, що можуть мати лише два стани. Позначимо стан ненамагніченості (або недзеркальності) для

оптичних носіїв як «0», а намагніченості (дзеркальності) як «1». За допомогою комбінацій таких «нулів» та «одиниць» можна описувати будь-які символи. Для цього достатньо встановити відповідність між кожним символом і конкретним набором «0–1». Але не завжди доцільно користуватись такою відповідністю. Наприклад, для виконання математичних обчислень виявилось більш раціональним всі розрахунки вести в так званій двійковій системі числення, в якій для запису чисел використовується всього 2 цифри: 0 і 1 (таблиця 1.1).

Таблиця 1.1

Запис чисел у різних системах числення

Десяткова система числення, $X_{(10)}$	**Двійкова система числення, $X_{(2)}$**	**Вісімкова система числення, $X_{(8)}$**	**Шістнадцяткова система числення, $X_{(16)}$**
1	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Для запису різних символів у вигляді двійкового коду існують спеціальні таблиці, а для запису двійкового коду чисел є різні методи переведення чисел у двійкову систему числення, наприклад: метод різниць, метод ділення, метод множення. Але простіше скористатися відповідною функцією калькулятора.

При створенні та прочитанні інформації в комп'ютері використовують певні кодування, які запрограмовані у драйверах (програмах). Найбільш відомі серед них кодування Unicod, Windows-1251, KOI8-Y та інші.

Універсальна система Unicod базується на 16-розрядному кодуванні символів. Ця 16-розрядна система забезпечує універсальні коди для 65536 різних символів, тобто в цій таблиці можуть розміститися символи мов більшості країн світу.

Кодування графічних зображень. Для кодування графічних даних застосовується, наприклад, такий метод кодування, як растр. Координати точок та їх властивості описуються за допомогою цілих чисел, які кодуються за допомогою двійкового коду. Так, чорно-білі графічні об'єкти можуть бути описані комбінацією точок із 256 градаціями сірого кольору, тобто для кодування яскравості будь-якої точки достатньо 8-розрядного двійкового числа.

Режим відображення кольорової графіки в системі RGB з використанням 24 розрядів (по 8 розрядів для кожного з трьох основних кольорів) називається повнокольоровим. Для такого режиму в системі CMYK необхідно мати 32 розряди (чотири кольори по 8 розрядів).

Кодування звуку. Неперервний електричний сигнал від джерела звуку (наприклад, мікрофона) має бути перетворений на дискретний сигнал. Розмір імпульсів дискретного сигналу має дорівнювати амплітуді аналогового сигналу від джерела звуку. Щоб перетворення на дискретний сигнал було достатньо точним, імпульси мають часто виникати один за одним.

Амплітуди імпульсів надалі наближено зображуються як двійкові числа. Весь діапазон зміни амплітуди розбивається на множину дискретних значень (зазвичай це 2^{16} = 65 536 значень), амплітуді імпульсу надається найближче дискретне значення. Внаслідок усіх цих перетворень для звукового сигналу утворюється послідовність двійкових чисел.

Зворотне перетворення дискретного сигналу на аналоговий виконується у спеціальних пристроях, підключених до комп'ютера, – звукових платах. За наявності звукової плати можемо прослуховувати на комп'ютері цифрові музичні диски. Сучасні звукові плати працюють зі звуком, який зображується як 16-бітні двійкові числа (звідси і кількість значень амплітуд звукового сигналу, що дорівнює 65 536).

1.4. Інформаційні процеси в економіці. Економічна інформація та її особливості

Для фахівців економічної галузі інтерес становить саме економічна інформація.

Економічну інформацію можна розглядати як ресурс, аналогічний матеріальним, трудовим і грошовим ресурсам.

Інформаційні ресурси – сукупність накопиченої інформації, зафіксованої на матеріальних носіях у будь-якій формі, що забезпечує її передачу в часі і просторі для вирішення наукових, виробничих, управлінських та інших завдань.

Під *економічною інформацією* розуміють сукупність даних (відомостей), що їх використовують для здійснення функцій управління народним господарством і його окремими ланками. До найважливіших властивостей економічної інформації належать:

- переважання алфавітно-цифрової форми подання;
- висока питома вага вихідних даних, що підлягають обробці;
- значний обсяг умовно-сталої інформації;
- дослідження таких інформаційних одиниць, як обліково-планові показники;
- висока точність результатів розв'язання задач;
- широке використання документної форми носіїв даних;
- переважання дискретної форми числових величин.

Крім загальних властивостей, кожний різновид економічної інформації характеризується властивостями, що випливають із методологістичних принципів розв'язання тих чи інших функціональних завдань управління.

На вдосконалення управлінської діяльності істотно впливають раціоналізація потоків інформації, процесів руху її та перетворення; організація систем інформації; поліпшення забезпечення інформацією керуючих елементів. Тому питання управлінської й інформаційної діяльності тісно взаємозв'язані.

Термін «інформація» означає повідомлення, відомості, знання, потрібні для прийняття рішень. Поняття якості інформації невіддільне від процедур прийняття рішень: тільки цілком вірогідна інформація може забезпечити якість рішення, допомогти

керівникові уникнути помилок. Економічна інформатика виступає тут як дисципліна, що створює технологію роботи з інформацією. Брак інформації призводить до того, що особа, яка приймає рішення, легко може припуститися помилки. Недоліки планування часто спричинені браком надійної інформації. Надмір інформації також може бути шкідливим і призводити до втрати цілісності, тому інформації потрібно стільки, скільки треба, не більше і не менше.

У процесі обробки даних на ПК широко використовується поняття структури інформації. Структурою визначається будова інформації та передбачається виділення певних її елементів (частин), що називаються одиницями. Одиниці бувають простими й складними. До простих належать такі елементи, які не можна поділити на частини. Складні одиниці – складені, утворені з інших інформаційних одиниць, простих або складних. При ієрархічній (багаторівневій) структурі економічної інформації одиницею нижчого рівня є реквізит, що становить одиницю інформації, яка не підлягає будь-якому членуванню. Реквізити являють собою слова або числа. Реквізити, що характеризують об'єкт управління якісно, називають ознаками, а кількісно – основами. Узяті окремо реквізити – ознаки й основи – не забезпечують всебічної характеристики явищ в економіці. Тому вони об'єднуються, утворюючи таку інформаційну одиницю, як показник. Він може бути простим (складатися з однієї основи й однієї ознаки) або складним (налічувати низку ознак). В управлінні використовуються також одиниці інформації, що складаються з самих реквізитів-ознак. Такі одиниці прийнято називати інформаційними повідомленнями. Інформаційне повідомлення – своєрідний показник, роль основи в якому відіграє провідний реквізит – ознака.

Вищий рівень інформаційної одиниці – набір даних, що є сукупністю однорідних показників і реквізитів-ознак на зовнішньому запам'ятовувальному пристрої. Набір даних називається файлом за термінологією ряду систем програмування.

Набір даних (файл) поділяється на частини, які не збігаються з одиницями інформації; сукупність наборів даних, що стосуються однієї ділянки управлінської роботи, часто називають інформаційним потоком.

Будь-які складові інформаційні одиниці (від окремих показників до інформаційної системи в цілому) можна розкладати, зрештою, на реквізити й тим самим підраховувати кількість мінімальних одиниць інформації, що лежать в основі її структурних побудов.

Економічну інформацію класифікують за рядом ознак. Залежно від здійснюваних в управлінні функцій розрізняють планову, облікову та регулюючу інформацію.

Планова в структурі економічної інформації займає 8–10 %. Вона містить директивні вказівки про розвиток конкретного об'єкта управління та його складових.

Облікова інформація в системі економічної інформації охоплює в середньому 88–90 %, відображаючи господарські процеси у вигляді натуральних, трудових і вартісних показників. Складовими облікової інформації є бухгалтерські, звітно-статистичні та оперативні дані.

Регулююча інформація займає в середньому майже 2 % від загального обсягу економічної інформації. На її основі приймають рішення щодо регулювання параметрів виробництва або планових завдань.

За стадіями утворення економічну інформацію поділяють на первинну та похідну.

Як первинна, так і похідна економічна інформація може бути змінною (робочою, оперативною) і постійною.

Інформація надходить працівникам апарату управління як із власних структурних підрозділів, так і від інших організацій. За цією ознакою інформацію поділяють на внутрішню і зовнішню.

Інформацію, що надходить до об'єкта управління, називають вхідною, а інформацію від об'єкта – вихідною.

До економічної інформації пред'являють такі вимоги: точність, достовірність, оперативність.

Точність інформації забезпечує її однозначне сприйняття всіма споживачами.

Достовірність визначає допустимий рівень спотворення як вхідної інформації, так і результатної (вихідної), при якому зберігається ефективність функціонування системи.

Оперативність відображає актуальність інформації для необхідних розрахунків і прийняття рішень в умовах, що змінилися.

Інформаційні процеси в економіці. Такі процеси, як збір, зберігання, опрацювання, передача інформації в числовій формі називаються інформаційними і здійснюються за допомогою інформаційних технологій.

Зовнішні інформаційні процеси визначаються взаємодією підприємства з економічними і політичними суб'єктами, що перебувають за межами підприємства. Це взаємодія підприємства з клієнтами і конкурентами: як реальними, так і потенційними. Внутрішні інформаційні процеси обробляють інформацію, яка описує процеси в колективі співробітників, а також відомості, породжувані у виробництві.

Підприємства мають і формують своє власне внутрішнє інформаційне середовище, у якому відбуваються інформаційні процеси з переробки вхідних і вихідних потоків інформації. Зовнішніми джерелами інформації для підприємства виступають держава, інформаційні центри і мережі, науково-дослідні організації, постачальники матеріалів, конкуренти, інфраструктура ринку тощо. Зовнішні джерела створюють вхідний потік інформації для підприємства. Вихідний інформаційний потік направляється підприємством у зовнішнє середовище і містить інформацію про свої виробничі можливості, вироблені товари (реклама), матеріальні, енергетичні, кадрові та інформаційні потреби.

Інформаційна система підприємства контролює інформаційні процеси, фільтрує інформаційний потік і виділяє інформацію, необхідну (релевантну) для життєдіяльності підприємства, перетворюючи її в зручну для прийняття рішень форму.

Особливістю інформаційних технологій є те, що в них і предметом, і продуктом опрацювання є інформація, а знаряддями праці – комп'ютери, засоби обчислювальної техніки і зв'язку.

У системах організаційно-економічного управління інформаційна технологія – це такий технологічний процес з опрацювання інформації, що забезпечує підготовку і прийняття управлінського рішення.

Об'єктом економічної інформатики виступають економічні інформаційні системи, мета функціонування яких – ефективне управління економікою, підприємством, організацією, установою.

Різноманітність завдань, що вирішуються з допомогою ІС, привела до появи безлічі різнотипних систем, що відрізняються принципами побудови і закладеними в них правилами опрацювання інформації. Інформаційні системи можна класифікувати за рядом різних ознак.

1. Класифікація ринку інформаційних систем за масштабністю системи, за класифікацією бізнес-завдань.

2. За кордоном були розроблені спеціальні програми Стандарти інформаційних систем управління підприємствами – системи *MRP, MRP-II, ERP, ERPII*.

MRP – це системи планування потреб у матеріальних ресурсах (забезпечує необхідний об'єм залишків матеріалів на складі).

MRP-II – призначені для планування виробничих ресурсів, тобто ресурсів, що використовуються для виробництва продукції.

ERP – призначена для планування і управління матеріальними, виробничими і людськими ресурсами. SAP R/3 – це ERP система (Enterprise Resourse Planning) управління ресурсами підприємства або SAP ER.

ERP II – призначена для управління ресурсами і зовнішніми зв'язками підприємств.

Інформаційні системи, що використовуються для планування і управління різними ресурсами, називаються інтегрованими системами управління або корпоративними інформаційними системами.

До основних компонентів інформаційних систем, що використовуються в економіці, відносять: програмно-апаратні засоби, бізнес-додатки і управління інформаційними системами.

1. Програмно-апаратні засоби інформаційних систем:

- технічні засоби опрацювання інформації (комп'ютери і периферійні пристрої);
- системне і сервісне програмне забезпечення (операційні системи і утиліти);
- прикладне програмне забезпечення офісного призначення (MS Office);

- комп'ютерні мережі (комунікаційне устаткування, мережеве ПЗ і мережеві застосування);
- бази і банки даних.

2. Бізнес-додатки (прикладні програми):

- локальні інформаційні системи (1С: Бухгалтерія, Інфін, Парус тощо);
- малі інформаційні системи (1С: Підприємство, Парус тощо);
- середні інформаційні системи (PEOPLE SOFT, BAAN, SCALA тощо);
- інтегровані системи управління (Галактика ERP).

3. Управління інформаційними системами призначене для управління і підтримки інформаційних процесів підприємства (управління персоналом, розвитком, якістю, безпекою, оперативне управління тощо).

Нині найбільш доцільним шляхом побудови економічної інформаційної системи є застосування готових рішень, які реалізовані у вигляді готових прикладних програм.

Офісна діяльність у системах організаційно-економічного управління полягає у забезпеченні ефективності процесів проходження документів (приймання документів, їх контроль, оформлення, забезпечення доступу до них, електронна пошта, автоматизація контролю за документоведенням, організація електронного документообігу, участь у нарадах з використанням засобів віддаленого доступу тощо).

Усі ці завдання можуть бути вирішеними за допомогою існуючих апаратно-програмних засобів, а саме ПК, програмних засобів текстового, табличного, графічного опрацювання інформації, засобів оперативного копіювання документів та засобів електронних комунікацій для взаємодії між різними типами ПК і мережами.

1.5. Класифікація та кодування інформації. Єдина система класифікації техніко-економічної інформації

Основні методи організації й пошуку економічної інформації в умовах застосування обчислювальної техніки потребують попередньої класифікації та кодування. Класифікація передає кодуванню номенклатуру об'єктів і являє собою впорядкування економічних явищ та предметів з урахуванням їх взаємозв'язків. Кожен об'єкт класифікації характеризується рядом властивостей, що називаються ознаками класифікації. У процесі класифікації утворюються множини та підмножини, які об'єднують частину об'єктів класифікації за однією чи кількома ознаками. Отже, визначаються класифікаційні групування. Розрізняють ієрархічну, фасетну та дискрипторну системи класифікації, до кожної з яких висувають такі вимоги:

- достатня місткість, що забезпечує урахування всіх об'єктів класифікації;
- гнучкість, що дає змогу розширювати множини класифікованих об'єктів і вносити зміни;
- можливість функціонування разом з іншими класифікаторами однорідних об'єктів;
- простота введення класифікатора.

Класифікатор – це систематизоване зведення назв класифікаційних групувань, їх кодових позначень.

Ієрархічна система класифікації передбачає встановлення між класифікаційними групуваннями відношення підпорядкування (ієрархії). Великі групування поділяються на ряд дрібних, послідовно конкретизуючи властивості об'єктів. Наприклад, група матеріалів, підгрупа і їх різновид. Отже, класифіковані поняття будуються у вигляді ієрархічного логічного дерева з послідовною конкретизацією властивостей об'єктів. У результаті створюється складна багаторівнева ієрархія понять.

Фасетна система класифікації – багатоаспектна, де класифікована множина утворює незалежні групування за цілим набором ознак, сформовані в паралельні незалежні фасети (групування).

Наприклад, ознаки об'єктів перепису населення. За фасетної системи класифікації набір ознак забезпечує розв'язання конкретних завдань. Наприклад, підрахунок кількості населення за національністю, статтю, віком. Окремі ознаки всередині фасетів при цьому часто утворюють ієрархічну структуру.

Дескрипторна система ґрунтується на застосуванні координарного методу інформування, який передбачає, що зміст документів або показників можна досить повно й точно відобразити за допомогою списку або переліку ключових слів. Дескриптор – це нормалізоване ключове слово або словосполучення. Дескриптори об'єднуються в слова або тезауруси.

Тезауруси – це систематизовані списки об'єктів, їх ознак (властивостей) з ключовими словами.

Усі системи класифікації мають свої переваги й недоліки. Вибір тієї чи іншої системи залежить від мети досліджень економічних даних і має проводитися з урахуванням вимог усього комплексу завдань управління.

Класифікація – основа кодування інформації.

Кодування – це процес присвоєння економічній інформації умовних позначень, підпорядкованих певним правилам. Сукупність правил, за якими здійснюється кодування, називається системою кодування. Код є позначенням ознаки об'єкта у вигляді знака або групи знаків згідно з прийнятою системою кодування. Розрізняють два типи кодів: машинні та економічні. Машинні коди використовують для керування машиною та подання команд, економічні об'єднують усі види кодів, використовуваних для подання техніко-економічної інформації. Знаки, що використовуються у процесі кодування, становлять абетку коду. Число знаків абеткового коду, що використовуються в кодовому позначенні, називається основою коду. Цифрова абетка коду складається з цифр, мішана – з букв і цифр.

В обчислювальній техніці існує система, яка має назву «двійкова система кодування», заснована на уявленні даних послідовністю усього двох знаків: 0 та 1. Іноді системи кодування називають системами числення. Також є вісімкова (від 0 до 7), шістнадцяткова (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) та десяткова (від 0 до 9) системи числення. Наприклад, у десятковій – 46, у двійковій – 101110, у вісімковій – 56, у шістнадцятковій – 2E.

Кодування інформації тісно пов'язане із застосуванням методів сортування, впорядкування, групування, пошуку. Ці методи визначають призначення й структуру кодів. Кодування і зворотний процес – декодування – виконують при автоматизованій обробці даних багаторазово, починаючи зі складання документів або інших носіїв інформації, передання даних каналами зв'язку і закінчуючи видачею результативних відомостей користувачеві.

Для кодування економічної інформації використовують такі системи кодування: порядкова, серійна, позиційна, шахова (матрична), комбінована.

Порядкова система кодування являє собою таке позначення позиційної номенклатури, яке відповідає її порядковим номерам. Розміщення для кодування може бути як випадковим, так і заздалегідь систематизованим. Порядковий код застосовується для ідентифікації малозначних, усталених списків назв. Переваги коду – легкість побудови, малозначність; недоліки – неможливість виділити класифікаційні групи та підгрупи ознак, неможливість розширення номенклатури у разі нових назв об'єкта.

Серійна система кодування – дальший розвиток порядкової. Ознаки попередньо групуються з урахуванням економічних вимог. Кожній групі відводиться серія номерів у порядку зростання, але з урахуванням резерву є вільні позиції на випадок появи нових об'єктів. Перевага цієї системи – легкість розширення номенклатури, коли прийнята система групування ознак не порушується, недолік – відсутність автоматичної можливості одержання кількох ступенів підсумків.

Позиційна система кодування застосовується для кодування багатоознакових номенклатур. Попередньо проводиться класифікація об'єктів з метою виділення груп, підгруп, різновидів ознак. Потім кожному виду ознак відводиться один, два або кілька видів, щоб старша в коді розміщувалася зліва, а нумерація молодшої ознаки починалася з першого номера. Переваги цієї системи – забезпечення групування даних за різними ознаками, можливість автоматичного стискання інформації, недоліки – багаторозрядність, громіздкість.

Шахова система кодування – різновид позиційної. Звичайно вона пов'язує одночасно дві ознаки у вигляді матриці, де одна ознака (старша) розміщується по горизонталі (по рядках), інша (молодша) – по вертикалі (по стовпцях). Код будується в комірках матриці як складової з двох характеристик.

Комбінована система кодування ґрунтується на поєднанні різних систем кодування з урахуванням їх переваг. Залежно від конкретних випадків доцільно використовувати, наприклад, серійно-позиційний код, порядково-серійний тощо.

Єдина система класифікації техніко-економічної інформації

Під час розв'язання економічних задач слід забезпечити їх порівнюваність та можливість використання цих результатів для розв'язування інших задач. Цього можна досягти за наявності єдиних систем угруповань, отриманих за єдиними класифікаційними ознаками.

Такі проблеми вирішують створенням Єдиної системи класифікації та кодування техніко-економічної інформації (далі – ЄСКК ТЕІ).

ЄСКК являє собою комплекс взаємопов'язаних ТЕІ, пристосованих до безпосередньої обробки засобами обчислювальної техніки з автоматизованою системою ведення цих класифікаторів. ЄСКК складається із сукупності взаємопов'язаних класифікаторів ТЕІ, систем їх ведення, науково-методичних і нормативно-технічних документів з розробки, ведення та впровадження, а також організацій і служб, які виконують роботи з класифікації та кодування.

ЄСКК ТЕІ охоплює широкий аспект об'єктів, інформацію про які використовують під час управління народним господарством. Перелік об'єктів ТЕІ, які відповідають класифікаційній множині класифікаторів, визначає їхні види, охоплює продукцію, що випускається в країні, структурні та адміністративні одиниці народного господарства (галузі, міністерства, відомства, об'єднання, підприємства, установи), адміністративно-територіальні одиниці, трудові ресурси і види діяльності, природні ресурси, документацію тощо. Особливим видом об'єктів у цьому переліку є техніко-економічні показники, які відображають діючий і плановий стан економіки.

Основною метою створення ЄСКК ТЕІ є стандартизація інформаційного забезпечення інформаційних процесів управління на основі застосування засобів обчислювальної техніки.

Залежно від рівня затвердження та сфери застосування класифікатори ТЕІ поділяють на три категорії:

- загальнодержавні;
- галузеві (відомчі);
- класифікатори об'єднань, підприємств та установ.

Згідно з установленими категоріями науково-технічної документації загальнодержавні класифікатори за статусом їх затвердження та застосування прирівнюються до державних стандартів, галузеві – до галузевих стандартів, класифікатори підприємств – до стандартів підприємств.

Загальнодержавні класифікатори ТЕІ мають затверджуватися Держстандартом України і обов'язково застосовуватися під час обміну інформацією між системами управління різних міністерств або відомств.

Галузеві (відомчі) класифікатори затверджуються відповідними міністерствами (відомствами) країни і застосовуються під час обміну інформацією між об'єднаннями, підприємствами та організаціями, підпорядкованими міністерству або відомству.

Класифікатори підприємств затверджуються керівництвом підприємств і застосовуються під час організації інформаційної взаємодії всередині підприємства.

Впровадження загальнодержавних класифікаторів ТЕІ передбачає або зміну класифікаторів, що використовуються у межах окремих систем, загальнодержавними, або застосування перекодувальних таблиць, що встановлюють відповідність кодів загальнодержавних і внутрішньосистемних класифікаторів на вході і виході системи, тобто під час організації інформаційної взаємодії систем.

Впровадження загальнодержавних класифікаторів передбачає їх використання під час кодування реквізитів форм економічних документів і кодування всіх номенклатур, використовуваних при виконанні економічних розрахунків.

1.6. Формалізація, алгоритмізація та автоматизована обробка економічної інформації

Підготовка реальних задач до автоматизованого розв'язування з використанням комп'ютера складається з таких етапів:

1. Постановка задачі та її змістовий аналіз (створення інформаційної моделі задачі).

2. Формалізація задачі (побудова математичної моделі, вибір методу розв'язування).

3. Складання алгоритму на основі вибраного методу.

4. Запис програми.

5. Введення, налагодження, тестування програми.

6. Виконання програми, аналіз результатів.

Формалізації будь-якого процесу, завдання чи задачі передує вивчення структури елементів, з яких складається цей процес, у результаті чого з'являється так званий змістовий опис процесу, тобто основа для створення формалізованої схеми процесу.

Змістовий опис включає в себе:

1. Аналіз умови задачі, а саме: що дано (які величини є вхідними та які їх значення допустимі), що потрібно зробити (які величини будуть вихідними або результатом і в якому вигляді вони мають бути представлені).

2. Проводиться запис в аналітичній формі всіх співвідношень між вихідними величинами і вхідними, описуються логічні умови та дії щодо реалізації процесу розв'язання.

3. Описується сукупність дій, необхідних для розв'язування задачі, тобто записують алгоритм розв'язування задачі.

Під алгоритмом розуміють послідовність дій, які необхідно виконати над вхідними даними, щоб розв'язати задачу і отримати результат або досягти поставленої мети.

Алгоритм має такі властивості: дискретність, визначеність, зрозумілість, універсальність, скінченність, результативність, формальність. Якщо алгоритм відповідає переліченим властивостям, то його виконання здійснюється формально, і це можна доручити комп'ютерам.

Розробка алгоритму є складним і трудомістким процесом. Алгоритмізація – це техніка розробки (складання) алгоритму для вирішення завдань на комп'ютері.

Ефективним методом побудови алгоритму є метод покрокової деталізації, при якому завдання розбивається на кілька простих підзадач (модулів) і для кожного модуля створюється свій власний алгоритм. Здебільшого модуль реалізує певний процес обробки інформації і застосовується як окремо, так і для включення модуля в інший алгоритм.

Застосування *модульності* під час створення алгоритмів дозволяє розбити великі задачі на незалежні блоки (модулі), усуває повторення стандартних дій і значно прискорює процес відлагодження алгоритму в цілому. Найчастіше алгоритм складається з головного модуля, який містить декілька інших модулів, створених раніше. Використовуючи модулі як складові великої конструкції, можна створювати алгоритми будь-якого ступеня складності, і при цьому не втрачати контролю за функціюванням алгоритму всієї задачі. Такий метод називається *структурованим* проєктуванням алгоритму «зверху-донизу», є універсальним і може використовуватися як для обчислювальних процесів (так зване системне програмування), так і для процесів реального життя.

Для запису алгоритму рішення задачі застосовуються такі способи їх подання:

- словесно-формульний опис;
- блок-схема (схема графічних символів);
- алгоритмічні мови;
- операторні схеми;
- псевдокод.

Для запису алгоритму існує загальна методика:

- кожен алгоритм повинен мати ім’я, яке розкриває його сенс;
- необхідно позначити початок і кінець алгоритму;
- описати вхідні і вихідні дані;
- вказати команди, які дозволяють виконувати певні дії над виділеними даними.

Загальний вигляд алгоритму:

- алгоритм: назва алгоритму;
- опис даних;
- початок;
- команди;
- кінець.

Формульно-словесний спосіб запису алгоритму характеризується тим, що опис здійснюється за допомогою слів і формул. Вміст послідовності етапів виконання алгоритмів записується на природній професійній мові наочно в довільній формі.

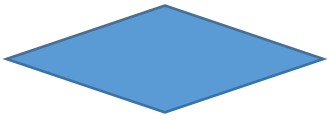
Графічний спосіб опису алгоритму (блок-схема) набув найбільшого поширення. Для графічного опису алгоритмів використовуються схеми алгоритмів або блокові символи (блоки), які з'єднуються між собою лініями зв'язку.

Кожен етап обчислювального процесу представляється геометричними фігурами (блоками). Вони діляться на арифметичні або обчислювальні (прямокутник), логічні (ромб) і блоки введення-виведення даних (паралелограм).

Основні елементи блок-схем.

Таблиця 1.2

Основні елементи блок-схеми алгоритму

Назва	Позначення	Значення
Термінатор		Початок або кінець програми, всередині елемента уточнюється
Процес		Виконання операцій, всередині записуються власне виконувані операції
Умовний перехід		Перемикач з одним або двома виходами, на які відбувається перехід залежно від виконання/невиконання умови, записаної всередині елемента
Підготовлений зовнішній процес		Операції, визначені в іншому місці програми, куди потрібно звернутись. Всередині елемента записується назва процесу і передані дані
Введення або виведення		Перетворення даних у форму для введення або виведення (відображення, друку)
Вузол		Елемент для переходу в іншу частину блок-схеми, якщо лінію перервано або частина схеми не помістилась у заданих межах
Лінії потоку		Для з'єднання блоків

Базові структури алгоритмів. У 1966 році вчені Бром і Джакопіні довели таку теорему: «Кожен алгоритм може бути побудований за допомогою лише трьох базових алгоритмічних структур – слідування, розгалуження, циклу».

Структура слідування, тобто лінійна, містить вказівки, що виконуються послідовно одна за одною.

Розгалужена структура передбачає вибір дії залежно від дотримання деякої умови, при цьому деякі вказівки можуть не виконуватися взагалі. Умови, що використовуються в розгалуженому алгоритмі, поділяються на прості і складені. Прості умови записують за допомогою двох виразів, поєднаних знаками відношень >, <, >=, <=, =, <>. Результатом перевірки умови є логічний вираз ІСТИНА (TRUE), якщо умова виконується, або ХИБНІСТЬ (FALSE), якщо умова не виконується.

Складена умова містить дві і більше простих умов, поєднаних знаками логічних операцій:

- і, &, and, наприклад – <умова1> & <умова2>, and(умова1; умова2) – результат буде істинним тоді і тільки тоді, коли обидві прості умови будуть дотримані;
- або, or, наприклад – <умова1> or <умова2>, or(умова1; умова2) – результат буде істинним тоді, коли хоча б одна з умов буде дотримана;
- не, not, наприклад – not(умова1) – це заперечення умови1, результатом буде твердження, протилежне умові1.

Таблиця 1.3

Таблиця істинності тверджень

Умова1	Умова2	and(Умова1; Умова2)	Or(Умова1; Умова2)	Not(Умова1)
T	T	T	T	F
F	T	F	T	T
T	F	F	T	F
F	F	F	F	T

Циклічна структура використовується у разі необхідності повторень виконання деяких дій з різними (новими) значеннями вхідних даних, тобто коли треба виконати перебір варіантів. Алгоритми циклічної структури описують цикли трьох видів: безумовний або арифметичний цикл (цикл ДЛЯ); умовний цикл з

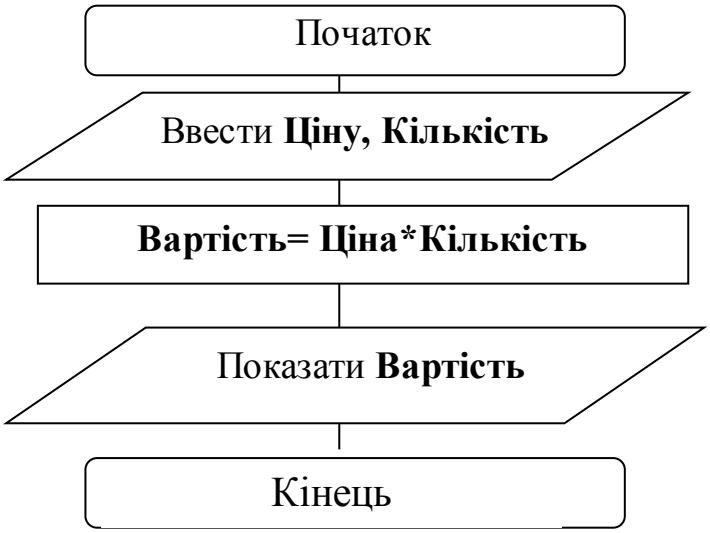
передумовою (цикл ПОКИ) та умовний цикл з післяумовою (цикл ДО). Як правило, умовні цикли використовують тоді, коли невідома кількість повторень вказівок циклу, такі цикли ще називають ітераційними.

Для побудови алгоритмів використовують структурний метод, що полягає у структурованому проєктуванні алгоритму «зверху-донизу» та метод покрокової деталізації, коли завдання розбивають на кілька простих підзадач (модулів) і для кожної підзадачі (кожного модуля) створюють власний алгоритм, який можна використовувати як окремо, так і включаючи в інший алгоритм, один з яких є головним, а інші – допоміжні, або підалгоритми. Якщо у процесі обчислення допоміжний алгоритм використовують багаторазово, то його оформляють у вигляді алгоритму-процедури, або алгоритму-функції. Особливість їх використання полягає в тому, що результатом алгоритму-функції є тільки одне значення – значення описаної функції.

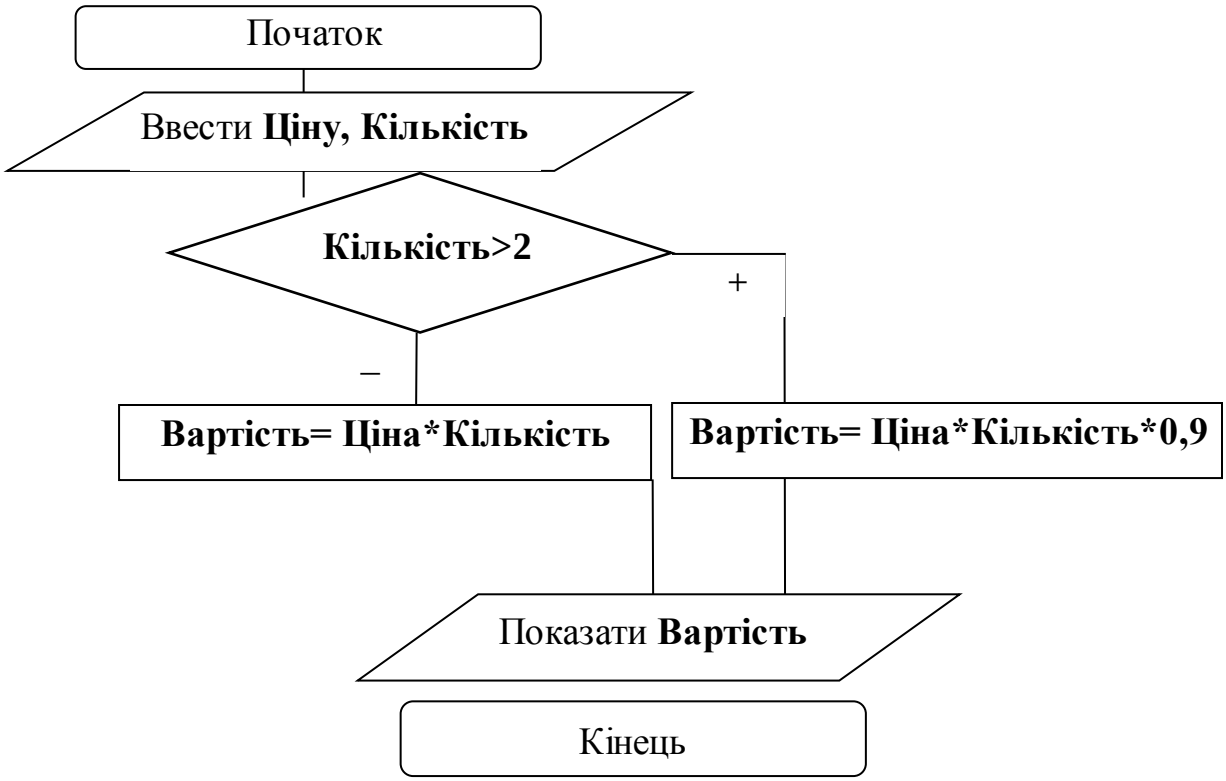
Комбінуючи базові структури алгоритмів, можна створювати алгоритми будь-якого ступеня складності.

Приклади алгоритмів

Алгоритм лінійної структури. Обчислення вартості товару за ціною і кількістю.



Алгоритм розгалуженої структури. Обчислення вартості товару за ціною і кількістю з урахуванням знижки у розмірі 10 %, якщо кількість більше 2.



Алгоритм циклічної структури. Циклом називають повторення послідовної кількості кроків алгоритму.

Розрізняють такі основні види циклів:

- цикл з лічильником;
- цикл з вхідним значенням;
- цикл з вихідним значенням (ітераційний цикл);
- цикл з накопиченням.

Під час кожного чергового виконання циклу перевіряється умова на продовження роботи, і, якщо умова набуває результату ІСТИНА, цикл виконується, а якщо ж умова набуває результату ХИБА – цикл не виконується. Перевірка умови може бути організована на початку циклу, і такий цикл називається циклом з передумовою, або у кінці циклу – тоді такий цикл називається циклом з післяумовою.

Різниця між такими циклами полягає в тому, що цикл з післяумовою виконується хоча б один раз, а цикл з передумовою може не виконуватися жодного разу.

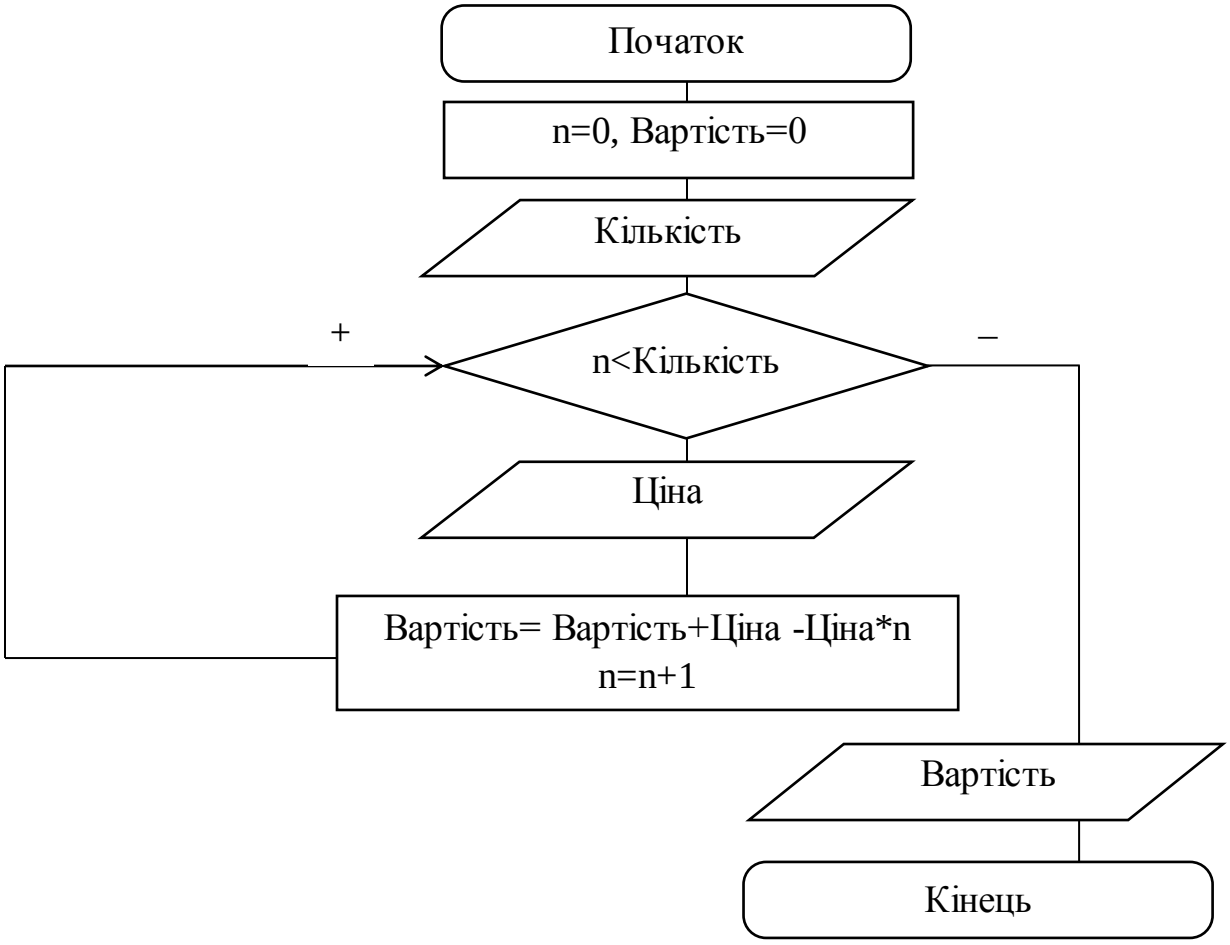
Цикл з лічильником характерний тим, що заздалегідь відома кількість повторень циклу, і цикл буде виконуватися, доки значення лічильника циклу не перевищить зазначену кількість повторень.

Цикл з вхідним значенням характерний тим, що відомі початкове та кінцеве значення вхідної змінної, а також закон (формула), за яким значення вхідної змінної змінюється від циклу до циклу. Цикл продовжує виконуватися, доки значення вхідної змінної лежатиме у межах від початкового до кінцевого значення, і кількість повторень циклу в явному вигляді невідома.

Цикл з вихідним значенням характерний тим, що він виконується, доки вихідне значення не досягне зазначеної величини, тобто так само, як і в попередньому випадку, кількість повторень циклу заздалегідь невідома.

Цикл з накопиченням характерний тим, що під час чергового виконання циклу нове значення вихідної змінної залежить від значення цієї ж змінної, визначеної на попередньому витку циклу. Накопичуватися може або сума значень, або добуток.

Алгоритм циклічної структури. Знайти значення накопичувальної знижки: за кожну наступну одиницю товару знижка збільшується на 1 %, але не може перевищувати 100 %. Для спрощення задачі будемо вважати, що кількість товару менше 100. Також введемо допоміжну змінну n – номер товару.



Отже, перш ніж починати розв'язання будь-якої задачі, потрібно:

- задачу формалізувати;
- розбити на кроки;
- кожному кроку дати відповідно розроблений модуль (або розробити його);
- скласти повний алгоритм;
- перевірити його функціональність;
- приступити до виконання.

Для цього зовсім необов'язково подавати алгоритм схематично або описувати словесно, якщо він досить простий, але у разі розв'язання складних задач це – обов'язкова умова успішного результату.

Запитання для самоконтролю

1. Що вивчає економічна інформатика?
2. Які основні об'єкти економічної інформатики?
3. Що таке інформація? Що таке економічна інформація?
4. Які особливості економічної інформації?
5. Що таке інформаційні системи?
6. Що називається алгоритмом?
7. Які типи алгоритмів існують?
8. Наведіть приклади базових алгоритмів у задачах економічного змісту.
9. Які процеси називаються інформаційними?
10. Які ознаки інформаційного суспільства?
11. Що таке цифрова економіка?
12. Які системи кодування використовуються в інформатиці?
13. Якими співвідношеннями зв'язані одиниці 1Кб, 1Мб, 1Гб, 1Тб, 1Пб, 1Еб?
14. Що таке система числення?

15. Яку мінімальну основу повинна мати система числення, якщо в ній можна записати числа: 341, 123, 222, 101?

16. Коли 2*2=100?

17. Які основні системи класифікації економічної інформації?

18. Що таке класифікатор техніко-економічної інформації?

ЛЕКЦІЯ 2
СИСТЕМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ В ЕКОНОМІЧНІЙ ДІЯЛЬНОСТІ

2.1. Апаратне забезпечення інформаційних процесів в економіці

Персональний комп'ютер (далі – ПК) – це малогабаритний комп'ютер індивідуального користування, реалізований на базі мікропроцесорних засобів. До його структури входить апаратна, програмна та інформаційна складова. Апаратну складову можна подати у вигляді такої схеми (рис. 2.1):

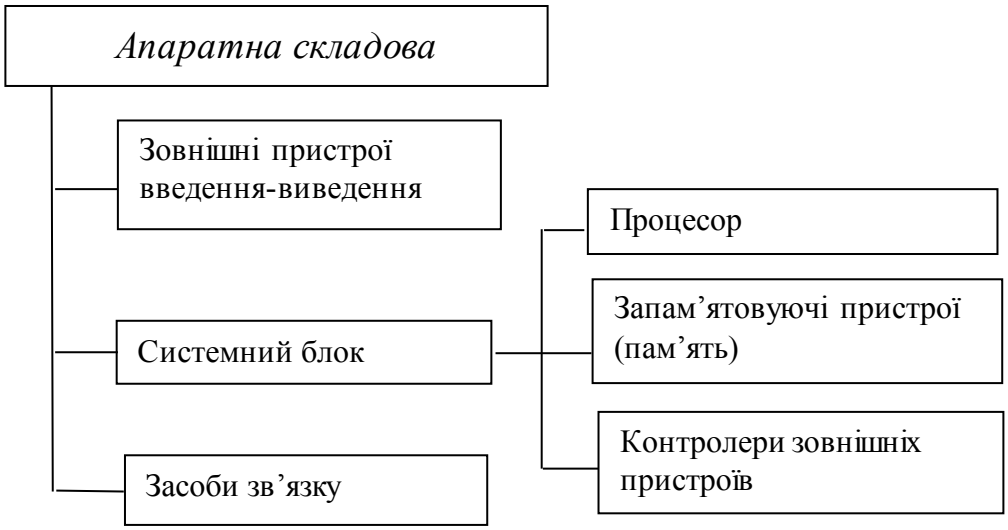


Рис. 2.1. Апаратна складова комп'ютера

Пристрої введення інформації – призначені для введення інформації до ПК. Прикладами є клавіатура, маніпулятори (миша, джойстик, трекбол тощо), сканер та інші.

Клавіатура – пристрій, призначений для введення користувачем інформації в комп'ютер. Стандартна клавіатура має більше 100 клавіш. Клавіші клавіатури розділяються на 6 груп:

- клавіші з позначеннями букв алфавіту;
- цифрові клавіші (перемикання режиму роботи здійснюється клавішею NumLock);

- клавіші редагування (Insert, Delete, Back Space);
- клавіші управління курсором (дві групи клавіш: чотири клавіші із стрілками і чотири клавіші: Home, End, Page Up, Page Down);
- спеціальні клавіші (Ctrl, Alt, Esc, Num Lock, Scroll Lock, Print Screen, Pause).
- функціональні клавіші F1 – F12 (розташовані у верхній частині клавіатури і призначені для виклику команд, що часто використовуються).

Для введення прописних букв та інших символів, розташованих на верхньому регістрі клавіатури, є клавіша [Shift]. Наприклад, щоб ввести прописну букву, треба натискувати клавішу [Shift] і, не відпускаючи її, натискувати клавішу з необхідним символом.

Клавіша [Caps Lock] слугує для фіксації режиму прописних букв. Клавіша [Space] – для створення пропуску між символами. Клавіша [Enter] при редагуванні тексту працює як «повернення каретки» на машинці, що пише. Крім того, натиснення цієї клавіші може означати закінчення введення команди або іншої інформації і звернення до комп'ютера.

Перемикання мови клавіатури (російський – український – англійський) можна здійснити за допомогою перемикача клавіатури, розташованого на панелі завдань, або за допомогою поєднань клавіш (Shift+ Ctrl або Shift+ Alt).

Маніпулятор миша – пристрій управління маніпуляторного типу. Невелика коробочка з клавішами (1, 2 або 3 клавіші). Переміщення миші по плоскій поверхні (наприклад, килимка) синхронізоване з переміщенням покажчика миші на екрані монітора.

Введення інформації здійснюється переміщенням курсора в певну область екрану і короткочасним натисненням кнопок маніпулятора або клацаннями (одинарними або подвійними). За принципом роботи маніпулятори діляться на механічні, оптомеханічні і оптичні.

У портативних ПК як миша використовуються трекболи і пойнтери. Комбінація монітора і миші забезпечують діалоговий режим роботи користувача з комп'ютером, це найбільш зручний і

сучасний тип інтерфейсу користувача. Корпорація Microsoft випустила новий набір з клавіатури і миші, призначений для настільних ПК. Продукт отримав назву Natural Ergonomic Desktop 7000, у нім використовується безпровідна технологія.

Пристрої виведення інформації – призначені для виведення даних (монітор, принтер, графопобудувачі, проєктор, динаміки та ін.).

Монітори – пристрої, які слугують для забезпечення діалогового режиму роботи користувача з комп'ютером шляхом виводу на екран графічної і символьної інформації. У графічному режимі екран складається з крапок (пікселів від англ. pixel – picture element, елемент картинки), отриманих розбиттям екрану на стовпці і рядки.

Кількість пікселів на екрані називається роздільною здатністю монітора в даному режимі. Нині монітори ПК можуть працювати в таких режимах: 480x640, 600x800, 768x1024, 864x1152, 1024x1280 (кількість пікселів по вертикалі і горизонталі).

Роздільна здатність залежить від типу монітора і відеоадаптера. Кожен піксель може бути забарвлений в один з можливих кольорів. Стандарти відображення кольору: 16, 256, 64К, 16М колірних відтінків кожного пікселя.

За принципом дії всі сучасні монітори розділяються на:

- монітори на базі електронно-променевої трубки (CRT);
- рідкокристалічні дисплеї (LCD);
- плазмові монітори;
- LED-монітори;
- 4K-монітори;
- IPS-монітори.

Найбільш поширеними є монітори з рідкокристалічними дисплеями (екранами) LED-монітори. Найвищу якість зображення мають сучасні 4K монітори та IPS-монітори. Стандартні монітори мають довжину діагоналі 14, 15, 17, 19, 20, 21, 22, 24 дюйми. Під час налаштування монітора необхідно встановлювати такі параметри роздільної здатності і режиму відображення кольору, щоб частота оновлення кадрів не перевищувала 85 Гц.

У моніторах LCD, LED, 4K зображення формується за допомогою матриці пікселів. Кожен піксель формується свіченням одного елементу екрана, тому кожен монітор має свій максимальний фізичний дозвіл. Так, наприклад, для моніторів 19 дюймів роздільна здатність 1280x1024.

Для того щоб виключити спотворення зображень на екрані, рекомендується використовувати монітори LCD, LED, 4K у режимах його максимального дозволу. Для моніторів LCD, LED, 4K частота зміни кадрів не є критичною. Зображення виглядає стійким (без видимого мерехтіння) навіть при частоті оновлення кадрів 60 Гц.

У плазмових моніторах зображення формується за допомогою матриці пікселів, як і в моніторах LCD. Принцип роботи плазмової панелі полягає в керованому холодному розряді розрядженого газу (ксенону або неону), що перебуває в іонізованому стані (холодна плазма).

Піксель формує група з трьох підпікселів, відповідальних за три основні кольори, які є мікрокамерами, на стінках яких є флюоресцентна речовина одного з основних кольорів. Це одна з найбільш перспективних технологій плоских дисплеїв.

Переваги плазмових моніторів полягають у тому, що в них відсутнє мерехтіння зображення, картинка має високу контрастність і чіткість по всьому дисплею, мають хорошу оглядовість під будь-яким кутом і малу товщину панелі. До недоліків потрібно віднести велику споживану потужність.

Принтер – це пристрій для виведення (друкування) повідомлень на папері. Існує три типи принтерів: матричні, струменеві та лазерні.

Матричні принтери найбільш поширені. За своєю будовою і принципами роботи вони нагадують друкарську машинку. Друкуюча головка складається з голок: у потрібний момент певна голка (або кілька голок) з силою виштовхується і ударяє через фарбувальну стрічку по паперу, залишаючи на папері точки у відповідних місцях. Усі роздруковані сторінки на матричному принтері складаються з таких точок. Чим більше голок на друкуючій головці, тим краща якість друкування.

Струменеві принтери дають високої якості чорно-білий і кольоровий друк. Друкуються зображення через струменевий принтер за допомогою крапельок фарбувальної речовини, які вилітають із сопел і залишають сліди (точки) на папері. З цих точок формується зображення. Швидкість роботи струменевого принтера така сама, як і матричного, а якість друкування набагато вища. Перевагою струменевих принтерів є безшумність роботи.

Лазерні принтери є найбільш швидкодіючими і дають найвищої якості чорно-білий і кольоровий друк. Швидкість друкування на лазерному принтері – одна сторінка тексту за 5 сек. – приблизно в 20 разів швидше, ніж на матричному чи струменевому принтері.

Процесор (центральний мікропроцесор) – це невелика мікросхема, що виконує всі обчислення і обробку інформації, це ядро ПК. У комп'ютерах типу IBM PC використовуються мікропроцесори фірми Intel і сумісні з ними мікропроцесори інших фірм.

Компоненти мікропроцесора:

- АЛУ виконує логічні і арифметичні операції;
- пристрій управління управляє всіма пристроями ПК;
- регістри використовуються для зберігання даних і адрес;
- схема управління шиною і портами – здійснює підготовку пристроїв до обміну даними між мікропроцесором і портом введення – виводу, а також управляє шиною адреси і управління.

Основні характеристики процесора:

- розрядність – число двійкових розрядів, що одночасно обробляються під час виконання однієї команди. Сучасні процесори – це 32-розрядні процесори, а також 64-розрядні процесори;
- тактова частота – кількість циклів роботи пристрою за одиницю часу. Чим вища тактова частота, тим вища продуктивність;
- наявність вбудованого математичного співпроцесора;
- наявність і розмір кеш-пам'яті.

Оперативна пам'ять (ОЗУ або RAM) – це область пам'яті, призначена для зберігання інформації протягом одного сеансу роботи з комп'ютером. Конструктивно ОЗУ виконано у вигляді інтегральних мікросхем.

З пам’яті процесор прочитує програми і вхідні дані для обробки в свої регістри, в пам’ять записує отримані результати. Назву «оперативна» ця пам’ять отримала тому, що вона працює дуже швидко, в результаті процесору не доводиться чекати при читанні або записі даних у пам’ять.

Проте швидкодія ОЗУ нижча за швидкодію регістрів процесора, тому перед виконанням команд процесор переписує дані з ОЗУ в регістри. За принципом дії розрізняють динамічну пам’ять і статичну.

Елементи динамічної пам’яті є мікроконденсаторами, які нагромаджують заряд на своїх обкладаннях. Елементи статичної пам’яті є тригерами, які можуть перебувати в двох стійких станах.

Основні параметри, які характеризують ОЗУ, – це ємкість і час звернення до пам’яті. ОЗУ типа DDR SDRAM (синхронна пам’ять з подвійною швидкість передачі даних) вважається найбільш перспективним для ПК.

Кеш-пам’ять слугує для забезпечення швидкого доступу до оперативної пам’яті, інакше мікропроцесор простоюватиме, і швидкодія комп’ютера зменшиться. Тому сучасні комп’ютери оснащуються кеш-пам’яттю або надоперативною пам’яттю.

За наявності кеш-пам’яті дані з ОЗУ спочатку переписуються в неї, а потім в регістри процесора. При повторному зверненні до пам’яті спочатку проводиться пошук потрібних даних у кеш-пам’яті і необхідні дані з кеш-пам’яті переносяться у регістри, тому підвищується швидкодія.

Контролери. Лише та інформація, яка зберігається в ОЗУ, доступна процесору для обробки. Тому необхідно, щоб в його оперативній пам’яті знаходилися програма і дані.

У ПК інформація із зовнішніх пристроїв (клавіатури, жорсткого диска і так далі) пересилається в ОЗУ, а інформація (результати виконання програм) з ОЗУ також виводиться на зовнішні пристрої (монітор, жорсткий диск, принтер і так далі).

Отже, у комп’ютері повинен здійснюватися обмін інформацією (уведення-виведення) між оперативною пам’яттю і зовнішніми пристроями. Пристрої, які здійснюють обмін інформацією між оперативною пам’яттю і зовнішніми пристроями, називаються контролерами або адаптерами, іноді картами. Контролери,

адаптери або карти мають свій процесор і свою пам'ять, тобто є спеціалізованим процесором.

Контролери або адаптери (схеми, керівники зовнішніми пристроями комп'ютера) знаходяться на окремих платах, які вставляються в уніфіковані роз'єми (слоти) на материнській платі.

Системна магістраль (шина) – це сукупність дротів і роз'ємів, що забезпечують об'єднання всіх пристроїв ПК в єдину систему і їх взаємодію.

Для підключення контролерів або адаптерів сучасні ПК забезпечені такими слотами, як PCI. Слоти PCI – E Express для підключення нових пристроїв до швидкіснішої шини даних. Слоти AGP призначені для підключення відеоадаптера.

Для підключення накопичувачів (жорстких дисків і компакт-дисків) використовуються інтерфейси IDE і SCSI. Інтерфейс – це сукупність засобів з'єднання і зв'язку пристроїв комп'ютера.

Підключення периферійних пристроїв (принтери, миша, сканери і так далі) здійснюється через спеціальні інтерфейси, які називаються портами. Порти встановлюються на задній стінці системного блоку.

Слоти (роз'єми) розширення конфігурації ПК призначені для підключення додаткових пристроїв до основної шини даних комп'ютера. До основних плат розширення, призначених для підключення до шини додаткових пристроїв, відносять:

- відеоадаптери (відеокарти);
- звукові плати;
- внутрішні модеми;
- мережеві адаптери (для підключення до локальної мережі);
- SCSI – адаптери.

Зовнішня пам'ять. Для зберігання програм і даних у ПК використовуються накопичувачі різних типів. Накопичувачі – це пристрої для запису і прочитування інформації з різних носіїв інформації. Розрізняють накопичувачі зі змінним і вбудованим носієм.

За типом носія інформації накопичувачі розділяються на накопичувачі на магнітних стрічках і дискові накопичувачі. До накопичувачів на магнітних стрічках відносять стримери та ін. Ширший клас накопичувачів становлять дискові накопичувачі.

За способом запису і читання інформації на носій дискові накопичувачі розділяються на магнітні, оптичні і магнітооптичні.

До дискових накопичувачів відносять:

- накопичувачі на флопи-дисках;
- накопичувачі на незмінних жорстких дисках (вінчестери);
- накопичувачі на змінних жорстких дисках;
- накопичувачі на магнітооптичних дисках;
- накопичувачі на оптичних дисках (CD-R CD-RW CD-ROM) з однократним записом;
- накопичувачі на оптичних DVD – дисках (DVD-R DVD-RW DVD-ROM і ін.).

Периферійні пристрої – це пристрої, які підключаються до контролерів ПК і розширюють його функціональні можливості.

За призначенням периферійні пристрої розрізняють як:

- пристрої введення (трекболи, джойстики, світлове пір'я, сканери, цифрові камери, діджитайзери, стилуси);
- пристрої виведення (плотери або графічні пристрої);
- пристрої зберігання (стримери, zip – накопичувачі, оптичні накопичувачі, накопичувачі HIFD, флеш-карти та ін.);
- пристрої обміну (модеми).

2.2. Програмне забезпечення інформаційних процесів в економіці

Програмне забезпечення ПК розподіляється на кілька рівнів.

Найнижчий рівень – базовий – представляє базове програмне забезпечення, яке відповідає за взаємодію з базовими апаратними засобами. Базові програмні засоби входять до складу базового обладнання і зберігаються у спеціальних мікросхемах – ПЗП (постійні запам'ятовуючі пристрої).

Системний рівень – перехідний. Програми цього рівня забезпечують взаємодію інших програм комп'ютера з програмами базового рівня і з апаратним забезпеченням. Наприклад, програми-драйвера пристроїв, програми, що забезпечують взаємодію з користувачем, – входять до складу програмного забезпечення системного рівня і утворюють ядро операційної системи.

Операційна система – це сукупність програм, які призначені для керування ресурсами комп'ютера, обчислювальними процесами, підтримки файлової структури зберігання даних, а також для організації взаємодії користувача з апаратурою комп'ютера.

Службовий рівень об'єднує службові програми, що називаються утилітами. Вони призначені для автоматизації робіт з перевірки, налагодження і настроювання комп'ютерної системи. Деякі з цих програм входять до складу ОС, інші функціонують автономно.

Прикладний рівень являє собою комплекс прикладних програм, за допомогою яких можна розв'язувати конкретні задачі.

Таблиця 2.1

Рівні програмного забезпечення комп'ютера

Прикладне ПЗ	Текстові редактори, текстові процесори, графічні редактори, СУБД, ЕТ, САП, видавничі системи, експертні системи, вебредактори, браузери, інтегровані системи забезпечення документообігу, бухгалтерські системи, фінансові аналітичні системи, навчальні, довідкові
Службове ПЗ	Диспетчери файлів, архіватори, засоби перегляду, засоби діагностики, засоби контролю, засоби комунікації, засоби забезпечення комп'ютерної безпеки, антивірусні програми, засоби електронного цифрового підпису
Системне ПЗ	ОС, драйвери пристроїв, оболонки ОС, системи програмування
Базове ПЗ	Bios

Програмне забезпечення (далі – ПЗ) персональних комп'ютерів можна поділити на дві основні частини: системне і прикладне ПЗ (рис. 2.2).

Системне ПЗ призначене для управління роботою комп'ютера, розподілу його ресурсів, підтримки діалогу з користувачем, надання йому допомоги в обслуговуванні комп'ютера тощо. Більшість із програм системного ПЗ постачаються разом із комп'ютером і мають відповідну експлуатаційну документацію. Системне ПЗ, у свою чергу, розподіляють на три основні складові: операційні системи (далі – ОС), системи програмування та сервісні програми (рис. 2.2).

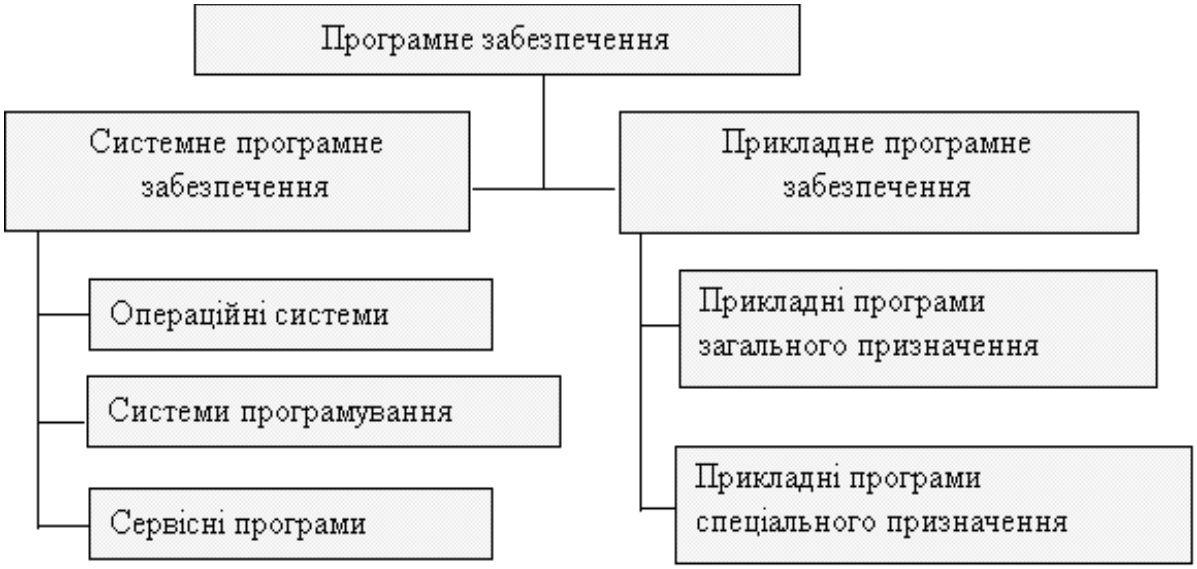


Рис. 2.2. Структура програмного забезпечення

Системи програмування призначені для полегшення та часткової автоматизації процесу розробки та налагодження програм. Основними компонентами цих систем є транслятори з мов високого рівня, наприклад Паскаль, Сі, Бейсик та ін. Особлива роль належить асемблерам. Програму мовою асемблера називають машинно-орієнтованою. Мовою асемблера користуються, як правило, системні програмісти.

Транслятори здійснюють перетворення програм із мов високого рівня на машинну мову. Крім того, транслятори зазвичай здійснюють синтаксичний аналіз програми, яка транслюється. Вони можуть також налагоджувати та оптимізувати програми, які одержують, видавати документацію на програму та виконувати ряд інших сервісних функцій.

Сервісні програми розширюють і доповнюють можливості ОС. Їх називають утилітами. Утиліти дозволяють, наприклад, перевірити інформацію в шістнадцятковому коді, яка зберігається в окремих секторах магнітних дисків; організувати виведення на принтер текстових файлів у визначеному форматі, виконувати архівацію та розархівацію файлів тощо.

Прикладне (проблемне) ПЗ призначене для розв’язання конкретних прикладних завдань виробничого, наукового, управлінського, навчально-тренувального характеру. Прикладне ПЗ комплектується у міру необхідності користувачем ПК. У структурі

прикладного ПЗ виділяють прикладні програми загального та спеціального призначення.

Прикладне ПЗ загального призначення включає комплекси програм, які застосовуються практично в будь-якій сфері діяльності людини, тобто є універсальними. Найпоширенішими серед ПЗ загального призначення є текстові редактори, графічні системи, електронні таблиці, системи управління базами даних та ін.

Текстові редактори дозволяють готувати текстові документи, починаючи з простих листів і закінчуючи складними технічними описами. Найбільш відомі такі текстові редактори: Лексикон, Write, Word.

Серед графічних систем виділяють системи ділової графіки (Microsoft Power Point, Lotus Freelance Graphics), художньої графіки, які ще називають просто графічними редакторами (Paintbrush), інженерної графіки та автоматизованого проєктування (Autodesk AutoCad), системи обробки фотографічних зображень (Adobe Photoshop), а також універсальні графічні системи (CorelDRAW).

Програми роботи з електронними таблицями дозволяють вирішувати велике коло завдань, зв'язаних із числовими розрахунками. Найширше використовують програми Supercalc, Microsoft Excel та Lotus 1-2-3.

Системи управління базами даних призначені для об'єднання наборів даних з метою створення єдиної інформаційної моделі об'єкта. Ці програми дозволяють накопичувати, поновлювати, редагувати, вилучати, сортувати інформацію, організовану спеціальним засобом у вигляді банку даних. Найпоширеніші з них: dBase III Plus, FoxBase+, Clipper, Access, FoxPro, Paradox.

Крім перерахованих систем, до складу прикладного програмного забезпечення загального призначення належать й інтегровані системи, що об'єднують можливості текстових редакторів, графічних систем, електронних таблиць та систем управління базами даних. Головна перевага інтегрованих систем щодо окремих систем прикладного програмного забезпечення загального призначення полягає у тому, що вони створюють єдині правила роботи як із текстом, так і з електронними таблицями та ін. Найвідоміші серед них: Microsoft Word, Microsoft Office, Perfect Office.

Прикладні програми спеціального призначення використовуються користувачами у специфічній професійній діяльності.

Інформаційна складова, або інформаційне забезпечення – сукупність програм і попередньо підготовлених даних, необхідних для роботи цих програм, наприклад, перевірка орфографії забезпечується наявністю спецсловників.

Сервісні програми для роботи з дисками. До службового програмного забезпечення належать і сервісні програми для роботи з дисками, для форматування диску, діагностики диску, дефрагментації та сканування диску. Вони входять до складу ОС Windows і дозволяють форматувати дискети, перевіряти гнучкі і жорсткі диски на наявність помилок, проводити дефрагментацію дисків, стиснення об'єму робочого простору, очистку дисків і багато інших операцій.

Форматування, або ініціалізація диска, являє собою процес поділу диска на сектори і доріжки.

Перевірка диска на наявність різних помилок і збоїв забезпечується спеціальною програмою, що виконує не тільки перевірку диска, а й виправлення програмних помилок. Після закінчення роботи програма повідомляє про те, який повний обсяг диска, скільки пам'яті зайнято програмами, а також, чи є на диску пошкоджені сектори.

У процесі роботи комп'ютера відбувається переміщення вмісту файлів у різні області простору диска. Як правило, файли не зберігаються в одному місці диска. Вони поділені на частини і знаходяться у різних областях диску. Такий спосіб збереження інформації збільшує час доступу до неї і утруднює роботу з великими файлами.

Програма *дефрагментації* диска переміщує файли таким чином, щоб усі частини одного файлу зберігалися у суміжних областях, тому після дефрагментації продуктивність комп'ютера підвищується при роботі з дисками.

Технологія *ущільнення дискового* простору застосовується для того, щоб отримати додаткове місце на диску для подальшого використання для збереження інформації, причому збільшення об'єму відбувається у 2–3 рази.

Програма очистки диска допомагає очистити простір на жорсткому диску, вона перевіряє диск і виводить перелік тимчасових файлів, файлів кеша інтернету, а також непотрібних програмних файлів, видалення яких не приведе до негативних наслідків.

«*Управління дисками*» – це системна службова програма для управління жорсткими дисками і розділами чи томами, що містяться на них. За допомогою цієї програми можна ініціалізувати нові диски (на фізичному диску може бути створено до чотирьох основних розділів), створювати томи, а також форматувати томи для використання файлових систем FAT, FAT32 або NTFS. Програма «Управління дисками» дозволяє виконувати задачі щодо роботи з дисками без перезавантаження комп'ютера; більшість змін відбувається одразу.

Прикладне програмне забезпечення офісного призначення. Нині на ринку програмного забезпечення є потужні програмні пакети, що отримали назву офісних систем. Найбільш поширеним на сьогодні є пакет Microsoft Office.

У MS Office входять текстовий процесор MS Word, табличний процесор MS Excel, засіб для створення та управління базами даних MS Access, а також спеціальні програми для організації роботи офісів. Серед цих програм: MS Outlook – засіб доступу до різноманітної інформації і її колективної обробки, MS PowerPoint – програма-додаток для підготовки і проведення презентацій, MS FrontPage та MS Publisher – програми для створення вебсторінок, публікацій та презентацій у глобальній мережі «Інтернет» і ряд інших.

У зв'язку з тим, що система Windows постійно модифікується фірмою-виробником, версії програмного пакета MS Office відповідно змінюються разом з нею.

Усі програми MS Office 2000, 2003, 2007, 2010 (за винятком Access) використовують однакові формати файлів, що дозволяє без додаткової обробки обмінюватися документами, а формати документів версії MS Office 2007 вже відрізняються, про що потрібно пам'ятати.

Важливою тенденцією в розвитку програмних продуктів цього пакета є підвищення «інтелектуальності».

До нього можна віднести подальший розвиток звичного інтерфейсу за рахунок функції, що дозволяє настроювати меню і панелі інструментів так, щоб відображалися тільки ті команди і кнопки, які використовуються найчастіше. Налаштування проводиться автоматично, відстежуючи закономірності використання команд і вносячи відповідні зміни до складу меню і панелей інструментів. При цьому до решти можливостей завжди можна дістати доступ.

Деякі дії у всіх пакетах MS Office несуть одне і те ж навантаження і виконуються аналогічно. Загальним принципом у роботі з офісними програмами є також можливість використання довідки. Щоб отримати довідку, відповідну ситуації (контекстну довідку), в якій програма перебуває в даний момент, необхідно натиснути на клавішу [F1] або на кнопку виклику помічника.

Використання єдиних форматів файлів, а також загальні принципи роботи, закладені в застосуваннях MS Office, дозволяють спільно використовувати дані, створені різними програмами. Рішення про те, який саме спосіб потрібно використовувати в тому або іншому випадку, залежить від поставленого перед користувачем завдання.

Якщо потрібно скопіювати або перемістити частину інформації з одного файлу в інший (незалежно від додатка, в якому вони створені), то доцільно скористатися способом *копіювання (переміщення) в буфер обміну* і вставки з нього даних у вказане місце.

З метою використання файлу, створеного в одному з додатків Office, застосовуються способи імпорту-експорту або вставки файлу. Якщо при цьому необхідне оновлення копії у разі зміни початкових даних, то інформація вставляється як *зв'язаний об'єкт.*

За необхідності створити посилання на дані, розташовані в одному документі або в різних файлах, використовують інструмент *гіперпосилання*.

У поточний документ за допомогою засобів вбудовування можна включити інформацію, створену в інших додатках, наприклад графіку, електронні таблиці, діаграми. Вбудовані об'єкти стають частиною документа, проте редагування вбудованого об'єкта проводиться в тому додатку, в якому він був створений.

Для можливості сумісного використання даних різними користувачами їх можна розмістити в загальній папці на сервері Microsoft Exchange, а також розіслати по електронній пошті.

Office дозволяє організувати співпрацю в інформаційних мережах. Якщо необхідно сумісне обговорення і робота з файлами декількох учасників, можна призначити і провести збори або вебобговорення за допомогою сервісної функції *Спільна робота*.

2.3. Класифікація та структура операційних систем

Операційна система становить основу програмного забезпечення ПК. Операційна система являє собою комплекс системних і службових програмних засобів, який управляє роботою комп'ютера, координує взаємодію між окремими частинами, забезпечує взаємодію користувача з комп'ютером і виконання всіх інших програм.

З одного боку, вона спирається на базове програмне забезпечення ПК, що входить у його систему BIOS, з іншого – вона сама є опорою для програмного забезпечення вищих рівнів – прикладних і більшості службових застосувань.

Для того щоб комп'ютер міг працювати, на його жорсткому диску має бути встановлена (записана) операційна система. Під час включення комп'ютера вона прочитується з дискової пам'яті і розміщується в ОЗУ. Цей процес називається завантаженням операційної системи.

Операційні системи розрізняються особливостями реалізації алгоритмів управління ресурсами комп'ютера, областями використання.

Так, залежно від алгоритму управління процесором операційні системи розподіляються на:

- однозадачні і багатозадачні;
- розраховані на одного користувача і розраховані на багато користувачів;
- однопроцесорні і багатопроцесорні системи;
- локальні і мережеві.

За числом одночасно виконуваних завдань операційні системи діляться на два класи:

1) однозадачні (MS DOS);
2) багатозадачні (OS/2, Unix, Linux, Windows).

У однозадачних системах використовуються засоби управління периферійними пристроями, засоби управління файлами, засоби спілкування з користувачами. Багатозадачні ОС використовують всі засоби, які характерні для однозадачних, і, крім того, управляють розділенням спільно використовуваних ресурсів: процесор, ОЗУ, файли і зовнішні пристрої.

Залежно від областей використання багатозадачні ОС підрозділяються на три типи:

1) системи пакетної обробки (ОС ЄС);
2) системи з розділенням часу (Unix, Linux, Windows);
3) системи реального часу (RT11).

Системи пакетної обробки призначені для вирішення завдань, які не вимагають швидкого здобуття результатів. Основною метою ОС пакетної обробки є максимальна пропускна спроможність або вирішення максимального числа завдань за одиницю часу.

Ці системи забезпечують високу продуктивність під час обробки великих об’ємів інформації, але знижують ефективність роботи користувача в інтерактивному режимі.

У системах з розділенням часу для виконання кожного завдання виділяється невеликий проміжок часу, і жодне завдання не займає процесор надовго. Якщо цей проміжок часу вибраний мінімальним, то створюється видимість одночасного виконання декількох завдань. Ці системи володіють меншою пропускною спроможністю, але забезпечують високу ефективність роботи користувача в інтерактивному режимі.

Системи реального часу використовуються для управління технологічним процесом або технічним об’єктом, наприклад, літальним об’єктом, верстатом і так далі.

За числом одночасно працюючих користувачів на ЕОМ ОС розділяються на розрахованих (MS DOS) на одного користувача і розрахованих на багато користувачів (Unix, Linux, Windows 95 – XP, Windows Vista, Windows 7, Windows 8, Windows 10).

У розрахованих на багато користувачів ОС кожен користувач набудовує для себе інтерфейс користувача, тобто може створити власні набори ярликів, групи програм, задати індивідуальну колірну схему, перемістити в зручне місце панель завдань і додати в меню *Пуск* нові пункти.

У розрахованих на багато користувачів ОС існують засоби захисту інформації кожного користувача від несанкціонованого доступу інших користувачів.

Багатопроцесорні і однопроцесорні операційні системи. Однією з важливих властивостей ОС є наявність у ній засобів підтримки багатопроцесорної обробки даних. Такі засоби існують в OS/2, Net Ware, Widows NT. За способом організації обчислювального процесу ці ОС можуть бути розділені на асиметричних і симетричних.

Однією з найважливіших ознак класифікації ЕОМ є розділення їх на локальні і мережеві. Локальні ОС застосовуються на автономних ПК або ПК, які використовуються в комп'ютерних мережах як клієнт.

До складу локальних ОС входить клієнтська частина ПО для доступу до видалених ресурсів і послуг. Мережеві ОС призначені для управління ресурсами ПК, включених у мережу з метою спільного використання ресурсів. Вони представляють потужні засоби розмежування доступу до інформації, її цілісності й інші можливості використання мережевих ресурсів.

Існує кілька класів ОС:

- DOS: MS – DOS, PC DOS, Novell DOS. оболонки Norton Commander, Xtree Pro Gold- ЕОМ;
- Windows: Windows 3.1 або 3.11, Windows 95, 98, NT, 2000, XP, Millenium, оболонки Norton Navigator – ПК, Windows Vista, Windows 7, Windows 8, Windows 10;
- Unix, Windows NT Server, NetWare, OS/2 – сервери локальних мереж.

Операційні системи класу UNIX: FreeBSD, Linux – це операційні системи типу UNIX. Вони працюють на комп'ютерах Intel x86 і Alpha. Добровольці з усього світу трудяться над їх удосконаленням і вихідний код цих систем доступний кожному:

- системами *FreeBSD* і *Linux* користуються найвідоміші компанії світу;

- системи надзвичайно стабільні;
- системи прості у роботі;
- системи відкриті.

Конструктивно операційна система *Unix* базується на безлічі невеликих програм, кожна з яких вирішує визначену задачу. Ці маленькі програми, за необхідності, можна об'єднати в єдине ціле для розв'язування задач будь-якої складності.

Відкритість системи означає, що всім користувачам надається вихідний текст абсолютно всіх програм, який можна змінити і виконати компіляцію як окремого модуля, так і всієї системи в цілому. Користувач (root) має можливість включати підтримку тільки тих пристроїв, що реально застосовуються на цьому комп'ютері, що зменшує необхідні ресурси комп'ютера і тим самим підвищує його продуктивність.

Відкритий вихідний код – це відкрита політика системи, яку може перевірити кожен, знайти і виправити помилки і повідомити про них інших. Саме така політика дозволяє створити операційну систему, здатну задовольнити професійних користувачів.

Успіх операційної системи залежить від її стабільності, можливості забезпечувати надійний захист документів користувачів. Така стійкість пояснюється вдало вибраною філософією системи, ключові елементи якої полягають у такому:

- наявність простих команд;
- можливість об'єднання команд у модулі (конвеєри);
- відкритість вихідного коду;
- простота стилю інтерфейсу;
- відсутність типів файлів.

Усе це сприяє подальшому розвитку цієї системи.

Текстовий інтерфейс UNIX має ряд недоліків, які відсутні у системах з графічним інтерфейсом. Програмісти всього світу створили графічний інтерфейс, що не поступається можливостям Windows.

Однією з найбільш популярних графічних оболонок є KDE, яка надає користувачу зручний інтерфейс для виконання різноманітних операцій над файлами і каталогами. Вона містить вбудовані текстовий і графічний редактори, утиліти для підтримки

файлової системи, архіватор, програми для управління мультимедійними пристроями. Все це сприяє просуванню операційної системи UNIX на ринок малих комп'ютерів для роботи у малому офісі і дома.

Не менш популярною графічною оболонкою можна вважати і Gnome. Ця оболонка незмінно поставляється з основним пакетом системи і, маючи простий інтерфейс, вона здатна виконувати всі необхідні дії над файлами і каталогами.

Але все одно стандартним інтерфейсом UNIX-систем залишається текстовий інтерфейс, у якому управління усіма процесами операційної системи виконується введенням різних команд у командному рядку.

Особливість UNIX полягає в тому, що параметри системи зберігаються у вигляді текстових файлів, які можна змінити як за допомогою найпростішого текстового редактора, так і за допомогою складних графічних програм (на відміну від Windows, де налаштування системи можна проводити тільки за допомогою спеціалізованих графічних програм).

Windows Vista – наступна версія операційної системи *Windows* фірми Microsoft що продовжує лінію Windows XP. Раніше вона була відома під кодовим ім'ям Longhorn (по імені бару Longhorn Saloon поблизу лижного курорту Whistler в Британській Колумбії). Назва «Vista» була обнародувана 22 липня 2005 року. Назва «Longhorn» збережеться за Windows Longhorn Server, що продовжує гілку Windows Server 2003. Випуск Vista було заплановано на другу половину року, але ця дата випуску (листопад 2006) – тільки для корпоративних користувачів. Для широкої публіки Vista стане доступна в січні 2007 року. Ці затримки Microsoft пояснює хронічним недоліком часу на збільшення безпеки нової ОС.

У перекладі на російську *Vista* означає «нові можливості», «перспективи, що відкриваються». Назва нової ОС вибиралася ретельно і послідовно. У результаті проведених досліджень назва була визначена і схвалена співробітниками Microsoft. Після цього Vista було протестовано в декількох регіонах світу методом fokus-grupp і «глибинних» інтерв'ю.

У лінії продуктів Windows NT, Windows Vista носитиме номер версії 6.0 (Windows 2000 – 5.0, Windows XP – 5.1, Windows Server 2003 – 5.2). Для позначення Windows Vista іноді використовують абревіатуру «WINVI», яка об'єднує і назву «Vista» і номер версії, записаний римськими цифрами.

Нова версія Windows забезпечує підвищену безпеку і надійність даних (вбудований захист від вірусів, гнучкіша настройка прав користувачів, шифрування важливих даних, контроль за роботою критично важливих служб ОС), легку роботу з інформацією і сумісність зі всіма сучасними засобами комунікації, спрощене управління і установку ОС на декілька комп'ютерів, оновлений інтерфейс, майстра синхронізації з мобільними пристроями і ін. Підвищена, порівняно з Windows XP, швидкодія, яка досягається декількома шляхами. Це і оновлена підсистема управління пам'яттю і введенням-виведенням, і можливість автозапуску служб і програм у фоновому режимі. З'явився в Windows Vista і новий сплячий режим, названий Sleep Mode, що поєднує в собі кращі риси режимів Standby (швидке «пробудження») і Hibernate (високе збереження даних). На думку фахівців Microsoft, назва Windows Vista покликана відобразити нові можливості операційної системи і її переваги для користувачів.

Фінальна версія Windows Vista існує як у 64-розрядному так і в 32-розрядному варіантах. Windows Vista має також новий логотип. Цей логотип графічно передає зміни в сприйнятті нової операційної системи і краще відображає її зовнішність і суть, зокрема новий «скляний» користувацький інтерфейс.

На сьогодні операційна система *Windows 10* є найновішою операційною системою сімейства Windows.

Однією з ключових особливостей «десятки» є голосова помічниця з елементами штучного інтелекту Cortana, яка істотно полегшує життя, виконуючи різні доручення типу планування зустрічей, пошуку потрібних файлів та інформації або бронювання столика в ресторані. Проте, на жаль, в Україні вона поки що недоступна. У Windows 10 також з'явився новий інтернет-браузер Microsoft Edge, що прийшов на зміну Internet Explorer.

Microsoft кілька місяців відточувала інтерфейс Windows 10 і їй вдалося домогтися цілісного сприйняття від операційної системи. Здається, що дизайнери компанії не упустили жодної дрібниці:

всі іконки, значки і найдрібніші елементи були намальовані наново. Акцентні кольори для меню і вікон обираються операційною системою самостійно залежно від встановлених шпалер. Але за бажання можна самому обрати будь-який інший колір.

Стандартні значки в області повідомлень приведені до загального дизайну – вони білі та схематично відображають те, за що відповідають. У налаштуваннях Wi-Fi можна швидко перемикатися між різними мережами, а у разі натискання на дату відкривається мініатюрний календар із напівпрозорим фоном. У планшетному режимі екран можна розділити на дві частини. Деякі програми підтримують масштабування вікон (рис. 2.3).

Провідник став більш функціональним: з'явилася панель швидкого доступу, на яку можна приколоти ті папки, які потрібні частіше за інші. Також з'явився прямий доступ до файлів, які лежать у сховищі OneDrive. Для цього достатньо авторизуватися і вже можна ділитися файлами та надавати спільний доступ.



Рис. 2.3. Робочий стіл Windows 10

Пошук у Windows 10 став ближчим – не в меню *Пуск*, а в панелі завдань, завжди під рукою. Він універсальний, шукає інформацію скрізь: у локальній пам'яті, хмарі, магазині додатків та інтернеті. В американській версії операційної системи доступна голосова помічниця Cortana. Вона доповнює пошук і дозволяє виконувати деякі дії, наприклад, створювати нагадування. Персональна помічниця покликана передбачати потреби користувача. Їй можна дати доступ до

ваших особистих даних, таких як електронна пошта, адресна книга, історія пошуків у мережі тощо. Усі ці дані вона буде використовувати для задоволення ваших потреб. Cortana замінить стандартну пошукову систему і буде викликатися натисканням кнопки *Пошук*.

Переваги Windows 10:

- Windows 10 доступний кожному, у кого є ліцензійний Windows 7 або 8/8.1 з найостаннішим пакетом оновлень;
- новий інтерфейс значків;
- оновлене меню *Пуск*;
- оновлення – в ОС Windows 10 включено автоматичне завантаження та встановлення оновлень. Тепер відмовитися від них вже буде неможливо;
- нова система безпеки, тобто паролі вже не підходять для входу в ПК, тепер це перевірка відбитка пальця, оболонки ока, а також рис обличчя;
- стабільна робота – ОС Windows 10 працює набагато швидше, збільшилася кількість драйверів;
- браузер Microsoft Edge – новий, універсальний та інтелектуальний браузер, створений спеціально для ОС Windows 10;
- інтелектуальні функції: *Cortana* – віртуальний помічник, який розрізняє ваші вимоги «на слух».

Недоліки Windows 10:

- обмежена безкоштовність;
- неможна відключити оновлення;
- несумісність більшості ігор з новою ОС;
- з приводу паролів – такі паролі діють лише за наявності спеціального обладнання;
- відсутність медіаплеєра Windows Media Center;
- оскільки нова версія – можливість нових глюків, багів і всього іншого;
- прискорена розрядка деяких ПК (ноутбуків);
- помічник Cortana не підтримує російську та українську мови, і в наступних оновленнях ОС не поставлена мета це змінити.

Зазвичай, на перший погляд, Windows 10 виглядає красивою і завершеною системою. Але якщо дивитися об’єктивно, то це тільки перші версії, краще почекати оновлень і виправлення недоліків.

2.4. Організація та робота з об'єктами файлової системи ОС MS Windows

Усі сучасні операційні системи забезпечують створення файлової системи, призначеної для зберігання даних на дисках та забезпечення доступу до них. При цьому всі дані зберігаються у файлах.

Файлова система – це система правил, згідно з якими операційна система забезпечує зберігання даних на диску. Ці правила визначають формат зберігання даних, їх розміщення на диску, адресацію й ідентифікацію.

Основним елементом файлової системи є *файл* – іменована область даних на диску, що містить логічно об'єднану інформацію. Кожному файлу дається *ім'я* на основі правил, що визначають файлову систему. Ім'я файлу складається з двох частин, розділених точкою, власне ім'я, точка і розширення файлу. Ім'я в ОС Windows може мати довжину до 255 символів, а саме: великі і малі літери алфавіту, цифри, пробіли, спеціальні символи. Не можна використовувати в іменах файлів такі символи: *:; ? / \ > < “.

Розширення вказує на те, в якій програмі було створено файл, наприклад:

Таблиця 2.2

Типи файлів

Розширення	Тип файлу	Значок	Приклади імен файлів
1	2	3	4
.SYS	системний		Io.sys
.EXE	виконуваний		MsPaint.exe
.ZIP	архівований		Архів.zip
.HLP	текстовий		Windows.hlp
.HTM	гіпертекст		Сайт.htm
.TXT	текстовий		Лист.txt
.PDF	текстовий		Файл.pdf
.JKS	цифровий підпис		pb_2673705626.jks
.MPG	відео та звук		Фільм.mpg
.WAV	музичний запис		Привітання.wav

Продовження таблиці 2.2

1	2	3	4
.CDR	графічний		Портрет.cdr
.CPT	графічний		Foto.cpt
.DOC,.DOCX	текстовий		Документ.doc
.XLS,.XLSX	електронна таблиця		Книга.xls
.PPT,.PPTX	презентація		Моя родина.ppt
.MDB,.ACCDB	база даних		Борей.mdb
.DOT	шаблон документа Word		Шаблон.dot

Файл також характеризується об’ємом і місцезнаходженням на диску (шлях).

Для зручності файли об’єднують у *каталоги*. Каталоги можуть містити не тільки файли, а й інші каталоги, що називаються *підкаталогами*.

Графічно подати файлову систему можна у вигляді *дерева каталогів***,** коренем дерева каталогів є каталог, що дав початок всім іншим. Його називають *кореневим каталогом***,** потім ідуть каталоги *першого рівня, другого рівня* тощо.

Принципи організації файлової системи залежать від операційної системи. Найбільш розповсюджений тип – табличний. Диск являє собою набір поверхонь. Кожна поверхня диска розділяється на кільцеві доріжки, а кожна доріжка – на сектори.

Розміри секторів фіксовані і дорівнюють 512 байт.

Щоб знайти на диску будь-який файл, потрібно знати, де він розміщений, тобто його адресу. Але в зв’язку з тим, що головки для зчитування переміщуються всі одночасно, то замість поняття доріжки використовують поняття *циліндра*.

Циліндр – сукупність усіх доріжок, які мають однакові номери, тобто рівновіддалені від осі обертання. Тому реально місцезнаходження файлу на жорсткому диску визначається номером циліндру, номером поверхні та номером сектора.

Сектор – це найменша одиниця зберігання даних, але для адресації вона використовується не у всіх ФС. Такі ОС, як MS-DOS, Windows, OS/2 – використовують більшу одиницю зберігання – кластер.

Кластер – це група сусідніх секторів. Розмір кластера залежить від розміру жорсткого диску. Типові значення: 8,16, 32, 64 сектора.

Дані про те, в якому кластері диску починається той чи інший файл, зберігаються в системній області диску в спеціальних *таблицях розміщення файлів* (FAT-таблицях). Оскільки порушення FAT-таблиці унеможливлює зчитування даних з диску, то до неї ставляться особливі вимоги надійності і вона існує у двох екземплярах, ідентичність яких перевіряється засобами ОС.

ФС FAT32 підтримує менші розміри кластерів, що дозволяє більш ефективно використовувати дисковий простір. Вона має 32-розрядні поля в таблиці розміщення файлів, що дозволяє розмістити 2^{32} записів· на відміну від FAT16 – $2^{16} = 65\,536$ записів.

ФС NTFS – це покращена ФС, яка забезпечує вищий рівень швидкодії і безпеки, а також додаткові можливості, яких не було в попередній ФС. У випадку збою роботи ПК цілісність файлової системи відновлюється за допомогою файлу журналу NTFS і даних про контрольні точки. В ОС WIN2000 і Win XP файлова система NTFS забезпечує також і такі додаткові можливості, як дозволи для файлів та папок (залежно від типу облікового запису – адміністратор, обмежені права, стандартні права), шифрування, дискові квоти і стиснення.

Наприклад, за допомогою команди *convert D: /fs:ntfs* можна виконати перетворення диску D у формат NTFS.

Під управлінням ОС здійснюються такі операції з обслуговування ФС:

- створення файлів і присвоювання їм імен;
- створення каталогів (папок) і присвоювання їм імен;
- перейменовування файлів і каталогів;
- копіювання і переміщення файлів;
- видалення файлів і каталогів;
- навігація по файловій структурі;
- управління атрибутами файлів.

2.5. Інформаційна безпека, основи захисту інформації

Побудувавши своє життя на роботі з інформацією, людина усвідомила, що завжди існує небезпека, що ще хтось може отримати до неї доступ. Тобто дізнатися про її зміст або внести якінебудь зміни. І першими скористалися з цього шахраї. Сьогодні відомо тисячі випадків шахрайства з банківськими картками. Але найбільші оператори карткового ринку – Visa, American Express і MasterCard – працюють над покращенням стандарту забезпечення захисту у мережі «Інтернет».

Крім злочинців, є ще й держава, яка теж хоче знати про громадян значно більше, ніж ті хочуть розповісти. У випадку боротьби з правопорушниками ці заходи виправдані, але таким чином може бути організоване спостереження за всіма без виключення. Існує апаратура для читання всіх листів, відправлених електронною поштою, чи визначити, хто, коли, на яких сайтах побував. Мобільні можна прослуховувати. Електронні платіжні документи – відслідковувати, тобто можна контролювати практично всю інформацію, що оточує людину. Інакше кажучи, може контролюватися все життя людини.

Отже, ІТ несуть як очевидні переваги, так і потенційну небезпеку.

Оптимальне співвідношення між загальним порядком і особистою свободою людство шукає давно. Розвиток ІТ веде як до збільшення свободи кожної людини, так і до жорстких її обмежень. Практично ми переходимо у нову еру, яка суттєво змінить наше існування. Поки що кожне окреме нововведення майже не змінює наше життя: безготівковий розрахунок, персональна карта замість ключа, автоматичний контрольно-пропускний пункт, переведення зарплати і пенсії на рахунок банку тощо. Але як тільки ці окремі технології будуть об’єднані в єдину систему, ми опинимося у якісно новому світі, який поставить ряд питань, на які повинні підготувати відповіді вже сьогодні:

- чи готові ми жити під постійним контролем;
- чи потрібно кому-небудь збирати, зберігати і опрацьовувати всі дані про нас;

- чи може хтось скористатися зібраною інформацією;
- чи готові увійти у друге коло несвободи;
- чи готові жити в оточенні цифр, які так легко запам'ятати?

Створення загального інформаційного простору, масове застосування персональних комп'ютерів та комп'ютерних систем викликали необхідність розв'язування комплексної проблеми захисту інформації.

Інформація є одним із найбільш цінних ресурсів будь-якої компанії, тому забезпечення захисту інформації є одною з найважливіших і пріоритетних завдань.

Безпека інформаційної системи – це здатність системи забезпечити її нормальне функціонування, тобто забезпечення цілісності та секретності інформації. Для забезпечення цілісності і конфіденційності інформації необхідно забезпечити захист інформації від випадкового знищення або несанкціонованого доступу до неї.

Під цілісністю розуміють неможливість несанкціонованого або випадкового знищення, а також модифікації інформації.

Під конфіденційністю інформації – неможливість витоку і несанкціонованого заволодіння інформацією, що зберігається, передається чи приймається.

Відомі такі джерела загроз безпеки інформаційних систем:

- антропогенні, викликані випадковими чи умисними діями суб'єктів;
- техногенні, що призводять до відмов і збоїв технічних і програмних засобів, викликаних застарілими програмними і апаратними засобами або помилками ПЗ;
- стихійні, викликані природними катаклізмами чи форс-мажорними обставинами.

У свою чергу, антропогенні джерела небезпек діляться на:

- внутрішні (дії з боку співробітників компанії) і зовнішні (несанкціоноване втручання сторонніх осіб із зовнішніх мереж загального призначення);
- ненавмисні (випадкові) і умисні дії суб'єктів.

Існує достатньо багато можливих напрямів витоку інформації і шляхів несанкціонованого доступу до неї в системах і мережах:

- перехоплення інформації;
- дистанційне фотографування;
- перехоплення акустичних хвильових випромінювань;
- зчитування інформації з масивів інших користувачів;
- модифікація інформації (вихідне повідомлення чи документ змінюється і відсилається адресату);
- підміна авторства інформації (хтось може надіслати листа чи документ від вашого імені);
- використання недоліків операційних систем і прикладних програмних засобів;
- копіювання носіїв інформації і файлів з визначенням засобів захисту;
- незаконне підключення до апаратури і ліній зв'язку;
- маскування під зареєстрованого користувача і присвоєння його повноважень;
- введення нових користувачів;
- зараження комп'ютерними вірусами тощо.

Об'єктами захисту інформації в системах обробки даних можна виділити такі:

- ПК користувачів;
- ПК адміністратора мережі;
- вузол зв'язку;
- засоби відображення інформації;
- засоби документування інформації;
- комп'ютерний зал і сховище носіїв інформації;
- зовнішні канали зв'язку і мережеве обладнання;
- накопичувачі і носії інформації.

Елементами захисту є блоки інформації в об'єктах захисту, тобто:

- дані і програми в основній пам'яті ПК;
- дані і програми на зовнішніх носіях;
- пакети даних, що передаються по каналах зв'язку;

• журнали призначення паролів та пріоритетів зареєстрованим користувачам;

• службові інструкції;

• архіви даних та ПЗ тощо.

Засоби розпізнавання та розмежування доступу до інформації. Для забезпечення безпеки інформаційних систем застосовують системи захисту інформації, які являють собою комплекс організаційно-технологічних заходів, програмно-технічних засобів та правових норм, направлених на протидію джерелам загрози безпеки інформації.

Функції захисту інформації реалізують:

• захист інформації від пошкодження з боку інших програм;

• захист інформації від руйнування у разі збоїв у роботі комп'ютера або відключення його від мережі;

• захист інформації від руйнування у разі помилкових дій користувача;

• захист від несанкціонованого доступу на рівні бази даних і всередині програми для користувачів (обмеження доступу на рівні папок документів, кореспондентів, об'єктів обліку, замовлень, документів).

Необхідно зазначити, що будь-яка система захисту інформації не є повністю безпечною. Завжди потрібно вибирати між рівнем захисту і ефективністю роботи інформаційних систем.

До засобів захисту інформації ІС від дій суб'єктів відносять:

• засоби захисту інформації від несанкціонованого доступу;

• захист інформації в комп'ютерних мережах;

• криптографічний метод захисту інформації;

• електронний цифровий підпис;

• захист інформації від комп'ютерних вірусів.

Одержання доступу до ресурсів інформаційної системи передбачає виконання трьох процедур: ідентифікація, аутентифікація і авторизація.

Ідентифікація – присвоєння користувачу унікального імені і коду (ідентифікаторів).

Автентифікація – встановлення відповідності (справжності) користувача присвоєному ідентифікатору чи перевірка того, що особа чи пристрій, що повідомили ідентифікатор, є дійсно тим, за кого себе видає. Найбільш розповсюдженим способом автентифікації є присвоєння користувачу пароля і зберігання його в комп'ютері.

Багатофакторна автентифікація – розширена автентифікація, метод контролю доступу до комп'ютера, в якому користувачеві для отримання доступу до інформації необхідно пред'явити більше одного «доказу механізму автентифікації». До категорій таких доказів належать:

- знання – інформація, яку знає суб'єкт. Наприклад, пароль, пін-код;
- володіння – річ, якою володіє суб'єкт. Наприклад, електронна або магнітна карта, токен, флеш-пам'ять;
- властивість, якою володіє суб'єкт. Наприклад, біометрія, унікальні природні відмінності: обличчя, відбитки пальців, райдужна оболонка очей, капілярні візерунки, послідовність ДНК.

Авторизація – перевірка повноважень чи перевірка права користувача на доступ до конкретних ресурсів і виконання визначених операцій над ними. Авторизація проводиться з метою розмежування прав доступу до мережевих і комп'ютерних ресурсів.

Локальні мережі підприємств досить часто підключаються до мережі «Інтернет». Для захисту локальних мереж компаній, як правило, використовують міжмережеві екрани – брандмауери (firewalls), які дозволяють розділити мережу на дві частини (межа проходить між локальною мережею і мережею «Інтернет») і сформувати набір правил, що визначають умови проходження пакетів із однієї частини в іншу. Екрани можуть бути реалізовані як апаратними засобами, так і програмними.

Криптографічний метод захисту інформації. Для забезпечення секретності інформації застосовують її шифрування або криптографію. Для шифрування використовується алгоритм чи пристрій, який реалізує визначений алгоритм. Управління шифруванням здійснюється за допомогою змінюваного коду ключа.

Добути зашифровану інформацію можна тільки за допомогою ключа. Криптографія – це досить ефективний метод, який підвищує безпеку передавання даних у комп'ютерних мережах і при обміні інформацією між віддаленими комп'ютерами.

Електронний цифровий підпис. Для виключення можливості модифікації вихідного повідомлення або підміни цього повідомлення іншим необхідно передавати повідомлення разом з електронним підписом. Електронний цифровий підпис – це послідовність символів, одержана в результаті криптографічного перетворення вихідного повідомлення з використанням закритого ключа, що дозволяє визначати цілісність повідомлення та належність його автору за допомогою відкритого ключа.

Повідомлення, зашифроване за допомогою закритого ключа, називається електронним цифровим підписом. Відправник передає незашифроване повідомлення у вихідному вигляді разом з цифровим підписом. Одержувач за допомогою відкритого ключа розшифровує набір символів повідомлення з цифрового підпису і порівнює їх з набором символів незашифрованого повідомлення. При повному збігу символів можна стверджувати, що одержане повідомлення не модифіковане і належить його автору.

Комп'ютерні віруси і сучасні антивірусні програмні засоби. Віруси розрізняються за такими чинниками:

Середовище перебування. Тут віруси поділяють на:

- файлові – ті, що додаються до файлів з розширенням exe, com;
- завантажувальні – ті, що додаються до boot-сектора;
- мережеві – ті, що поширюються по комп'ютерній мережі;
- макровіруси – ті, що заражають файли Microsoft Office. Вони пошкоджують копію шаблону Normal.dot, який завантажується в оперативну пам'ять комп'ютера під час роботи і внаслідок чого всі файли, з якими проводиться робота, стають ураженими.

Способи зараження комп'ютера. Тут існують віруси:

- резидентні – ті, що вміщуються в оперативну пам'ять і додаються до всіх об'єктів (файлів, дисків), до яких звертається ОС;
- нерезидентні – ті, що додаються до оперативної пам'яті і є активними лише короткий час.

Функціональні можливості. Тут є такі групи вірусів:

• нешкідливі – ті, що не впливають на роботу комп'ютера (наприклад, збільшують розмір файлу);

• безпечні – ті, що заважають роботі, але не пошкоджують інформацію (наприклад, дають якісь повідомлення, перезавантажують комп'ютер тощо);

• небезпечні – ті, що пошкоджують інформацію;

• файлів, зумовлюючи «зависання» комп'ютера;

• дуже небезпечні – ті, що зумовлюють утрату програм, знищення інформації із системних областей, форматування жорсткого диска.

Особливості алгоритму. За цією ознакою віруси поділяють на такі групи;

• віруси-супутники – віруси, які не змінюють файли, але створюють однойменні файли з розширенням com, що завантажуються першими;

• віруси-черв'яки – віруси, що поширюються автоматично в комп'ютерній мережі за знайденою адресою в адресній книзі;

• віруси-паразити – віруси, які розпізнаються за зміненим змістом дискових секторів і файлів;

• Stealth-віруси – ті, що фальсифікують інформацію, яка читається з диска. Вірус перехоплює вектор переривання int 13h і видає активній програмі хибну інформацію, яка показує, що на диску все гаразд. Цей засіб використовується як у файлових, так і в завантажувальних вірусах;

• віруси-мутанти – віруси, що мають зашифрований програмний код;

• ретровіруси – звичайні файлові віруси, які намагаються заразити антивірусні програми, щоб знищити їх або зробити недієздатними.

Антивірусні програми, що дають змогу виявити вірус, відкоректувати або вилучити пошкоджені файли, поділяються на детектори, фаги (лікарі), ревізори, сторожі, вакцини.

Детектори (сканери) перевіряють оперативну або зовнішню пам'ять на наявність вірусу за допомогою розрахованої контрольної суми або сигнатури (частина коду, що повторюється) і складають

список ушкоджених програм. Якщо детектор – резидентний, то програма перевіряється, і тільки в разі відсутності вірусів вона активізується. Детектором є, наприклад, програма MS AntiVirus.

Фаги (поліфаги) – виявляють та знешкоджують вірус (фаг) або кілька вірусів. Сучасні версії поліфагів, як правило, можуть проводити евристичний аналіз файлу, досліджуючи його на наявність коду, характерного для вірусу (додання частини цієї програми в іншу, шифрування коду тощо). Фатами є, наприклад, програми Aidstest, DrWeb.

Ревізори – програми, що контролюють можливі засоби зараження комп'ютера, тобто вони можуть виявити вірус, невідомий програмі. Ці програми перевіряють стан BOOT-сектора, FAT-таблиці, атрибути файлів (ємність, час створення тощо). У разі виявлення будь-яких змін користувачеві видається повідомлення (навіть у разі відсутності вірусів, але наявності змін). Ревізором є, наприклад, програма Adinf.

Сторожі – резидентні програми, які постійно зберігаються у пам'яті й у визначений користувачем час перевіряють оперативну пам'ять комп'ютера (включаючи додаткову та розширену), файли, завантажувальний сектор, FAT-таблицю. Сторожем є, наприклад, програма AVP, що може виявити понад 30 тис. вірусів.

Вакцини – програми, які використовуються для оброблення файлів та завантажувальних секторів з метою передчасного виявлення вірусів.

Заходи захисту інформації на електронних носіях. Резервування інформації. Основними способами резервування інформації є:

- її зберігання у захищених місцях (спеціальних приміщеннях, сейфах та ін.);
- зберігання інформації в територіально розподілених місцях.

Технічні заходи. Один із технічних заходів захисту інформації – використання безперебійних джерел живлення (UPS), які дають змогу коректно закінчити роботу і вийти із програми в разі перебою електропостачання. Ці пристрої залежно від складності задачі та потужності встановленого комп'ютерного обладнання можуть підтримувати роботу системи від 20 хв. до кількох годин. Більш надійна робота забезпечується у разі підключення до запасної енергопідстанції. На підприємствах, що мають неперервний

робочий цикл перероблення інформації (наприклад, головні, банки), потрібно використовувати власні енергогенератори.

Адміністративні заходи. Керівники інформаційних відділів повинні:

- чітко визначити функції всіх учасників інформаційного процесу;
- досліджувати й аналізувати ризики безпеки інформації;
- створити інструкції щодо дій персоналу в разі виникнення загроз безпеці інформації;
- мінімізувати ризик для тих, хто працює із важливою інформацією, та їх родин із метою запобігання їх викраденню та вимаганню інформації;
- визначити стратегію резервування, створити окрему інструкцію з резервування (наприклад, цю інформацію копіювати кожен день о 12 год). При цьому потрібно враховувати фізичне руйнування магнітних носіїв з часом. Копій має бути як мінімум дві, одна з яких зберігається у вогнетривкому сейфі біля комп'ютера, інша – якнайдалі від офісу (на випадок вибуху, пожежі, землетрусу).

Захист інформації від несанкціонованого доступу. Захистити інформацію від несанкціонованого доступу можна за допомогою апаратно-програмних, програмних, біометричних, адміністративних, технічних засобів.

Апаратно-програмні засоби. До них належать:

- спеціальні криптографічні плати, що вбудовуються в комп'ютер, за допомогою яких інформацію можна зашифрувати, створити електронний підпис, а також автентифікувати користувача (аутентифікація – процес ідентифікації користувачів, пристроїв або будь-якої іншої одиниці, що бере участь в інформаційному обміні, перед початком якого треба мати дозвіл на доступ до даних);
- SmartCard – магнітна картка для зберігання секретного ключа, шифрування паролей;
- пристрої ActivCard для введення паролей, де пароль не вводиться, а розраховується (динамічний пароль), а також SmartReader для зчитування паролей. У цих пристроях усередині

вмонтовано мікропроцесор, у пам'яті якого зберігається секретний код. Пароль, що вводиться користувачем (чотири цифри), в комп'ютері перераховується, тобто створюється спеціальний код.

Програмні заходи. Вони включають:

- вбудовані у програми функції захисту даних. Наприклад, система Netware після трьох спроб користувача ввійти в мережу з неправильним паролем блокує ідентифікатор цього користувача, і тільки адміністратор мережі має змогу розблокувати доступ;
- спеціальні криптографічні розробки.

За принципом побудови існуючі засоби захисту інформації, в яких використовуються криптографічні методи захисту, можна поділити на два типи:

1) засоби, в основі роботи яких лежать симетричні алгоритми для побудови ключової системи і системи автентифікації;

2) засоби, основу роботи яких складають асиметричні алгоритми, що застосовуються для тих самих цілей.

У засобах *першого типу* обов'язковою є наявність центру розподілу ключів, що відповідає за їх створення, розповсюдження та вилучення. При цьому носії ключової інформації передаються абонентам із використанням фізично захищених каналів зв'язку. Ключі мають змінюватися досить часто, кількість абонентів має бути значною, тому ці засоби негнучкі та дорогі. Питання автентифікації вирішується довір'ям користувачів один одному, цифровий підпис неможливий. Центр розподілу ключів контролює всю інформацію. Захист інформації досить низький.

У засобах *другого типу* ключі для шифрування автоматично генеруються, розповсюджуються і вилучаються для кожного сеансу зв'язку. Функції служби розповсюдження ключів виконує сертифікаційний центр, де користувач реєструється, встановлюється його автентифікація, після чого ключі вилучаються. У таких засобах можливими є організація цифрового підпису та його перевірка. Протокол установлення аутентичного зв'язку відповідає певному стандарту.

Автентифікація є простою та суворою. При простій автентифікації відбувається обмін паролями між абонентами, які встановили зв'язок, із подальшою перевіркою відповідності цих

паролів еталонним. При суворій автентифікації кожен абонент має два криптографічних ключі – секретний, відомий тільки цьому абоненту, та відкритий – той, що передається у банк. Використовуючи секретний ключ і спеціальний алгоритм, абонент формує цифровий підпис – послідовність бітів, яка однозначно відповідає документу, що підписується. Перевірка відповідності підпису виконується за допомогою відкритого ключа.

Біометричні засоби. До них належать:

- візерунки сітчатки ока;
- відбитки пальців;
- геометрія руки;
- динаміка підпису.

Адміністративні заходи. Вони включають:

- систему електронних перепусток для персоналу і відвідувачів;
- системи відеоспостереження та відеореєстрації, що дають змогу вести цілодобовий візуальний нагляд як за периметром об'єкта, так і всередині з можливістю запису інформації на відеомагнітофон або комп'ютер;
- розподіл доступу до інформації. Тут необхідним є чітке визначення осіб, які мають право на ту чи іншу інформацію. Наприклад, програмісти не повинні мати доступу до БД, а користувачі – до програмного забезпечення;
- систематичний аналіз мережевого протоколу роботи, блокування спроб введення паролів кілька разів;
- ретельний підбір співробітників, навчання, стажування, тренування. Кандидат повинен мати задовільні свідоцтва й атестати з попередніх робочих місць, не мати нахилу до зловживання наркотиками та алкоголем, не мати вагомих заборгованостей, не виявляти недоброзичливості до наймачів.

Технічні заходи. Їх можна поділити на такі групи:

1) заходи захисту від підслуховування, що включають:

- установлення фільтрів на лініях зв'язку;
- обстеження приміщень із метою виявлення підслуховувальних пристроїв;

- використання звукопоглинальних стін, стелі, підлоги;
- застосування систем віброакустичного й акустичного зашумлення для захисту мовної інформації від прослуховування за допомогою акустичних мікрофонів, стетоскопів, лазерних та інфрачервоних систем відбору інформації;

2) заходи захисту від електромагнітного випромінювання, до яких належать:

- використання оптико-волоконного кабелю;
- застосування захисної плівки на вікнах;
- користування захищеними дисплеями;

3) заходи захисту від поновлення вилучених даних.

Запитання для самоконтролю

1. Які принципи фон Неймана покладено в основу роботи сучасних комп'ютерів?
2. Які пристрої входять до апаратної складової сучасних комп'ютерів?
3. Які характеристики має процесор?
4. Які характеристики має пам'ять комп'ютера?
5. Яка структура головного вікна операційної системи MS Windows?
6. Які основні об'єкти робочого столу?
7. Яка структура головного меню операційної системи MS Windows?
8. Які операції можна виконувати над об'єктом ?
9. Які типи вікон використовуються у середовищі операційної системи MS Windows?
10. Що таке контекстне меню?
11. Яке призначення програми *Проводник*?
12. Як виконуються операції пошуку об'єктів і за якими критеріями?
13. Що означають шаблони Р*. ?уст та А*е у тексті для пошуку?
14. Яке призначення має програма *Калькулятор*?

15. Як здійснити запуск програми *Блокнот*?

16. Яку структуру має вікно програми *Paint*?

17. Яке призначення має програма *WordPad*, основні функції та можливості використання?

18. В чому суть принципів вбудовування та зв'язування об'єктів?

19. Які службові програми для роботи з дисками входять до складу ОС Windows?

20. Для чого потрібна дефрагментація диска?

21. Яке призначення системної службової програми *Управление дисками*?

22. Що таке архівований файл?

23. Для чого потрібне резервне копіювання даних?

24. Для чого проводять архівацію файлів?

25. Який принцип стиснення інформації при архівації файлів?

26. Що таке комп'ютерний вірус?

27. Як захистити комп'ютер від «зараження» комп'ютерним вірусом?

28. Як діє антивірусна програма?

ЛЕКЦІЯ 3
ПРОГРАМНІ ЗАСОБИ РОБОТИ З БАЗАМИ ТА СХОВИЩАМИ ДАНИХ УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСІВ

3.1. Поняття бази даних та бази знань

Поняття «база даних» (далі – БД) означає сукупність даних, призначених для спільного використання. База даних – сукупність спеціальним чином організованих даних і об'єктів, що відображають стан і їхні взаємозв'язки у предметній області. *Під предметною областю* прийнято розуміти частину реального світу, який підлягає вивченню для організації управління і в кінцевому рахунку автоматизації, наприклад, установа, вищий навчальний заклад, підприємство.

Розрізняють документальну БД (сукупність текстових документів вільних форматів) і фактографічну БД (сукупність відомостей, що зберігаються в інформаційній системі й організованих відповідно до вимог якого-небудь фіксованого набору форматів).

Дані, збережені в документальних і фактографічних БД, поміщаються в сховища, модифікуються і використовуються.

На всіх етапах життєвого циклу БД складається з двох компонентів: структури та даних. Під *структуризацією* розуміється введення угод про способи представлення даних. Неструктурованими можуть вважатися дані у текстовому файлі. Прикладом структурованих даних може виступати таблиця.

Логічну структуру даних, що зберігаються у базі даних, називають моделлю представлення даних. На різних рівнях абстракції опису БД її структура зображується по-різному, як показано на рисунку 3.1.

На концептуальному інфологічному рівні структура БД ефективно відображається у вигляді ER-діаграми – діаграми «Entity – Relationship» («Сутність – зв'язок»), запропонованої у 1976 р. П. Ченом; на концептуальному даталогічному рівні вона є або ієрархічною, або мережевою, або реляційною, або об'єктно-орієнтованою,

або об’єктно-реляційною, або багатовимірною, або гібридною моделлю даних. На фізичному рівні структура БД – це структура файлів даних і допоміжних файлів.

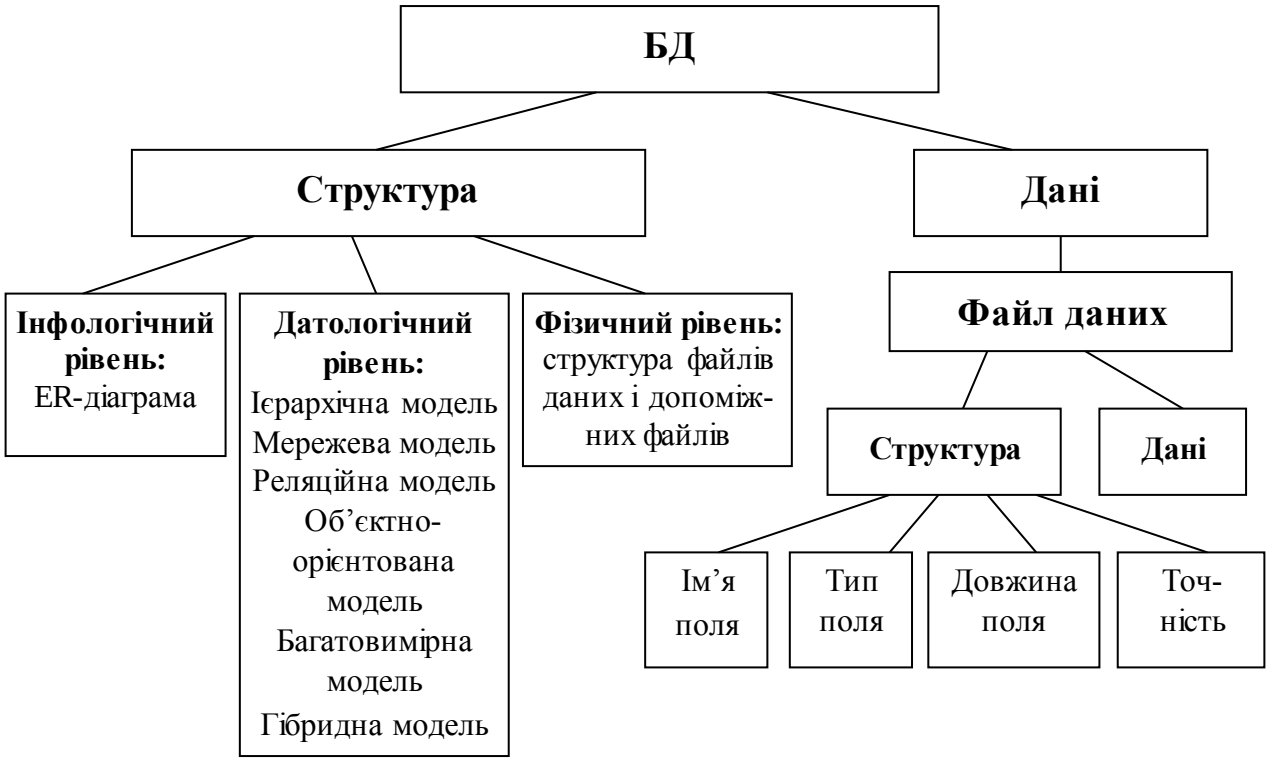


Рис. 3.1. Складові БД та файлу даних у реляційній БД

Самі файли даних також складаються з двох компонентів: структури та даних. Структура файлу в реляційній моделі – це ім’я, тип поля, його довжина, точність (для числових полів).

Розробленням структури БД на всіх етапах займається системний аналітик, який забезпечує БД усіма необхідними властивостями.

Дані – це компонент БД, яким маніпулює користувач ІС, актуалізуючи БД.

Дані сучасних БД поділяються на:

- алфавітно-цифрові (використовуються практично в усіх ІС);
- графічні образи (застосовуються в різних системах автоматизованого проєктування, так званих САПР);
- гіпертекстові, гіпермедійні, мультимедійні (використовуються в ІС на основі технологій Internet/Intra-net/Extranet);
- просторові або картографічні (застосовуються в геоінформаційних системах);
- ряди даних (як правило, хронологічні ряди, що використовуються у багатовимірних БД, які становлять основу сховищ даних).

База даних містить словник мета даних, що сам по собі є БД, де розміщені дані про дані, тобто інформація про саму БД, і є інструментом адміністратора БД. У словнику корисно мати такі дані: ім'я, тип, довжину, точність, допустимі значення кожного елемента даних; назви файлів даних і допоміжних файлів, що забезпечують доступ до даних; адреси цих файлів, кількість записів, які зберігаються (мають формуватися автоматично при роботі з даними); дати реєстрації змін у структурах даних (теж мають формуватися автоматично); дані про адміністраторів, які керують файлами, та іншу інформацію. Система управління БД разом з обчислювальною системою та словником БД є інструментальним засобом розроблення та супроводу БД.

Адміністратор БД (АБД) – особа або група осіб, які відповідають за загальне керування БД. До обов'язків адміністратора БД належать: вироблення вимог до БД, її проєктування, створення, ефективне використання і супровід, визначення інформаційного змісту БД, структури зберігання та стратегії доступу, взаємодія з користувачами, визначення контролю повноважень і процедур перевірки вірогідності даних, а також стратегії відновлення даних, керування ефективністю БД.

У процесі експлуатації АБД стежить за функціонуванням інформаційної системи, забезпечує захист від несанкціонованого доступу, контролює надмірність, несуперечність і вірогідність збереженої в БД інформації. Для однокористувальницьких інформаційних систем функції АБД, зазвичай, покладаються на осіб, що безпосередньо працюють з додатком БД.

В обчислювальній мережі АБД, як правило, взаємодіє з адміністратором мережі. До обов'язків останнього належать контроль за функціонуванням апаратно-програмних засобів мережі, реконфігурація мережі, відновлення програмного забезпечення після збоїв і відмовлень устаткування, профілактичні заходи і забезпечення розмежування доступу.

Обслуговуючий персонал виконує функції підтримки технічних і програмних засобів у працездатному стані. Він проводить профілактичні, регламентні, ремонтні й інші планові роботи, а також роботи за необхідності.

Додаток – це програма або комплекс програм, що забезпечують автоматизацію обробки інформації для прикладної задачі. Додатки можуть створюватися або у середовищі, або поза середовищем СУБД – за допомогою системи програмування, що використовує засоби доступу до БД, наприклад Delphi, C++, Builder. Додатки, розроблені в середовищі СУБД, часто називають додатками СУБД, а додатки, розроблені поза СУБД, – зовнішніми додатками.

Для роботи з базою даних найчастіше достатньо засобів СУБД, не потрібно використовувати додатки, створення яких вимагає програмування. Додатки розробляють в основному у випадках, коли потрібно забезпечити зручність роботи з БД некваліфікованим користувачам або коли інтерфейс СУБД не влаштовує користувачів.

База знань – упорядкована сукупність спеціальним чином організованих правил і фактів, що відображають їхній стан і взаємозв'язки у предметній області, механізмів виведення та програмних засобів, та призначена для подання нагромаджених у ній знань. База знань широко використовується в автоматизованих експертних системах для вирішення експертних завдань за допомогою правил логічного виведення.

Знання – це особлива форма інформації, яка являє собою сукупність структурованих теоретичних і емпіричних положень предметної області, які представлені у різній формі, мають певні властивості, пов'язані синтаксичними, семантичними, прагматичними відносинами і дозволяють розв'язувати прикладні задачі.

Знання існують у таких формах: у пам'яті людини (експерта); матеріалізовані або канонізовані знання у підручниках, наукових монографіях тощо та розташовані на різних матеріальних носіях; напівформалізована структурована модель у вигляді «поля знань»; формалізовані знання на деякій мові представлень і БЗ.

Процедура створення БЗ може мати такий вигляд (рис. 3.2):

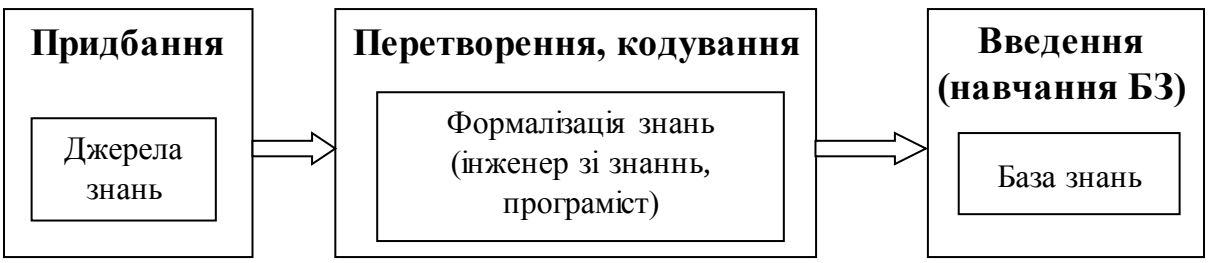


Рис. 3.2. Процедура створення бази даних

Сховище (склад, банк даних (БнД)) даних (англ. data warehouse) – предметно-орієнтований, інтегрований, незмінний набір даних, що підтримує хронологію і здатний бути комплексним джерелом достовірної інформації для оперативного аналізу та прийняття рішень.

Сховища даних можна представити так:

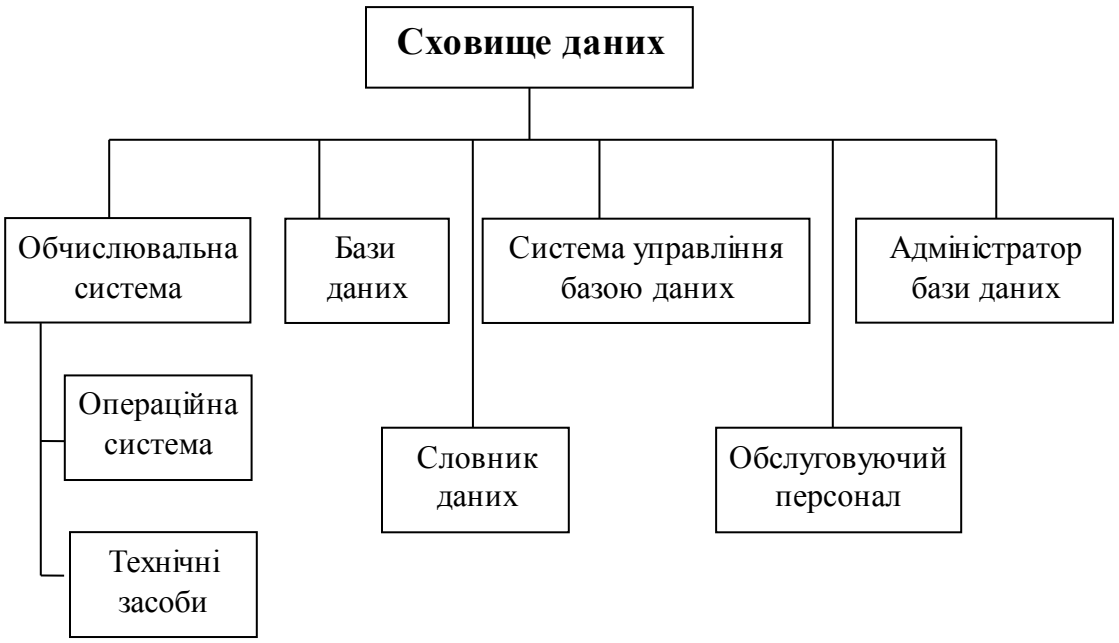


Рис. 3.3. Сховище даних

Описи баз даних належать до метаінформації, тобто інформації про інформацію. Опис баз даних часто називають *схемою*. Крім того, в сховищі даних можуть бути присутніми описи окремих частин баз даних, з погляду окремих користувачів – *підсхеми*.

Централізоване сховище метаінформації називається *словником даних*.

Крім опису баз даних, до складу метаінформації може включатися інформація про предметну область, яка необхідна для проєктування системи, про користувачів сховища даних, про проєктні рішення і деяка інша інформація.

3.2. Різновиди моделей даних. Типи зв'язків

До основних теоретичних знань з баз даних слід віднести визнання *типів* (видів) та *моделей подання даних*.

Характер зв'язків між елементами бази даних визначає тип організації даних (*модель подання даних*) – *структуру даних*.

Ядром будь-якої бази даних є модель даних. *Модель даних* (англ. Data model) – абстрактне представлення реального світу, що відображає тільки ті об'єкти, що безпосередньо стосуються проблеми, визначає специфічну групу об'єктів, їх атрибутивне значення і відношення між ними. Модель даних являє собою множину структур даних, обмежень цілісності і операцій маніпулювання даними. За допомогою моделі даних можуть бути представлені об'єкти предметної області і взаємозв'язку між ними. Модель даних – сукупність структур даних та операцій їх обробки. Модель даних – це формальний спосіб відображення взаємовідношень даних у системі управління базою даних.

Приведені вище визначення доповнюють одне одного. Важливо розуміти, що модель даних – це не самі дані, а дані про дані (метадані), що характеризують принцип організації одиниць інформації і припустимих операцій над ними.

До класичних моделей належать *ієрархічні, мережеві, реляційні* моделі подання даних.

Крім того, в останні роки з'явилися і стали більш активно впроваджуватися на практиці такі моделі даних: *постреляційна, багатовимірна, об'єктно-орієнтована*.

Розглянемо три основні типи моделей даних: ієрархічну, мережеву та реляційну моделі.

Спочатку використовувались ієрархічні моделі. Ієрархічна структура являє собою сукупність елементів, пов'язаних між собою за визначеними правилами. Об'єкти, які пов'язані ієрархічними відношеннями, створюють орієнтовний граф (перевернуте дерево) (рис. 3.4).

До основних понять ієрархічної структури відносять: рівень, елемент (вузол), зв'язок. Вузол – це сукупність атрибутів даних, які описують деякий об'єкт. На схемі ієрархічного дерева вузли

представлені вершинами графа. Кожен вузол на більш низькому рівні пов'язаний лише з одним вузлом, який перебуває на більш високому рівні. Ієрархічне дерево має тільки одну вершину (корінь дерева), яка непідвладна ніякій іншій вершині, вона знаходиться на самому верхньому (першому) рівні. Всі інші вершини залежні.

У кожному записі бази даних існує тільки один (ієрархічний) шлях від кореневого запису.

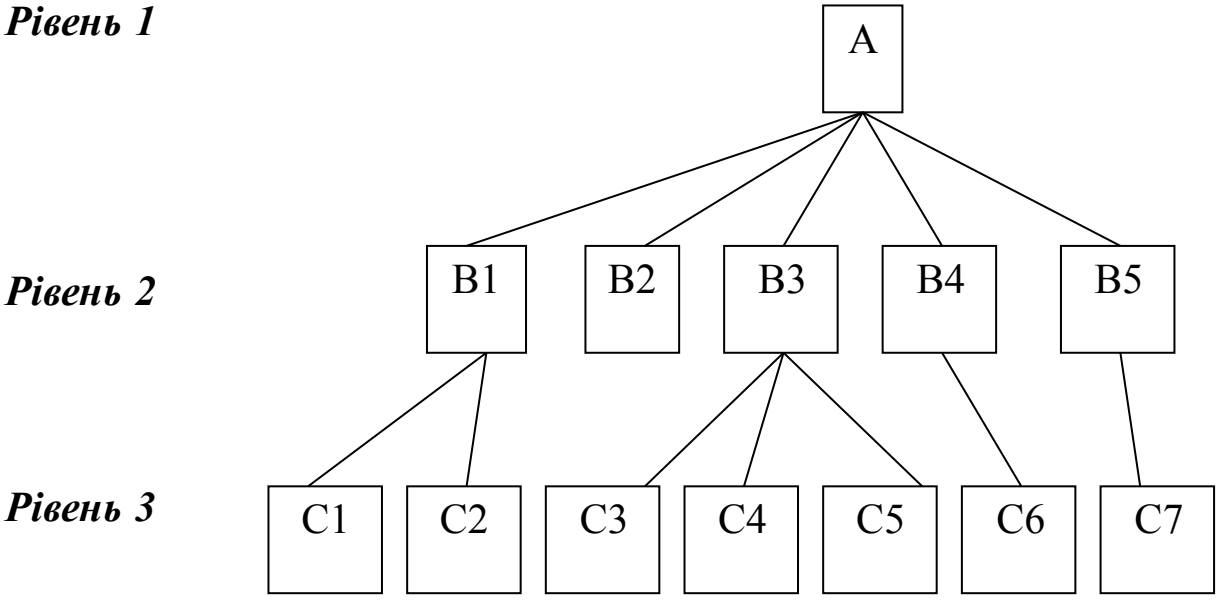


Рис. 3.4. Графічне відображення ієрархічної структури бази даних

Перевагами ієрархічної моделі є простота організації, наявність заздалегідь заданих зв'язків між сутностями, подібність з фізичними моделями даних. Це дозволяло домогтися прийнятної продуктивності ієрархічних СУБД повільних ЕОМ з дуже обмеженими обсягами пам'яті. Але якщо дані не мали деревоподібної структури, то виникала маса складностей під час побудови ієрархічної моделі і за бажання домогтися потрібної продуктивності.

У мережевій структурі за тих самих основних понять (рівень, вузол, зв'язок) кожен елемент цієї бази даних може бути пов'язаний з будь-яким іншим елементом.

На рис. 3.5. продемонстровано мережеву структуру бази даних у вигляді графа.

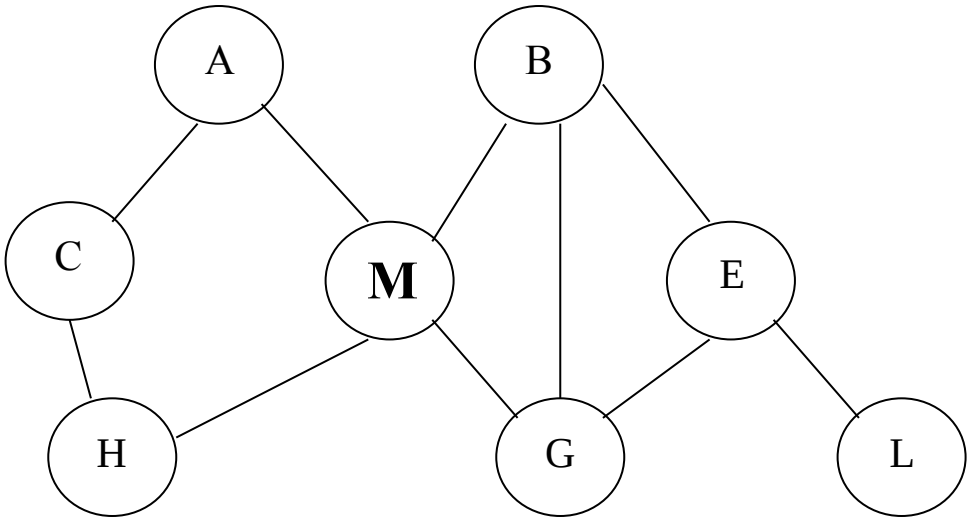


Рис. 3.5. Мережева структура бази даних у вигляді графа

У мережевій моделі дані організовуються у вигляді довільного графа. Недоліками є жорстка структура та висока складність реалізації.

Недоліком як ієрархічної, так і мережевої моделей є те, що структура даних задається на етапі проєктування бази даних та не може бути змінена при організації доступу до неї.

В орієнтованих базах даних окремі записи є об'єктами. Між записами бази даних та функціями їх обробки встановлюються взаємозв'язки за допомогою механізмів, що подібні відповідним зв'язкам у ООП мовах програмування. Об'єктно-орієнтовані моделі сполучають особливості мережевої та ієрархічної моделі та використовуються для створення великих баз даних зі складною структурою.

Реляційна модель отримала свою назву від англійського слова relation та була запропонована в 70-х роках Е. Коддом. Реляційна база даних – це сукупність таблиць, що пов'язані відношеннями. Перевагами реляційної моделі є простота, гнучкість структури, зручність реалізації на ЕОМ, наявність теоретичного опису. Більшість сучасних баз даних є реляційними.

Типи зв'язків. Усі інформаційні об'єкти предметної області пов'язані між собою. Розрізняють зв'язки декількох типів, для яких введені такі позначення:

- один до одного (1:1);
- один до багатьох (множини) (1:∞);
- багато до багатьох (множина до множини) (∞:∞).

Розглянемо ці типи зв'язків на прикладі.

Приклад. Дано сукупність інформаційних об'єктів, яка відображає навчальний процес у ЗВО:

СТУДЕНТ (*Номер*, Прізвище, Ім'я, По батькові, Стать, Дата народження, Група).

СЕСІЯ (*Номер*, Оцінка 1, Оцінка 2, Оцінка 3, Оцінка 4, Результат).

СТИПЕНДІЯ (*Результат*, Відсоток).

Викладач (*Код викладача*, Прізвище, Ім'я, По батькові).

Зв'язок один до одного (1:1) передбачає, що в кожен момент часу одному екземпляру інформаційного об'єкта А відповідає не більше одного екземпляра інформаційного об'єкта В і навпаки.

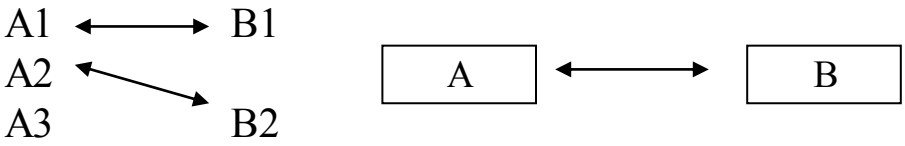


Рис. 3.6. Графічне представлення реального відношення 1:1

Приклад. Прикладом зв'язку 1:1 може слугувати зв'язок між інформаційними об'єктами СТУДЕНТ і СЕСІЯ

СТУДЕНТ ↔ СЕСІЯ

Кожен студент має визначений набір екзаменаційних оцінок за сесію.

При зв'язку один до багатьох (множини) (1:M) одному екземпляру інформаційного об'єкта А відповідає 0, 1 або більше екземплярів об'єкта В, але кожен екземпляр об'єкта В пов'язаний не більше ніж з одним екземпляром об'єкта А

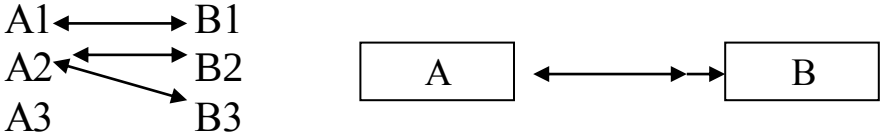


Рис.3.7. Графічне представлення реального відношення 1:∞

Приклад. Прикладом зв'язку 1: ∞ може слугувати зв'язок між інформаційними об'єктами СТИПЕНДІЯ і СЕСІЯ

СТИПЕНДІЯ ↔→ СЕСІЯ

Встановлений розмір стипендії за результатами сесії може повторюватися багато разів для різних студентів.

Зв'язок багато до багатьох (M:M) передбачає, що в кожну одиницю часу одному екземпляру інформаційного об'єкта A відповідає 0, 1 або більше екземплярів об'єкта B і навпаки.

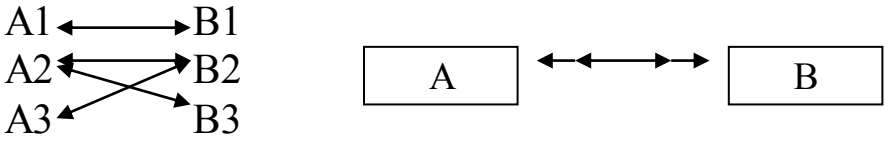


Рис.3.8. Графічне представлення реального відношення ∞:∞

Приклад. Прикладом зв'язку M:M може слугувати зв'язок між інформаційними об'єктами СТУДЕНТ і ВИКЛАДАЧ

СТУДЕНТ  ВИКЛАДАЧ

Один студент навчається у багатьох викладачів, і один викладач навчає багатьох студентів.

3.3. Проєктування реляційної бази даних економічної діяльності: метод нормальних форм; метод суть-зв'язок (ER-діаграм); засоби автоматизації проєктування

На першому етапі проєктування бази даних необхідно *визначити мету* створення бази даних, *основні її функції* та *дані*, які вона повинна містити. Після чого потрібно починати опис предметної області. Такий опис повинен охоплювати весь клас сутностей предметної області (реальні об'єкти, процеси і явища), інформація про яких повинна міститися в БД і забезпечувати реалізацію можливих запитів до БД і рішення задач.

Проєктування передбачає етапи створення проєкту бази даних від концепції до реального втілення.

Етапи проєктування бази даних:

- дослідження предметної області;
- аналіз даних;
- визначення відносин між елементами даних. Визначення первинних і вторинних (зовнішніх) ключів відносин. Реальні наочні області є системами взаємозв'язаних об'єктів, наприклад облік товарів, які реалізовані різним фірмам з декількох складів. Проєктуючи таку базу даних, користувач повинен передбачити, яка інформація йому може знадобитися в майбутньому.

Етапи розробки бази даних:

1) інфологічна модель;
2) даталогічна модель;
3) фізична модель;
4) база даних.

Розглянемо більш докладно інфологічний етап проєктування.

У предметній області (діловому середовищі) існують *сутності*, інформацію про які необхідно зберігати, і ці сутності пов'язані одна з одною певними зв'язками.

Для того щоб область зберігання інформації розглядалася як база даних, у ній повинні міститися не тільки дані, але і відомості про зв'язки між цими даними.

Значення поняття бази даних таке, що користувач – або людина, що взаємодіє з нею, або прикладна програма – не повинен піклуватися про те, яким чином дані фізично зберігаються у зовнішній пам'яті комп'ютера. Користувач висловлює запити маніпулювання даними на мові взаємовідношення даних. Програмні ж засоби, звані *системами управління базами даних* (далі – СУБД), організовують зв'язок між запитом користувача і фізичною областю зберігання інформації.

Перед початком проєктування бази даних необхідно визначити інформаційні об'єкти (сутності) і взаємозв'язки, що існують між ними.

Взаємовідношення між даними не залежать від моделі даних і СУБД, яка застосовується. А модель даних, навпаки, визначається системою управління базою даних.

Сутності і їх атрибути. Сутність (entity) – це щось, про що зберігається інформація. Клієнт – це сутність, як і товар, що зберігається на складі. Зовсім не обов'язково, щоб сутності були матеріальні. Так, деяка подія, наприклад концерт, є сутність; звернення до лікаря – теж сутність.

Сутність описують даними, званими атрибутами (attributes). Якщо сутність є економічним об'єктом, то атрибутами є його реквізити. Наприклад, клієнт як *сутність* звичайно описується номером, ім'ям, прізвищем, вулицею, містом, країною, поштовим індексом і номером телефону. *Сутність* концертів можна описати назвою, датою, місцем проведення і ім'ям виконавця.

При представленні сутності в базі даних фактично зберігаються тільки атрибути. Кожна група атрибутів, що описують один реальний прояв сутності, є екземпляром сутність.

Ідентифікатори сутностей. Єдиною метою розміщення в базі даних інформації, що описує сутність, є подальше прочитування цієї інформації. Отже, необхідно мати певний метод, що дозволяє відрізняти одну сутність від іншої і тим самим забезпечувати прочитування потрібної сутності. Для цього кожній сутності привласнюються певні значення атрибутів, що відрізняють її від будь-якої іншої сутності бази даних; сукупність значень називається ідентифікатором сутності (entity identifier).

Припустимо, що у фірми є два клієнти з ім'ям Петро Сидоров. Якщо службовець шукає пункти замовлення Петра Сидорова, відомості про якого з Петрів Сидорових вибере СУБД? У даному випадку – про обох. Оскільки не існує способу розрізнити двох клієнтів, результати запиту будуть неточні.

Ця проблема може бути вирішена створенням унікальних кодів (номерів) клієнтів. Це досить поширений спосіб ідентифікації екземплярів сутностей, коли у даних не передбачений будь-який простий унікальний ідентифікатор.

Іншим рішенням може стати об'єднання імені і прізвища клієнта з його телефонним номером. Подібна комбінація (складений ідентифікатор) також однозначно визначає кожного клієнта. Проте тут є два недоліки. По-перше, такий ідентифікатор довгий і громіздкий, і при введенні будь-якого з його компонентів велика

вірогідність помилки. По-друге, у разі зміни номера телефону клієнта ідентифікатор теж повинен змінитися. Зміна ідентифікатора сутності може призвести до виникнення серйозних проблем у базі даних.

У деяких сутностей, наприклад у *рахунків фактур*, є природні ідентифікатори (номери рахунків-фактур). Іншим сутностям – переважно людям, населеним пунктам і предметам – привласнюються унікальні номери без певного значення. Для третіх необхідні складені ідентифікатори.

При збереженні екземпляра сутності в базі даних потрібно, щоб СУБД забезпечувала наявність у нового екземпляра унікального ідентифікатора. Це є прикладом обмеження в базі даних – правила, якому повинні відповідати дані. Реалізація різних обмежень допомагає підтримувати узгодженість і точність інформації.

Однозначні і багатозначні атрибути. У такому випадку кінцевим висновком процесу проєктування є створення реляційної бази даних, тому атрибути нашої моделі даних повинні бути однозначними. Іншими словами, у кожного атрибуту конкретного екземпляра сутності може бути тільки одне значення. Наприклад, сутність *Клієнт* допускає наявність тільки одного телефонного номера для кожного клієнта. Якщо у клієнта декілька телефонних номерів, які потрібно включити в базу даних, то сутність *Клієнт* не зможе вміщати їх всі.

Хоча правильним є те, що модель взаємовідношення сутностей у базі даних не залежить від формальної моделі даних, що використовується для відображення структури даних у СУБД, часто рішення про моделювання даних ухвалюються, виходячи з вимог тієї формальної моделі, яка використовуватиметься згодом. Одне з подібних рішень – видалення багатозначних атрибутів. Аналогічний приклад буде наведено під час розгляду взаємостосунків «багато до багатьох» між сутностями.

Наявність декількох телефонних номерів перетворює атрибут телефонного номера в багатозначний. Оскільки в реляційній базі даних сутність не може мати багатозначні атрибути, необхідно упорядкувати їх, створивши сутність для їх зберігання.

У цій ситуації можна створити сутність телефонних номерів. У кожен екземпляр цієї сутності потрібно включити номер клієнта, якому належить телефонний номер, і сам телефонний номер. Якщо у клієнта три телефонні номери, то для нього існуватимуть три екземпляри сутності телефонних номерів. Ідентифікатор такої сутності потрібно складати з номера клієнта і з телефонного номера. Уникнути використання телефонного номера як елемента ідентифікатора сутності телефонних номерів неможливо.

Як правило, при зустрічі з багатозначним атрибутом рекомендується створити ще один атрибут. Єдиним способом роботи з декількома значеннями одного атрибута є створення сутності, декілька екземплярів якого можна зберегти в базі даних, поодинці для кожного значення атрибута.

Про сукупність сутностей. На початку роботи з сутностями може здатися, що природа сутності досить туманна. Розглянемо, наприклад, запас товарів. Чи є «запас товарів» сутністю? Ні. Цей запас складається з окремих товарів, з якими працює магазин. Справжньою сутністю є окремий товар. Товарний запас визначається переглядом усіх екземплярів сутності окремих товарів як єдиного цілого.

Щоб прояснити ситуацію, розглянемо атрибути, які необхідні у разі створення сутності запасу товарів: число товарів, назва товару, кількість на складі, роздрібна ціна тощо. Оскільки весь запас товарів ми намагаємося описати за допомогою однієї сутності, для кожного з цих атрибутів потрібно по декілька значень. Проте, як було показано вище, атрибути не можуть бути багатозначними, тобто запас товарів сутністю бути не може. Його необхідно представляти у вигляді сукупності екземплярів сутності окремих товарів.

Як інший приклад, розглянемо медичну історію хвороби людини, заповнювану лікарем. Подібно запасу товарів, історія хвороби є сукупністю декількох сутностей. Вона складається зі звернень до лікаря і подій, що відбуваються під час цього обігу. Отже, реально історія є сукупністю екземплярів сутностей обігу і сутностей курсів лікування. «Історія» – це вихідна інформація, яку додаток бази даних може одержати, вибравши дані, що зберігаються в екземплярах базових сутностей,

Документування сутностей і атрибутів. Діаграми взаємозв'язків сутностей дозволяють документувати сутності в базі даних, а також атрибути, що описують ці сутності. Є декілька типів ER-діаграм. Найчастіше використовують діаграми Чена (Chen) (на ім'я винахідника ER-моделювання) і діаграми інформаційного проєктування (Information Engineering). Дозволяється застосовувати діаграми будь-якого типу, головне, щоб той, хто використовує діаграму, розумів її символіку.

У обох моделях для представлення сутностей застосовуються прямокутники. Ім'я кожної сутності вказується у прямокутнику і ставиться в однині, як показано на рис. 3.9.

У початковій ER-моделі (entity relationship) Чена не передбачена можливість відображення атрибутів безпосередньо на ER-діаграмі. Проте багато хто розширює цю модель, вносячи атрибути і укладаючи їх в овали, наприклад як на рис. 3.9.

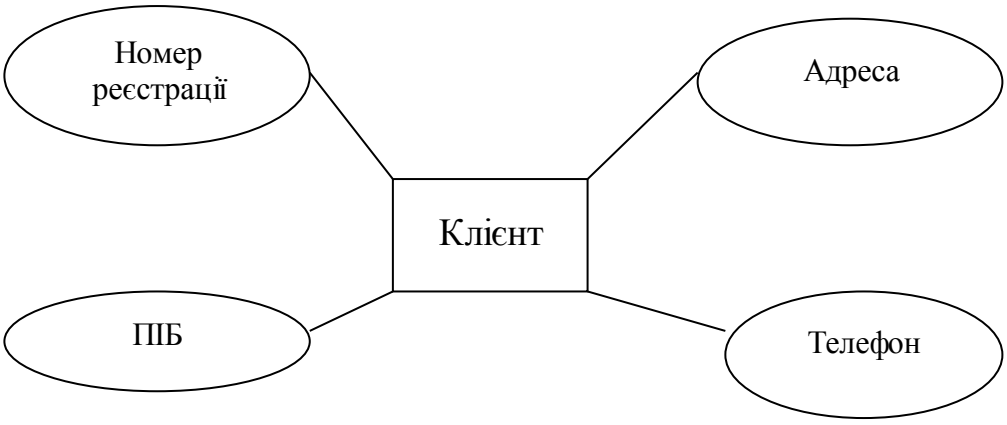


Рис. 3.9. Зображення сутності *Клієнт* та її атрибутів

Ідентифікатор (ключ) сутності – це атрибут, що позначається зірочкою (*Номер реєстрації).

У моделі інформаційного проєктування на рис. 3.10 атрибути вказуються в прямокутнику разом із сутністю.

Клієнт
*Код реєстрації
ПІБ
Адреса
Телефон

Рис. 3.10. Зображення сутності та її атрибутів у моделі інформаційного проєктування

Модель інформаційного проєктування дозволяє будувати компактніші діаграми.

Домени. У кожного атрибута є домен (domain) – вираз або перелік значень, дозволених для цього атрибута. Домен може бути дуже малий. Так, у магазині, значеннями атрибута Size (розмір) товарів можливо можуть бути лише S, M, L, XL і XXL; ці значення і становлять весь домен. І навпаки, домен імені клієнта досить великий, і його можна вказати тільки як ”текст” або ”імена людей”.

У СУБД домен реалізується за допомогою обмеження. Кожного разу при записі деякого значення в базу даних СУБД перевіряє його відповідність домену, вказаному для його атрибута. Хоча у багатьох випадках невеликі домени визначити неможливо, домен, принаймні, забезпечує отримання даних коректного типу. Наприклад, СУБД може заборонити користувачу запис значення 123 × 50 в той атрибут, доменом якого є значення грошових одиниць. Крім цього, в більшості СУБД за допомогою доменів забезпечується досить жорстка перевірка атрибутів дати і часу, що допомагає уникнути появи таких дат, як, наприклад, 30 лютого.

Документування доменів. Найчастіше домени не вказуються безпосередньо на ER-діаграмах, а записуються в спеціальний документ (зазвичай в словник даних). Проте у варіанті методу Чена з вказівкою атрибутів домени можуть бути задані. Вираз, що визначає домен, розміщується нижче за кожен атрибут.

Взаємозв’язки між сутностями відображаються за допомогою ER-діаграми (рис. 3.11).

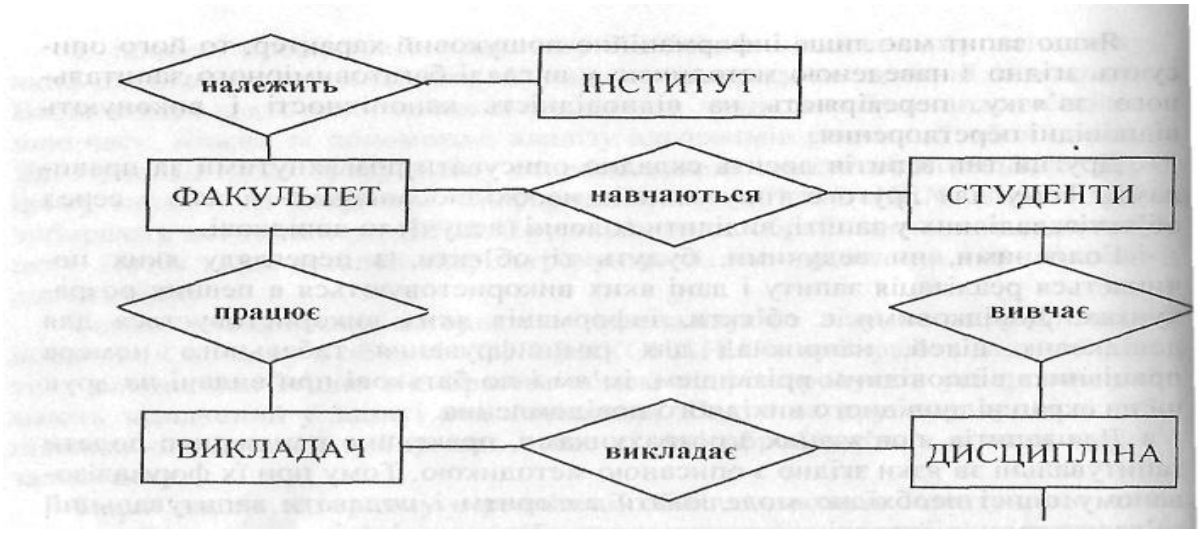


Рис. 3. 11. Приклади ER-діаграми

Метод Чена і метод інформаційного проєктування, що використовуються для побудови ER-діаграм, по-різному відображають взаємозв'язки. Кожен з методів має свої переваги з погляду об'єму інформації і складності діаграм, що створюються. Основними поняттями ER-моделі є с*утність*, *зв'язок* та *атрибут*. Кожна з частин такої діаграми повідомляє дещо про структуру даних або про те, як ці дані співвідносяться з іншими.

Метод Чена. На використанні різновидностей ER-моделі ґрунтується більшість сучасних підходів до проєктування реляційних баз даних. Модель була запропонована Ченом (Chen) у 1976 році. Моделювання предметної області базується на використанні графічних діаграм, які включають невелике число різновидних компонентів. У зв'язку з наочністю подання концептуальних (логічних інформаційних) схем баз даних ER-моделі набули значного поширення в системах CASE, які підтримують автоматизоване проєктування реляційних баз даних.

У методі Чена для зображення взаємозв'язків використовуються ромби, а для зображення типу взаємовідношення між сутностями – лінії зі стрілками. Наприклад, на рис. 3.12 представлено взаємовідношення між фірмою–клієнтом і замовленням.



Рис. 3.12. Зображення взаємозв'язків між сутностями у методі Чена

Одиночна стрілка, направлена на сутність Фірма–клієнт, указує на те, що замовлення належить максимум одному клієнту, подвійна стрілка, направлена на сутність Замовлення, означає, що клієнт може зробити один або декілька замовлень.

У рамках методу Чена існують два альтернативні способи представлення взаємозв'язків. Перший (мал. 3.13а) припускає використання стрілок з цифрами і буквами. Тут "1" указує на те, що замовлення надходить від одного клієнта, а "∞" – на те, що клієнт може робити багато замовлень.



Рис. 3.13а. Використання стрілок з цифрою і буквою для зображення взаємовідношень між сутностями на ER-діаграмі

Під час використання другого способу усувається недолік, пов'язаний із легкістю для читання взаємовідношення в обох напрямах, коли ім'я взаємовідношення перебуває усередині ромба. Вираз «Клієнт робить замовлення» має цілком певний сенс, але «Замовлення робить клієнта» – нісенітниця. Для вирішення цієї задачі ім'я взаємовідношення видаляється з ромба і додається як до взаємовідношення, так і до його інверсії на діаграмі (рис. 3.13б). Цей варіант діаграми можна читати в будь-якому напрямі: «Клієнт робить багато замовлень» і «Замовлення робиться одним клієнтом».

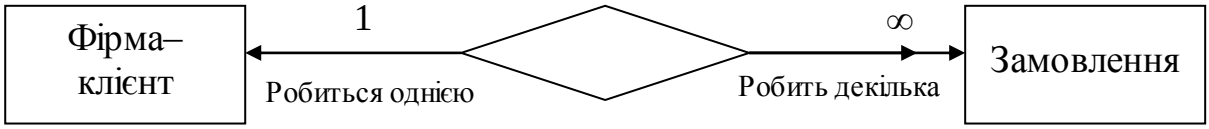


Рис. 3.13б. Винесення імені взаємовідношення за межі ромба на ER-діаграмі для зображення взаємовідношень між сутностями

При зображенні ER-діаграм за допомогою методу Чена існує одне важливе обмеження: відсутній очевидний спосіб вказівки слабких сутностей і обов'язкових взаємозв'язків. Так, замовлення не повинно бути присутнім у базі даних без клієнта. Отже, замовлення є слабкою сутністю, і його взаємовідношення з клієнтом обов'язкове. Саме тому багато розробників баз даних, що використовують метод Чена, почали застосовувати для слабкої сутності новий символ – прямокутник з подвійною межею (рис. 3.14).



Рис. 3.14. Зображення слабкої сутності

Метод інформаційного проєктування. У методі інформаційного проєктування додаткова інформація на ER-діаграмі відображається графічними символами на кінцях ліній.

На кінцях ліній діаграми інформаційного проєктування можуть використовуватися такі символи, які наведено у таблиці 3.1.

Таблиця 3.1

Графічні символами ER-діаграми

□□	один і лише один (обов'язкове взаємовідношення)
0 □	нуль або один
>□	один або більш (обов'язкове взаємовідношення)
>0	нуль, один або більш

Кінці ліній указують, які з взаємозв'язків є обов'язковими. Як приклад, розглянемо рисунок 3. 15.

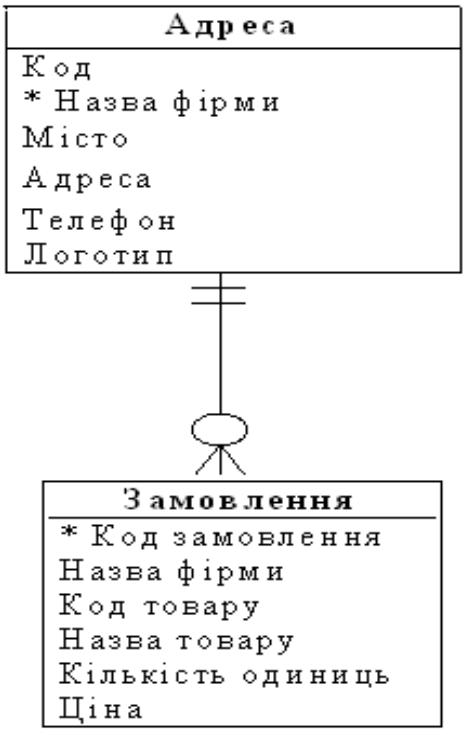


Рис. 3.15. Взаємовідношення типу "один до багатьох" на діаграмі, побудованій методом інформаційного проєктування

Подвійна лінія нижче за сутністю ”Адреси” означає, що кожне замовлення пов’язано з однією і лише однією фірмою–клієнтом. Оскільки нуль не є одним із варіантів, це взаємовідношення обов’язкове. Навпаки, 0 і комбінація з трьох ліній, сполучена з сутністю ”Замовлення”, указують на те, що клієнт може зробити нуль, один або декілька замовлень.

Ці символи часто зображаються повернеными на 90° (рисунок 3.15).

При використовуванні методу інформаційного проєктування атрибути часто указуються безпосередньо на діаграмі. Ідентифікатори сутностей позначені зірочками.

На рис. 3.16 наведено інформаційну діаграму.

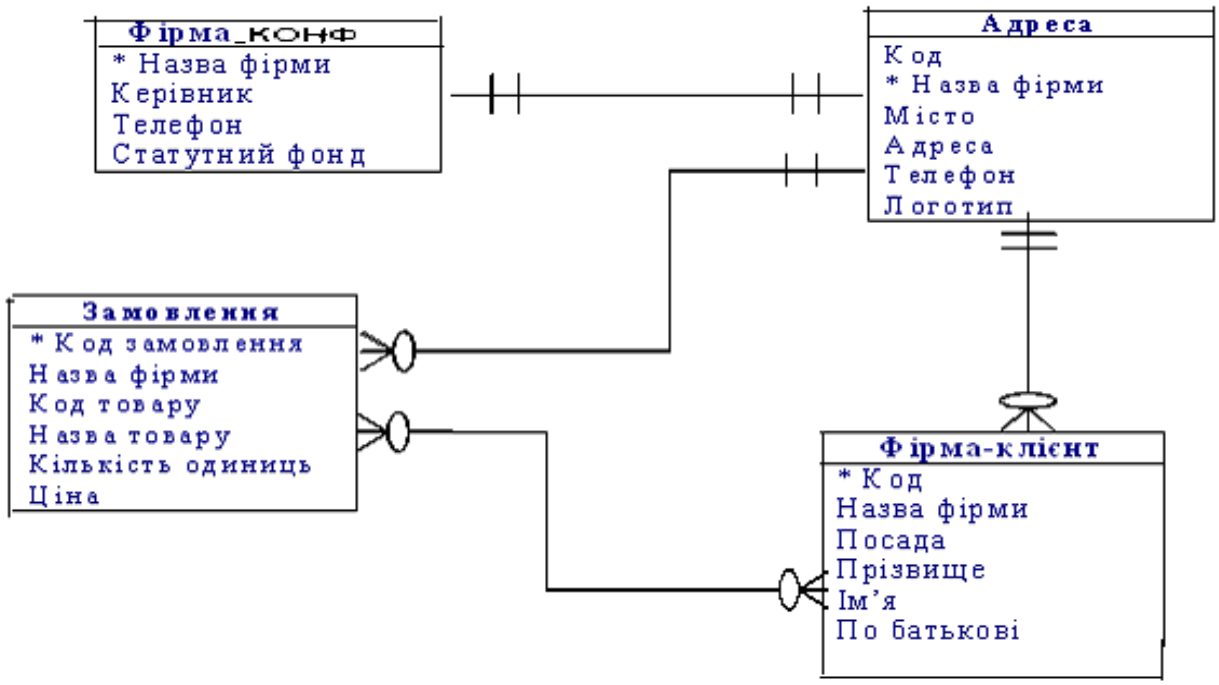


Рис. 3.16. Інформаційна діаграма

3.4. Програмні засоби роботи з базами даних. СУБД в економіці

Система управління базами даних (СУБД) – це комплекс мовних і програмних засобів, призначений для створення, ведення і спільного використання БД багатьма користувачами. Нині найбільшою популярністю користуються СУБД спроєктовані на основі реляційної моделі даних, і які називаються реляційними СУБД. Нині кількість різних СУБД обраховується тисячами. СУБД належить до набору засобів програмного забезпечення, створених для роботи з фактографічними даними, структура яких набагато складніша за структуру неформатованих текстових документів. Однак існують СУБД, що можуть працювати з різними типами БД (у цих випадках звичайно зазначають, що до складу СУБД включений текстовий редактор). Існують СУБД, створені спеціально для збереження і пошуку текстових документів, а також СУБД для збереження графічних об'єктів (карт, креслень).

Системою управління базами даних називають програмну систему, яка призначена для створення на ЕОМ загальної бази даних, яка використовується для розв'язування задач. Подібні системи слугують для підтримки баз даних в актуальному стані і забезпечують ефективний доступ користувачів до даних, які зберігаються в базі даних, у рамках надання користувачам повноважень.

СУБД призначена для централізованого управління базою даних у інтересах всіх працюючих у цій системі.

За ступенем універсальності розрізняють два класи СУБД:

1) системи загального призначення;

2) спеціалізовані системи.

СУБД загального призначення не орієнтовані на якусь предметну область або на інформаційні потреби деякої групи користувачів. Кожна така система реалізується як програмний продукт, який здатний функціонувати на деякій моделі ЕОМ у визначеній операційній системі і доставляється багатьом користувачам як комерційний виріб. Такі СУБД володіють засобами налаштування на роботу з конкретною БД. Використання СУБД загального призначення як

інструментального засобу для створення автоматизованих інформаційних систем, заснованих на технології БД, дозволяє суттєво скоротити строки розробки, зекономити трудові ресурси.

Спеціалізовані СУБД створюються у рідких, виняткових випадках за неможливості або недоцільності використання СУБД загального призначення.

СУБД загального призначення – складні програмні комплекси, призначені для виконання всієї сукупності функцій, пов'язаних зі створенням і експлуатацією БД інформаційної системи.

Ринок програмного забезпечення має велику кількість різнобічних за своїми функціональними можливостями комерційних систем управління БД загального призначення, а також засобами їх оточення практично для всіх моделей машин і для різних операційних систем.

Сучасні СУБД володіють засобами забезпечення цілісності даних і надійної безпеки, що дає можливість розробникам гарантувати велику безпеку даних за менших затрат сил на низькорівневе програмування. Продукти, які функціонують у середовищі WINDOWS, відрізняються зручним користувацьким інтерфейсом і вбудованими засобами підвищення продуктивності.

Розглянемо основні характеристики деяких СУБД – лідерів на ринку програм, які призначені як для розробників інформаційних систем, так і для кінцевих користувачів.

До складу групи таких програмних продуктів увійшли:

- dBASE IV 2.0, компанії Borland International;
- Microsoft Access;
- Microsoft FoxPro (Microsoft Corp);
- Paradox (Borland);
- Oracle, компанії Oracle, – перша у світі СУБД, розроблена спеціально для работи у мережах розподілених обчислень.

Продуктивність СУБД. Продуктивність СУБД оцінюється:

- часом виконання запитів;
- швидкістю пошуку інформації в неіндексованих полях;
- часом виконання операцій імпортування БД з інших форматів;

- швидкістю створення індексів і виконання таких операцій, як оновлення, вставка, знищення даних;
- максимальним числом паралельних звернень до даних у багатокористувацькому режимі;
- часом генерації звіту.

На продуктивність СУБД здійснює вплив два фактори:

1) продуктивність власних прикладних програм дуже залежить від правильного проєктування і побудови бази даних;

2) СУБД, які слідкують за дотриманням цілісності даних, несуть додаткове навантаження, яке не відчувають інші програми.

Найшвидші програмні вироби не володіють найрозвиненішими функціональними можливостями на рівні процесора СУБД.

Забезпечення цілісності даних на рівні БД. Під цією характеристикою розуміють наявність засобів, які дозволяють переконатись, що інформація у БД завжди залишається коректною і повною. Повинні бути встановлені правила цілісності, і вони мають зберігатися разом з БД і дотримуватись на глобальному рівні. *Цілісність даних* повинна забезпечуватись незалежно від того, яким чином ці дані заносяться у пам'ять (в інтерактивному режимі, засобами імпорту або за допомогою спеціальних програм).

До засобів забезпечення цілісності даних на рівні СУБД відносять:

- вбудовані засоби для призначення первинного ключа, зокрема засоби для роботи з типом полів з автоматичним прирощуванням, коли СУБД самостійно присвоюють нове унікальне значення;
- засоби підтримки цілісності зносок (зсилок), які забезпечують запис інформації про зв'язки таблиць і автоматично припиняють будь-яку операцію, яка призводить до порушення зсилкової цілісності.

Деякі СУБД мають добре розроблений процесор СУБД для реалізації таких можливостей, як унікальність первинних ключів, обмеження операцій і каскадне оновлення та знищення інформації. У таких системах перевірка коректності, яка призначена полю або таблиці, буде проводитись завжди після змін даних, а не тільки під час введення інформації за допомогою екранних форм.

Цю властивість можна налагоджувати для кожного поля і запису, що дозволяє контролювати не лише значення окремих полів, але і взаємозв'язки між декількома полями цього запису.

Access та Paradox краще інших СУБД відповідають реляційній моделі за надійністю збереження цілісності даних на рівні БД; правила зберігаються сумісно з БД і автоматично дотримуються.

СУБД FoxPro зовсім не володіє такими засобами, і введення у програму процедур, які забезпечують виконання правил цілісності, покладаються на програміста.

Забезпечення безпеки. Деякі СУБД мають засоби безпеки даних. Ці засоби забезпечують виконання таких операцій:

- шифрування прикладних програм;
- шифрування даних;
- захист паролем;
- обмеження рівня доступу (до бази даних, до таблиці, до словника, для користувача).

Найвищий рівень безпеки даних реалізовано у СУБД dBASE. Адміністратор може призначати системі різні права доступу на рівні файлу, поля, а також організовувати автоматичне шифрування даних.

Хорошими характеристиками забезпечення безпеки відрізняється ACCESS 2.0. У ньому передбачено призначення паролів для індивідуальних користувачів або груп користувачів і присвоєння різних прав доступу окремо таблицям, запитам, звітам, макрокомандам або новим об'єктам на рівні користувача або групи.

Робота у багатокористувацьких середовищах. Практично всі розглянуті СУБД призначені для роботи у багатокористувацьких середовищах, але володіють різними можливостями.

Обробка даних у багатокористувацьких середовищах передбачає виконання програмним продуктом таких функцій:

- блокування БД, файлу, запису, поля;
- ідентифікацію станції, яка встановила блокування;
- оновлення інформації після модифікації;
- контроль за терміном та повторення звернення;
- обробку транзакцій;
- роботу з мережевими системами (LAN Manager, Net Ware, Unix).

Трансакція – послідовність операцій користувача над базою даних, яка зберігає її логічну цілісність.

Найкращими можливостями для роботи у багатокористувацьких середовищах володіють СУБД Paradox, Access, dBASE.

Імпорт–експорт. Ця характеристика відображає:

- можливість обробки СУБД інформації, яка підготовлена іншими програмними засобами;
- можливість використання іншими програмами даних, які сформовані засобами СУБД, яка розглядається.

Особливий інтерес являють собою такі формати файлів: ASII–файли,.DBF,.WK*,.XLS.

Доступ до даних за допомогою мови SQL. Мова запитів SQL(Structured Query Language) реалізована у популярних СУБД для різних типів ЕОМ або як базова, або як альтернативна. Маючи широке використання, є міжнародним стандартом мови запитів. Мова SQL надає розвинені можливості як кінцевим користувачам, так і спеціалістам у сфері обробки даних.

СУБД мають доступ до даних SQL у таких випадках:

- бази даних сумісні з ODBC (Open Database Connectivity – відкрите з'єднання баз даних);
- реалізована природна підтримка SQL-баз даних;
- можлива реалізація SQL-запитів локальних даних.

Багато СУБД можуть «прозоро» підключатись до вхідних SQL-підсистем за допомогою ODBC або драйверів, які є їх частиною, тому існує можливість створення прикладних програм для них.

Access і Paradox працюють з джерелами SQL-даних, які сумісні з системою ODBC.

FoxPro постачається з додатковими бібліотеками, які забезпечують доступ до SQL-баз даних, спроможних працювати сумісно з системою ODBC, але ця можливість менш інтегрована, ніж засоби первинного введення інформації в Access і Paradox.

Можна безпосередньо управляти базами даних Access за допомогою мови SQL і передавати наскрізні SQL-запити сумісним з специфікацією ODBC SQL-базам даних, таким як MS SQL Server і Oracle, так що Access здатна слугувати засобом розробки систем клієнт-сервер, які маштабуються.

Можливості запитів та інструментальні засоби розробки прикладних програм. СУБД, які орієнтовані на розробників, володіють розвиненими засобами для створення додатків. До елементів інструментарію розробки додатків можна віднести:

- потужні мови програмування;
- засоби реалізації меню, екранних форм введення-виведення даних і генерації звітів;
- засоби генерації додатків (прикладних програм);
- генерацію файлів, які виконуються.

Функціональні можливості моделей даних доступні користувачу СУБД завдяки її мовним засобам.

Реалізація мовних засобів інтерфейсів може бути здійснена різними способами. Для висококваліфікованих користувачів (розробників складних прикладних систем) мовні засоби частіше за все представляються в їх явній синтаксичної формі. В інших випадках функції мов можуть бути доступні непрямим, опосередкованим чином, коли вони реалізуються у формі різних меню, діалогових сценаріїв або таблиць, які заповнює користувач. За такими вхідними даними інтерфейсні засоби формують адекватні синтаксичні конструкції мови інтерфейсу і передають їх на виконання або включають у програмний код додатка (програми), який генерується. Інтерфейси з ненаявним використанням мови широко використовуються у СУБД для персональних комп'ютерів. Прикладом такої мови є QBE (Query By Example).

Мовні засоби використовуються для виконання двох основних функцій:

1) описання представлення бази даних;

2) виконання операцій маніпулювання даними.

Перша з цих функцій забезпечується *мовою описання* (визначення) *даних* (МОД). Описання бази даних засобами МОД називається *схемою бази даних*. Воно включає описання структури бази даних і накладених на неї обмежень цілісності в рамках тих правил, які регламентовані моделлю даних СУБД, яка використовується. МОД деяких СУБД забезпечує також можливості накладання обмежень доступу до даних або повноважень користувачів.

МОД не завжди синтаксично оформлюється у вигляді самостійної мови. Вона може бути складовою частиною єдиної мови даних, яка поєднує можливості визначення даних і маніпулювання даними.

Мова маніпулювання даними (далі – ММД) дозволяє запитувати передбачені в системі операції над даними з БД.

Є багаточисельні приклади мов СУБД, які об'єднують можливості описання даних і маніпулювання даними в єдиних синтаксичних рамках. Популярною мовою є реляційна мова SQL.

3.5. Структура сховищ даних та програмні засоби роботи зі сховищами даних

Сховище даних – це аґреґований інформаційний ресурс, що містить консолідовану інформацію з усієї проблемної області та використовується для підтримки прийняття рішень.

Консолідована інформація – це одержані з декількох джерел та системно інтеґровані різнотипні інформаційні ресурси, які в сукупності наділені ознаками повноти, цілісності, несуперечності та становлять адекватну інформаційну модель проблемної області з метою її аналізу опрацювання та ефективного використання в процесах підтримки прийняття рішень.

Типове сховище даних, зазвичай, відрізняється від традиційної реляційної бази даних. По-перше, традиційні бази даних призначені для того, щоб допомогти користувачам виконувати повсякденну роботу, тоді як сховища даних призначені для прийняття рішень. Наприклад, продаж товару і виписування рахунку здійснюється з використанням бази даних, призначеної для опрацювання транзакцій, а аналіз динаміки продажу за декілька років, що дозволяє спланувати роботу з постачальниками, – за допомогою сховища даних.

По-друге, традиційні бази даних характеризуються постійними змінами у процесі роботи користувачів, а сховище даних порівняно стабільне: дані у ньому зазвичай оновлюються за розкладом (наприклад, щотижня, щодня або щогодини – залежно від потреб). В ідеалі процес поповнення (або як далі ми будемо називати

завантаження) є просто додаванням нових даних за певний період часу без зміни попередньої інформації, що вже міститься в сховищі.

І, по-третє, традиційні бази даних найчастіше є джерелом даних, що потрапляють у сховище. Крім того, сховище може поповнюватися за рахунок зовнішніх джерел, наприклад, статистичних звітів. Дані, що надходять до бази даних з іншої бази, невеликого обсягу (тисячі записів), мають ту ж схему даних, що і база даних-приймач. На відміну від них сховища даних у визначені терміни отримують значно більші обсяги даних, які можуть відрізнятися від приймача форматом, а іноді і типом, що вимагає застосування додаткових процедур трансформування та завантаження даних (так звані процедури Extract, Transform, Load).

Як бази даних, так і сховища даних, можуть будуватись на основі певної системи керування базами даних (СКБД) (реляційна, постреляційна тощо). СКБД забезпечує загальний репозиторій для зберігання і опрацювання структурованих даних. СКБД підтримує набір взаємозв'язаних послуг і дозволяє розробникам зосередитись на специфічних проблемах їх застосування, а не на завданнях, які виникають за потреби в узгодженому й ефективному керуванні великими обсягами даних. Проте СКБД вимагають, щоб всі дані перебували під єдиним адміністративним керуванням і відповідали єдиній схемі. У відповідь на задоволення цих обмежень СКБД можуть забезпечити розвинені засоби маніпулювання даними та опрацювання запитів зі зрозумілою і строгою семантикою, а також строгі транзакційні ґарантії оновлень, паралельного доступу і довготривалого зберігання (так звані властивості ACID).

Враховуючи специфіку, сховище даних має такі особливості проєктування та побудови:

1) отримання інформації з різних джерел даних (зокрема і з реляційних баз даних) у деталізованому та аґреґованому вигляді (зберігаються результати застосування функцій агрегації – суми, середнього значння, максимуму, мінімуму тощо);

2) багатовимірне подання інформації – ігноруються деякі вимоги нормалізації (дотримують максимум 3-ої нормальної форми), що значно підвищує швидкість опрацювання інформації, оскільки зменшує кількість операцій з'єднання;

3) наявність метаданих для опису джерел метаданих та структури самого сховища даних – у базах даних також використовують словники для опису структур даних, а у сховищах даних метадані (словники, дані про дані) повинні будуватися за класифікаційною схемою Захмана. За цією схемою описують об'єкти (що?), суб'єкти (хто?), місцезнаходження (де?), час (коли?), фактори впливу, чинники (чому?), способи (як?);

4) наявність пакетного завантаження даних у сховище даних та вивантаження даних;

5) наявність процедур аналізу даних та отримання нових даних;

6) орієнтованість даних на аналітичне, а не на статичне опрацювання.

Сховища даних краще пристосовані до зберігання та аналітичного опрацювання великих обсягів даних, переважно є інтеграцією реляційної та багатовимірної моделей. На сьогодні є такі архітектури побудови сховищ даних: *корпоративна інформаційна фабрика Білла Інмона, шина Ральфа Кімбола, зведення даних корпорації TDAN*. Вони мають розвинені засоби інтеграції даних з різних джерел та дозволяють працювати як з деталізованою, так і агрегованою інформацією.

Парадигма для реляційних даних у сховищі даних (парадигма *корпоративної інформаційної фабрики* КІФ – Corporate Information Factory, CIF) розроблена Інмоном і передбачає, що дані повинні перебувати на низькому рівні деталізації і в третій нормальній формі (3НФ, 3NF). Білл Інмон підтримує повторний або спіральний підхід до розвитку великого сховища даних. За цим підходом розвиток сховища відбувається ітераційно, тобто у разі виникнення потреби додається одна таблиця за один раз, що забезпечує лише незначну зміну схеми даних. Тому такий підхід до проєктування сховища ще називають *спіральним підходом*.

Є альтернативний підхід до архітектури сховищ даних, відомий як *Сховище даних з архітектурою шини*, або *підхід Ральфа Кімбола*, або Просторове сховище. У цій моделі первинні дані перетворяться в інформацію, придатну для використання, на етапі підготовки даних. При цьому обов'язково приймаються до уваги вимоги до швидкості опрацювання інформації і якості даних.

Як і в моделі Білла Інмона, підготовка даних починається зі скоординованого добування даних із джерел. Ряд операцій відбувається централізовано, наприклад підтримка і зберігання загальних довідкових даних, інші дії можуть бути розподіленими.

Область подання просторово структурована, при цьому вона може бути централізованою або розподіленою. Просторова модель сховища даних містить ту саму атомарну інформацію, що й нормалізована модель, але інформація структурована по-іншому, щоб полегшити її використання й виконання запитів. Ця модель включає як атомарні дані, так і узагальнювальну інформацію (агрегати у зв'язаних таблицях або багатомірних кубах) відповідно до вимог продуктивності або просторового розподілу даних. Запити в процесі виконання звернені до нижчого рівня деталізації без додаткового перепрограмування з боку користувачів або розроблювачів застосування.

На відміну від підходу Білла Інмона просторові моделі будуються для обслуговування бізнес-процесів (які, у свою чергу, пов'язані з бізнес-показниками або бізнес-подіями), а не бізнес-відділів. Наприклад, дані про замовлення, які повинні бути доступні для загальнокорпоративного використання, вносяться в просторове сховище даних тільки один раз, на відміну від КІФ-підходу, у якому їх довелося б тричі копіювати у вітрини даних відділів маркетингу, продажу і фінансів. Після того, як у сховищі з'являється інформація про основні бізнес-процеси, консолідовані просторові моделі можуть видавати їхні перехресні характеристики.

Матриця корпоративного сховища даних з архітектурою шини виявляє й підсилює зв'язок між показниками бізнес-процесів (фактами) й описовими атрибутами (вимірами).

Гібридна архітектура, яка поєднує особливості реляційної та багатовимірної моделей, запропонована Дугласом Хекні. Іншою назвою моделі є *Узгоджувана вітрина даних*. За такої архітектури передбачається подвійне проєктування схеми сховища даних – розроблення нормалізованого центрального (корпоративного сховища) та багатовимірних (побудованих за архітектурою шини) вітрин даних. Корпоративне нормалізоване сховище дозволяє коректно зберігати дані, а ненормалізовані вітрини – швидко виконувати запити користувачів.

Зведення даних – предметно орієнтована, історична й унікально зв'язана множина нормалізованих таблиць, які підтримують одну або більше функціональних предметних областей. Це гібридний підхід, що поєднує кращі особливості 3-ої нормальної форми (3НФ) і схеми «зірка». Модель гнучка, масштабується, послідовна і пристосована до потреб різних предметних областей. Вона відповідає потребам сховища даних і відкидає потребу у використанні вітрин даних та, на відміну від гібридного підходу Хекні, не вимагає подвійної роботи для надбудови архітектури шина над архітектурою корпоративної фабрики.

Зведення даних може керувати масивними наборами гранульованих даних в меншому, більш нормалізованому фізичному просторі, наприклад 3НФ і схемі «зірка». Базується на математичних принципах, які підтримують нормалізовані моделі даних. Внутрішня частина моделі зведення даних – близькі структури, які відповідають традиційним визначенням схеми «зірка» і 3НФ, що включають виміри, зв'язки багато до багатьох, і стандартні табличні структури. Відмінності полягають у поданні зв'язків, структуризації поля і гранульованому, пов'язаному з часом зберіганні даних.

Є такі підвиди сховища даних: вітрина даних, оперативне сховище даних.

Вітрина даних (далі – ВД) – зріз сховища даних, масив тематичної, вузьконаправленої інформації, що орієнтований, наприклад, на користувачів однієї робочої групи або департаменту.

Дворівнева архітектура сховища даних передбачає побудову вітрин даних без створення центрального сховища, при цьому інформація надходить із реєстраційних систем і обмежена конкретною предметною областю. При побудові вітрин використовуються основні принципи побудови сховищ даних, тому їх можна вважати сховищами даних у мініатюрі.

Побудова повноцінного корпоративного сховища даних, зазвичай виконується в *трирівневій архітектурі*. На першому рівні розташовані різноманітні джерела даних – внутрішні системи, що реєструють, довідкові системи, зовнішні джерела (дані інформаційних агентств, макроекономічні показники). Другий рівень містить центральне сховище, куди надходить інформація від усіх джерел з першого рівня, і, можливо, оперативне сховище даних

(сховище поточної інформації розглянуто далі), що не містить історичних даних і виконує дві основні функції:

1) воно є джерелом аналітичної інформації для оперативного керування;

2) тут підготовлюються дані для наступного завантаження в центральне сховище.

Операційне сховище даних (далі – ОСД) – це предметно-орієнтований, інтегрований, змінюваний набір консолідованих даних, який містить поточну (неісторичну) деталізовану інформацію.

На перший погляд, операційне сховище даних схоже на сховище за структурою і змістом. Зазвичай за деякими характеристиками ОСД і сховище даних дуже схожі, але ОСД має ряд властивостей, які істотно відрізняють його від сховища. Як ОСД, так і сховище даних є предметно-орієнтованим інтегрованим набором консолідованих даних. З цієї точки зору вони подібні, оскільки як в одному, так і в іншому випадку дані повинні бути завантажені з трансакційних систем. Але на цьому їх подібність закінчується. ОСД *містить дані, що змінюються,* тоді як у сховищі дані після завантаження не змінюються. Інша відмінність полягає у тому, що операційне сховище *містить тільки дані, актуальні на певний момент час*у, тоді як у сховищі містяться як поточні, так і історичні дані. При цьому актуальність даних у сховищі значно нижча, ніж в операційному сховищі. Як правило, в сховищі містяться дані, завантажені протягом останніх 24 годин, тоді як актуальність даних в ОСД може вимірюватися секундами. Ще однією відмінністю ОСД від сховища є те, що в ньому містяться тільки *детальні дані*, тоді як сховище містить як детальні, так і агреговані дані.

Архітектура сховища даних. Архітектуру сховища даних розглядають за чотирма рівнями:

– рівень користувача – описує програмний інтерфейс доступу користувачів до сховища даних;

– рівень застосувань (ППЗ) – описує засоби роботи з даними;

– рівень даних – представляє засоби для розробки структур та моделей даних, правила збереження даних, режими контролю доступу до даних;

– рівень калькуляції – містить сумарні, підсумкові дані, що полегшує та прискорює доступ до даних.

Джерела інформації. Джерелами даних для сховища даних ДПС є дані систем сегментів операційної діяльності ДПС:

– реєстрація платників податків;

– обробка податкової звітності та платежів;

– облік платежів;

– податковий аудит;

– управління персоналом;

– обслуговування платників податків;

– апеляції платників податків;

– погашення податкового боргу;

– внутрішній контроль;

– досудове слідство (кримінальні розслідування);

– розробка автоматизованої системи декларування, обліку і контролю виробництва, зберігання, руху та споживання підакцизних товарів;

– вирішення справ у господарських судах за участю органів ДПС;

– матеріально-технічне та фінансове забезпечення;

– зв'язки з громадськістю;

– розробка проєктів нормативно-правових актів та впровадження законодавства;

– внутрішні файли, що прямо не пов'язані з оперативними системами – індивідуальні робочі електронні таблиці, робочі журнали;

– зовнішні джерела – дані від зовнішніх організацій відповідно до угод/протоколів/наказів про інформаційну взаємодію ДПА України з відповідною організацією: Мінфін, Мінекономіки, Держмитслужба, Держказначейство, ГоловКРУ, Держкомстат, Держпідприємництво, НБУ інші.

Вимоги до даних. Основа сховища даних – дані, які зберігаються в ньому і які є ключем для реалізації процесу прийняття рішень. Тому до даних у сховищах даних висуваються такі вимоги:

- достовірність;
- повнота;
- несуперечливість;
- ненадлишковість;
- актуальність;

- цілісність;
- надіндексованість;
- денормалізованість.

Індекси використовуються для швидкого доступу до даних і є критичними для виконання запитів, що обробляються в сховищі даних. У сховищі даних створюється набагато більше індексів, ніж в операційних системах.

Сховище даних містить свідомо надлишкову інформацію (тобто містить архіви та історичні дані), що знаходиться в базах чи файлах оперативних систем. Денормалізація сприяє швидкому виконанню запитів. Процес нормалізації використовується для групування атрибутів спеціальними засобами, для мінімізації збитковості і функціональної залежності даних. Для сховища даних цей процес має суб'єктивний характер, оскільки висока нормалізація даних уповільнює виконання запитів.

Інформаційна структура сховища даних. Сховище даних є одним з елементів інформаційної інфраструктури ДПС України. Сховище даних буде побудовано з використанням відповідної програмно-апаратної платформи, специфікація якої розробляється на підставі детального аналізу вимог користувачів до нього. Супроводжуватиметься сховище даних ДПС ІТ-персоналом Департаменту інформатизації процесів оподаткування ДПА України.

До інформаційної інфраструктури ДПС, яка включає сховище даних, у загальному вигляді входять:

- існуючі оперативні системи (АРМи та АІС) як джерела даних;
- проміжне середовище як місце очищення та перетворення даних;
- аналітичне ППЗ як засоби створення запитів до даних та отримання звітів, результатів розрахунків складних математичних та статистичних моделей;
- вітрини даних як сукупність даних відповідної сфери діяльності;
- сховище даних як сукупність інтегрованих даних, що є джерелом даних для вітрин даних та аналітичної обробки інформації ДПС.

Опис інформації. Важливий аспект у розробці сховища даних пов'язаний із створенням репозиторію метаданих – "дані про дані". Застосовуючи дане визначення до сховищ даних, мають на увазі, що метадані – це "мапа" розташування даних у сховищі даних.

Виділяють три види метаданих сховища:

1) метадані оперативних систем – використовуються в процесах управління завантаженням та доступом до джерел даних;

2) метадані кінцевих користувачів – описують розміщення та структуру даних, об'єми даних та алгоритми, тобто є навігатором по даних сховища для кінцевого користувача;

3) метадані управління сховищем – включають усі правила отримання, очищення, перетворення, агрегування, передачі та завантаження даних до сховища даних та належать як до даних, що розміщують у самому сховищі даних, так і до проміжного середовища, де дані готуються до завантаження чи вивантаження.

Послуги, що надаються кінцевому користувачу. Для побудови сховища даних використовуються декілька варіантів представлення даних і засобів їх аналізу. Основними вважаються:

- OLAP (On-line Analytical Processing), інтерактивний оперативний аналіз даних – застосування, що підтримує багатовимірне представлення даних з метою оперативного аналізу і підготовки звітів, полегшує навігацію користувачів у множині вимірів та ієрархій всередині вимірів;
- Reporting – формування звітів за шаблонами попередньо визначених форм та за довільними формами;
- Ad-Hoc Reporting – отримання інформаційних виборок у відповідь на довільні запити аналітиків;
- Business Intelligence, BI – процес перетворення даних в інформацію, а інформацію в знання, тобто метою BI є перетворення множини даних у корисну для роботи інформацію;
- Data Mining – інтелектуальний аналіз даних – процес аналізу великих масивів даних застосовується для виявлення зв'язків між різними їх елементами та пошуку схованих закономірностей;
- побудова складних математичних та статистичних моделей.

Служби доступу до даних. Основним призначенням служб доступу до даних є надання програмним засобам інтерфейсу, який дозволяє передавати SQL-запити до сховища даних для отримання потрібної інформації. При розробці власних застосувань вибір типу та постачальника інтерфейсу доступу до даних (JDBC, ODBC, ADO, OLE DB) залежить від обраної програмної платформи сховища даних. Тип інтерфейсу доступу до даних додатково залежить від типу об'єктів, до яких звертається застосування (наприклад, для доступу до каталогів OLAP на СКБД ORACLE доречно використовувати інтерфейс JDBC:OCI). Для стандартизації доступу до ресурсів даних рекомендовано використовувати служби імен та каталогів (JNDI, ODI, AD), вибір яких теж залежить від обраної програмної платформи.

Служби ETL. До категорії служб ETL належать такі служби, які забезпечують процеси регулярних (регламентних) завантажень даних з оперативних систем та визначених джерел до сховища даних за попередньо визначеною схемою.

Для виконання регулярних завантажень можливо використовувати один з методів – створення власних програм (алгоритмів), використання готового спеціалізованого інструментарію ETL, використання інструментарію СКБД або операційних систем. Ініціювання завантаження може відбуватись як за розкладом, так і за системною подією.

Планування і побудова сховища даних. Для побудови сховища даних планується використати методологію ітераційного швидкого методу розробки (RAD). Його перевага полягає в тому, що проєкт розпочинається з реалізації простої задачі, кожна наступна ітерація покращує і розширює реалізацію початкової задачі.

Побудова сховища даних складається з таких етапів: планування, опис вимог, аналіз, проєктування, побудова, тестування, введення в експлуатацію. Ці етапи відображають реальний життєвий цикл будь-якої інформаційної системи. Деталізований опис життєвого циклу сховища даних наведено в табл. 3.2.

Таблиця 3.2

Деталізований опис життєвого циклу сховища даних

Етапи побудови сховища даних	Деталізація етапів побудови сховища даних (рекомендований підхід)
1	**2**
Планування	– визначення основних бізнес-цілей, для досягнення яких реалізується проєкт; – визначення команди розробників та виконавчого керівника; – оцінка умов реалізації; – визначення основних джерел інформації; – визначення показників успішності побудови сховища даних
Опис вимог	– визначення предметної області та цілей розробки; – визначення технічної архітектури та архітектури сховища даних; – методи доступу до даних
Аналіз	– вимоги користувачів до інформації, збору даних та доступу до даних; – визначення реляційної чи багатомірної моделей сховища даних та вітрин даних; – визначення відповідних інструментів для компонентів збору даних, якості даних, адміністрування, метаданих та доступу до даних
Проєктування	– перетворення вимог, визначених на етапі аналізу у специфікації; – повна інсталяція технічної архітектури; – створення дизайнів бази даних сховища даних (логічна та фізична моделі), репозиторію метаданих; – моделювання процесів початкового та регламентного завантаження даних; – створення планів тестування: інтеграційного, системного, регресивного, об’ємів, ad hoc запитів
Побудова	– створення структур бази даних, модулів збору даних, модулів адміністрування сховища, модулів метаданих, модулів доступу до даних та звітів і запитів; – оптимізація структур бази даних для відповідності стандартам проєктування та цілям виконуваних робіт; – передача документації для використання та супроводу сховища даних

Продовження таблиці 3.2

1	2
Тестування	– тестування продуктивності; – тестування якості даних; – тестування процедур та регламенту ведення сховища даних; – тестування застосувань, засобів доступу до даних та інтерфейсу користувача
Введення в експлуатацію	– інсталяція сховища даних; – навчання користувачів роботи зі сховищем даних та моніторинг ефективності і доступу кінцевих користувачів; – подальший розвиток сховища даних з урахуванням потреб користувачів

Запитання для самоконтролю

1. Що таке база даних, СУБД? У чому полягає відмінність між цими поняттями?

2. Що таке «модель даних»? Перерахуйте різновиди моделей даних.

3. Чи можна використовувати термін «база даних» і «банк даних» як еквівалентні?

4. Що розуміють під терміном «сутність»?

5. Що розуміють під терміном «атрибут»?

6. Як позначаються сутності на ER-діаграмі за методом Чена? Інформаційного методу?

7. Що являє собою ідентифікатор сутності?

8. Як позначається на ER-діаграмі за інформаційним методом ідентифікатор сутності?

9. Які типи взаємозв’язків (взаємовідносин) можуть існувати між сутностями і як вони позначаються на ER-діаграмі за методом Чена?

10. Як позначаються взаємовідносини на ER-діаграмі за інформаційним методом?

11. Які сутності називають слабкими і як вони позначаються на ER-діаграмі?

12. Як зображуються взаємовідносини між сутностями на ER-діаграмі за методом Чена?

13. Чим датологічні документальні моделі відрізняються від фактографічних?

14. Як виконати сортування записів у базі даних?

ЛЕКЦІЯ 4
РОБОТА З БАЗОЮ ДАНИХ У СЕРЕДОВИЩІ MS ACCESS 2013

4.1. Технології конструювання бази даних та обробки економічної інформації в СУБД Microsoft Access 2013

База даних – це сукупність структурованих взаємозалежних даних, призначена для забезпечення інформаційних запитів у визначеній предметній області. Підприємства вносять відомості у бази даних і з допомогою систем управління базами даних виконують оперативну обробку економічної інформації.

СУБД Access дозволяє керувати всіма відомостями з одного файлу бази даних, що містить всі об'єкти бази даних. Об'єкти – це компоненти, що використовуються для збереження і представлення даних.

Таблиця – це об'єкт, що використовується для збереження даних. Кожна таблиця містить у собі інформацію про об'єкт визначеного типу.

Форма – це об'єкт, призначений здебільшого для введення даних, відображення їх на екрані чи керування роботою додатка.

Запит – це об'єкт, що дозволяє користувачу одержати потрібні дані з однієї чи декількох таблиць.

Звіт – це об'єкт, призначений для створення документа, що згодом може бути роздрукований чи включений у документ іншого додатка.

Microsoft Access являє собою потужний інструмент, призначений для створення й роботи з базами даних (БД).

Реляційна модель даних подає дані у вигляді множини таблиць. Структура бази даних визначається встановленням зв'язків між таблицями замість моделювання схеми даних відповідно до способу їхнього фізичного розташування.

Перевага її полягає у тому, що для кожного самостійного набору даних створюється власний файл. У нашому прикладі бази даних підприємства це був би окремий файл найменувань

товарів й окремий файл постачальників. У файлі найменувань товару адреса постачальника вказується у вигляді посилання – номера блоку даних файлу постачальників, що містить реальну адресу. Цим номером однозначно визначається блок даних (адреса) у файлі постачальників.

Номер блоку даних називається також первинним ключем. Багато програм банків даних, у тому числі й Access, можуть призначати такий номер автоматично, наприклад у міру уведення записів у базу – просто нумеруючи їх.

Проєктування таблиці. Після створення нового банку даних за допомогою директиви File/New Database (Створити новий банк) або відкриття існуючого банку за допомогою File/Open Database (Відкрити банк) на екрані в рамках вікна Access з'являється вікно банку даних.

У меню File виберіть директиву New (Створити новий об'єкт), а в підменю – опцію Table (Таблиця).

Призначення імен полів. Кожен рядок специфікації визначає характеристики одного поля запису. У колонку Field Name задається ім'я поля. Воно може мати довжину до 64 символів і містити кирилицю, пробіли й спеціальні символи, за винятком крапок, знак оклику й кутових дужок. Природним обмеженням є заборона на наявність в одній таблиці двох полів з однаковими іменами.

Установка типу даного поля. Тип даних вноситься в стовпчик Data Type, причому його можна вибрати зі списку доступних типів: Text, Memo, Number, Date/Time, Currency, Counter, Yes/No, OLE Object.

Визначення розміру поля. Для числових полів параметр Field Size може мати одне з таких значень:

- Byte – зберігає числа від 0 до 255 (тільки цілі), займає 1 байт;
- Integer – зберігає числа від –32768 до 32767 (тільки цілі), займає 2 байти;
- Long Integer – зберігає числа від –2147483648 до 2147483647 (тільки цілі), займає 4 байти;
- Single – зберігає числа із шестирозрядною точністю від 3,402823E38 до 3.402823E38, займає 4 байти;

- Double – зберігає числа з десятирозрядною точністю від – 1.79769313486232E308 до 1,79769313486232E308, займає 8 байтів (стандартна установка);
- визначення параметрів поля. Характеристики кожного поля визначаються рядом параметрів. Ці параметри регламентують способи обробки, збереження й індикації даних:
- Field Size (Розмір поля) – установлює максимальну довжину текстового поля або спосіб подання чисел у поле типу Number;
- Format (Формат) – визначає спосіб подання даних. Поряд з визначеними форматами допускається використання власних форматів користувача;
- Decimal Places (Десяткові розряди) – встановлює кількість розрядів праворуч від десяткової коми;
- Caption (Напис) – визначає напис, що буде використовуватися як найменування поля у формулярі або звіті. Якщо для цього параметра не буде задано значення, то замість напису буде за замовчуванням використане ім'я поля;
- Default Value (Значення за замовчуванням) – встановлює значення, що буде автоматично уведено в поле при генерації блоку даних;
- Validation Rule (Обмеження уведення) – правило, що обмежує припустимі для введення в поле дані;
- Validation Text (Повідомлення про порушення) – про спробу ввести в поле дані, які не задовольняють правилу, сформульованому в Validation Rule;
- Indexed (Індексовано) – ознака індексування.

Додавання й видалення полів. У готову специфікацію можна вносити зміни. Зокрема, можна змінювати параметри окремих полів, додавати поля в запис у потрібних місцях і видаляти зайві. Але при цьому варто постаратися внести всі виправлення в специфікацію до початку заповнення банку даних, тому що спроба змінити параметри полів заповненої бази може викликати втрату або перекручування даних.

1. Якщо видаляє поле, що містить дані, то з появою попереджуючого повідомлення з питанням про те, чи дійсно користувач хоче виконати видалення, натисніть на кнопку *Cancel*.

2. З меню *Edit* виберіть директиву Undo Delete (Скасувати видалення). Однак скасувати операцію видалення й відновити вихідний стан таблиці можна тільки у випадку, якщо після видалення не було зроблено жодних змін у структурі або вмісті бази. Access гарантує можливість скасування, але тільки для останньої виконаної операції.

3. Закрийте вікно таблиці й натисніть командну кнопку *No* (Немає) у діалоговому вікні запиту про необхідність збереження змін. Однак у цьому випадку не беруться до уваги й всі інші зміни, виконані за цей сеанс роботи з таблицею.

Установка первинного ключа. Після того як дане визначення всім полям, варто вибрати щонайменше одне поле для використання як первинний ключ. Оголошення первинного ключа перешкоджає уведенню повторюваних блоків даних, оскільки поле таблиці, використовуване як первинний ключ, містить однозначний ідентифікатор для кожного блоку даних. Це поле не може містити однакову величину у двох різних записах.

Первинний ключ може бути визначений тільки в режимі проєктування таблиці. Промаркуйте поле, що повинне стати полем первинного ключа й викличте директиву Set Primary Key (Установити первинний ключ) у меню *Edit*. Промарковане поле негайно позначається піктограмою ключа в селекторному стовпчику (це і є ознакою того, що поле оголошене первинним ключем) і відповідно індексується.

Якщо до моменту виходу з режиму проєктування первинний ключ для створюваної таблиці не буде оголошений, то Access запитає, чи варто включити в таблицю поле первинного ключа. Якщо користувач відповість позитивно (Yes), то Access створить особливе поле з ім’ям ID, у яке для кожного блоку даних буде вводитися свій оригінальний номер.

Створення нової БД. Для того щоб створити нову БД, необхідно запустити Microsoft Access 2013. Можна скористатися декількома способами:

1) пуск – програми – Microsoft Office – Microsoft Office Access 2013;

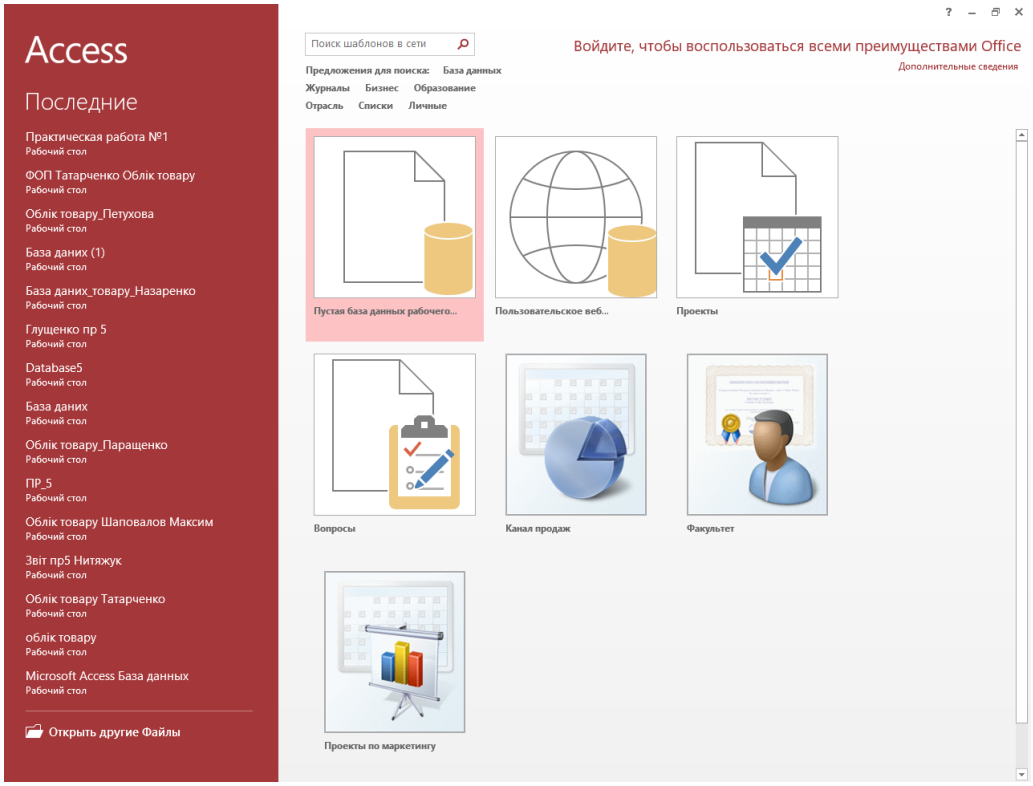


Рис. 4.1. Запуск Microsoft Office Access 2013

2) клацнути правою клавішею миші по вільному місцю – створити – Microsoft Office Access 2013.

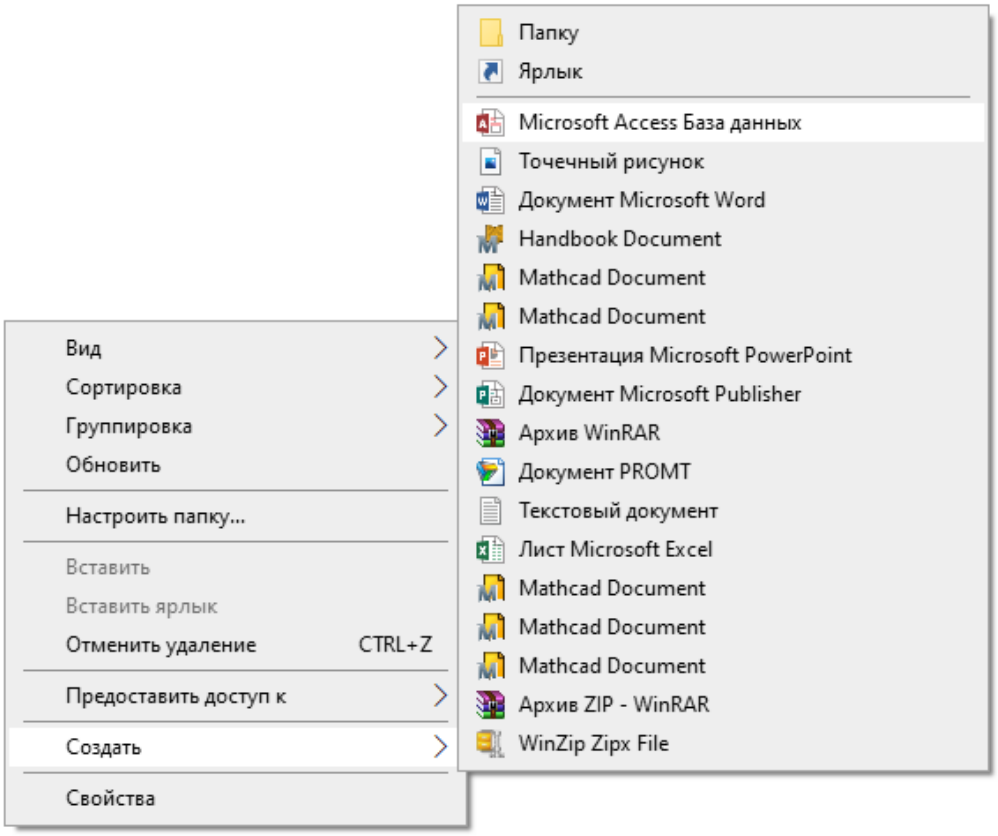


Рис. 4.2. Створення бази даних в Microsoft Office Access 2013, 1-й крок

Після того, як завантажено Microsoft Access 2013, створюємо нову БД:

1) натиснути меню *Файл* – команда *Создать*;

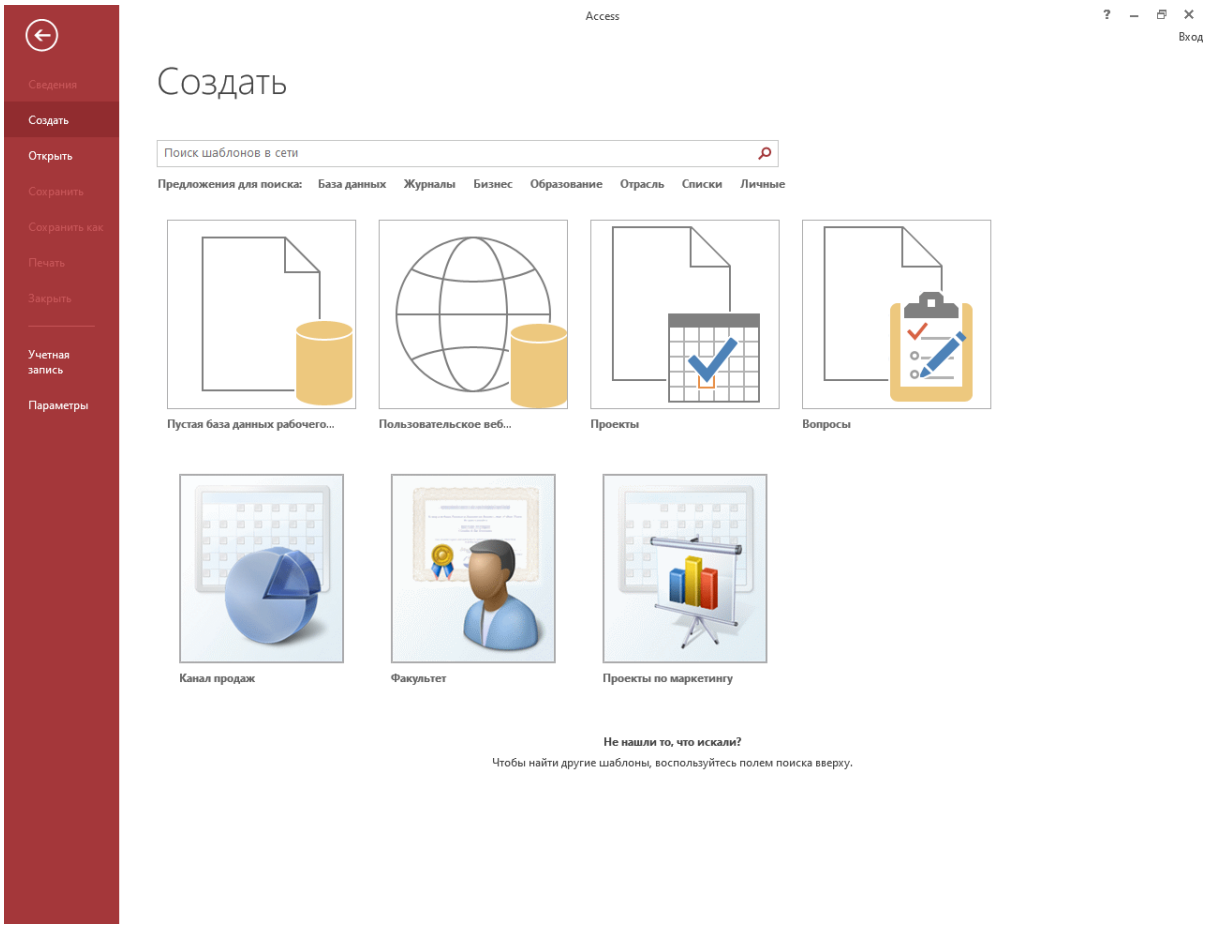


Рис. 4.3. Створення бази даних у Microsoft Office Access 2013, 2-й крок

2) у верхній частині вікна необхідно виконати команду *Новая база данных*;

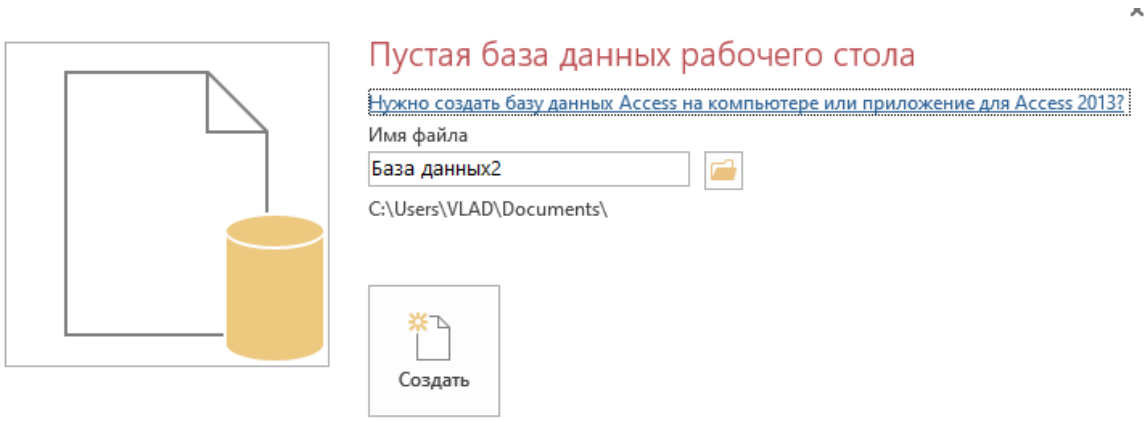


Рис. 4.4. Створення бази даних у Microsoft Office Access 2013, 3-й крок

3) тепер необхідно задати ім’я базі даних та шлях її збереження.

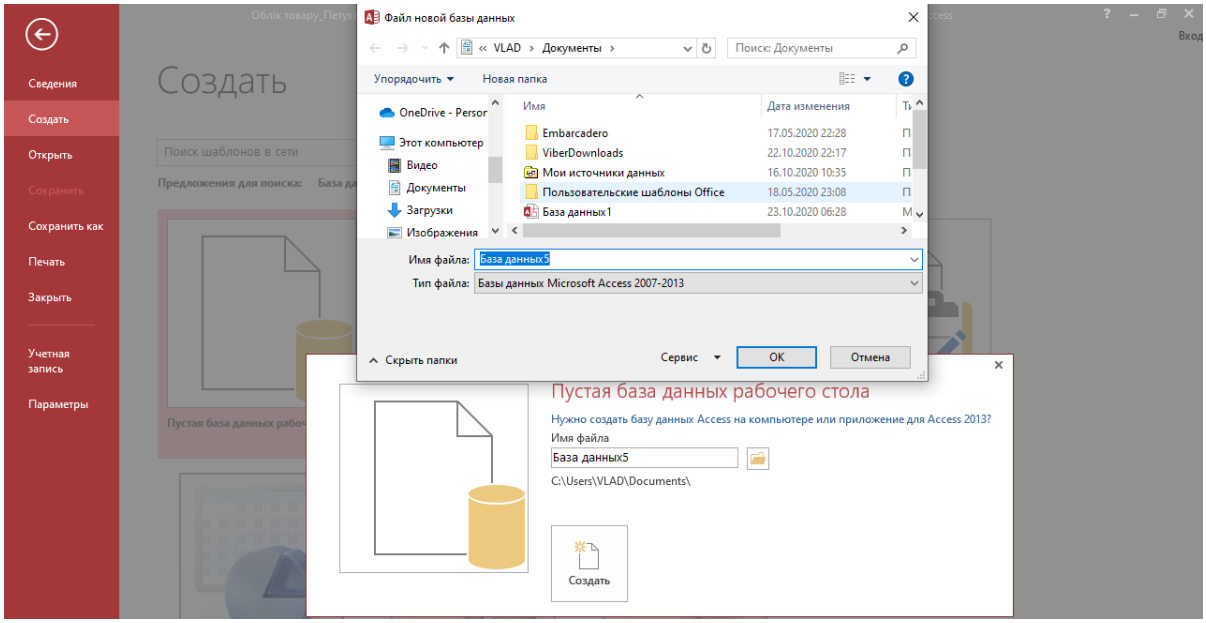


Рис. 4.5. Створення бази даних в Microsoft Office Access 2013, 4-й крок

У цьому вікні діалогу зі списку, що розкривається, потрібно вибрати теку, в якій зберегти створювану базу даних, а в полі *Ім’я файлу* – ввести ім’я бази даних. Після цього можна створити за допомогою *Майстра* базу даних певної структури зі всіма необхідними таблицями, формами і звітами.

Інтерфейс Microsoft Access 2013. Після створення нової БД можна розглянути інтерфейс програми. Вікно програми складається з вікна програми та робочого вікна.

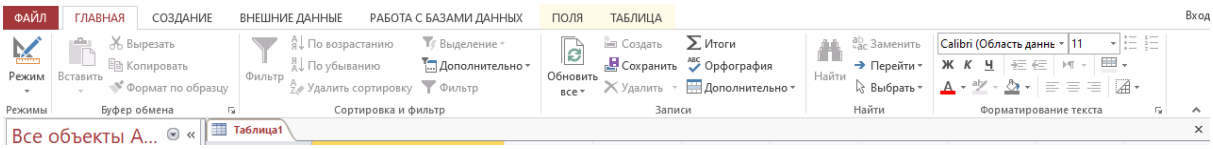


Рис. 4.6. Інтерфейс Microsoft Office Access 2013

Панель інструментів має такі команди: створити, відкрити, зберегти, пошук файлів, друк, попередній перегляд, орфографія, вирізати, копіювати, вставити, відміна, зв’язки з Office, аналіз, програма, редактор сценаріїв, властивості, схема даних, автоформа, довідка.

Створення таблиць. Отже, на наступному кроці створюються таблиці, в які згодом вводитиметься інформація. Надалі дані в таблиці можуть доповнюватися новими даними, редагуватися або виключатися з таблиці. Такі дані можна переглядати в таблицях або упорядковувати їх за деякими ознаками.

Інформація, що міститься в таблицях, може бути використана для складання звітів. Крім того, можна задати графічну інтерпретацію інформації, що міститься в базі даних. З вирішенням цих завдань можна познайомитись у подальших главах.

Створення таблиці в MS Access 2013 здійснюється у вікні бази даних.

Таблиця – це об'єкт бази даних, набір інформації з конкретної теми. Дані в таблиці розміщені в стовпцях та рядках.

Послідовність дій при створенні таблиці в новій базі даних:

1) необхідно відкрити вікно створеної бази даних і перейти на вкладку *Таблиці*;

2) натиснути кнопку *Создание* у вікні бази даних.

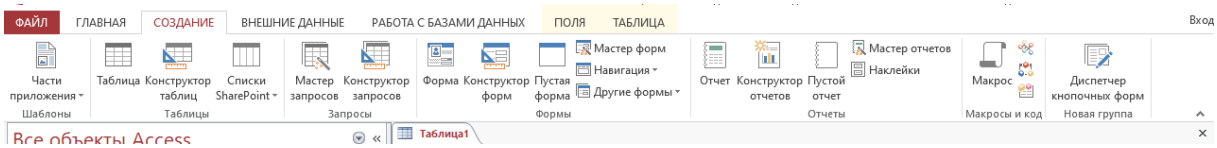


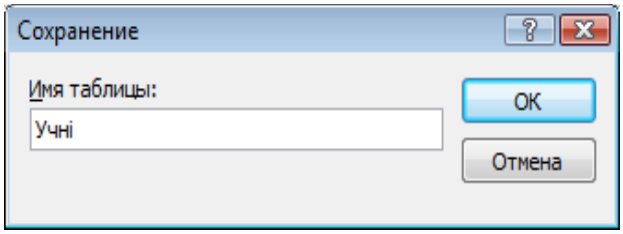
Рис. 4.7. Створення таблиці бази даних

Відкриється вікно діалогу *Таблицы*, в правій частині якої розміщений список варіантів подальшої роботи:

- режим таблиці – дозволяє створити нову таблицю в режимі таблиці;
- конструктор – дозволяє створити нову таблицю в конструкторі таблиць.

Потрібно вибрати з цієї таблиці відповідний варіант створення таблиці і натиснути кнопку *ОК*.

Для того щоб пов'язати таблицю з інформацією, що міститься в ній, кожній таблиці привласнюється ім'я. У вікні діалогу *Збереження* задається ім'я, після чого потрібно натиснути кнопку ОК.



Ім’я таблиці, так само як і імена бази даних, можна не обмежувати вісьмома символами. Ім’я таблиці, як і імена інших об’єктів бази даних, зберігається в самій базі даних.

Створення таблиці за допомогою Майстра. MS Access 2013 містить цілий ряд таблиць, які можна використовувати як прототип потрібних таблиць. При використанні майстра можна не лише заощадити час на створенні таблиць, але і забезпечити стандартні імена і типів даних полів таблиць.

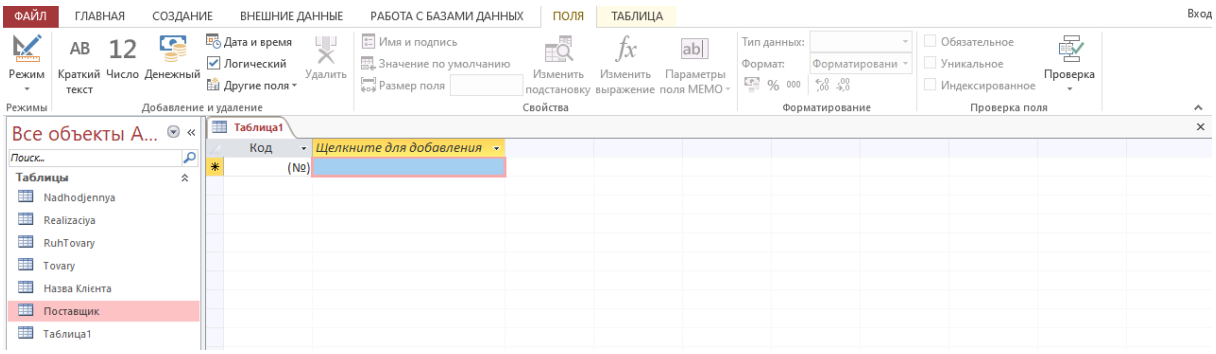


Рис. 4.8. Створення таблиці за допомогою майстра

Безпосереднє введення даних у таблицю – після завершення робіт *Майстра* на екрані відкривається режим таблиці, що дозволяє здійснювати введення даних у таблицю.

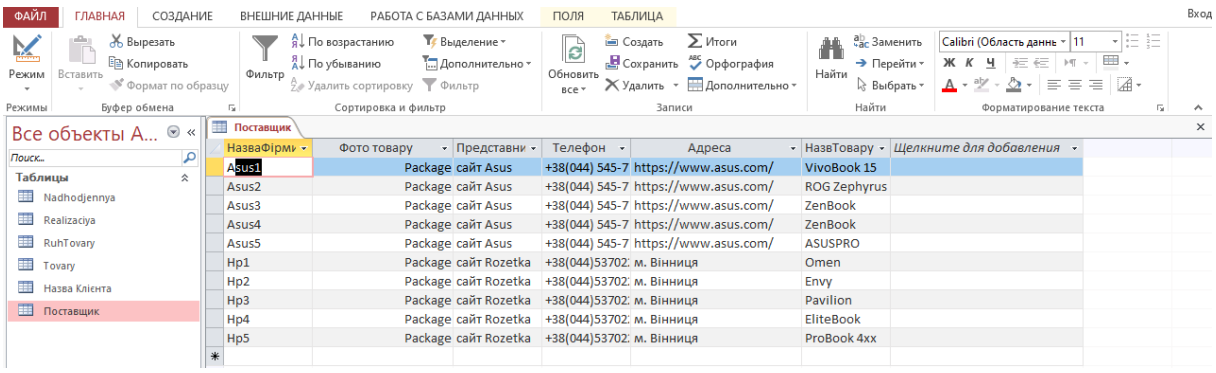


Рис. 4.9. Введення даних у таблицю

Створення таблиці у вікні Конструктора. Створення таблиць у режимі Конструктора надає більш широкі можливості з визначенням параметрів створюваної таблиці. Щоб перейти в режим Конструктор відкритої таблиці, потрібно відкрити список режимів, вибрати із списку варіантів значення Конструктор і натиснути кнопку ОК. У результаті виконання цих дій відкриється вікно Конструктора таблиці.

У верхній частині вікна діалогу розміщена таблиця, яка містить такі атрибути створюваної таблиці: найменування поля, тип даних і опис. Крім цих основних атрибутів, кожне поле таблиці володіє додатковими властивостями, що відображуються в нижній частині конструктора і визначають умови введення даних. Найменування кожного з полів таблиці, як правило, вибирається довільно, але так, щоб відобразити характер інформації, яка зберігатиметься в цьому полі. Тип поля визначається типом даних, що зберігаються в цьому полі.

Найменування поля. При завданні найменувань полів потрібно слідувати таким правилам:

- найменування поля може містити до 64 символів, але не слід зловживати цією можливістю, задаючи довгі імена;
- найменування поля може містити букви, цифри, пропуски і спеціальні символи, за винятком крапки (.), знаку (!) оклику, прямих дужок ([]) і деяких символів, що управляють (з кодами ASCII 0-31);
- найменування поля не може починатися з пропуску;
- два поля в одній таблиці не можуть мати однакових найменувань;
- для найменування поля не рекомендується використовувати абревіатури або короткі назви.

Недотримання цих правил відстежується засобами СУБД MS Access 2013, але в деяких випадках це може призвести до важко визначуваних помилок, тому рекомендується на практиці самостійно контролювати дотримання вищеперелічених правил. Бажано використовувати імена, що відрізняються стислістю – для полегшення їх ідентифікації при перегляді таблиць.

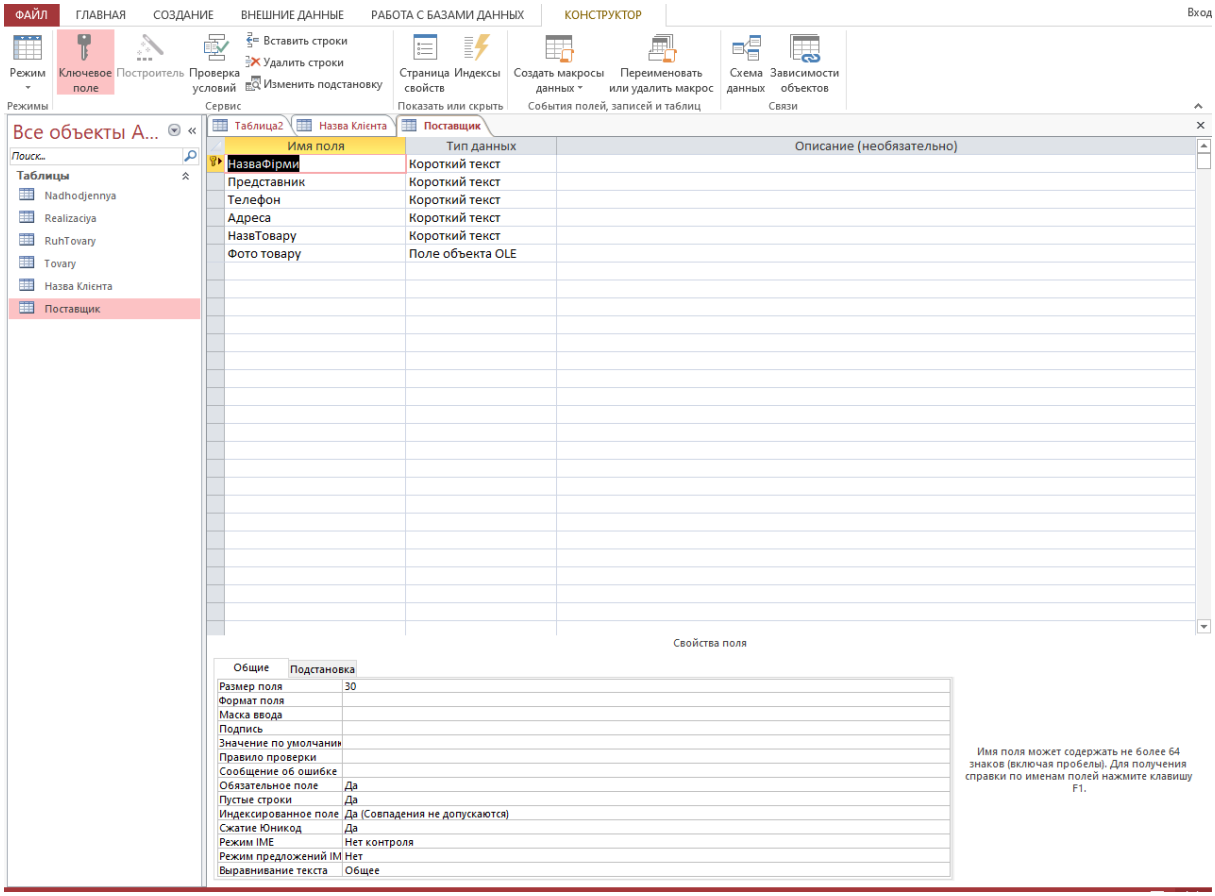


Рис. 4.10. Створення таблиць у вікні Конструктора

Типи даних. Найменування поля використовується для посилання на дані таблиці. Для визначення типа даних, що зберігаються, використовується тип даних. Тип даних поля вводиться в поле введення стовпця Тип даних. У MS Access 2013 допустимими є дані таких типів:

- текстовий – символьні, текстові дані, об'єм яких не повинен перевищувати 255 символів, за умовчанням 50;
- числовий – символьні дані, за умовчуванням 50;
- грошовий – Різновид типу даних для зберігання, грошових еквівалентів, розміром 15 розрядів до коми і 4 розряди після;
- лічильник – є різновидом числового типу, може мати лише ціле значення, автоматично збільшується на заданий крок, за умовчанням +1, значення не можуть повторюватися;
- дата/час – відображає дані дати або часу;
- логічний – призначений для зберігання логічних значень, для команд і операцій: хиба/істина, так/ні, true/false, 1/0;
- поле МЕМО – текстовий тип даних, обмеження до 64 000 символів, поля цього типу не індексуються;

- поле об’єкта OLE – призначений для зберігання об’єктів (файлів), які створені в інших застосуваннях. Поле не індексується;
- майстер підстановок – дозволяє підставляти значення полів у вигляді простого або комбінованого списку;
- для зміни ширини поля потрібно в рядку «Розмір поля» розділу «Властивості поля» задати число, що визначає ширину поля (від 1 до 255).

Кожен з типів даних наділений власними властивостями, які відображуються у розділі «Властивості поля» вікна Конструктора.



Рис. 4.11. Властивості полів бази даних

Числові поля. У текстових полях допустимі символи, цей тип поля може містити цифри. Як основні аргументи щодо використання числових полів, приведемо такі міркування:

Під час введення даних числового типу автоматично відбувається перевірка даних. Якщо дані містять текстові або спеціальні символи, MS Access 2013 видасть попередження, – і помилкові дані не будуть внесені.

Лише над числовими полями можливе виконання математичних операцій.

Перш ніж встановити розмір поля потрібно поміркувати, які значення зберігатимуться в ньому. Вибравши оптимальне значення, заощаджується місце для зберігання даних. Для вказівки кількості десяткових знаків використовується властивість поля Число десяткових знаків. Воно може набувати значення від 0 до 15. Атрибут *Авто даної властивості* слугує для автоматичної установки кількості знаків після коми. Використовуючи значення властивості *Розмір поля*, можна встановити такі формати для полів числового типу:

Поля типу Лічильник. Поля типу Лічильник призначені для зберігання даних, значення яких не редагуються, а встановлюються автоматично у разі додавання кожного нового; запису в таблицю. Їх значення є унікальними, послідовно зростаючими на 1 при додаванні кожному новому запису або можуть бути будь-якими випадковими числами. Такі поля дозволяють проводити автоматичну перенумерацію даних незалежно від того, в яке місце таблиці вставляються дані, що буває зручно при веденні різних кодифікаторів, ключів і впорядкованих масивів даних, оскільки це створює умови для вживання алгоритмів швидкого сортування і пошуку даних.

Поля дати/часу. Використовуються для зберігання дат і часу полів, в яких дані подані в спеціальному форматі, надає ряд додаткових переваг. MS Access 2013 дозволяє вводити і відображувати дати в декількох форматах. Конкретний варіант відображення дати/часу встановлюється у властивості *Формат поля* вікна *Конструктора таблиць*. Під час зберігання даних ці формати автоматично перетворяться у внутрішнє представлення даних. При введенні дат MS Access 2013 здійснює перевірку формату даних і допускає введення лише правильних дат. У полі цього типу може міститися будь-яка дата і будь-який час: від 1 січня 100 р. до 31 грудня 9999 р. Дані цього типу займають 8 байт.

Логічні поля. Логічні поля використовуються для зберігання даних, які можуть приймати одне з двох можливих значень. Властивість *Формат поля* для логічного поля дозволяє використовувати спеціальні формати або один з трьох вбудованих: *Істинне/Хибне, Так/Ні або Вкл/Викл*. При цьому значення *Істинне, Так і Вкл* еквівалентні логічному значенню True, а значення Хибне, Ні і Викл еквівалентні логічному значенню False. Якщо вибрати вбудований формат, а потім вводити еквівалентне логічне значення в таблицю, то введене значення відображуватиметься у вибраному форматі. Наприклад, якщо значення *True* вводити у відповідне поле логічного типу, для якого у властивості *Формат поля* вказано *Так/Ні*, то введене значення тут же перетвориться в *Так*;

Текстові поля довільної довжини. Текстові поля довільної довжини (поля МЕМО) можуть містити ті самі дані, що і прості текстові поля. Відмінність між цими полями полягає в тому, що

довжина полів типу МЕМО може бути досить велика порівняно з розміром текстових полів. Може містити довгий текст або комбінацію тексту і чисел;

Поля об'єкта OLE. MS Access 2013 дозволяє зберігати в таблицях графічні зображення й інші двійкові дані (наприклад, електронну таблицю MS Excel 2013, документ MS Word 2013, малюнок, звукозапис), зв'язаний або упроваджений у таблицю Microsoft Access 2013. Для цих цілей слугує тип даних *Поле об'єкта* OLE. Фактичний об'єм даних, який можна ввести в поле цього типу, визначається об'ємом жорсткого диска комп'ютера (до 1 Гігабайта);

Майстер підстановок. Вибір цього типу даних запускає *Майстер підстановок*, який створює поле, в якому пропонується вибір значень зі списку, що розкривається, містить набір постійних значень або значень з іншої таблиці. Вибір цього параметра зі списку, що розкривається, запускає *Майстра підстановок*. Розмір цього поля збігається з розміром ключового поля, використовуваного як підстановка (4 байти).

Створення структури таблиці. Після розгляду типів даних в Access 2013 і окремих властивостей полів таблиці можна приступити до створення структури таблиці. Натиснути клавішу *Tab* або *Enter*, щоб перейти у стовпець *Тип даних*. При цьому зверніть увагу, що в нижній частині вікна діалогу з'являється додаткова інформація в розділі *Властивості поля*.

У стовпці *Тип даних* з'явилося значення *Текстовий*. Потрібно натиснути кнопку розкриття списку в правій частині прямокутника і побачити список, що містить усі типи даних. З цього списку за допомогою миші або клавішами вгору і вниз вибираємо значення (наприклад, *Лічильник*) і натискуємо на клавішу *Tab* для переходу в стовпець *Опис*. Стовпець *Опис* – це пояснення до значень і вмісту полів. Якщо в майбутньому працювати з цією таблицею, цей опис з'являтиметься в нижній частині екрану MS Access 2013 кожного разу, коли курсор буде у цьому полі, і тим самим нагадає призначення цього поля.

Необхідно ввести пояснюючий текст у стовпець *Опис* і натиснути клавішу *Tab* або *Enter*, щоб перейти до введення інформації про наступне поле. Аналогічно можна ввести опис усіх полів таблиці.

Завершивши введення структури таблиці, необхідно зберегти її, виконавши команду *Файл/Зберегти*.

Таблиця 4.1

Приклади виразів, які використовуються в умовах на значення

Вираз	Значення
”NY” or ”NJ” or ”PA”	Дані, які вводяться, мають бути або ”NY”, або ”NJ”, або ”PA”
Between #15/4/10# and #31/12/10#	Повинна бути правильна дата між 15/4/10 та 31/12/10
<=#15/12/10#	Повинна бути дата до 15/12/10
Like 112M####	Значення повино вміщувати вісім символів, які починаються з ”112M”
>=4.50 and <=12.30	Значення має бути між 4.50 та 12.30
<=[основна зарплата]*2	Щоб значення поля не перевищувало [основна зарплата]*2

Пов’язування таблиць. Створення зв’язків

Процес пов’язування таблиць може розпочатись на етапі визначення підстановок. Завершує цей процес робота з визначенням схеми даних. Зв’язки між таблицями можна редагувати, видаляти. Ця робота починається з натискання на значок *Схема данных*, що розташований на панелі інструментів. У результаті цього відкривається діалогове вікно *Схема данных*, де необхідно додати таблиці.

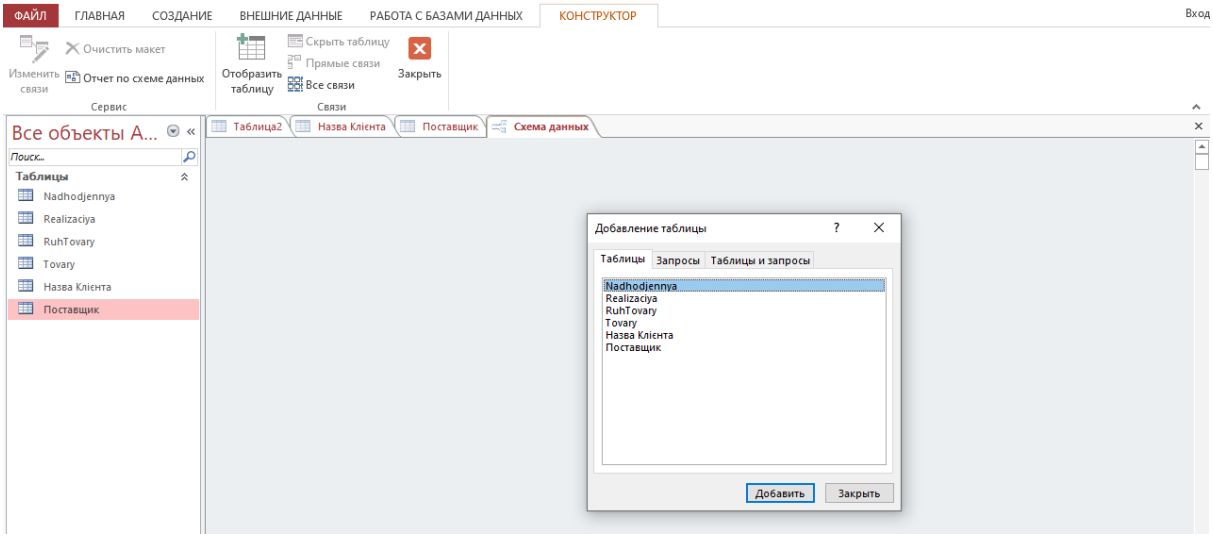


Рис. 4.12. Вікно Схема даних

Для того, щоб додати таблиці, необхідно клацнути правою клавішею миші на вільному місці та вибрати команду *Добавление таблицы*. Після виконання цієї команди відкриється діалогове вікно *Добавление таблицы*, де необхідно виділити необхідні таблиці та додати їх до схеми даних.

Після того, як таблиці додались у схему даних, можна почати зв'язування таблиць. Цей процес починається шляхом перетягування ключового поля однієї таблиці до підлеглого поля іншої.

Після перетягування ключових полів відкривається вікно «Изменения связей», де пропонується «обеспечение целостности данных», «каскадное обновление связанных полей» та «каскадное удаление связанных полей».

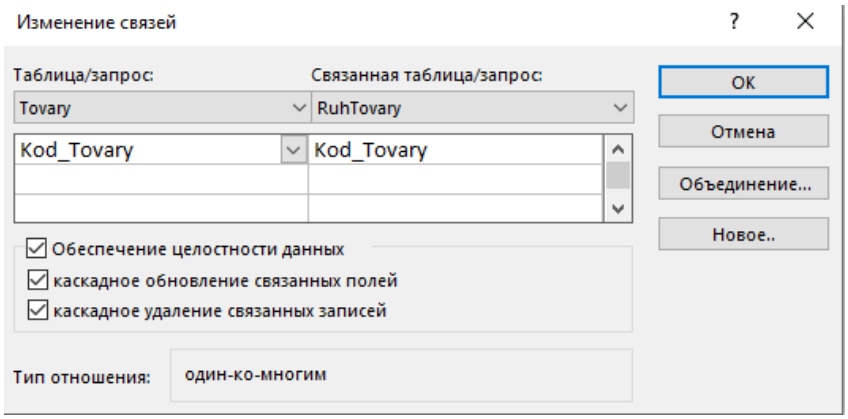


Рис. 4.13. Встановлення зв'язків між таблицями

Забезпечення цілісності даних БД означає виконання ряду обмежень, додержання яких необхідне для підтримки несуперечності даних, що зберігаються.

Якщо у діалозі *Изменение связей* в поле *Обеспечение целостности данных* встановити прапорець, а в полях *Каскадное обновление связанных полей* та *Каскадное удаления связанных полей* ні, то програма не дозволить змінити або вилучити цільову сутність за умови наявності пов'язаних з нею записів в іншій таблиці.

А якщо встановити прапорець *Каскадное обновление связанных полей*, то програма буде автоматично змінювати дані в усіх пов'язаних записах під час зміни значення цільової сутності.

Якщо встановити прапорець *Каскадное удаление связанных полей*, то програма автоматично буде вилучати пов'язані записи в процесі цільової сутності.

У результаті пов’язування всіх таблиць отримано таку схему даних:

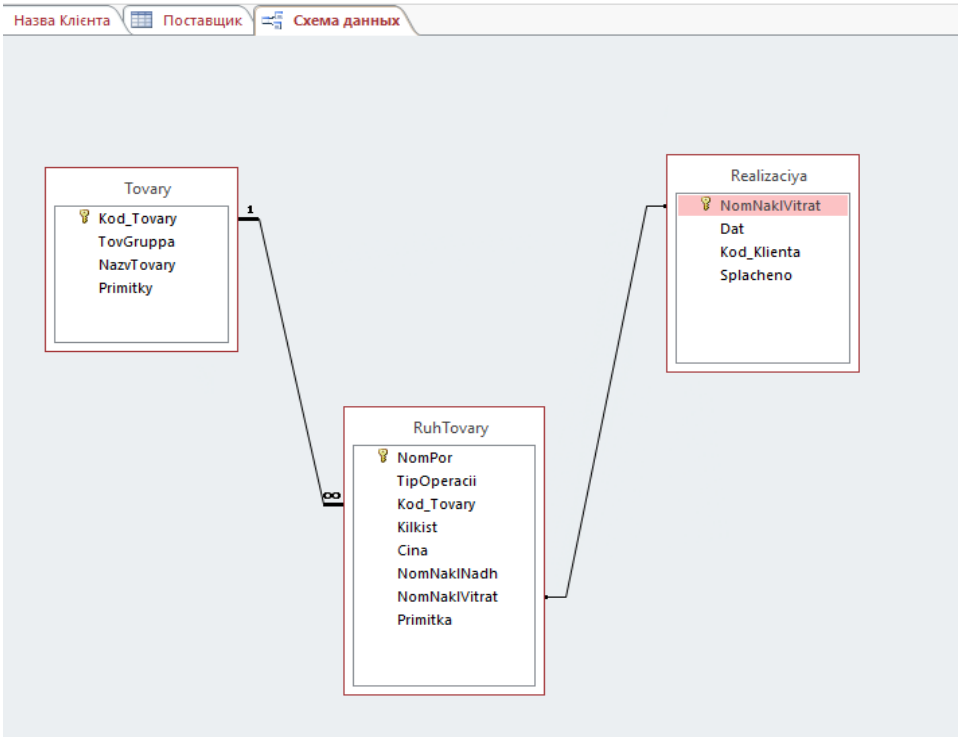


Рис. 4.14. Схема даних

Вибірка даних. На практиці часто потрібно вибрати з вихідної таблиці частину записів, що задовольняють певним критеріям, і упорядкувати вибірку. Критерії можуть визначатися поєднанням ряду умов. Наприклад, необхідно вибрати записи про певних учнів групи і упорядкувати їх в алфавітному порядку. Для вирішення таких завдань призначені майстри запитів і конструктор запитів, за допомогою яких можна:

– формувати складні критерії для вибору записів з однієї або декількох таблиць;

– вказати поля, що відображуються для вибраних записів;

– виконувати обчислення з використанням вибраних даних.

Запит за зразком. Раніше були розглянуті загальні положення, що належать до баз даних, де зазначалося, що одним з основних призначень баз даних є швидкий пошук інформації і здобуття відповідей на будь-які питання. Питання, що формулюються стосовно бази даних, називаються запитами. У MS Access 2013 для формування запитів використовується конструктор запитів і інструкція SELECT мови MS Access 2013.

Запит за зразком – це інтерактивний засіб для вибору даних з однієї або декількох таблиць. Під час формування запиту необхідно вказати критерії вибірки записів у вихідній таблиці. Щоб не набирати з клавіатури критерії відбору записів таблиці спеціальною мовою SQL, запит за зразком дозволяє просто заповнити бланк, який розташовується у вікні Конструктора запитів. Метод формування запиту шляхом заповнення бланку простий для вивчення і розуміння. Він сприяє ефективному використанню можливостей MS Access 2013 користувачами, що мають навіть мінімальні навички роботи з додатком або якщо не мають його зовсім.

Для створення простих запитів можна використовувати *Майстер запитів*, який послідовно пропонує найменування таблиць, використовуваних у запиті, перелік полів таблиць і деякі додаткові параметри. Для створення більш складних запитів може знадобитися *Конструктор запитів*, який є простим і наочним засобом створення досить складних запитів.

Створення запитів за допомогою майстра. Відмінною рисою MS Access 2013 є наявність майстрів, які допомагають під час виконання більшості дій, не вимагаючи глибоких знань у сфері баз даних. Запити також можуть створюватися за допомогою *Майстра запитів*. Для створення запиту у вікні бази даних необхідно перейти на вкладку «Запросы» і натиснути кнопку «Создать».

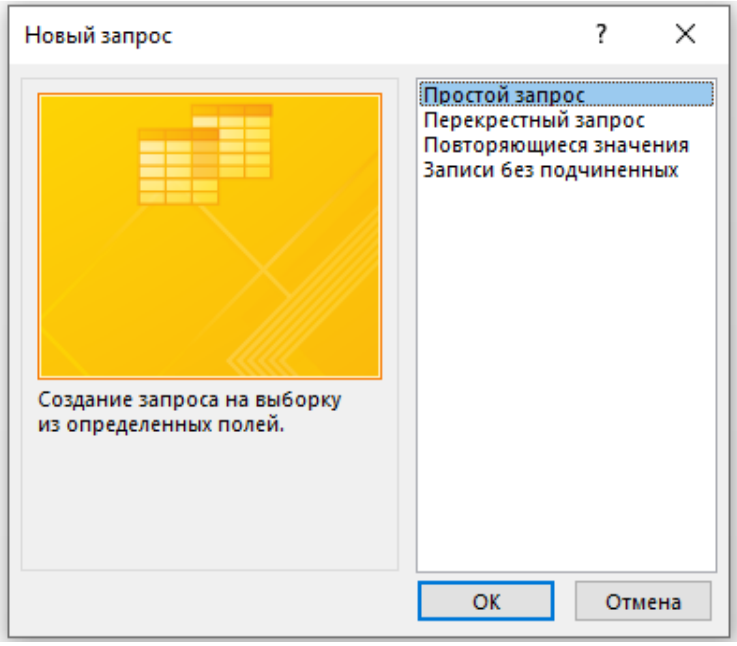


Рис. 4.15. Створення запитів за допомогою Майстра

Відкриється вікно діалогу, в якому можна вибрати опцію створення запиту за допомогою Конструктора запитів або один з майстрів створення запиту. Для знайомства із запитами краще вибрати опцію *Простий запит*.

У цьому випадку буде запропоновано вказати список таблиць і їх полів, і на підставі цих даних буде створений запит.

У разі створення простого запиту відкриється вікно майстра, в якому можна вибрати потрібні поля з однієї або декількох таблиць. Якщо вибрано більш за одну таблицю, *Майстер* автоматично зв'язує таблиці.

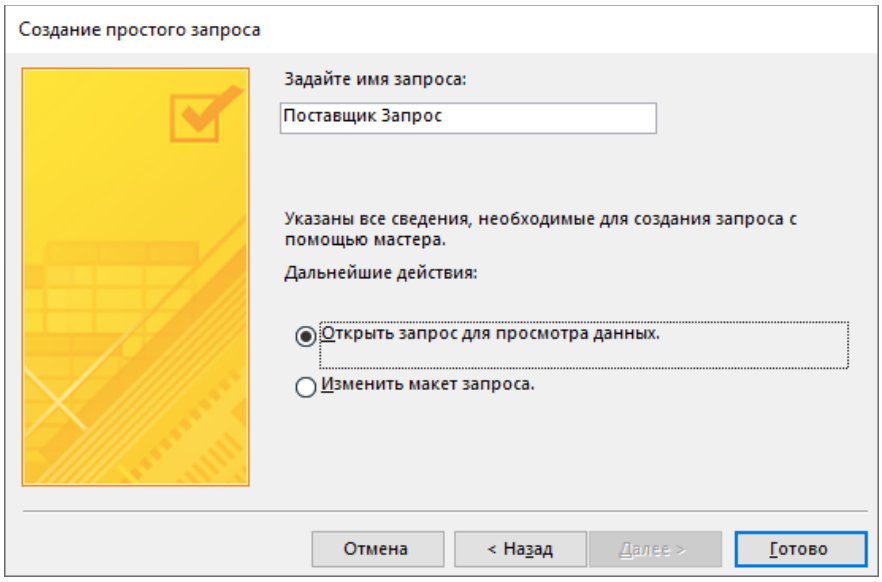


Рис. 4.16. Створення простого запиту

Після цього можна переглянути запит у режимі Таблица. Або перейти в режим Конструктор та задати певні критерії:

Поле. У цьому рядку поміщаються ті поля, які використовуються для створення запиту, кожне у своєму елементі таблиці.

Ім'я таблиці. Цей рядок показує, з якої таблиці (або запиту) вибрано це поле.

Сортування. У цьому рядку вказуєтьсч тип сортування інформації, в запиті, за збільшенням (від А до Я, від більшого до меншого, від ранішого до пізнішого і так далі) або по спаданню (від Я до А і так далі);

Вивід на екран. Якщо необхідно, щоб MS Access 2013 показував інформацію, знайдену в полі, встановіть прапорець перегляду поля. Якщо ж поле використовується лише для завдання умови вибору даних, яке повертає запит, краще залишити його порожнім.

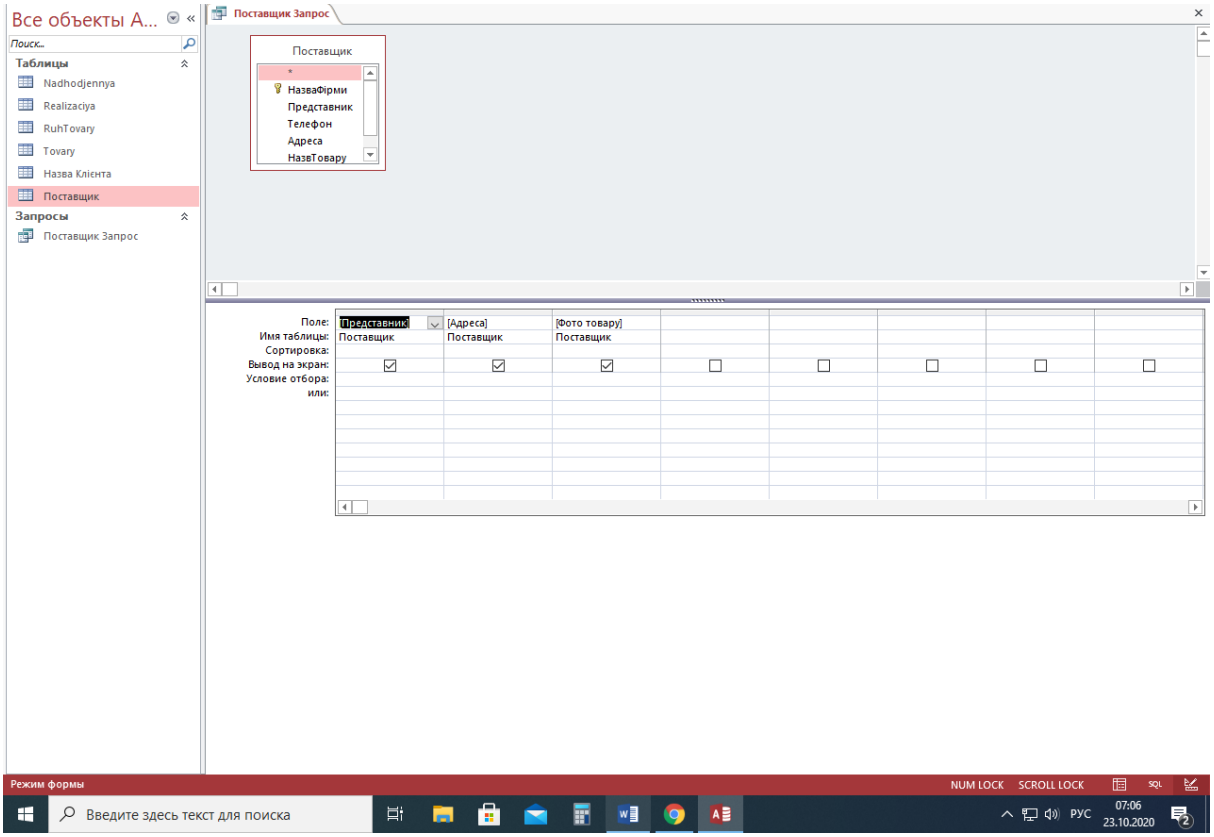


Рис. 4.17. Перегляд запиту в режимі Конструктор

Якщо, наприклад, потрібно відібрати всіх представників підприємства, які живуть у Києві, то поле *Місто* у запиті використовується для обмеження результатів вибірки; проте якщо не потрібно знову і знову переглядати значення поля з назвами інших міст, то після завершення роботи *Майстра* переходимо в режим *Конструктор* та задаємо умови відбору.

Умова відбору. Це найбільш важлива частина конструювання запиту. У цьому рядку (і в рядку, розташованому нижче за нього) вводяться обмеження пошуку – умови, які прийнято також називати критерієм пошуку.

Після відкриття запиту в режимі *Конструктор* у поле *Условие отбора* під полем *Місто* потрібно ввести *Київ*, зберегти, тоді перейти в режим *Таблиці* і переглянути результат.

Збереження запиту. Створений запит можна використовувати надалі. Для цього потрібно зберегти його. Збереження запиту здійснюється командою *Файл/Сохранить как экспорт*, який відкриє вікно діалогу для введення імені запиту. MS ACCESS 2013 за замовчуванням запропонує ім’я запиту, але краще самостійно підібрати ім’я.

Створення Форми. Для більш зручного відображення інформації, що міститься в базі даних, використовуються форми. У повсякденному житті зазвичай заповнюються форми, наприклад, коли вносять плату за квартиру або телефон.

Вживання різних елементів інтерфейсу дозволяє наочніше представити запис. Додавання до форми Windows Media Player дозволяє відтворювати файли .avi (Audio-Video Interleaved) і.asf (ACTIVEX Streaming Format) з тим, щоб користувачі могли розслабитися в процесі роботи. Як фонове зображення, можна використовувати графічні файли з розширенням .gif і jpg.. Форми зазвичай використовують для таких цілей:

– для введення нових даних у таблицю, їх редагування і видалення;

– для вибору даних, їх попереднього перегляду і для друку звіту;

– для відкриття інших форм або звітів.

Дані, наведені у формі, беруться з полів базової таблиці. Зв'язок між формою і джерелом даних можуть забезпечувати елементи управління, наприклад поля. Форму можна переглядати в трьох режимах:

– *Конструктор* (Design View);

– *Режим таблиці* (Datasheet View);

– *Режим форми* (Form View).

У відкритому вікні бази даних потрібно відкрити вкладнику *Формы* і натиснути кнопку *Создать*. У результаті відкривається вікно *Новая форма*.

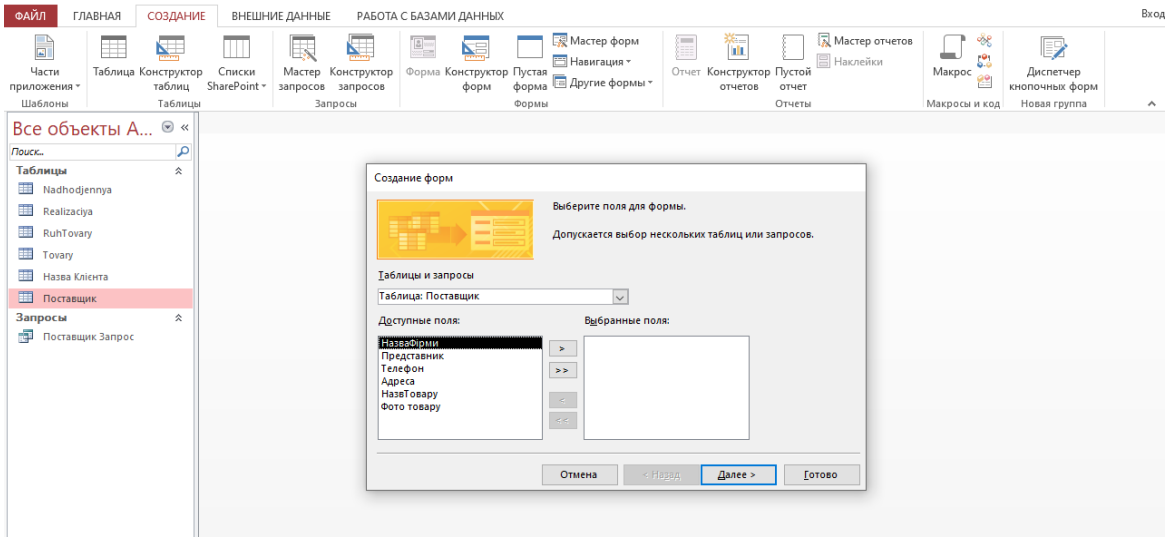


Рис. 4.18. Створення форми

У цьому вікні потрібно відкрити список таблиць і вибирати одну з них, а саме ту, з якої буде створюватися форма.

Створення першої стандартної форми – у стовпчик. Для цього виконаємо команду *Автоформа*: у стовпчик.

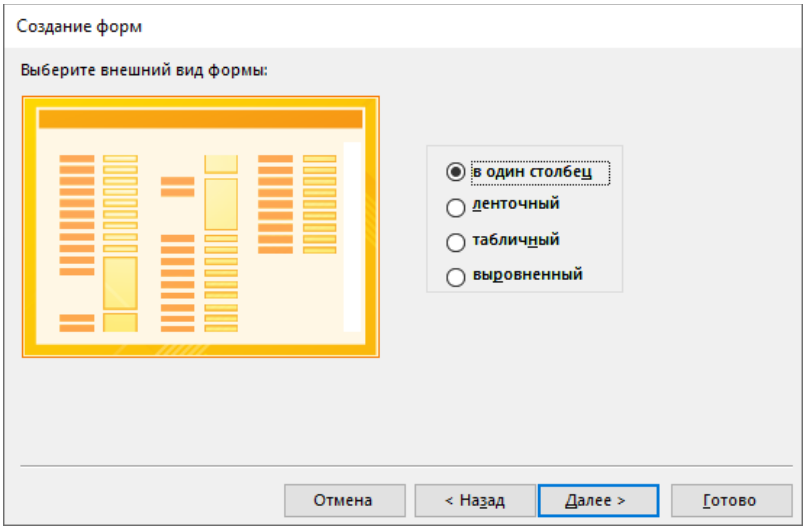


Рис. 4.19. Стандартна форма: у стовпчик

У результаті буде створена форма, зображена на рисунку.

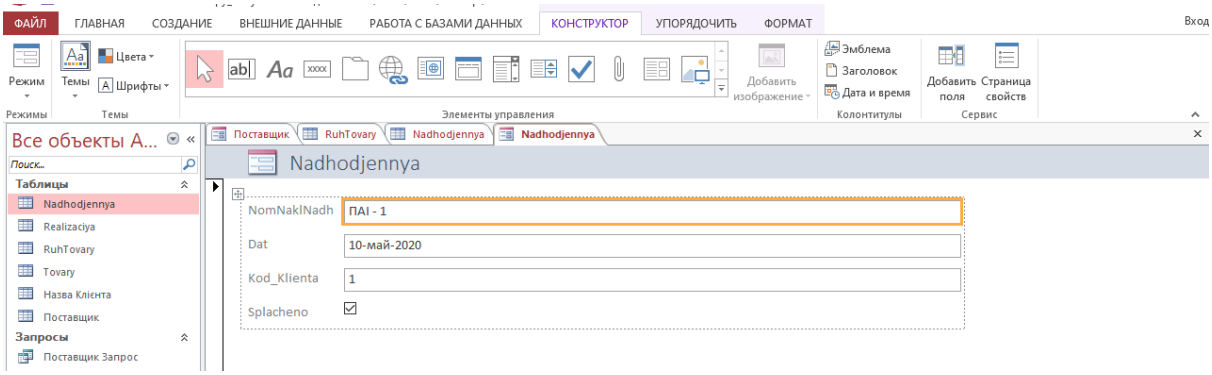


Рис. 4.19. Результат створення форми у стовпчик

У цьому вікні всі поля розташовані у стовпчик і виведено тільки один запис. Перегляд послідовних записів таблиці *Список* здійснюється за допомогою кнопок прокрутки.

Створення звітів. Звіти – представлення інформації у друкованому вигляді. Для створення звітів MS Access 2013 дозволяє використовувати такі засоби:

– Майстер звітів, який дозволяє досить швидко створювати звіт на основі відбору полів з вхідних інформаційних таблиць;

– Конструктор звітів, у якому користувач самостійно розробляє потрібний із заданими властивостями;

– Автозвіт: стрічковий, який дозволяє автоматично створювати найпростіший табличний звіт на основі однієї таблиці;

– Автозвіт: у стовпчик, який дозволяє автоматично створювати звіт з полями, назви яких розташовуються в один або декілька стовпчиків;

– Майстер діаграм;

– Поштові наклейки.

Для створення звіту будь-яким із зазначених способів необхідно виконати такі дії:

1) відкрити базу даних;

2) у вікні бази даних вибрати вкладку *Отчеты* і натиснути відповідну кнопку;

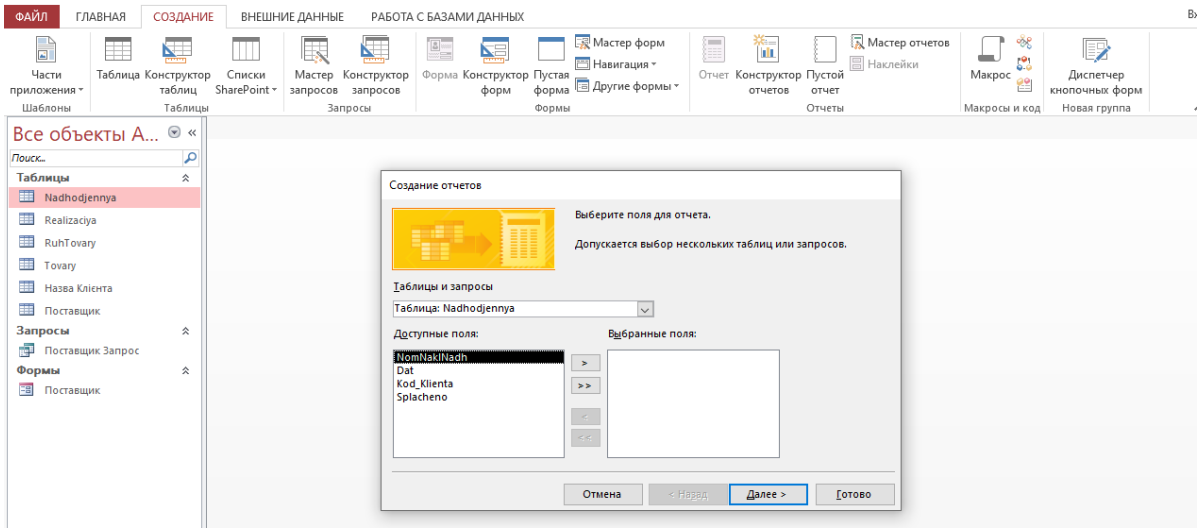


Рис. 4.20. Створення звітів

3) у діалоговому вікні *Новый отчет*, що з'явилось на екрані, вибрати один з можливих засобів формування звіту.

Створення найпростішого звіту за допомогою засобу Автоотчет. Найпростішим способом створення табличного звіту є використання засобу *Автоотчет*. З цією метою у діалоговому вікні «Новый отчет» потрібно вибрати *Автоотчет: ленточный* і як джерело для побудови звіту у цьому ж вікні вибрати таблицю або запит, на основі якого й буде побудовано звіт. На екрані з'явиться готовий до використання звіт.

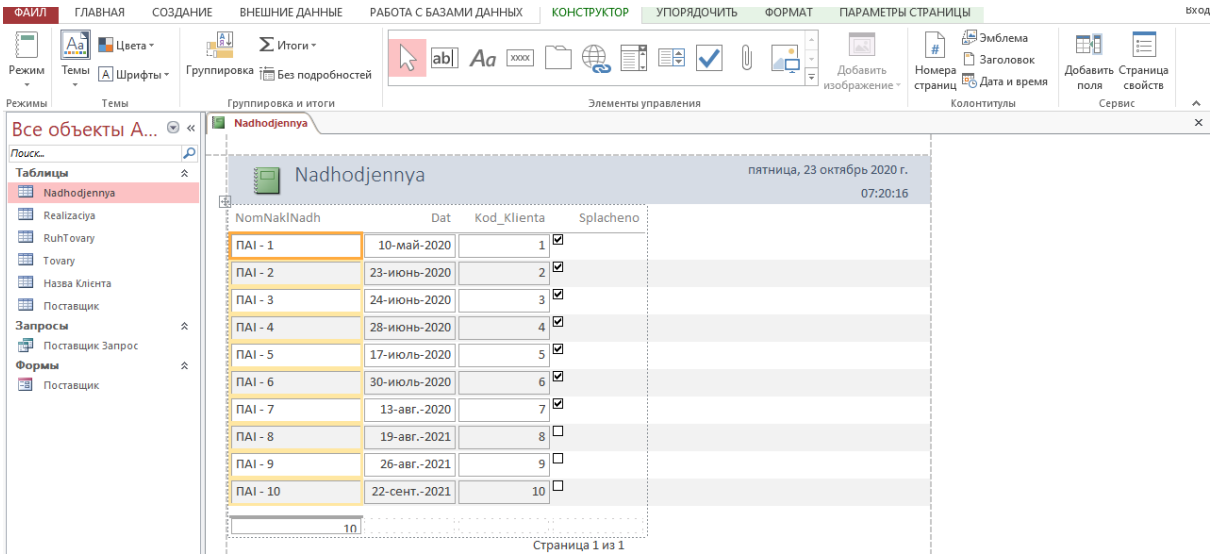


Рис. 4.21. Результат створення звіту

У цей звіт автоматично будуть включені усі поля вибраної таблиці або запиту. Назви полів розташовуються зверху звіту в рядок. На кожній сторінці звіту розміщуються декілька записів, що робить зручним перегляд даних та їх порівняння.

Створення звіту з використанням Майстра звітів. Створення звіту за допомогою *Майстра* зводиться до вибору таблиць, які мають скласти інформаційну основу звіту, визначення списку полів звіту та послідовність їх розміщення.

Для запису *Майстра звітів* у вікні діалогу *Новый отчет* потрібно зі списку можливих варіантів побудови звіту виділити *Мастер отчетов* і натиснути на кнопку *ОК*.

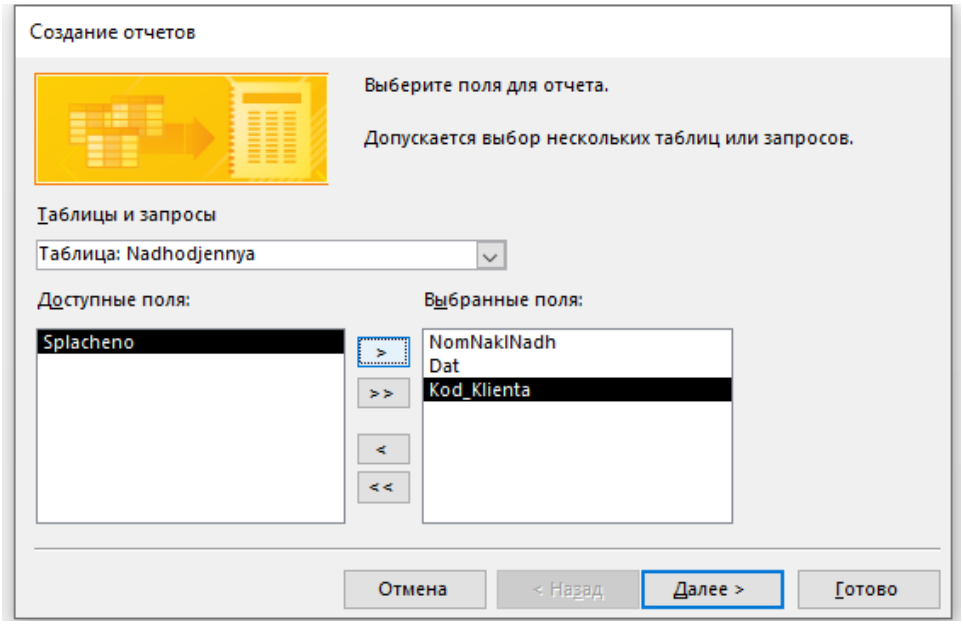


Рис. 4.22. Створення звіту у режимі Майстер

Потрібно вибрати зі списку, що з'явився, таблицю або запит, на базі якої (якого) створюється звіт. Зі списку полів необхідно перенести у список *Выбранные поля* ті поля, які бажаєте розмістити у створюваному звіті.

Якщо звіт створюється на основі декількох таблиць, то аналогічно вибираються і переносяться у створюваний звіт потрібні поля з інших таблиць. Якщо після цього натиснути на кнопку *Готово*, то на екрані з'явиться попередній варіант звіту.

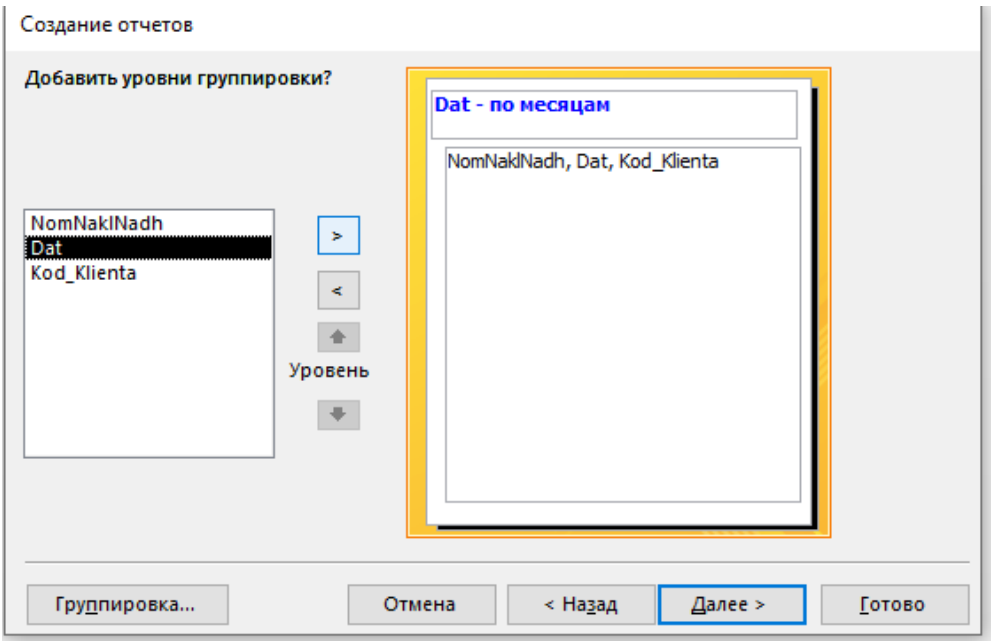


Рис. 4.23. Попередній перегляд звіту у режимі Майстер

Якщо натиснути на кнопку *Далее*, то з'явиться наступне вікно *Майстра звітів*. На другому кроці створення звіту за допомогою *Майстра* необхідно визначити, чи потрібно групувати дані за будь-яким полем. Поля, за якими необхідно виконати групування, вибираються зі списку полів, що висвічуються у лівій частині вікна.

При цьому вони автоматично з'являються у верхній частині правого списку в окремій рамці й виділяються синім кольором. MS Access 2013 пропонує варіант групування даних. Якщо він не задовольняє, то є можливість за допомогою кнопок вікна діалогу задати свій варіант, пам'ятаючи, що MS Access допускає до 10 рівнів групування.

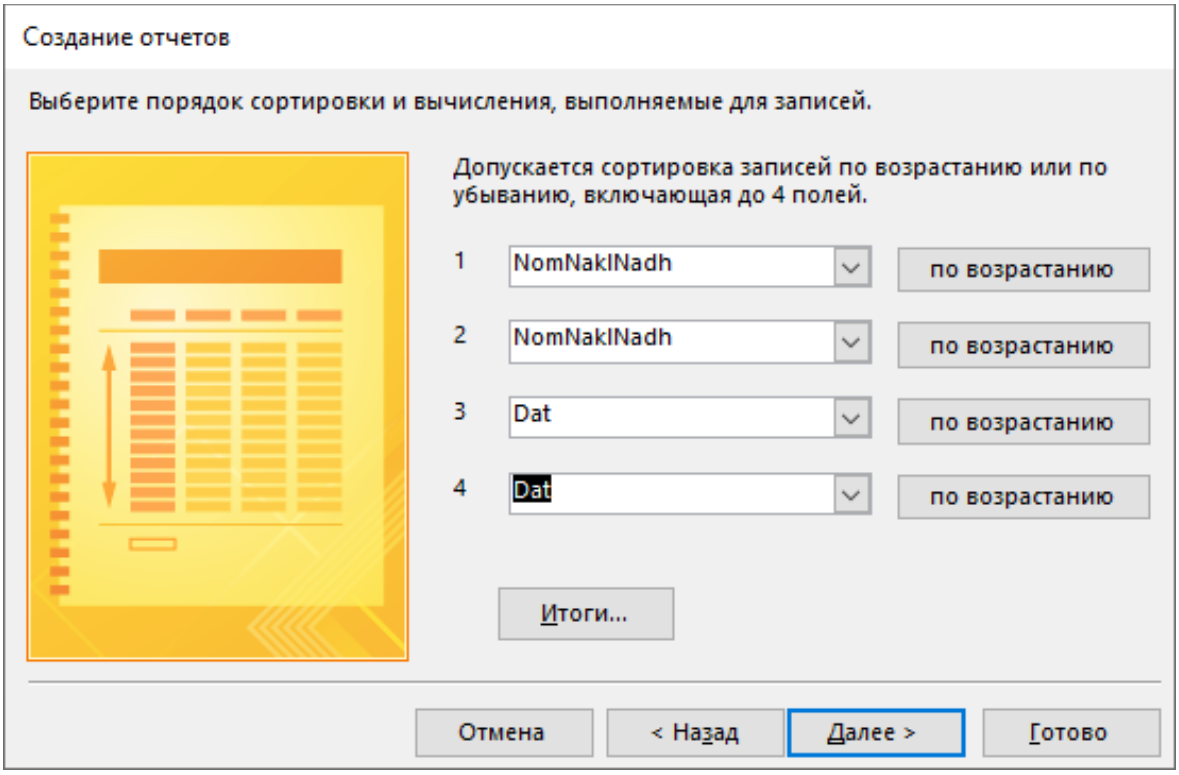


Рис. 4.24. Групування даних звіту у режимі Майстер

У наступному вікні *Майстра звітів* є можливість виконати сортування записів і підрахунок підсумків у полях звіту. З метою сортування записів звіту у цьому вікні необхідно вибрати поля, за якими має відбутись сортування та порядок сортування. Дозволяється вибирати до 4 полів сортування.

Щоб виконати підрахунки підсумків у полях звіту, потрібно натиснути кнопку *Итоги* та у діалоговому вікні *Итоги*, що з'явиться після цього, за допомогою прапорців здати для потрібних полів тип операції для підсумкових підрахунків (Sum, Avg (середнє), Min, Max). Після цього натиснути на кнопку *ОК*, щоб повернутися у вікно сортування.

Наступні два кроки формування звіту за допомогою *Майстра* дозволяють визначити вид макета і стиль оформлення звіту.

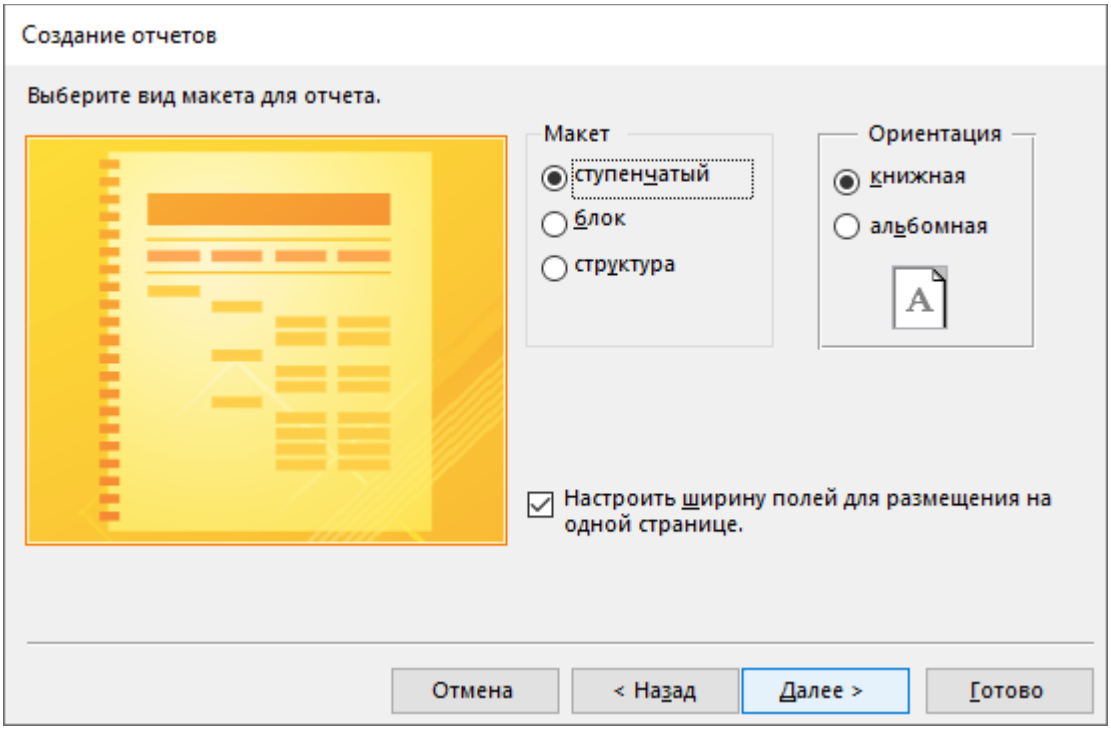


Рис. 4.25. Формування макету звіту у режимі Майстер

На останньому кроці створення звіту є можливість задати ім’я звіту й вибирати один з двох варіантів продовження роботи:

а) перегляд і друкування звіту;

б) модифікація звіту за допомогою *Конструктора звітів*.

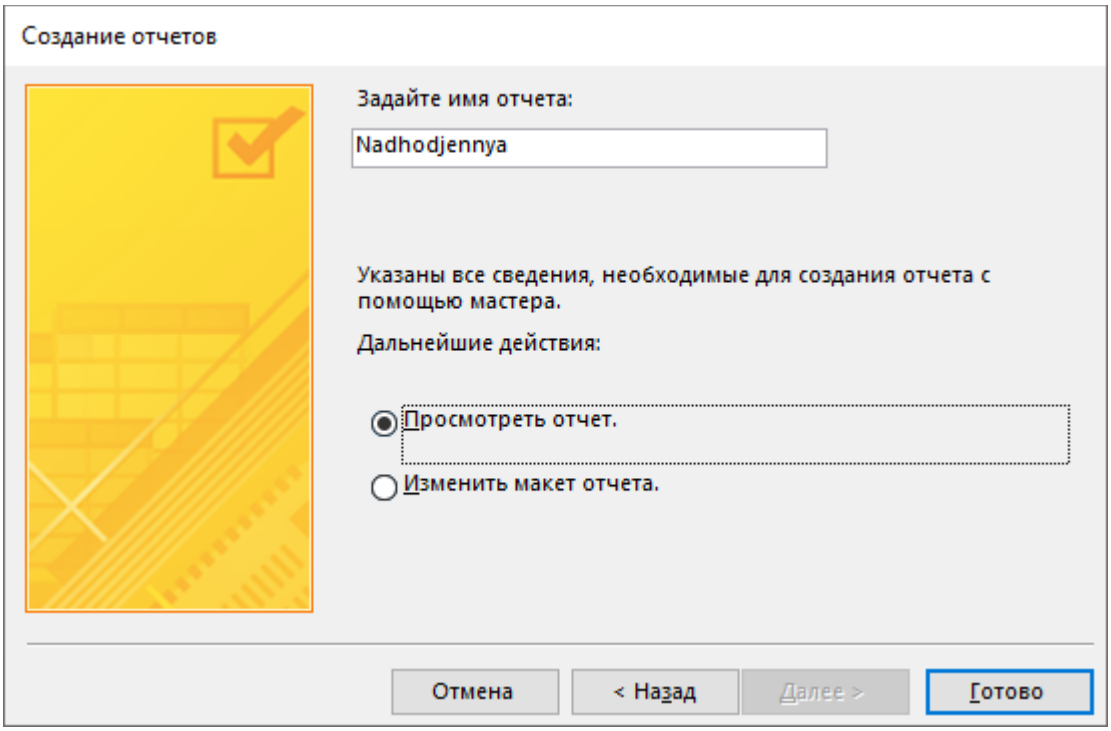


Рис. 4.26. Завершальний етап створення звіту у режимі *Майстер*

Побудований за допомогою *Майстра* звіт може повністю задовольнити ваші вимоги. Проте у деяких випаках виникає потреба внести зміни у структуру і форму звіту, дещо поліпшити його зовнішній вигляд. У цьому випадку є можливість використати створений звіт як чорновий варіант і скористатись *Конструктором звітів* для його модифікації та поліпшення.

Використання Конструктора звітів для створення та модифікації звітів. Основними об'єктами звіту є пояснювальний текст і поля. У звітів також можуть бути присутні лінії, прямокутники, малюнки тощо.

У процесі створення модифікації звітів *Конструктор звітів* дозволяє додавати, вилучати, переміщати зону звіту разом із розміщеними у них об'єктами. Також можна встановлювати колір і керувати параметрами відображення будь-яких елементів і зон звіту.

Для відкриття вікна *Конструктор звітів* з метою модифікації раніше створеного звіту потрібно виконати такі дії:

1. У вікні вибрати потрібний звіт.
2. Натиснути на кнопку *Конструктор*.

На екрані відкриється вікно *Конструктора звітів*, яке нагадує вікно *Конструктора форм* і містить майже ті самі об'єкти (написи, поля) та сітку, що й вікно *Конструктора форм*.

У вікні *Конструктора звітів*, як і у формах, мають місце і заповнюються за допомогою засобів *Конструктора* такі зони: заголовок звіту, верхній і нижній колонтитули, зони даних і приміток звіту. У разі проведення групування даних у вікні *Конструктора* додатково з'являються зони заголовка групи та примітки групи.

– Заголовок звіту розміщується на початку звіту і роздруковується на першій сторінці звіту. Використовується під час створення титульної сторінки звіту.

– Верхній колонтитул розміщується на початку кожної сторінки і під заголовком звіту. У цій зоні розміщуються назви стовпців звіту.

– Заголовок групи розміщується перед першим записом кожної групи.

– Зона даних використовується для відображення даних окремих записів з джерела записів для звіту, тобто з таблиці або запиту.

– Примітка групи розміщується після останнього запису зони даних відповідної групи. Тут, як правило, розміщують підсумковий рядок групи.

– Нижній колонтитул розміщується і друкується у нижній частині кожної сторінки. Найчастіше використовується, щоб вставити номер сторінки.

– Примітка звіту розміщується і друкується наприкінці звіту. Тут розміщують підсумковий рядок звіту.

Конструктор звітів надає користувачу у розпорядження велику кількість засобів для формування нових звітів та поліпшення звітів, створених раніше у режимах *Автозвіту* або *Майстра звітів*.

Створення макросів. Макрос – певна послідовність операцій. Вони зберігаються у вікні, що нагадує таблицю, в такому порядку, в якому їх необхідно виконувати. При запуску макросу Access виконує ці дії.

Макрос може істотно полегшити використання Access 2013. Однією з корисних особливостей макросів є можливість їх прив'язки до кнопок, які поміщаються у форми. Ці кнопки можуть виконувати операції, що часто повторюються (наприклад, відкривати діалогові вікна для пошуку, змінювати порядок сортування даних або друкувати звіт).

Користувачі Access 2013 можуть використовувати певні кнопки для виконання цих дій, навіть не знаючи всіх подробиць операцій, що виконуються макросом. Використовуючи макроси, можна створювати завершені застосування з призначеним для користувача меню і діалоговими вікнами.

Перший макрос виконуватиме досить просте завдання – вказувати комп'ютеру на необхідність видавати звуки за допомогою вбудованого динаміка. Створюється він так:

1) необхідно клацнути на посиланні *Макроси*, розташованої у вікні бази даних;

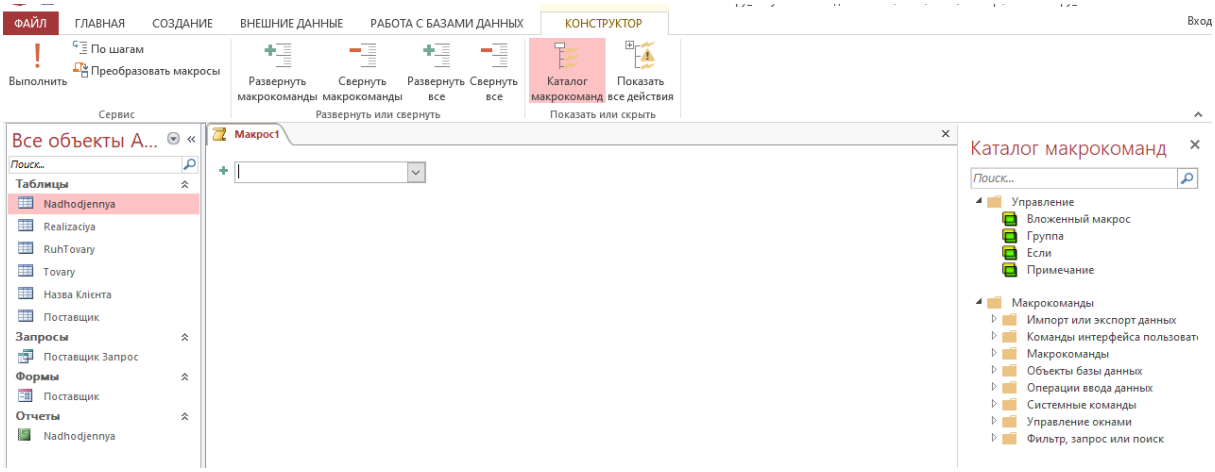


Рис. 4.27. Створення макросу

2) для відкриття таблиці конструктора, призначеної для створення нового макросу, потрібно клацнути на кнопці *Створити*, розташованої на панелі інструментів;

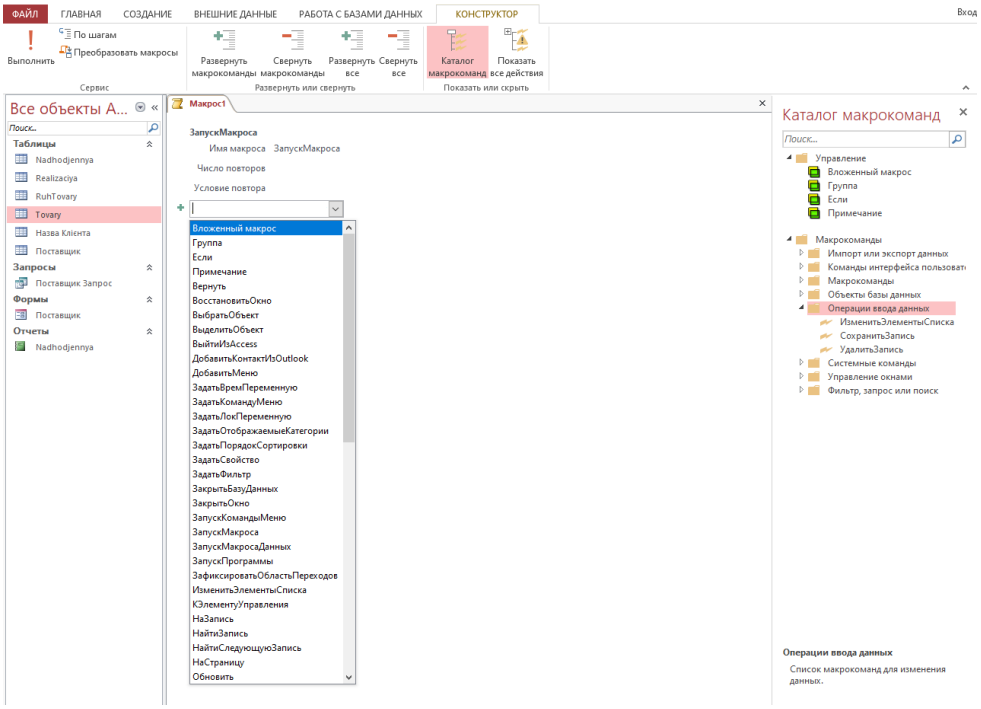


Рис. 4.28. Вибір дії для макросу

3) після цього клацнути на списку, що розкривається, в першому рядку стовпця *Макрокоманда* і вибрати із списку команду *Сигнал*;

4) виконати команду *Файл – Сохранить*;

5) у діалоговому вікні *Сохранение* потрібно привласнити макросу ім’я *Сигнал* і клацнути на кнопці *ОК*.

Створення модулів. Модуль Access складається з описів, інструкцій і підпрограм (процедур), написаних мовою програмування VBA (Visual Basic for Application) та збережених під загальним ім'ям. Visual Basic був розроблений Microsoft як мова високого рівня для створення додатків на базі операційної системи Windows.

У VBA процедури є замкнутими програмними одиницями. Вони забезпечують більш широкі можливості, ніж макроси. Кожна форма або звіт у базі даних містить вбудовані процедури обробки подій, наприклад, процедура обробки події дозволяє відкрити іншу форму у разі натиснення певної кнопки у формі.

У Microsoft Access 2013 існують модулі двох типів: стандартний модуль і модуль класу. Стандартний модуль містить процедури Sub і Function, не пов'язані з конкретним об'єктом: формою або звітом. Вони доступні для всіх процедур у базі даних.

Стандартні модулі використовуються також для оголошення доступних зі всіх модулів додатка глобальних змінних. Для перегляду списку стандартних модулів натискуйте на панелі інструментів кнопку Модулі (Modules) у вікні База даних (Database).

Модулі класів, крім процедур, містять опис нового об'єкта і програми, які є локальними для цих об'єктів. Як приклад модуля класу, можна назвати модулі форм і звітів. Модуль форми створюється і зв'язується з формою, щоб створити процедуру обробки подій для цієї форми.

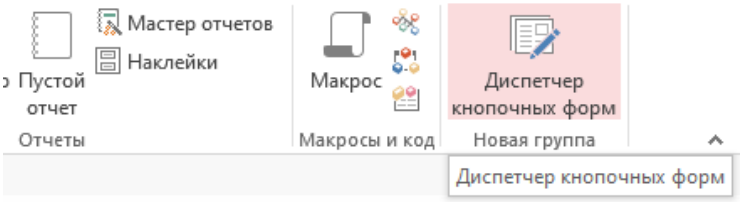
Процедури із стандартного модуля, якщо вони не описані явно як локальні для модуля, що містить їх, розпізнаються і можуть викликатися процедурами з інших модулів у тій саме базі даних або з баз даних, що адресуються.

Створення кнопкової форми. Одним з найбільш ефективних інструментів, використовуваних для організації об'єктів баз даних Access 2013, є головна кнопкова форма. Головна кнопкова форма запускається під час відкриття бази даних і містить кнопки для активізації найбільш важливих об'єктів. Вона має набагато більш дружній інтерфейс, ніж основне вікно бази даних.

На додавання кнопкової форми до бази даних не потрібно багато часу, оскільки компанія Microsoft вбудувала в Access 2013 диспетчер кнопкових форм. Нижче вказаний диспетчер буде

застосований для створення кнопкової форми бази даних. Для цього необхідно виконати такі дії:

1) запустити Access 2013 і завантажити базу даних;
2) виконати команду *Диспетчер кнопочной формы*;



3) оскільки диспетчер завантажується вперше, з'явиться вікно із запитом на створення нової кнопкової форми. Потрібно клацнути на кнопці *Так*;

4) відкривається вікно *Диспетчер кнопочной формы*. Для однієї бази даних можна створити декілька кнопкових форм. Під час запуску диспетчера за замовчуванням створюється головна кнопкова форма. Потрібно клацнути на кнопці *Изменить* для відкриття діалогового вікна *Изменение* сторінки кнопкової форми;

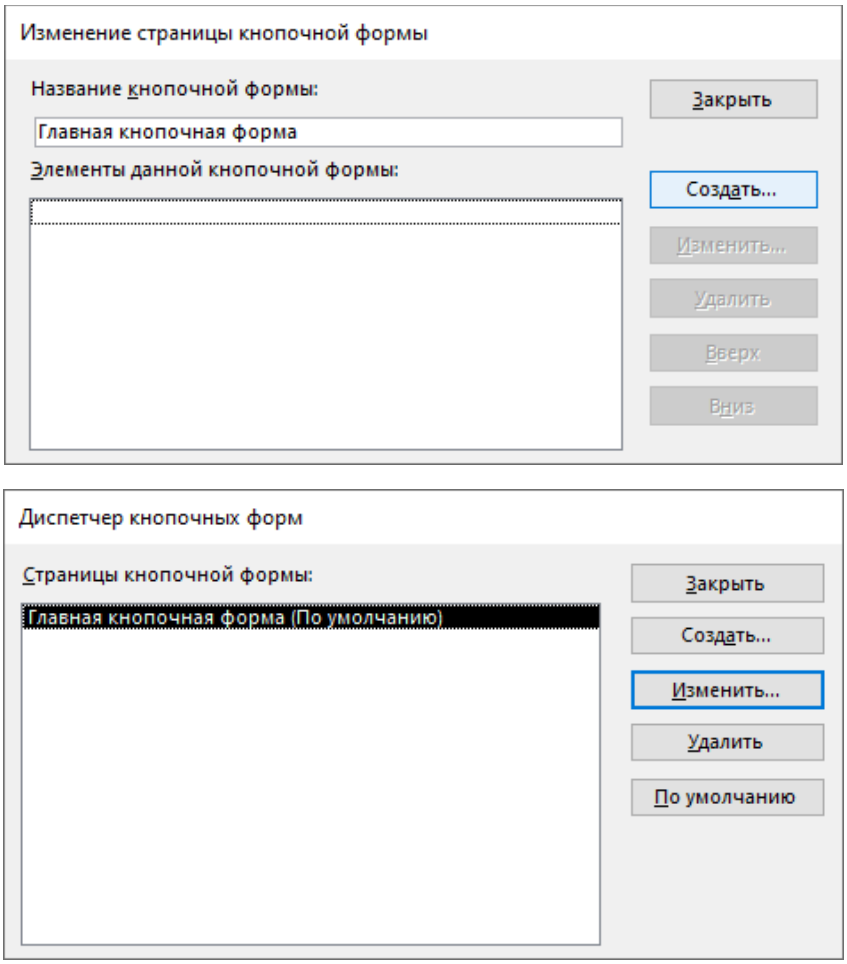
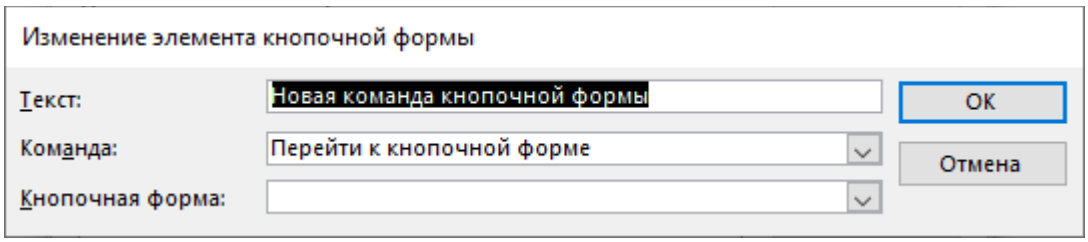


Рис. 4.29. Створення кнопкової форми

5) у діалоговому вікні, що відкрилося, демонструються всі елементи кнопкової форми. Досі ще жоден з них не вибраний. Потрібно клацнути на кнопці *Создать* для відкриття діалогового вікна *Зміна елементу кнопкової форми*;

6) ввести в поле *Текст* назву форми і у полі із списком *Команда* вибрати команду *Відкрити форму* для зміни, потім у списку *Форма* вибрати елемент. На рисунку нижче діалогове вікно вже заповнене. Клацнути на кнопці ОК для його закриття;



7) клацнути на кнопці *Закрити* у діалоговому вікні *Зміна сторінки кнопкової форми*;

8) клацнути на кнопці *Закрити* у вікні *Диспетчер* кнопкової форми;

9) клацнути на посиланні *Форми* у вікні бази даних;

10) двічі клацнути на опції *Кнопкова форма*, щоб вивести форму на екран;

11) клацнути на кнопці – і буде відкрито форму.

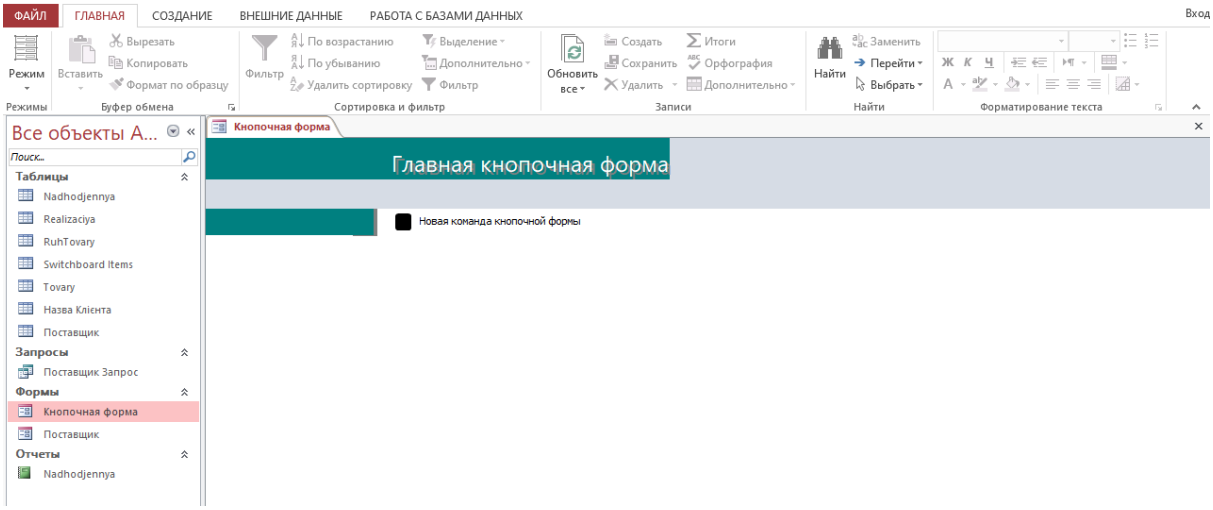


Рис. 4.30. Побудована кнопкова форма

Зміна кнопкової форми. Диспетчер кнопкової форми може використовуватися багато разів для однієї і тієї ж кнопкової форми. Якщо знадобиться внести якісь зміни, наприклад додати новий елемент, змінити або видалити його, потрібно відкрити вікно *Диспетчер* кнопкової форми. Отже, всі раніше встановлені параметри залишаться чинними.

4.2. Модель бази даних *Особові рахунки* для банківської установи

Розрахункові та касові операції належать до основних, найбільш трудомістких та відповідальних у банківській діяльності. Тому для виконання цих операцій застосовується обчислювальна техніка. Сьогодні в інтегрованих банківських системах комплекс розрахункових і касових операцій становить підсистему, призначення якої – автоматизувати облік, контроль та складання звітності про рух коштів на рахунках клієнтів банку.

Згідно з призначенням такої підсистеми виконуються її функції: відкриття особових рахунків, їх перегляд, доповнення й коригування, формування довідкової фінансової та технологічної інформації про стан особових рахунків через такі процедури, як блокування особового рахунку на установлений термін, вилучення закритих рахунків.

Для автоматизації розрахунково-касових операцій і контролю за їх здійсненням в інтегрованих банківських системах створюються такі автоматизовані робочі місця (далі – АРМ): операціоніста, контролера, технолога, бухгалтера, спеціаліста валютного відділу, касира, кредитного інспектора, керівника (відділу, банку), адміністратора бази даних. Кожне з перелічених АРМ має різний набір функцій, різний рівень доступу до бази даних. В інтерфейсах адміністратора бази даних і головного бухгалтера передбачено найповніший перелік функцій.

Інформаційну базу (далі – ІБ) розрахунково-касових операцій утворює сукупність певним чином структурованої (як документи чи файли) інформації, використовуваної під час виконання завдань підсистеми.

Усю наявну ІБ поділяють на зовнішню та внутрішню. Зовнішня ІБ – це сукупність вхідних повідомлень – документів та файлів, що надходять від клієнтів цього банку, інших комерційних банків, державних структур, різних юридичних і фізичних осіб. До них належать платіжні доручення, грошові чеки, векселі, меморіальні ордери, оголошення про внесення готівки, касовий план, реєстри платіжних документів тощо, файли платіжних доручень, що надходять від інших банків через електронну пошту НБУ та від клієнтів банку через підсистему *Клієнт-банк*.

Внутрішня ІБ як сукупність даних на машинних носіях, використовуваних для виконання завдань з обліку й контролю розрахунково-касових операцій, містить у собі файли з довідковою, оперативною та архівною інформацією. До файлів довідкової інформації належать довідники клієнтів банку, банків, балансових рахунків і операцій з коштами, довідники каси.

Файли оперативної інформації протягом робочого дня формуються або на підставі вхідних повідомлень, які надходять операціоністу у вигляді паперових чи електронних (мережею) документів, або під час обробки оперативної інформації.

Архівні файли – це файли результатної інформації, сформовані після складання балансу та записані в архів. За допомогою файлів довідкової, оперативної та архівної інформації виконуються завдання з обліку та контролю розрахункових і касових операцій.

Автоматизація розрахункових операцій виконується за допомогою програмного комплексу *Операційний день банку* (далі – ОДБ), який дає змогу в будь-який момент його функціонування дістати інформацію про рух коштів на розрахункових, поточних, кредитних, депозитних та інших рахунках. Залежно від конструктивних характеристик програмного комплексу ОДБ облік і контроль за розрахунковими операціями здійснюються в національній валюті та валюті інших країн в одному ОДБ. У більшості програмних пакетів ОДБ передбачається, що головними виконавцями автоматизації розрахункових операцій є операціоністи, контролери, технологи. Для них у банківській системі створюються відповідні АРМ. Конкретний перелік операцій для кожного АРМ залежить від запровадженого технологічного процесу, наявності технічних засобів, обсягу документообігу тощо, але деякі роботи жорстко прив’язані до виконавців.

Технологічний процес автоматизації розрахункових операцій здійснюється за допомогою програм, які об'єднані у модуль під назвою *Особові рахунки*.

Базовою функцією під час автоматизації розрахункових операцій є відкриття нових особових рахунків. Виконує її головний бухгалтер або за його дорученням провідний технолог через меню системи ОДБ. Увійшовши до пункту меню *Особові рахунки*, переходять до пункту *Відкриття особових рахунків*. При цьому зазначають такі реквізити: тип особового рахунку, який вибирається з переліку, що відкрився за допомогою функціонального клавіша; рівень секретності рахунка, що задається обмеженнями доступу до нього; вид валюти рахунку, який вибирається курсором з висвітленого на екрані переліку; реєстраційний номер клієнта в банку; номер балансового рахунку; номер особового рахунку клієнта; назва особового рахунку, дата відкриття; підстава для відкриття рахунку; тип особового рахунку для податкової адміністрації; ознаки звітності; ознаки для розрахунку платежів за розрахунково-касове обслуговування; код відповідального виконавця, операціоніста; нарахування процентів за залишки.

Супроводження особових рахунків виконується на робочих місцях відповідальних виконавців-операціоністів і технолога. Крім відповідальних виконавців, доступ до особових рахунків з різними повноваженнями мають працівники валютного відділу, каси, головний бухгалтер та керівники служб банку. Усім зазначеним працівникам доступна функція перегляду особових рахунків, виконувати яку можна по-різному. Найчастіше переглядається окремий особовий рахунок, для чого насамперед входять до меню другого рівня (пункт *Перегляд особових рахунків*). Перш ніж увійти в цей режим, необхідно в екранному вікні набрати номер рахунку.

Функція перегляду особових рахунків дає змогу:

– переглянути реквізити рахунку, зокрема номер балансового рахунку, номер особового рахунку, тип рахунку, його призначення та інші довідкові ознаки;

– визначити оперативний стан особового рахунку, а саме: вхідний залишок, поточні обороти коштів, вихідне сальдо рахунку.

Банки України використовують такі програмні комплекси ОДБ: ИСАОД, 4K-Bank, Profix, SmartBank, Мебиус, Скрудж, SyBridge, B2 та інші.

З різних причин, зокрема високої вартості програмних комплексів, відсутності технічних можливостей та з міркувань безпеки, студенти закладів вищої освіти не мають можливості під час навчання отримати навички роботи в жодному з комплексів ОДБ; винятками є лише окремі домовленості вишів про проходження короткотермінових ознайомчих курсів або екскурсій.

Тому актуальним завданням для якісної підготовки студентів спеціальності «Фінанси» є розробка навчальних середовищ, подібних за своїм функціоналом до автоматизованих робочих місць працівника банку.

Принципова схема руху коштів у банку між рахунками клієнтів

Як зазначалось вище, технологічний процес автоматизації розрахункових операцій у банку здійснюється за допомогою програм ОДБ, які об'єднані у модуль під назвою «Особові рахунки».

Для розробки навчального модуля «Особові рахунки» використаємо систему управління базами даних Access.

1. Створимо базу даних з іменем РУХ_КОШТІВ.accdb відповідно до схеми даних на рис. 4.31.

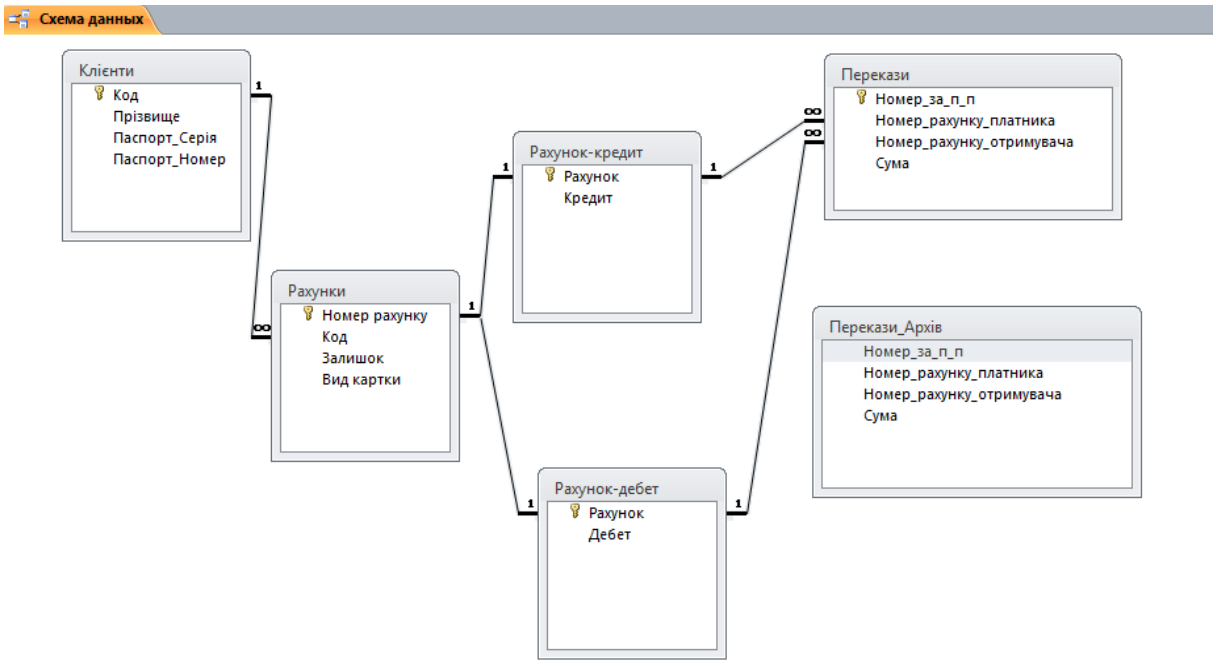


Рис. 4.31. Схема даних бази РУХ_КОШТІВ.accdb

1.1. Для побудови таблиць *Клієнти, Рахунки, Рахунок-кредит, Рахунок-дебет, Перекази, Перекази_Архів* скористаємось режимом *Конструктор*, як показано на рис. 4.32, 4.33, 4.34, 4.35, 4.36, 4.37.

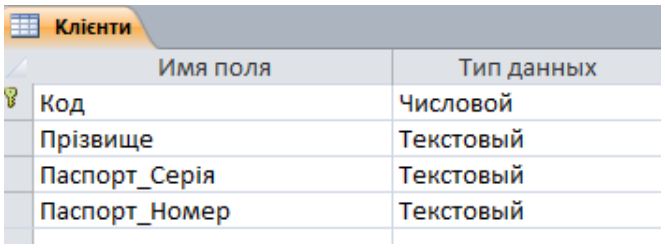


Рис. 4.32. Таблиця *Клієнти* в режимі *Конструктор*

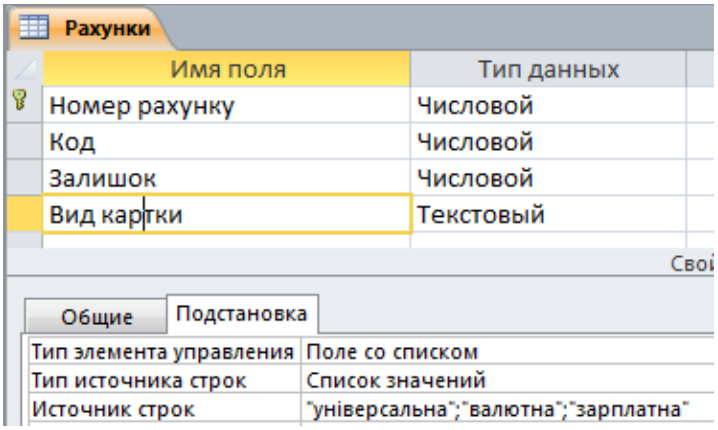


Рис. 4.33. Таблиця *Рахунки* в режимі *Конструктор*

Примітка: поле *Залишок* має властивість *за замовчуванням 0.*

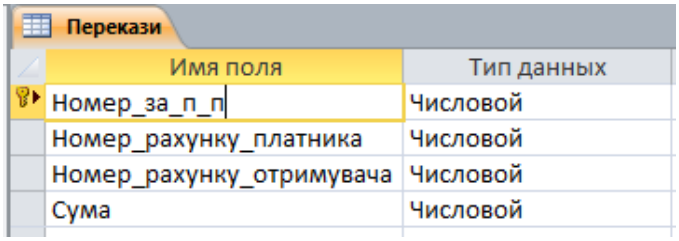


Рис. 4.34. Таблиця *Перекази* в режимі *Конструктор*

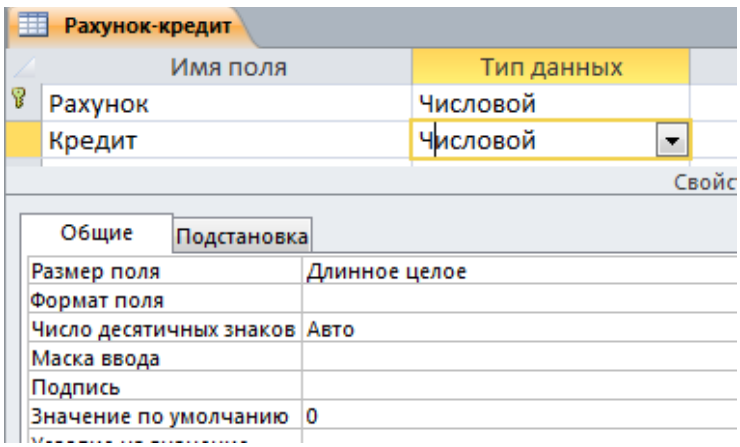


Рис. 4.35. Таблиця *Рахунок-кредит* у режимі *Конструктор*

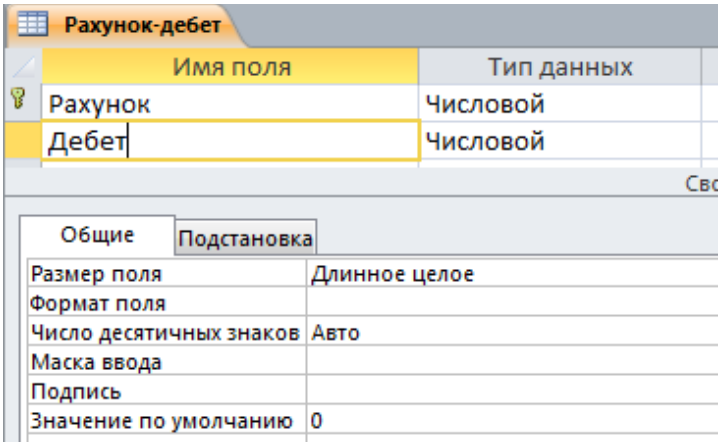


Рис. 4.36. Таблиця *Рахунок-дебет* у режимі *Конструктор*

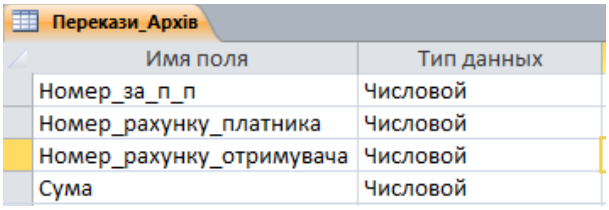


Рис. 4.37. Таблиця *Перекази_Архів* у режимі *Конструктор*

Примітка: таблицю *Перекази_Архів* можна отримати шляхом копіювання таблиці *Перекази*.

1.2. У вікні *Схема даних* створимо зв'язки між таблицями, як показано на рис. 4.31.

1.3. Заповнимо таблиці даними, як показано на рис. 4.38, 4.39, 4.40.

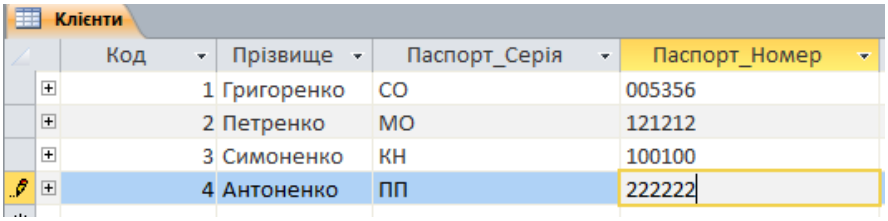


Рис. 4.38. Таблиця *Клієнти* в режимі *Таблиця*

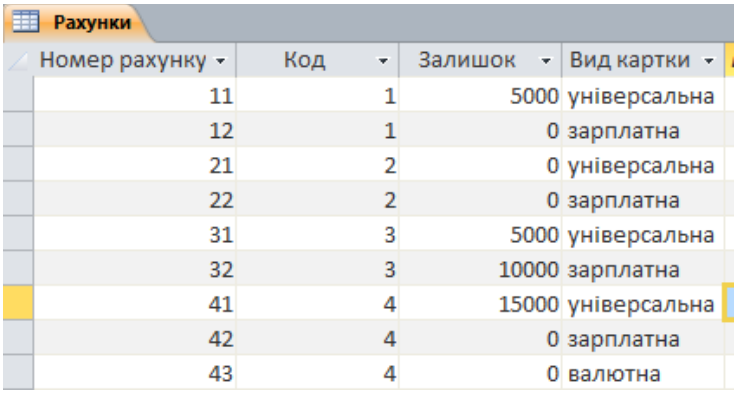


Рис. 4.39. Таблиця *Рахунки* в режимі *Таблиця*

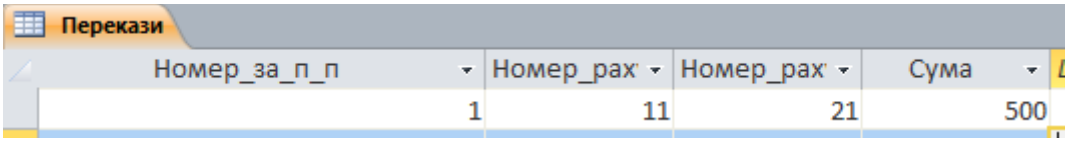


Рис. 4.40. Таблиця *Перекази* в режимі *Таблиця*

Примітка: таблиці *Рахунок-кредит, Рахунок-дебет, Перекази_Архів* спроєктовано не для заповнення, а для допоміжних дій у процесі переведення коштів з одного рахунку на інший.

Проєктування запитів для реалізації розрахункових операцій між клієнтами банку.

2. Таблицю *Рахунок-кредит* спроєктовано для списання *Суми* з одного рахунку; таблицю *Рахунок-дебет* спроєктовано для нарахування *Суми* на інший рахунок; таблицю *Перекази_Архів* спроєктовано для збереження історії переведення коштів.

Для реалізації вказаних дій необхідно сконструювати запити.

2.1. На рис. 4.41 показано запит на оновлення в режимі *Конструктор*, який передає у таблицю *Рахунок-кредит* суму списання коштів (500) з рахунку (11).

2.2. На рис. 4.42 показано запит на оновлення, який передає у таблицю *Рахунок-дебет* суму нарахування коштів (500) на рахунок (21).

2.3. На рис. 4.43 показано запит на оновлення, який проводить обчислення нового залишку на рахунках у таблиці *Рахунки* за формулою:

Новий залишок = Залишок – Кредит + Дебет.

У результаті виконання цього запиту:

на рахунку (11) буде сума 5000 – 500 + 0 = 4500,

на рахунку (21) буде сума 0 – 0 + 500 = 500.

2.4. На рис. 4.44 показано запит на оновлення, який обнуляє значення допоміжних таблиць *Рахунок-кредит, Рахунок-дебет* після їхнього використання.

2.5. На рис. 4.45 показано запит на додавання, який додає запис із таблиці *Перекази* у таблицю *Перекази_Архів*.

2.6. На рис. 4.46 показано запит на видалення, який видаляє запис із таблиці *Перекази* (після чого можна починати новий переказ коштів з одного рахунку на інший).

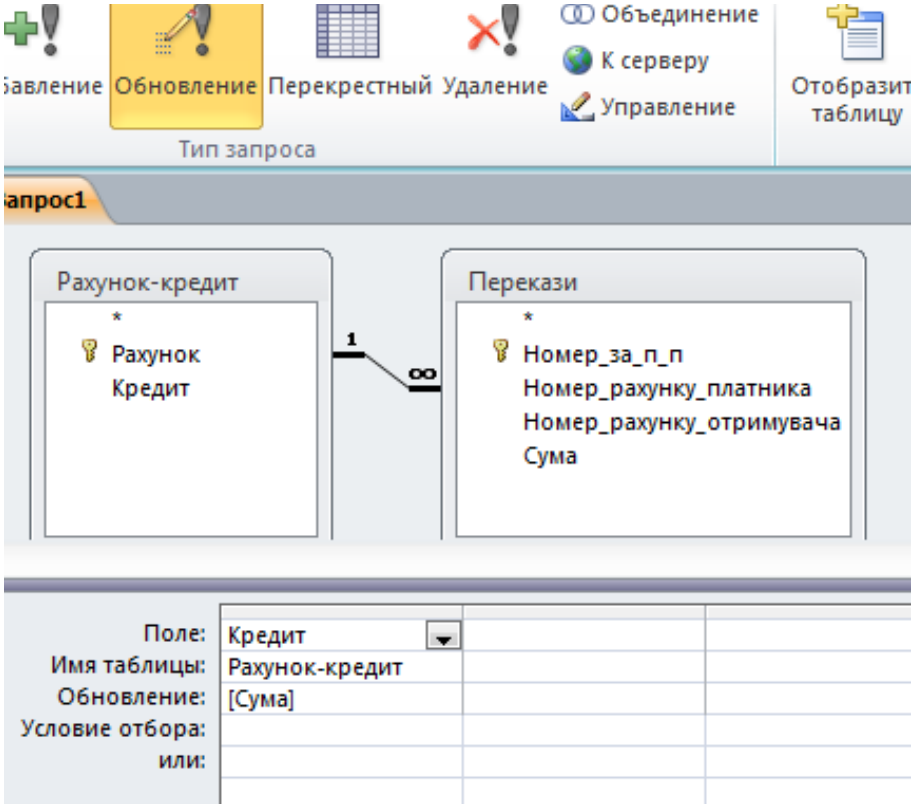


Рис. 4.41. Запит на оновлення *Запрос 1*

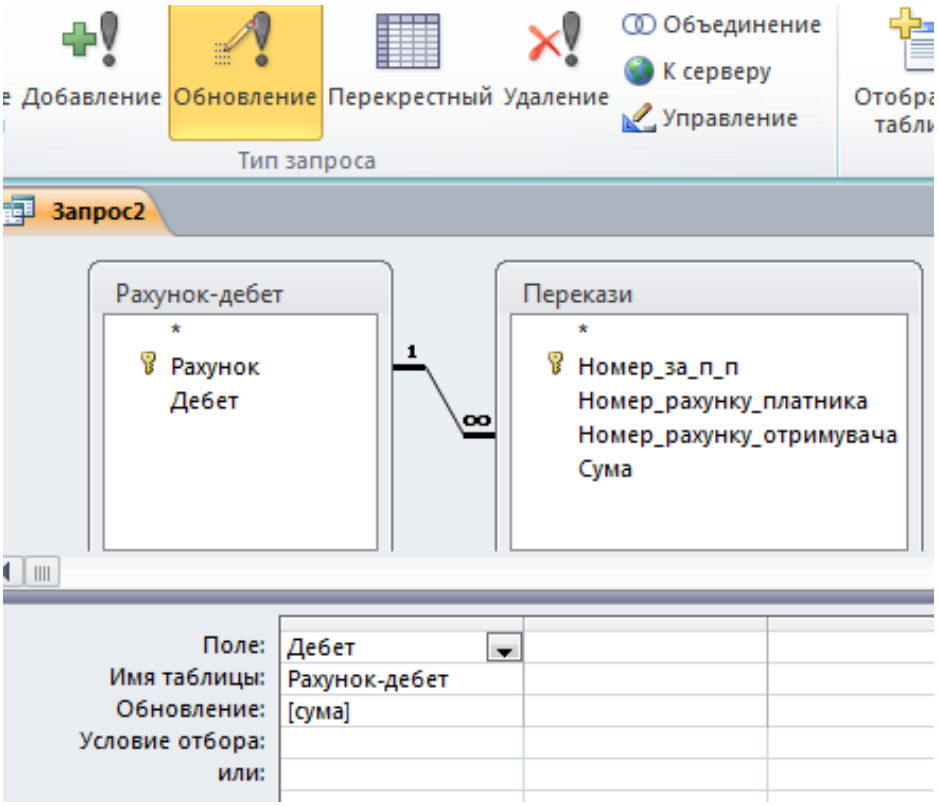


Рис. 4.42. Запит на оновлення *Запрос 2*

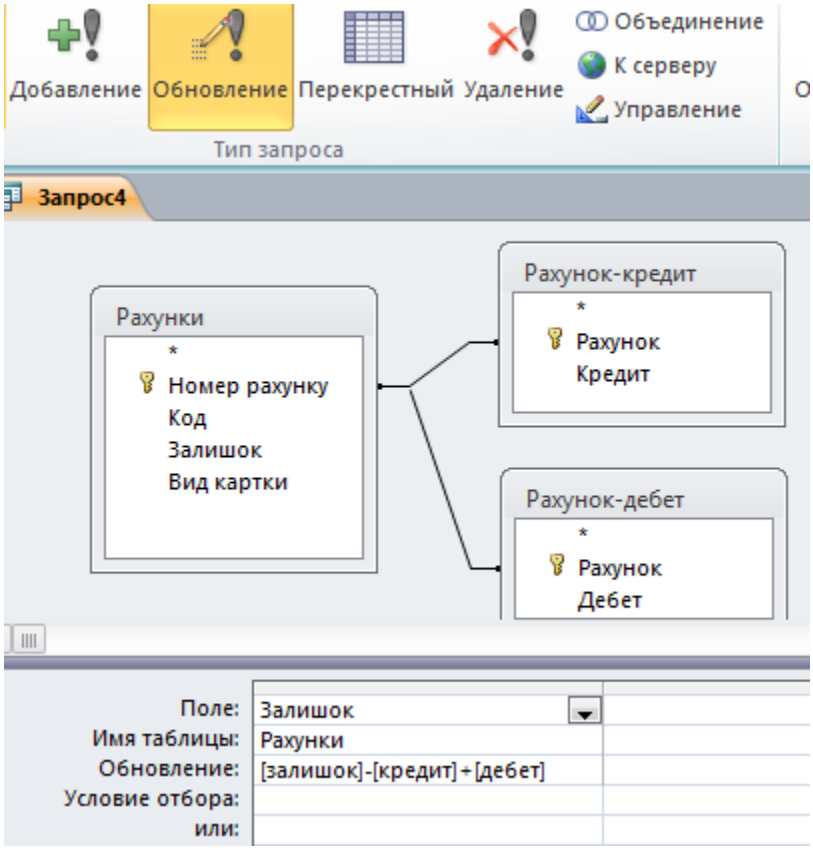


Рис. 4.43. Запит на оновлення *Запрос 4*

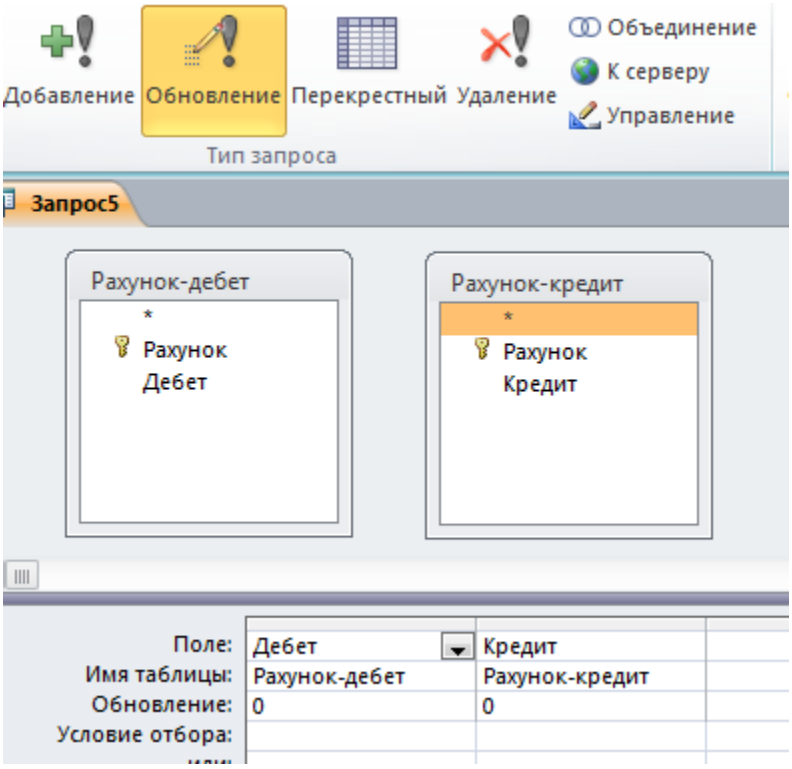


Рис. 4.44. Запит на оновлення *Запрос 5*

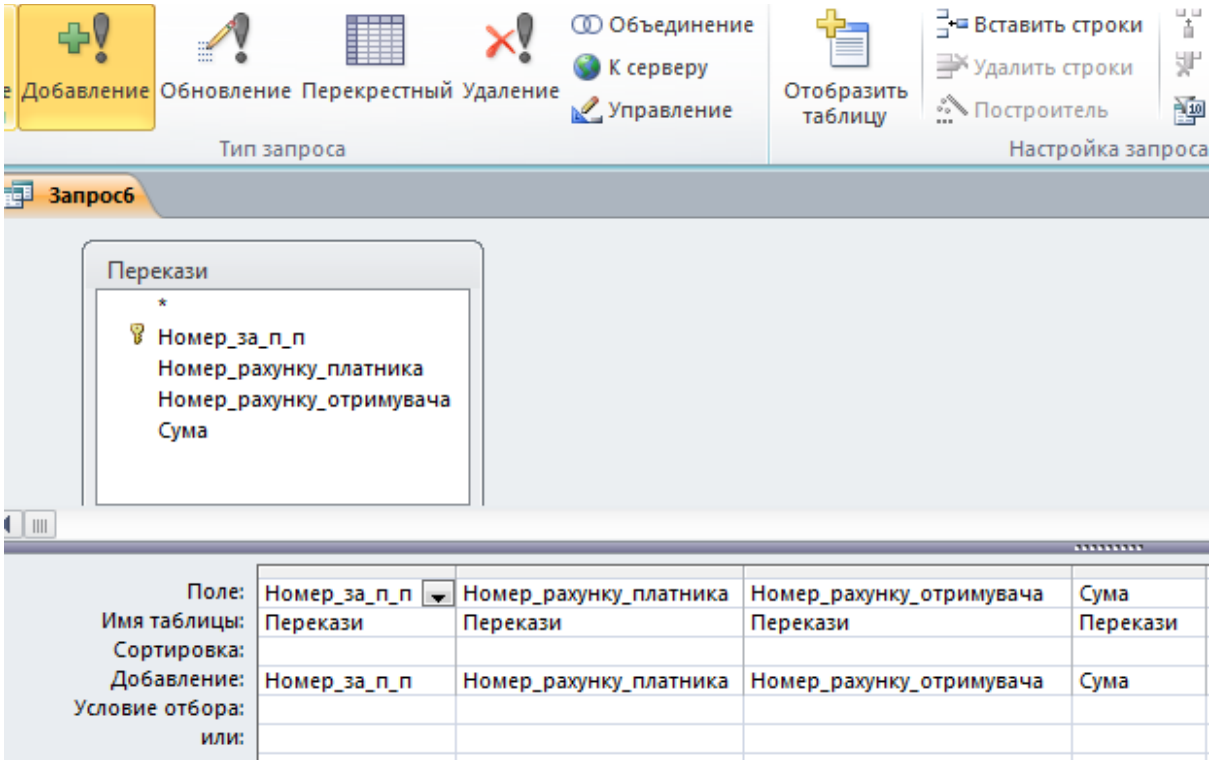


Рис. 4.45. Запит на додавання *Запрос 6*

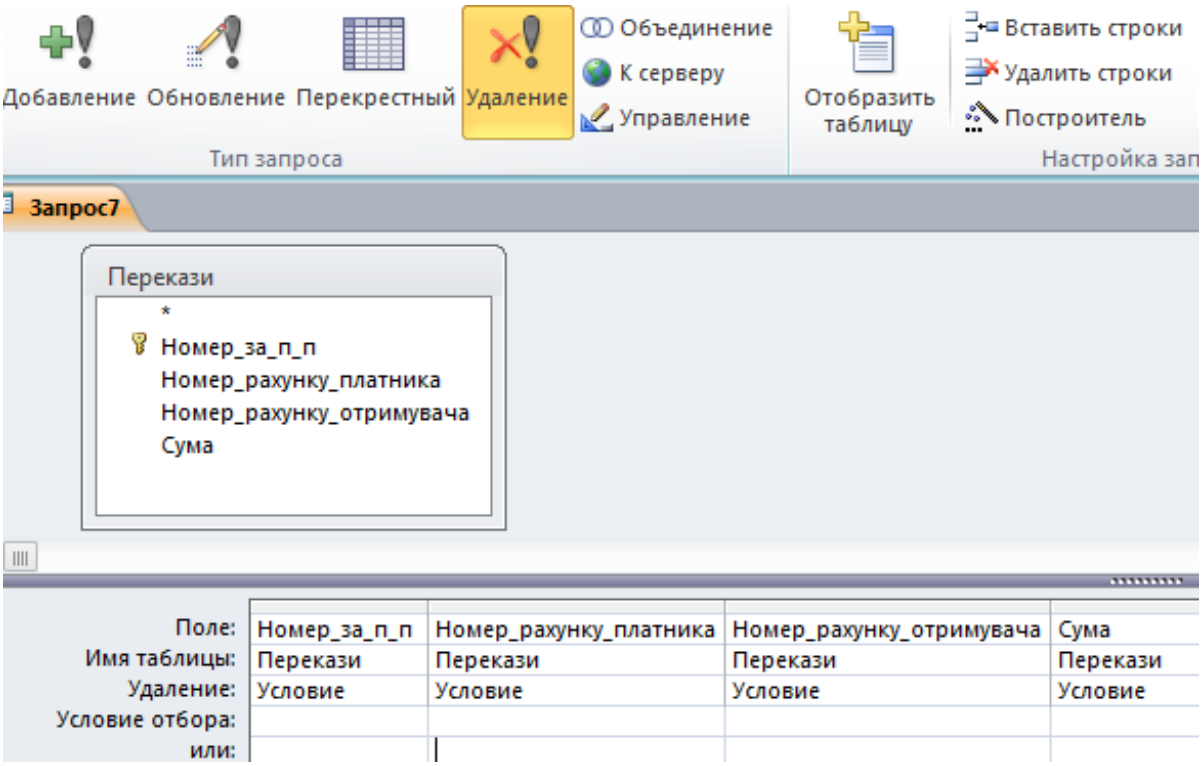


Рис. 4.46. Запит на додавання *Запрос 7*

Тестування розрахункових операцій між клієнтами банку

Перевіримо функціонал побудованої бази даних РУХ_КОШТІВ.accdb

Для цього виконаємо запити 1–7 у правильній послідовності.

Перевіримо стан усіх таблиць після виконання запитів. Таблиці *Рахунки, Перекази* та *Перекази_Архів* матимуть такі зміни (рис. 4.47).

Рахунки

Номер рахунку	Код	Залишок	Вид картки	Щелкни
11	1	4500	універсальна	
12	1	0	зарплатна	
21	2	500	універсальна	
22	2	0	зарплатна	
31	3	5000	універсальна	
32	3	10000	зарплатна	
41	4	15000	універсальна	
42	4	0	зарплатна	
43	4	0	валютна	

Перекази

Номер_за_п_п	Номер_рах	Номер_рах	Сума
*			

Перекази_Архів

Номер_за_п_п	Номер_рах	Номер_рах	Сума
1	11	21	500
*			

Рис. 4.47. Результат виконання запитів

Автоматизація виконання розрахункових операцій між клієнтами банку

Для того, щоб під час наступних операцій з переведення коштів автоматизувати виконання запитів, потрібно побудувати макрос, як показано на рис. 4.48.

Крім того, необхідно позбутися зайвих діалогових вікон, які гальмують виконання макроса. Таке налаштування можна виконати на вкладці *Файл* у вікні *Параметри* (рис. 4.49).

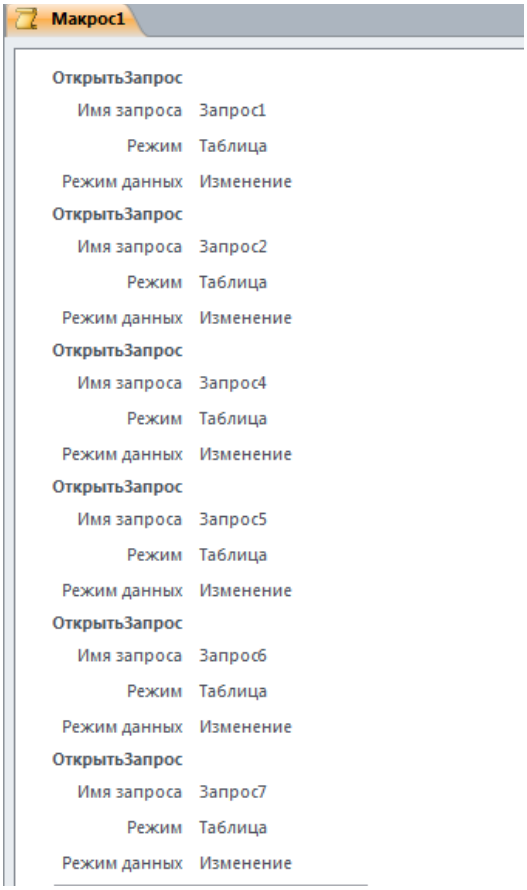


Рис. 4.48. Макрос для автоматизації виконання запитів

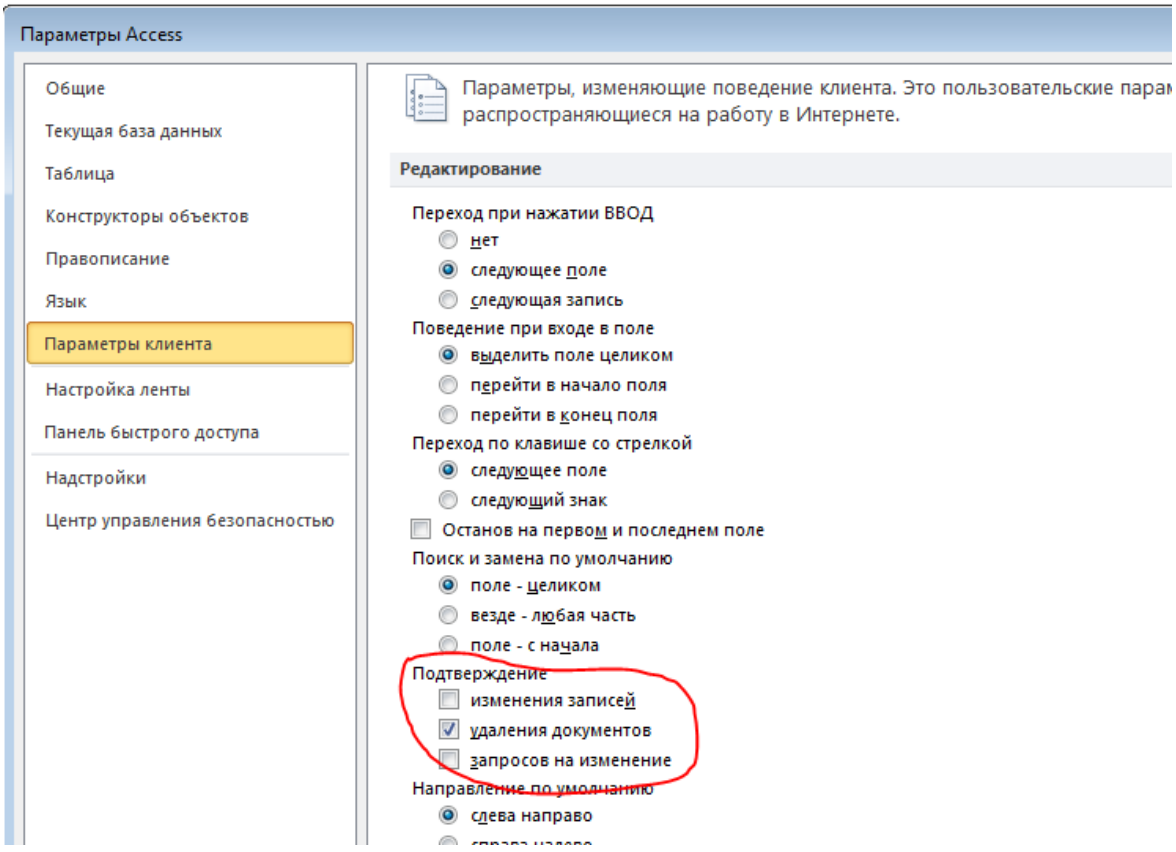
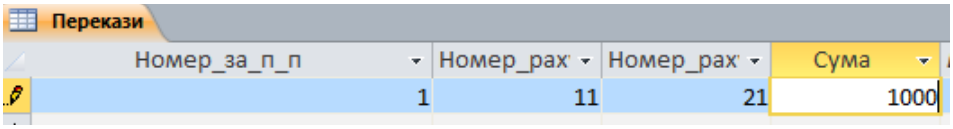


Рис. 4.49. Налаштування виконання запитів у вікні *Параметри*

У таблиці *Перекази* зробимо новий запис.



Виконаємо *Макрос1*. Перевіримо результат у таблицях *Рахунки, Перекази* та *Перекази_Архів*.

Зазвичай під час заповнення або перегляду бази даних замість таблиць використовують форми. Це зумовлено рядом причин, серед яких:

– *зручність* (екранні форми можна зробити точною копією паперових бланків, з яких відбувається введення даних; завдяки цьому набагато зменшується кількість помилок при введенні і значно знижується стомлюваність персоналу);

– *конфіденційність даних* (різні люди можуть мати різні права доступу до інформації, яка зберігається в таблицях. Наприклад, один має право вводити тільки імена й адреси клієнтів, інший – лише номера їхніх розрахункових рахунків, а третій – грошові суми, що зберігаються на цих рахунках. Для введення даних їм дають різні форми, хоча дані з форм можуть надходити в одну таблицю).

Отже, створимо форму для таблиці *Перекази*.

Найпростішим способом можна створити форму в режимі *Автоформа*. Для цього виділимо таблицю *Перекази* та натиснемо кнопку 

на вкладці *Главная*. Тоді буде створено форму, як показано на рисунку 4.50.

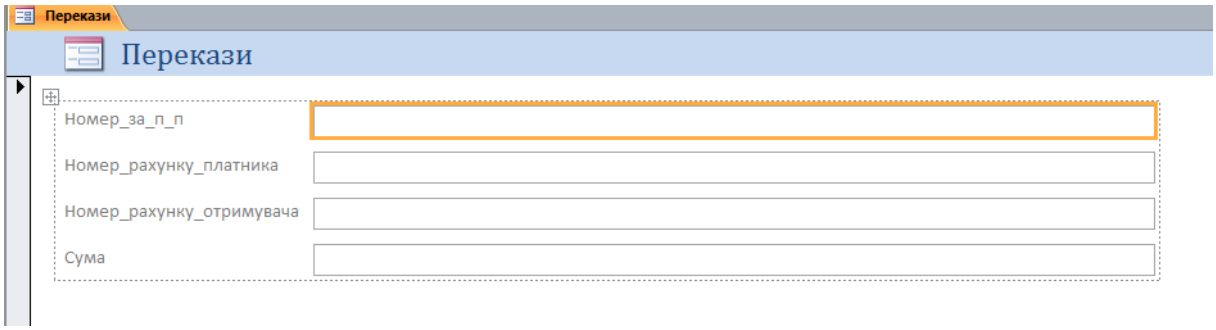


Рис. 4.50. Форма для таблиці *Перекази*, побудована в режимі Автоформа

Щоб створити на формі кнопку для запуску *Макрос1,* перейдемо у режим *Конструктор*, як показано на рисунку 4.51.

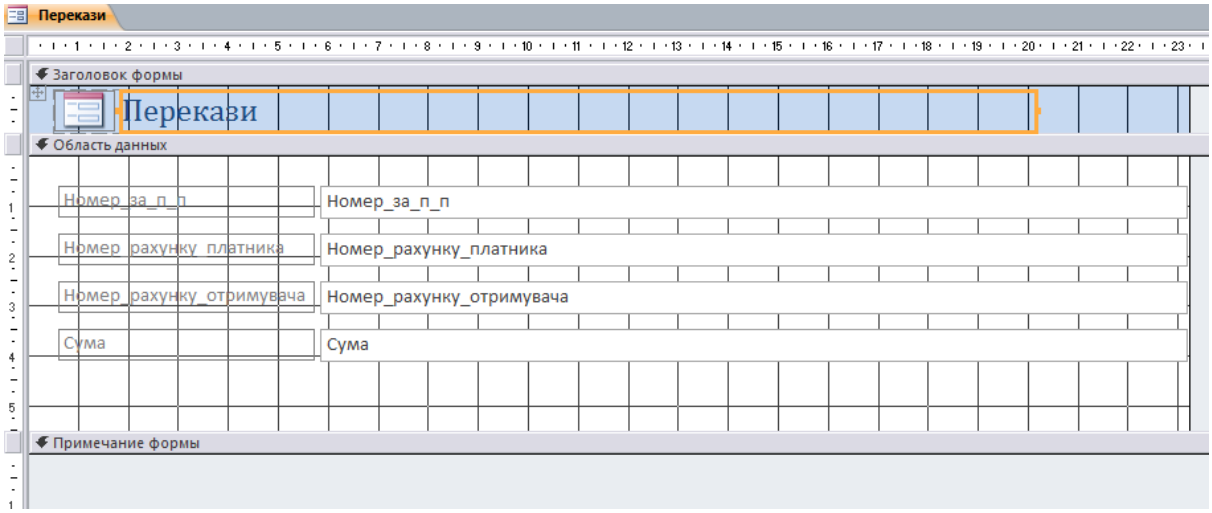


Рис. 4.51. Форма для таблиці *Перекази* в режимі *Конструктор*

Вставимо елемент *Кнопка* на форму та виберемо категорію *Разное,* дії *Выполнить макрос,* як показано на рисунку 4.52. На наступному кроці (рис. 4.53) зробимо зрозумілий напис на кнопці, щоб пояснити її призначення, наприклад *Виконати переказ*. Результатом конструювання кнопки стане рисунок 4.54.

Для правильного функціонування форми недостатньо кнопки *Виконати переказ;* потрібно також побудувати кнопку *Сохранить запись,* як показано на рисунку 4.55. Збережемо форму (наприклад, закриваючи її).

Перейдемо у режим заповнення форми (рис. 4.56). Заповнимо поля форми. Натиснемо послідовно кнопки *Сохранить запись, Виконати переказ*. Закриємо форму. Перевіримо стан усіх таблиць після виконання запитів.

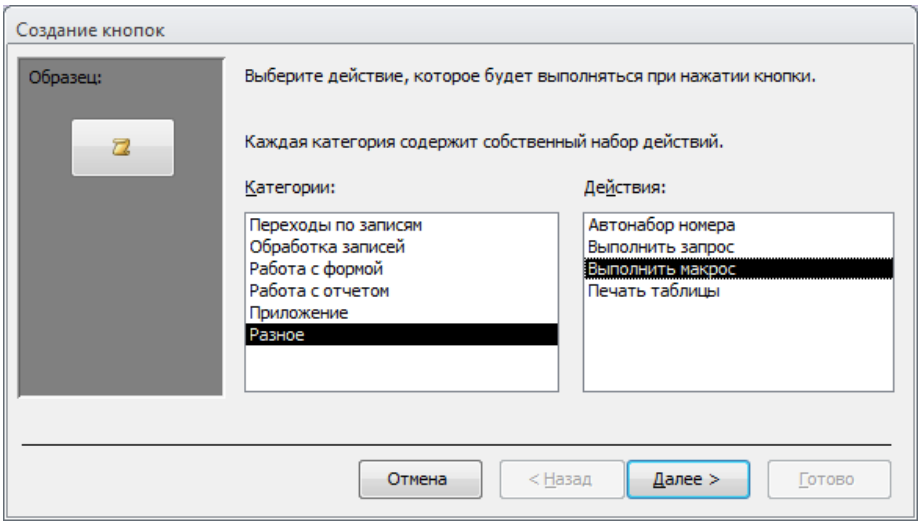


Рис. 4.52. Конструювання кнопки на формі

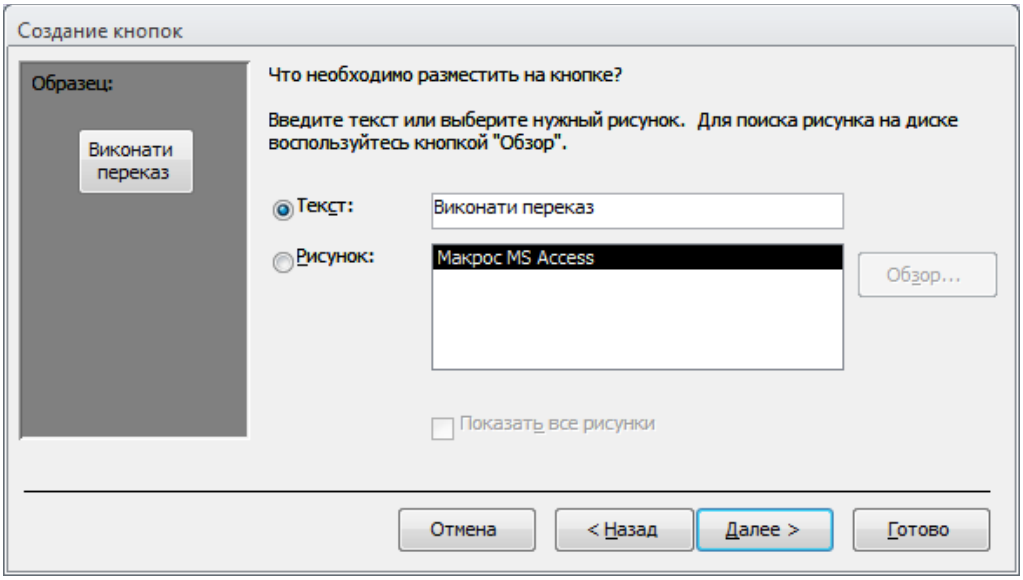


Рис. 4.53. Налаштування напису на кнопці форми

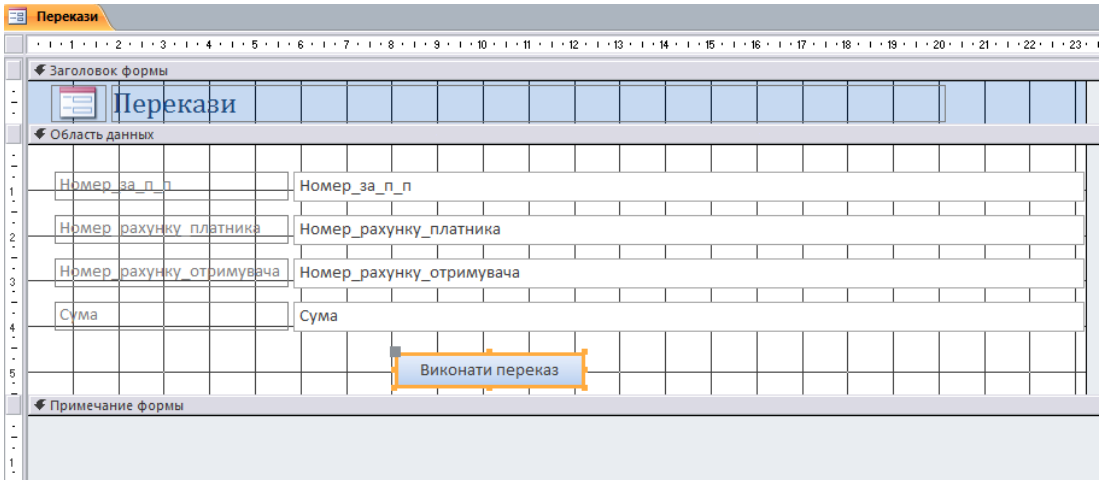


Рис. 4.54. Результат конструювання кнопки *Виконати переказ* на формі

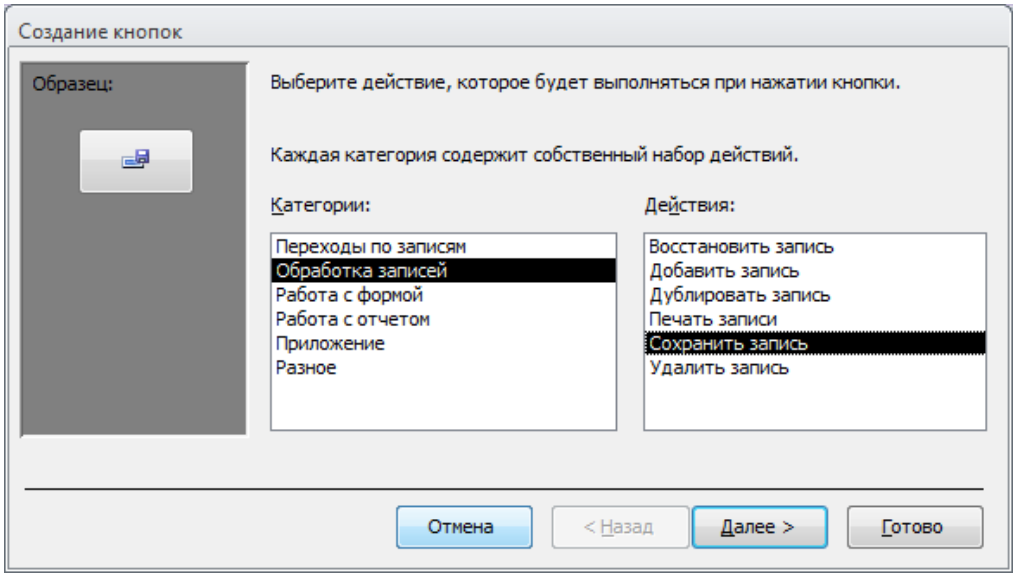


Рис. 4.55. Конструювання кнопки *Сохранить запись*

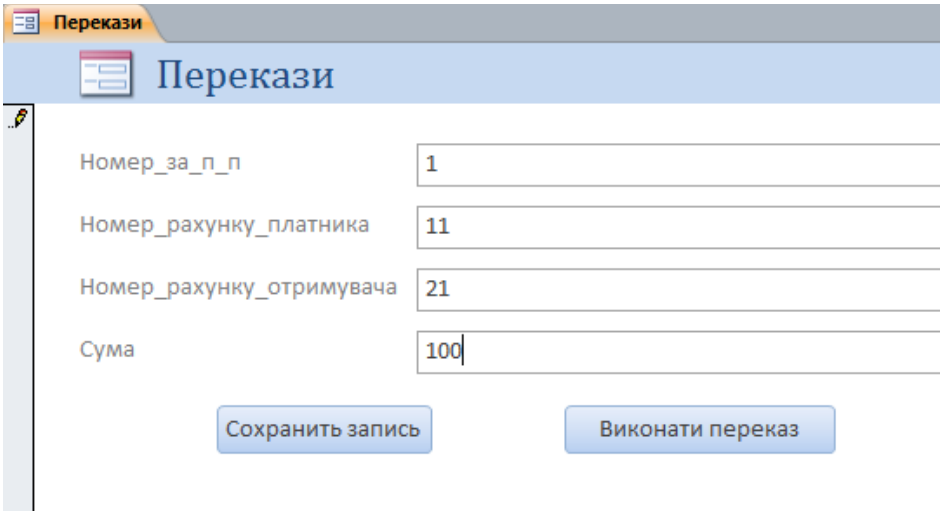


Рис. 4.56. Форма *Перекази* в режимі заповнення

Отже, у розділі 4.2 запропоновано принципову схему руху коштів між особовими рахунками клієнтів банку та реалізовано засобами системи управління базами даних Access.

Функціонал побудованої бази даних відповідає функціоналу модуля «Особові рахунки» програмного комплексу «Операційний день банку». Функціонал дає змогу:

– переглянути реквізити рахунку, зокрема номер балансового рахунку, номер особового рахунку, тип рахунку, його призначення та інші довідкові ознаки;

– визначити оперативний стан особового рахунку, а саме залишок на рахунку;

– перевести кошти з одного рахунку на інший;

– переглянути архів операцій руху коштів між особовими рахунками клієнтів банку.

Під час розробки бази даних було вирішено такі завдання:

– розробка інфологічної моделі;

– побудова таблиць з даними;

– проєктування запитів, форм, звітів, макросів.

Побудована база даних може бути корисною для якісної підготовки студентів спеціальності «Фінанси» як навчальне середовище для засвоєння теоретичних знань та отримання практичних навичок роботи у комп'ютерних програмах працівників банку.

Ця база має досить простий і зручний інтерфейс, забезпечений СКБД Microsoft Access, що дозволяє будь-якому користувачу вільно користуватись та модифікувати його.

Запитання для самоконтролю

1. Яку базу даних називають реляційною?
2. Які об’єкти існують у реляційній базі даних?
3. Які способи створення таблиць існують у Microsoft Access 2013?
4. Як створити таблицю в режимі Таблиці?
5. Як створити таблицю в режимі Конструктора?
6. Як виконати імпортування таблиці?
7. Які типи полів можна використовувати в Microsoft Access 2013?
8. Які основні властивості мають текстові поля і як їх задати?
9. Які властивості мають числові поля і як їх задати?
10. Скільки символів можна ввести в поле текстового типу?
11. Які властивості має поле МЕМО? Що зберігається в полі МЕМО?
12. Що таке маска введення, як її задати?
13. Які поля мають тип даних поле об’єкта OLE?
14. Яке поле таблиці можна вважати унікальним полем?
15. Як надати унікальному полю статусу первинного ключа таблиці?
16. Поле якого типу є ключем у більшості таблиць?
17. Яку назву має запис таблиці реляційної бази даних?
18. Яку назву має поле запису таблиці реляційної бази даних?
19. Як змінити назву поля у таблиці, його властивості чи вилучити його з таблиці?
20. Які режими відображення таблиці існують у Microsoft Access 2013 і яке їх призначення? Як встановити відповідний режим?
21. Чи потрібно переходити в режим Конструктора у разі вилучення деякого запису в таблиці?
22. Як і з якою метою пов’язують таблиці в MS Access 2013?
23. Що означають прапорці *Обеспечение целосности данных, Каскадное обновление* і *Каскадное удаление* даних у діалоговому вікні *Изменение связей*?

24. Який зміст має тип даних *Мастер подстановок*?

25. Які типи відношень між таблицями існують у реляційній базі даних?

26. Який зміст має відношення «один до багатьох»?

27. Який зміст має відношення «один до одного»?

28. Як активізувати вікно *Схема даних*?

29. Як виконати зв'язування таблиць?

30. Як додати до *Схеми даних* нову таблицю?

31. Як вилучити із *Схеми даних* таблицю?

32. Як розірвати зв'язок між таблицями?

33. Яке призначення мають звіти?

34. На основі яких об'єктів бази даних створюються звіти?

35. Які способи створення звітів можливі в MS Access 2013?

36. Чи можна при розробці звітів виконувати групування та обчислення і створювати підсумки?

37. Що являє собою звіт у режимі *Конструктора*?

38. Які структурні одиниці має звіт у режимі *Конструктора*?

39. У якій частині звіту розміщуються імена полів?

40. У якій частині звіту розміщуються підсумки за групами?

41. У якій частині звіту розміщуються загальні підсумки?

42. Які функції використовуються у групових підсумкових операціях?

43. Які функції використовуються у загальних підсумках?

44. Які об'єкти управління присутні у звітах, відображених у режимі Конструктора?

45. Як створити звіт у режимі Конструктора?

46. Яке призначення мають запити і які різновиди запитів існують у MS Access 2013?

47. Як створити запит на вибірку?

48. Як створити запит на оновлення?

49. Як створити запит на додавання записів?

50. Як створити запит на вилучення записів?

51. Як створити запит на створення нової таблиці?

52. Як створити запит на вибірку?

53. Як створити перехресний запит у базі даних в MS Access 2013?

54. Чи можна пов’язати таблиці, в яких не визначені ключові поля?

55. Як створити розрахункове поле у запиті?

56. Які способи створення виразів у запитах існують у MS Access 2013?

57. Які різновиди обчислень можна виконувати в запитах?

58. У чому перевага завдання обчислень саме в запитах?

59. Де зберігаються запити і як їх застосувати?

60. Які об’єкти бази даних створюються на основі запитів?

61. Що таке мова SQL?

ЛЕКЦІЯ 5
АНАЛІЗ ЧИСЛОВИХ ДАНИХ У СЕРЕДОВИЩІ MS EXCEL 2013

5.1. Технологія роботи із структурованими документами у табличному процесорі MS Excel 2013

MS Excel 2013 належить до програмних продуктів, які класифікуються як електронні таблиці. *Електронна таблиця* – це інтерактивна програма, що складається з набору рядків і стовпців, зображених на екрані в спеціальному вікні. Зона, що міститься на перетинанні рядка і стовпця, називається *чарунка (клітинка)*. У чарунці можуть бути число, текст чи формула, за допомогою якої здійснюють обчислення, що належать до однієї чи кількох чарунок. Чарунки можна копіювати, переміщати, змінювати їхній розмір, форматувати, а також редагувати відображувану в них інформацію. У разі зміни значення чарунки відбувається автоматичне перерахування значень усіх чарунок, що використовують у формулах змінену чарунку. На основі груп чарунок створюються діаграми, зведені таблиці і карти. Електронну таблицю можна зберігати в окремому файлі для подальшого її використання.

Мова XML дозволяє організувати структуру й працювати з даними й книгами способами, які раніше були неможливими або складними. Використовуючи схеми XML, можна визначати й виділяти певні частини бізнес-даних зі звичайних бізнесів-документів.

Наприклад, рахунок-фактура, що містить ім’я й адресу покупця або звіт з фінансовими показниками за попередній квартал не будуть більше статичними звітами, відомості, що містяться в них, можуть бути перенесені в базу даних або використані ще за межами книги.

Використання налагоджених схем XML. Після створення або відкриття книги в Microsoft Excel 2013 до неї можна приєднати налагоджену схему XML. Потім за допомогою сфери завдань *XML-джерело* варто зіставити комірки з елементами схеми. Після зіставлення елементів XML з аркушем можна ефективно імпортувати й експортувати дані XML у зіставлені комірки або з них.

Нижче перелічені деякі з дій, які можна виконувати за допомогою можливостей мови XML, реалізованих у Microsoft Office Excel 2013.

- Відкриття файлів, що містять дані XML, у новій книзі.
- Розширення можливостей існуючих шаблонів за допомогою зіставлення елементів XML існуючим полям. Це полегшує додавання й видалення даних у шаблонах, не вимагаючи їхнього повного переконструювання.
- Використання даних XML як вхідних даних для існуючих схем обчислення шляхом зіставлення елементів XML схемам обчислення, розташованим на аркуші.
- Зіставлення схем, що набудовують, XML з даними книги.
- Вставка даних XML, отриманих від вебслужб, на аркуш Microsoft Excel 2013.

Використання схем таблиць XML. Звичайну книгу Microsoft Excel 2013 можна зберегти у форматі таблиці XML. Microsoft Excel 2013 використає власні XML-схеми (XMLSS) під час розміщення XML-тегів, які зберігають різні відомості, наприклад про властивості файлу, і визначають структуру книги.

Функціональні можливості й обмеження формату XML-таблиці. Під час збереження електронної таблиці Microsoft Excel 2013 у форматі XML-таблиці зберігаються багато функціональних можливостей Excel, але не все. Вони докладно описані в наведеній нижче таблиці.

Збереження й експорт даних XML. Існує кілька способів використання й експортування вмісту аркуша у файл даних XML.

Збереження вмісту зіставленого діапазону у файл даних XML

1. У меню *Файл* виберіть команду *Зберегти*.

При цьому всі зміни, внесені в книгу, не будуть загублені у разі збереження у файл даних XML.

2. У меню *Файл* виберіть команду *Зберегти як*.
3. У поле *Ім'я файлу* введіть ім'я файлу даних XML.
4. У списку *Тип файлу* виберіть пункт *XML-дані*.
5. Натисніть кнопку *Зберегти*.
6. Під час одержання повідомлення про те, що збереження файлу у вигляді *XML-даних* може призвести до втрати можливостей аркуша, натисніть кнопку *Продовжити*.

7. У діалоговому вікні *Експорт XML-даних* виберіть карту XML, яку необхідно використати, і натисніть кнопку *OK*.

Діалогове вікно *Експорт XML-даних* буде відображено тільки у разі додавання в книгу більше ніж однієї карти XML.

Примітка. Поточна книга тепер є файлом даних XML. Щоб продовжити роботу у вихідній книзі, закрийте файл даних XML і відкрийте вихідну книгу знову.

Експорт умісту зіставленого діапазону у файл даних XML

1. На панелі інструментів *Список* натисніть кнопку *Експорт*.
2. У діалоговому вікні *Експорт XML-даних* виберіть карту XML, яку необхідно використати, і натисніть кнопку *OK*.

Діалогове вікно *Експорт XML-даних* буде відображено тільки у разі додавання в книгу більше ніж однієї карти XML.

3. У поле *Ім'я файлу* введіть ім'я файлу даних XML.
4. Натисніть кнопку *Експорт*.

Відкриття файлу, що містить дані XML.

1. У меню *Файл* виберіть команду *Відкрити*.
2. У списку, що розкривається, Папка виберіть диск, папку або адресу в інтернеті, де перебуває файл, що потрібно відкрити.
3. У списку папок знайдіть і відкрийте папку, що містить потрібний файл.
4. Клацніть значок файлу й натисніть кнопку *Відкрити*.

З'явиться діалогове вікно *Відкриття XML*.

5. Виберіть один з таких варіантів і натисніть кнопку *OK*.

XML-список. У новій робочій книзі створюється XML-список. У цей XML-список імпортується вміст файлу, що відкриває. Якщо файл даних XML не посилається ні на яку схему, то в Microsoft Excel буде побудована схема цього файлу.

Книга, доступна тільки для читання.

Файл даних XML відкривається як робоча книга, доступна тільки для читання. Структура цього файлу стає плоскою.

Використати сферу завдань XML-джерела.

Схема файлу даних XML відображається у сфері завдань *XML-джерело*. Для відображення елементів схеми на аркуші їх можна перетягнути на аркуш за допомогою миші.

Під час відкриття файлу даних XML, що не посилається ні на яку схему, в Microsoft Excel буде побудована схема цього файлу.

Збереження й експорт даних XML. Існує кілька способів використання й експортування вмісту аркуша у файл даних XML.

Збереження інформації у файл даних XML

1. У меню *Файл* виберіть команду *Зберегти*.

При цьому всі зміни, внесені в книгу, не будуть загублені під час збереження у файл даних XML.

2. У меню *Файл* виберіть команду *Зберегти як*.
3. У поле *Ім'я файлу* введіть ім'я файлу даних XML.
4. У списку *Тип файлу* виберіть пункт *XML-дані*.
5. Натисніть кнопку *Зберегти*.
6. У разі одержання повідомлення про те, що збереження файлу у вигляді XML-даних може призвести до втрати можливостей аркуша, натисніть кнопку *Продовжити*.
7. У діалоговому вікні *Експорт XML-даних* виберіть карту XML, яку необхідно використати, і натисніть кнопку *OK*.

Діалогове вікно *Експорт XML-даних* буде відображено тільки у разі додавання в книгу більше ніж однієї карти XML.

Примітка. Поточна книга тепер є файлом даних XML. Щоб продовжити роботу у книзі, закрийте файл даних XML і відкрийте книгу заново.

Експорт інформації у файл даних XML

1. На панелі інструментів *Список* натисніть кнопку *Експорт*.
2. У діалоговому вікні *Експорт XML-даних* виберіть карту XML, яку необхідно використати, і натисніть кнопку OK.

Діалогове вікно *Експорт XML-даних* буде відображено тільки у разі додавання в книгу більше ніж однієї карти XML.

3. У поле *Ім'я файлу* введіть ім'я файлу даних XML.
4. Натисніть кнопку *Експорт*.

Сфера завдань XML-джерело

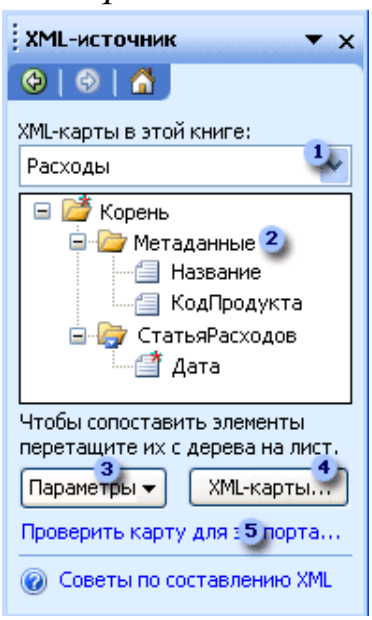


Рис. 5.1. ***Сфера завдань XML-джерело***

1. Список карт XML, доданих у книгу.
2. Ієрархічний список елементів карти XML.
3. Установка параметрів сфери завдань *XML-джерело*.

Переглянути дані у сфері завдань. Якщо були імпортовані дані XML, зіставлені з картою XML у поточній версії Microsoft Excel, то під час установки цього прапорця відображаються зразки даних у списку елементів.

Сховати довідковий текст у сфері завдань. Якщо цей прапорець установлений, довідковий текст, розташований під списком елементів, ховається.

Автоматично поєднувати елементи під час зіставлення. Якщо цей прапорець установлений, списки XML автоматично розширюються у процесі перетаскування елемента в клітку, суміжну зі списком XML.

Дані із заголовками. Якщо цей прапорець встановлений, то існують дані, що можуть бути використані як заголовки стовпців у разі зіставлення з аркушем повторюваних елементів.

Сховати границі неактивних списків. Якщо цей прапорець установлений, границі списку, списку XML або окремо зіставленого осередку не відображаються під час виділення осередку за межами списку.

4. Відкриття діалогового вікна *Карти XML*, у якому можна додавати, видаляти й перейменовувати карти XML.

5. Перевірка, чи можливий експорт даних XML за допомогою поточної карти XML.

Зіставлення елементів XML з аркушем

1. Для зіставлення одного або декількох елементів з аркушем виберіть ці елементи у сфері завдань *XML-джерело*. Для вибору несуміжних елементів клацніть один елемент, а потім, утримуючи клавішу CTRL, клацніть інші потрібні елементи.

2. Перетягнете обрані елементи туди, де вони повинні відображатися.

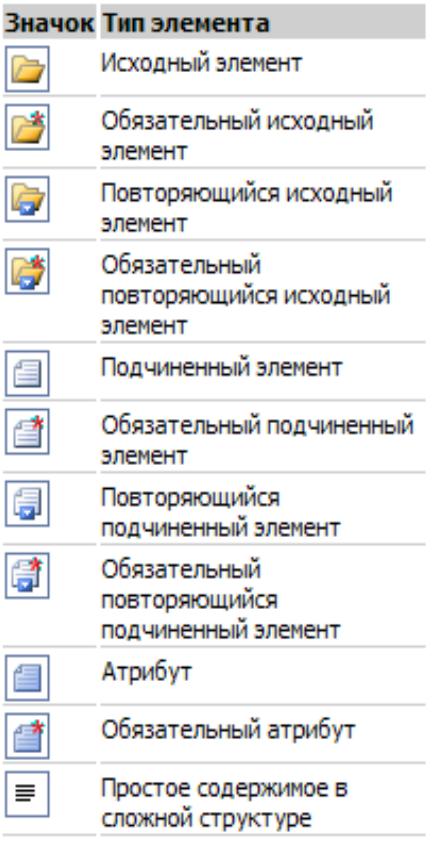


Рис. 5.2. Позначення в списку елементів

Обмеження на розташування списків XML. Список XML створюється у процесі відображення на аркуші повторюваного елемента XML. Списки XML полегшують імпорт, експорт, сортування, фільтрування й печатку даних, заснованих на джерелі XML-даних. Однак існують деякі обмеження на розташування списків XML на аркуші.

Списки XML організовані за рядками, іншими словами, вони ведуться вниз від рядка заголовка. Не можна додавати нові записи над існуючими рядками. Неможливо також транспонувати список XML так, щоб нові записи додавалися праворуч.

Зіставлення XML-елементів з аркушем

Примітка. Коди XML, за винятком збереження файлів у форматі таблиць XML, доступні тільки в Microsoft Office Professional версія 2003 й окремому випуску Microsoft Office Excel 2003.

1. У меню *Дані* виберіть команду *XML*, а потім команду *Джерело XML*, щоб відкрити сферу завдань *XML-джерело*.

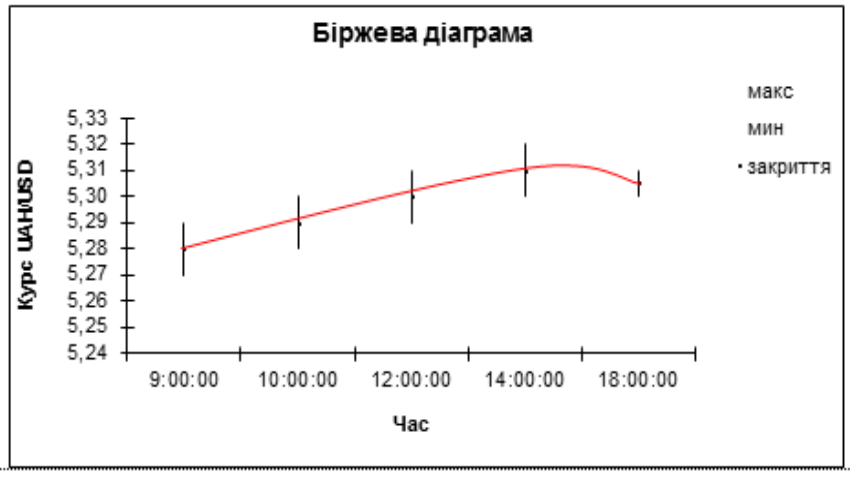
2. Для зіставлення одного або декількох елементів з аркушем виберіть ці елементи у сферу завдань *XML-джерело*. Для вибору несуміжних елементів клацніть один елемент, а потім, утримуючи клавішу CTRL, клацніть інші потрібні елементи.

3. Перетягнете обрані елементи туди, де вони повинні відображатися.

Таблиця 5.1

Типи діаграм у табличному процесорі Excel та їх призначення

Стандартні	Нестандартні
Гістограма	Блоки із зонами
Лінійчата	Вирізані сектори
Графік	Гістограма зони
Кругова	Гладкий графік
Точкова	Графік гістограма
Із зонами	Дерев'яна
Кільцева	Конуси
Пелюсткова	Логаріфмічна
Поверхня	Трубчаста
Пузирькова	Кольорові графіки
Біржова	Бруски
Конічна	Широка гістограма
Циліндрична	
Пірамідальна	



Діаграми *призначені для ілюстрації функціональної залежності однієї величини від іншої або для порівняння двох і більше величин, а також з метою виявлення тенденцій змін будь-якого параметра в часі, відображення відсоткового вмісту ряду компонентів у деякому об'єкті.*

Технологія створення діаграм та графіків на основі табличних даних.

Створення діаграми

1. Вибір типу та вигляду діаграми, а також перегляд її зразка.
2. Вибір або зміна діапазону комірок, що містять дані, на основі яких буде побудовано діаграму. Визначення способу формування її рядків.
3. Введення заголовків діаграми та її осей. Встановлення будь-яких параметрів діаграми, пов'язаних з її зовнішнім оформленням.
4. Вибір варіанта розміщення діаграми (на поточному або окремому листі).

Таблиця 5.2

Приклад. Біржова діаграма

час	**макс.**	**мін.**	**закриття**
9:00:00	5,29	5,27	5,280
10:00:00	5,30	5,28	5,290
12:00:00	5,31	5,29	5,300
14:00:00	5,32	5,30	5,310
18:00:00	5,31	5,30	5,305

5.2. Побудова реляційної бази даних бізнес-процесів у MS Excel 2013

У загальному значенні термін база даних можна застосувати до будь-якої сукупності пов'язаної упорядкованої інформації, об'єднаної разом за певною ознакою, наприклад телефонні списки, списки клієнтів, транзакцій, активів, пасивів тощо. Основним призначенням баз даних є швидкий пошук інформації, що міститься в них.

За допомогою Microsoft Excel 2013 можна створювати й обробляти бази даних. База даних у Microsoft Excel – таблиця, що складається з однотипних записів (рядків). Стовпці таблиці є полями запису в базі даних. Під імена полів виділено перший рядок у базі даних. Наприклад, якщо базою даних вважати телефонний довідник, то полями запису будуть прізвища, номера телефонів і адреси абонентів.

	A	B	C	D	E
1	Код замовлення	Товар	Ціна	Кількість	Знижка
2	78653	Quesco Cabrales	99,00₴	18	0%
3	78654	Valkoinen Suklaa	187,00₴	14	0%
4	78656	Louisiana Fiery Hot	17,00₴	23	0%
5	78657	Mozzarella di Glovanni	189,00₴	11	10%

Рис. 5.3. Загальний вигляд таблиці Excel

Для роботи з базою даних необхідно спочатку створити відповідну таблицю. Якщо виділити вічко в таблиці і вибрати одну з команд обробки баз даних у меню *Данные,* Microsoft Excel автоматично визначає й обробляє всю таблицю. Дані, розташовані в стовпцях і рядках робочого аркушу, обробляються як набір полів, що утворюють записи.

Excel 2013 має у своєму розпорядженні набір функцій, призначених для аналізу списку. Однією з найчастіше розв'язуваних за допомогою електронних таблиць є обробка списків. Унаслідок цього Microsoft Excel 2013 має багатий набір засобів, що дають змогу значно спростити обробку таких даних. Нижче наведено кілька порад з роботи зі списками.

У Excel 2013 бази даних розміщуються в таблицях. Кожна таблиця складається з рядків і стовпців, що у базах даних називаються записами і полями.

Інформація в базах даних має постійну структуру. Кожен рядок можна розглядати як одиничний запис. Інформація в межах кожного запису міститься в полях.

Працюючи з базами даних в Excel 2013, насамперед потрібно ввести заголовки стовпців. Після цього можете ввести інформацію в базу даних. Уведення даних і перегляд інформації можна здійснювати за допомогою команди *Данные/Форма*.

На робочому аркуші можна обробляти кілька списків, але краще помістити кожен список на окремий аркуш робочої книги.

Засоби обробки списків Excel 2013 можна використовувати, не створюючи унікальні заголовки стовпців, але все-таки це краще зробити.

Ефективність списку обумовлена такими характеристиками. Кожен стовпець повинен містити однорідну інформацію. Наприклад, у списку персоналу можна відвести один стовпець для особистих номерів працівників, другий – для їхніх прізвищ, третій – для імен, четвертий – для дати прийому на роботу тощо. Один чи два верхні рядки у списку мають містити мітки, що описують призначення відповідного стовпця. Необхідно уникати порожніх рядків і стовпців усередині списку.

В ідеалі на робочому аркуші не повинно бути нічого, крім списку. Якщо це неможливо, то список потрібно відокремити від інших даних принаймні одним порожнім рядком і одним порожнім стовпцем.

Якщо планується фільтрувати список, не вміщуйте якісь дані на рядки, в яких він розташований.

Щоб підкреслити заголовки, скористайтеся вкладками *Шрифт* або *Рамка* діалогового вікна *Формат ячеек*. (Воно відкривається за командою *Формат/Ячейки*.) Не вставляйте окремий рядок з дефісів чи знаків рівності, тому що Excel може витлумачити таке підкреслення як дані.

Нові дані можна додавати безпосередньо в кінець списку. Досить зручним може виявитися порядкове редагування списку за допомогою стандартної екранної форми. Для цього виділіть якусь комірку в списку та виконайте команду *Данные/Форма*.

Примітка. Перед виконанням команди *Данные/Форма* повинна бути виділена тільки одна комірка в списку, інакше у формі можуть з'явитися неправильні заголовки стовпців.

Порада. В Excel є команда *Текст по столбцам*, що дає можливість створювати списки з даних, які містяться в текстових файлах. Також можна скористатися *Майстром перетворень*.

Команда *Форма* відображає на екрані форму, що являє собою вікно діалогу, призначене для перегляду та редагування записів у базі даних, а також для додавання нових і видалення існуючих записів. Крім того, за допомогою форми можете здійснити пошук конкретних записів на підставі складних критеріїв.

Кожному стовпцю таблиці у формі відповідає поле, найменування якого визначається введеним заголовком стовпця. Крім таких полів, у формі можуть бути присутніми поля, що обчислюються за допомогою різноманітних обчислень. Значення цих полів автоматично заповнюються Excel на підставі інформації, що міститься в базі даних. Ви не можете безпосередньо ввести чи змінити значення в полях, що обчислюються, тому в режимі редагування ці поля у формі не відображаються. Крім того, можете самостійно визначити деякі з полів таблиці як редаговані. У результаті ці поля також не відображатимуться у формі в режимі редагування.

5.3. Засоби роботи з базою даних у MS Excel 2013

Фільтрування даних. Пошук даних називають фільтрування даних. Фільтрація списку дає можливість знаходити й відбирати для обробки частину записів у списку, таблиці, базі даних.

В Excel 2013 для фільтрації даних використовуються команди *Автофильтр* і *Расширенный фильтр*. У випадку простих критеріїв для вибірки потрібної інформації досить команди *Автофильтр*, у комірках виділених заголовків повинні з'явитися нові елементи. За їх допомогою можна встановлювати критерії фільтрації даних. Виклик команди фільтрації – команда <*Дані-Сортування*>.

Ідея принципу фільтрації полягає в тому, що в якийсь момент часу в таблиці візуально відображаються лише ті дані, які задовольняють заданим критеріям. Ці критерії можуть міститися в комірках самої таблиці або бути умовою, що вказується користувачем.

Таблиця 5.3

Початковий стан бази даних підприємства (до фільтрації)

№	Замовник	Шифр товару	Кіль-кість, шт	Дата замовлення	Ціна, грн	Сума, грн
1.	ВАТ «Старт»	ШТ-22	4	01.02.2002	22,00	88,00
2.	ТОВ «Атом»	ШТ-23	5	04.01.2002	12,00	60,00
3.	МП «Еталон»	ШТ-15	34	12.05.2002	2,00	68,00
4.	ПП «Маяк»	ШТ-22	23	01.02.2002	22,00	506,00
5.	ПП «Маяк»	ШТ-23	11	04.01.2002	12,00	132,00
6.	НВО «Протон»	ШТ-15	7	12.05.2002	2,00	14,00
7.	ТОВ «Атом»	ШТ-22	45	01.02.2002	22,00	990,00
8.	ВАТ «Старт»	ШТ-23	3	04.01.2002	12,00	36,00
9.	НВО «Протон»	ШТ-15	23	12.05.2002	2,00	46,00
10.	ПП «Маяк»	ШТ-22	45	01.02.2002	22,00	990,00
11.	ВАТ «Старт»	ШТ-23	67	04.01.2002	12,00	804,00

Приклад

1. Відбір за критерієм поля *Замовник* Дані – Автофільтр. У спадному меню поля *Замовник* вказати те підприємство, яке вас цікавить. Відбір за іншими умовами аналогічно.

2. Відбір за умовою. Вказати умову. Припустимо, всі товари за ціною 12 грн або ті, які були замовлені в певний період.

3. На наступному рисунку (рис. 5.4) показано результат фільтрації за умовою: кількість більше 7.

Команда *Фильтр* меню *Данные* дозволяє відшукувати і використовувати потрібну підмножину даних у переліку. У відфільтрованому переліку виводяться на екран тільки ті рядки, що містять визначене значення або відповідають визначеним критеріям, при цьому інші рядки виявляються приховані. Для фільтрації даних використовуються команди *Автофильтр* і *Расширенный фильтр* з пункту *Фильтр* меню *Данные*.

	A	B	C	D	E	F	G
1	№	Замовник	Шифр товару	Кількість,шт	Дата замовлення	Ціна,грн.	Сума.грн.
4	3	МП "Еталон"	ШТ-15	34	12.05.2002	2,00 грн.	68,00 грн.
5	4	ПП "Маяк"	ШТ-22	23	01.02.2002	22,00 грн.	506,00 грн.
6	5	ПП "Маяк"	ШТ-23	11	04.01.2002	12,00 грн.	132,00 грн.
8	7	ТОВ "Атом"	ШТ-22	45	01.02.2002	22,00 грн.	990,00 грн.
10	9	НВО "Протон"	ШТ-15	23	12.05.2002	2,00 грн.	46,00 грн.
11		ПП "Маяк"	ШТ-22	45	01.02.2002	22,00 грн.	990,00 грн.
12		ВАТ "Старт"	ШТ-23	67	04.01.2002	12,00 грн.	804,00 грн.

Рис. 5.4. Результат фільтрації за умовою: кількість більше 7

Автофильтр. Команда *Автофильтр* установлює кнопки прихованих переліків (кнопки зі стрілкою) безпосередньо в рядок з іменами стовпців (рис. 5.5). За їх допомогою можна вибрати записи бази даних, що слід вивести на екран. Після виділення елементу в переліку, що розкривається, рядки, що не містять даний елемент, будуть приховані. Наприклад, якщо у прихованому переліку поля *Ціна* вибрати 99 грн, то будуть виведені тільки записи, в яких у полі *Ціна* міститься значення 99 грн.

	A	B	C	D	E
1	Код замовлення	Товар	Ціна	Кількість	Знижка
2	78653	Quesco Cabrales	99,00₴	18	0%
3	78654	Valkoinen Suklaa	187,00₴	14	0%
4	78656	Louisiana Fiery Hot	17,00₴	23	0%
5	78657	Mozzarella di Giovanni	189,00₴	11	10%

Рис. 5.5. Кнопки прихованих переліків для фільтрації даних

Якщо у прихованому переліку вибрати пункт *Условие …*, то з'явиться вікно *Пользовательский автофильтр*. У верхньому правому переліку варто вибрати один з операторів (дорівнює, більше, меньше та ін.), а у полі праворуч вибрати одне зі значень. У нижньому правому переліку можна вибрати інший оператор, і у полі ліворуч – значення. Коли увімкнений перемикач *И*, то будуть виводитися тільки записи, які задовольняють обидві умови. У разі увімкненого перемикача *ИЛИ* будуть виводитися записи, які задовольняють одну з умов. Наприклад, у вікні введені умови для виведення записів з виробів за ціною більше 99 грн і менше 187 грн.

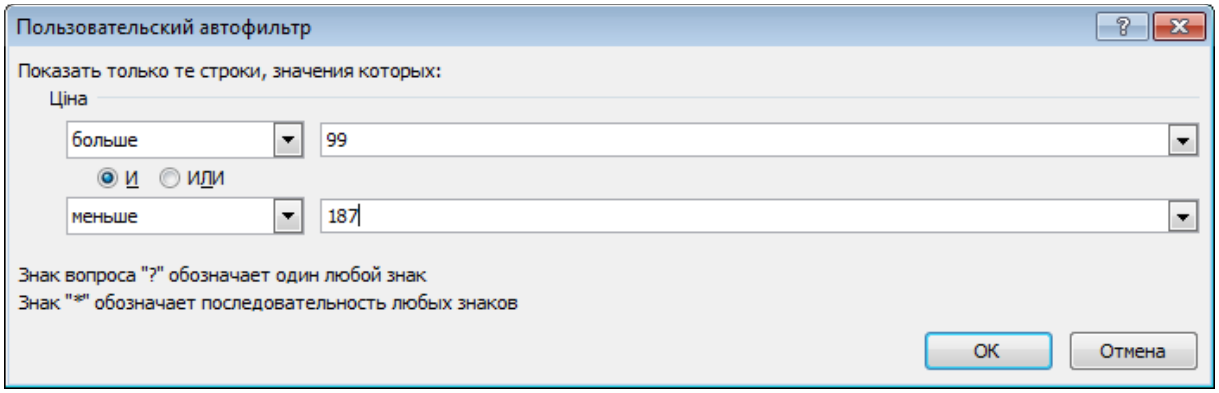


Рис. 5.6. Конструювання умови для фільтрації даних

Для виведення декількох записів з найбільшими або найменшими значенням за будь-яким полем варто у прихованому переліку поля вибрати пункт *Первые 10*. У діалоговому вікні *Наложение* условия по списку у першому полі з лічильником необхідно вибрати кількість записів, а у полі праворуч вибрати найбільших або найменших.

Щоб вивести усі дані переліку, необхідно викликати команду *Отобразить* все або скасувати команду Автофильтр меню Данные підменю Фильтр.

Розширений фільтр. Команда *Расширенный фильтр* дозволяє фільтрувати дані з використанням інтервалу критеріїв для виводу тільки тих записів, що задовольняють визначеним критеріям. У разі повторної фільтрації будуть проглядатися усі рядки: і приховані, і відкриті. Значення перемикачів і полів вікна *Расширенный фильтр такі*:

- *Фільтрувати список на місці* – перемикач, що приховує рядки, які не задовольняють зазначеному критерію;
- *Скопіювати результат в друге місце* – копіює відфільтровані дані на інший робочий аркуш або на інше місце на тому самому робочому аркуші;
- *Исходный диапазон* – поле, що визначає інтервал, якій містить перелік, що підлягає фільтрації;
- *Диапазон условий* – поле, що визначає інтервал вічок на робочому аркуші, який містить необхідні умови;
- *Поместить результат в диапазон* – поле, що визначає інтервал вічок, в який копіюються рядки, що задовольняють визначеним умовам; це поле активно тільки в тому випадку, якщо обраний перемикач *Скопировать результат в другое место*;

- *Только уникальные записи* – перемикач, що виводить тільки рядки, що задовольняють критерію і не містять неповторюваних елементів. Якщо інтервал критеріїв не визначений, то в цьому випадку усі рядки переліку, що містять дублікати, будуть приховані.

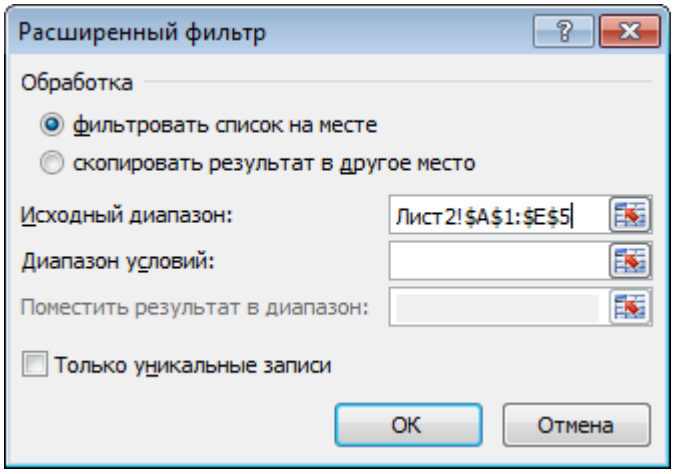


Рис. 5.7. Вікно розширеного фільтра

Для установлення складних критеріїв необхідно:

1) уставити декілька рядків у верхній частині робочого аркуша;

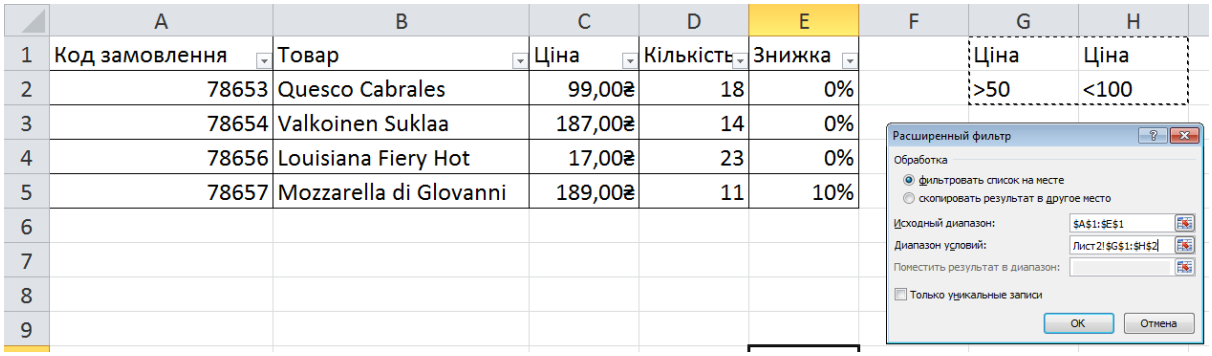
2) в одному із уставлених порожніх рядків увести імена стовпців, за якими потрібно відфільтрувати перелік;

3) у разі використання критеріїв порівняння імена критеріїв повинні бути ідентичні іменам стовпців, що перевіряються;

4) у рядках, розташованих під рядком з іменами стовпців, що перевіряються, увести критерії, яким повинні відповідати вічка стовпців, що перевіряються;

5) вибрати в меню *Данные* підменю *Фильтр*, а потім команду *Расширенный фильтр*, і в діалоговому вікні увести умови фільтрації.

Для об'єднання критеріїв за допомогою умовного оператора И потрібно зазначити критерії в одному і тому самому рядку, а для об'єднання критеріїв за допомогою умовного оператора ИЛИ варто подати критерії в різних рядках. Наприклад, інтервал критеріїв виводить на екран усі записи, що мають у стовпці *Ціна* значення більше 50 і менше 100.



	A	B	C	D	E	F	G	H
1	Код замовлення	Товар	Ціна	Кількість	Знижка		Ціна	Ціна
2	78653	Quesco Cabrales	99,00₴	18	0%		>50	<100
3	78654	Valkoinen Suklaa	187,00₴	14	0%			
4	78656	Louisiana Fiery Hot	17,00₴	23	0%			
5	78657	Mozzarella di Glovanni	189,00₴	11	10%			
6								
7								
8								
9								

Рис. 5.8. Підготовка до використання розширеного фільтра

Після використання команд *Автофильтр* або *Расширенный фильтр* таблиця переходить у режим фільтрації. У цьому режимі багато команд Microsoft Excel впливають тільки на зримі вічка. Після застосування фільтра для виводу тільки потрібних рядків можна скопіювати отриману підмножину даних в інше місце для подальшого аналізу.

Щоб знову вивести усі записи, слід у меню *Данные* вибрати пункт *Фильтр* і потім пункт *Отобразить все.*

Сортування даних. Команда <Дані-Сортування>.

Сортування використовується для упорядкування рядків у списку відповідно до вмісту окремого стовпця. Сортуванню підлягають дві категорії даних – текстові і числові. Якщо сортуванню підлягають дані в стовпці, що пов'язаний з іншими стовпцями за рядками, то виділяти необхідно всі ці стовпці. Якщо немає потреби в завданні особливих умов сортування, то можна користатися кнопками «сортування за зростанням» і «убування», у панелі інструментів «стандартна». Розширені можливості для сортування надаються в меню «дані» командою «сортування». Можна здійснити послідовне сортування за декількома незалежно обумовленими критеріями.

Сортування списків можна виконувати за значеннями комірок одного чи декількох стовпців. Рядки чи стовпці або окремі комірки у процесі сортування переупорядковуються відповідно до заданого користувачем порядком сортування. Списки можна сортувати у зростаючому (від 1 до 9, від А до Я) чи спадаючому (9 до 1, від Я до А) порядку.

За замовчуванням списки сортуються за абеткою. Види сортування:

- сортування рядків за зростанням даних в одному стовпці;
- сортування даних за убуванням даних в одному стовпці;
- сортування рядків за двома чи більше стовпцями;
- сортування стовпців за вмістом рядків;
- сортування місяців, днів тижня або списків, що потрібно відсортувати в порядку, відмінному від алфавітного.

Наприклад, щоб упорядкувати дані за ціною виробу. Для сортування даних варто виділити одну клітинку таблиці і викликати команду *Сортировка* із меню *Данные*.

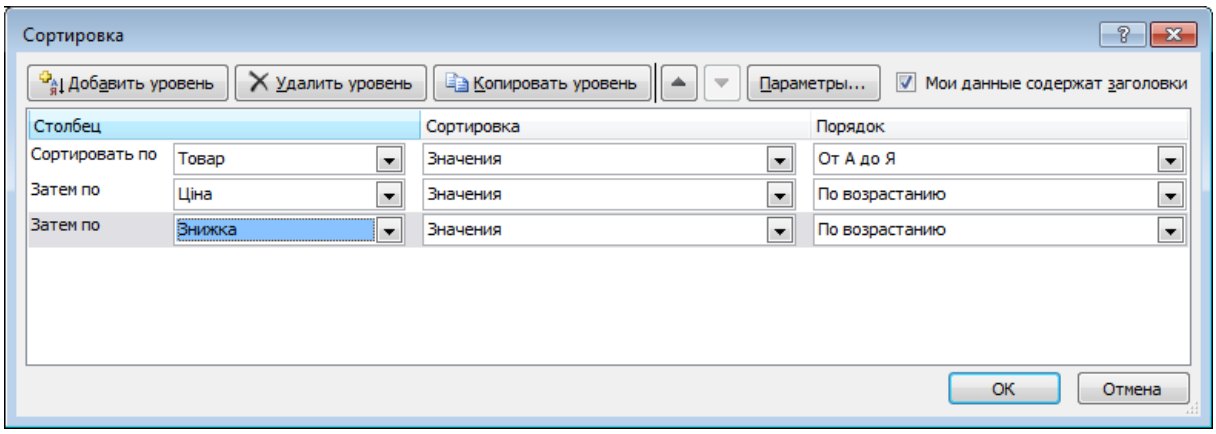


Рис. 5.9. Сортування даних таблиці

У полі прихованого переліку *Сортировать по* вибирається поле, за яким будуть відсортовані дані, і тип сортування:

- за зростанням – цифри сортуються за зростанням, текст за абеткою, логічному виразу *ХИБА* передує *ІСТИНА*;
- за спаданням – сортування в зворотному порядку.

У полі прихованого переліку *Затем по* указується поле, за яким будуть відсортовані дані, що мають однакові значення в першому ключовому полі. У другому полі *Затем по* указується поле, за яким будуть відсортовані дані, що мають однакові значення в перших двох ключових полях.

Для сортування даних також використовуються кнопки. Перед їх використанням варто виділити стовпець, за яким необхідно сортувати таблицю.

Під час сортування в одному стовпці рядки з однаковими значеннями в цьому стовпці зберігають колишнє упорядкування. Рядки з порожніми вічками в стовпці, за яким ведеться сортування, розташовуються наприкінці переліку, що сортується. Microsoft Excel 2013 дозволяє також сортувати не всю таблицю, а тільки виділені рядки або стовпці.

Форми даних. У Microsoft Excel 2013 не варто робити нічого особливого для того, щоб використовувати перелік як базу даних. У разі виконання характерних для баз даних операцій, таких як пошук, сортування, підведення підсумків, Microsoft Excel автоматично розглядає таблицю як базу даних.

У процесі перегляду, зміни, додавання і видалення запису в базі даних, а також під час пошуку конкретних записів за визначеним критерієм зручно використовувати *форми даних*. У разі звертання до команди *Форма* меню *Данные* Microsoft Excel 2013 читає дані й створює діалогове вікно форми даних. У формі даних на екран виводиться один запис. У разі уведення або зміни даних у полях цього вікна змінюється вміст відповідних вічок у базі даних.

Для використання форм даних таблиця повинна мати імена стовпців. Імена стовпців стають іменами полів у формі даних. Поле відповідає кожному стовпцю в таблиці. Форма даних автоматично розгортається так, щоб вивести на екран відразу усі поля в цій таблиці, до 32 полів за один раз. За допомогою смуги прокручування можна прокручувати записи в базі даних. Позиція виведеного запису вказується у верхньому правом куті. Пересуватись полями форми можна за допомогою миші та клавіш Tab (униз), Shift+Tab (угору). Праворуч розташовані такі кнопки:

- Добавить – очищує поля для уведення нового запису бази даних. Якщо знову натиснути кнопку *Добавить*, то уведені дані будуть додані як новий запис у кінець бази даних.
- Удалить – видаляє виведений запис, інші записи бази даних зсуваються. Видалені записи не можуть бути відновлені.
- Вернуть – відновлює відредаговані поля у виведеному запису, видаляючи зроблені зміни. Щоб відновити запис, необхідно зробити це перед натисканням клавіші *Enter* або перед переходом до іншого запису.

• Назад – виводить попередній запис у переліку. Якщо був визначений критерій за допомогою кнопки *Критерии*, то кнопка *Назад* виведе попередній запис з тих, що задовольняють заданому критерію.

• Далее – виводить наступний запис у базі даних.

• Критерии – очищає поля для уведення критеріїв порівняння з операторами порівняння для пошуку необхідної підмножини записів.

• Правка – слугує для виходу з режиму уведення критеріїв. Доступна тільки тоді, коли натиснута кнопка *Критерии*.

• Очистить – видаляє існуючий критерій із вікна діалогу. Доступна тільки тоді, коли натиснута кнопка *Критерии*.

• Закрыть – закриває форму даних.

Для додавання запису до бази даних необхідно:

– виділити клітинку в таблиці, до якої потрібно додати запис;

– у меню *Данные* вибрати команду *Форма*;

– натиснути кнопку *Добавить*;

– заповнити поля нового запису;

– для переміщення до наступного поля натиснути клавішу *Tab*;

– після уведення даних натиснути клавішу *Enter* для додавання запису;

– після додавання усіх необхідних записів натиснути кнопку *Закрыть*.

Для пошуку за допомогою форми даних записів, що відповідають критерію, необхідно:

• виділити клітинку в таблиці;

• у меню *Данные* вибрати команду *Форма*;

• натиснути кнопку *Критерии*;

• у полях редагування увести критерії для пошуку даних;

• для виводу на екран першого запису, що відповідає критерію, натиснути кнопку *Далее*;

• для виводу на екран попереднього запису, що відповідає критерію, натиснути кнопку *Назад*;

• для пошуку записів у переліку за іншим критерієм натиснути кнопку *Критерии* і ввести новий критерій;

• після закінчення натиснути кнопку *Закрыть*.

Щоб знову одержати доступ до усіх записів переліку, необхідно натиснути кнопку *Критерии*, а потім – кнопку *Правка*.

Зведені таблиці. Зведеною називається спеціальна таблиця, яка підсумовує інформацію з окремих полів списку або бази даних. Під час створення зведеної таблиці за допомогою майстра можна задати потрібні поля, структуру таблиці (її макет), тип обчислень, що виконуються. У разі потреби після побудови таблиці можна приховувати або відображати її структурні рівні. Розглянемо основні етапи роботи із зведеною таблицею.

Створення зведеної таблиці. Вибравши будь-яку комірку бази даних, виконайте команду *Зведена таблиця* з пункту меню *Дані*. Майстер зведених таблиць виводить перше з трьох вікон діалогу. У ньому потрібно зазначити, на основі яких даних створюється таблиця. Як правило, це список Excel 2013, тому залиште перемикач у списку або в базі даних Microsoft Excel 2013 встановленим. Натисніть кнопку *Дальше*. Відкривається друге вікно майстра, в якому пропонується ввести діапазон, що містить вихідні дані. Якщо вікно майстра правильно задає вихідні дані, натисніть кнопку *Дальше*, інакше введіть правильний діапазон.

У третьому вікні діалогу майстра зведених таблиць сформуйте макет нової зведеної таблиці. Для цього натисніть кнопку Макет І перемістіть кнопки з назвами полів у потрібну зону рядків, стовпців або даних чи сторінок. Зони *Стрічки*, *Стовпець*, *Дані* повинні мати відповідне поле. Інформація в зоні Дані, як правило, обробляється математично. Тому в ній розміщують числові поля. У разі переміщення кнопки поля в зону *Далі програма* Excel 2013 вказує тип операції, яку здійснить з даними. За замовчуванням обчислюється сума. Для зміни типу операції варто перемістити кнопку поля в зону *Далі* і двічі клацнути лівою кнопкою миші, на ній відкриється вікно, в якому можна вибрати потрібну функцію обробки даних.

Заповнювати зону Сторінка не обов’язково. Натисніть кнопку Ж для закриття вікна діалогу і повернення у третє вікно майстра зведених таблиць. Виберіть місце розташування зведеної таблиці – наявний або новий робочий аркуш. Натисніть кнопку

Параметри і введіть назву зведеної таблиці. Натисніть кнопку. Після натискання кнопки *Готово* майстер створить таблицю. Відкриється також панель інструментів *Зведені таблиці*. За допомогою цієї панелі можна змінювати вигляд і параметри вже побудованої таблиці. До готової таблиці можна застосувати будь-який автоформат Excel 2013. Для нього потрібно виконати команду *Автоформат* з меню *Формат* / вибрати відповідний формат таблиці. Якщо вибраний автоформат не забезпечує потрібного числового форматування, можна змінити числовий формат даних зведеної таблиці. Для цього слід виділити будь-яку комірку таблиці, на панелі інструментів *Зведені таблиці* натиснути кнопку *Зведена таблиця*, вибрати команду *Параметри поля*, у вікні діалогу натиснути кнопку Формат, вибрати потрібний формат і двічі натиснути кнопку ОК.

Зміна розміщення полів. Вигляд таблиці можна змінити, перегустивши кнопки полів в інші зони, а також змінивши порядок відображення поля в зоні стовпців або рядків зведеної таблиці. Для видалення поля із зведеної таблиці кнопку поля треба перемістити за межі макету. Оновлення зведеної таблиці.

Зведена таблиця динамічно пов’язана з базою даних, яку було використано під час її створення. Якщо значення в базі даних змінилися, треба виконати команду *Обновити дані* з пункту меню *Далі* або натиснути кнопку *Обновити дані* на панелі інструментів *Зведені таблиці*. Проте цей спосіб не підходить, якщо в базі даних з’являються нові рядки або стовпці. У цьому разі необхідно повернутися до майстра зведених таблиць і зазначити новий діапазон записів, що додаються до таблиці.

Перейменування полів і елементів. Імена полів і елементів, що призначені майстром зведених таблиць, можна змінити, відредагувавши їх заголовки у зведеній таблиці.

Сортування елементів. Майстер зведених таблиць впорядковує елементи за їх збільшенням. Якщо треба впорядкувати елементи за їх зменшенням або за значеннями поля даних, необхідно скористатись командою *Сортування* з пункту меню *Дані*.

Відображення або приховування елементів поля рядка чи стовпця зведеної таблиці. Можна приховати певні елементи полів

у зведеній таблиці. Для цього слід розгорнути список поля і встановити (зняти) прапорці для елементів, які потрібно відобразити (відповідно приховати).

Відображення або приховування елементів внутрішніх полів. Зведена таблиця для кожного елемента зовнішнього поля повторює елементи внутрішнього поля. Щоб приховати елементи внутрішнього поля, потрібно двічі клацнути на відповідному елементі зовнішнього поля. Щоб вивести на екран елементи внутрішнього поля, знову слід двічі клацнути на заголовку зовнішнього поля.

Відображення або приховування детальних даних. Можна побачити детальну інформацію, що використовується для обчислення значення поля даних, якщо двічі клацнути на цьому значенні. При цьому здійснюється копіювання детальних даних на новий аркуш книги.

Групування елементів внутрішнього поля. Майстер зведених таблиць автоматично групує елементи внутрішнього поля для кожного заголовка зовнішнього поля і в разі потреби створює проміжні підсумки для кожної групи елементів внутрішнього поля. Проте іноді зручніше групувати елементи в інший спосіб. Щоб створити групу, слід виділити її елементи і виконати команду *Дані/Структура/Згрупувати*. Створення діаграми на основі зведеної таблиці.

Діаграму на основі зведеної таблиці можна побудувати так само, як і для будь-яких інших типів даних робочого аркуша, скориставшись кнопкою *Майстер діаграм* на панелі інструментів або командою *Діаграма* з меню *Вставка*. Щоб досягти найкращих результатів у разі створення діаграми на основі зведеної таблиці, дотримуйтесь таких рекомендацій: видаліть з таблиці проміжні та загальні підсумки; прослідкуйте, щоб таблиця мала не більше двох полів у зоні стовпців і рядків, приховайте всі зайві елементи.

Функції Excel. Excel має велику кількість вмонтованих функцій робочих таблиць, які користувачі можуть застосовувати в своїх формулах. До них належать як досить поширені функції, такі як СУММ (SUM), СРЗНАЧ (AVERAGE) і КОРЕНЬ (SQRT), так і функції, призначені для виконання спеціальних обчислень,

наприклад статистичні функції або прикладні функції. Функції можуть спростити формули і зробити їх більш зрозумілими, крім того, вони здебільшого дають змогу робити обчислення, які без них виконати досить важко.

В Excel 2013 входить більше ніж 300 вмонтованих функцій.

Функції – це вмонтовані інструменти, що використовуються у формулах. Вони дають змогу:

- спрощувати формули;
- виконувати за формулами такі обчислення, які без них зробити неможливо;
- прискорювати виконання деяких завдань редагування.

Крім того, вони дозволяють «умовне» виконання формул, завдяки чому користувач може реалізовувати алгоритми прийняття рішень.

Аргументи функцій. У наведених прикладах користувач міг помітити, що в усіх функціях використовуються круглі дужки. Інформація, поміщена в дужках, називається *аргументом*. Функції відрізняються залежно від того, які аргументи вони використовують. Залежно від типу функції можуть бути:

- без аргументів;
- з одним аргументом;
- з невизначеною кількістю аргументів;
- з необов'язковим аргументом.

Прикладом функції, що не використовує аргумент, є функція **СЛЧИС** (**RAND**), яка генерує випадкове число в діапазоні від 0 до 1. Синтаксис цієї функції такий (хоч і немає аргументу, дужки обов'язкові):

=СЛЧИС() або **=RAND()**

Якщо у функції використовується більше ніж один аргумент, то кожен з них відокремлюється крапкою з комою (;).

Як аргументи функцій в Excel 2013 можна використовувати різноманітні вирази. У них можуть бути включені інші функції. Функції, які застосовуються у формулах як аргументи інших функцій, інколи називають *вкладеними*. Спочатку Excel обчислює значення найглибше вкладеного виразу.

Способи введення функцій. Є два шляхи введення функції в формулу: *ручний* або з допомогою *Мастера функций* Excel 2013.

Перший метод введення функцій у клітинки робочої таблиці полягає в тому, що необхідно просто ввести з клавіатури ім'я функції і список її аргументів. Часто цей спосіб є найефективнішим.

Мастер функций в Excel 2013 дає змогу ввести функцію та її аргументи в напівавтоматичному режимі. Використання *Мастера функций* забезпечує правильне синтаксичне написання функції і перелік усіх її аргументів у правильному порядку.

Мастер функций в Excel 2013 має два діалогових вікна, які допомагають користувачеві крок за кроком правильно вводити потрібну функцію. Користувач може звернутися до *Мастера функций* одним із зазначених методів:

Виконати команду *Вставка* та її опцію *Функция (Insert Function)*.

Клацнути мишею на кнопці виклику *Мастера функций* (*Function Wizard*), розміщеній на стандартній панелі інструментів.

Натиснути клавіші <Shift+F3>. Під час ручного введення або редагування формули клацнути мишею на кнопці виклику *Мастера функций* (*Function Wizard*), розміщеній в рядку формул.

У будь-якому випадку на екрані з'явиться перше з двох діалогових вікон *Мастера функций* (*Function Wizard*) (рис. 5.10)

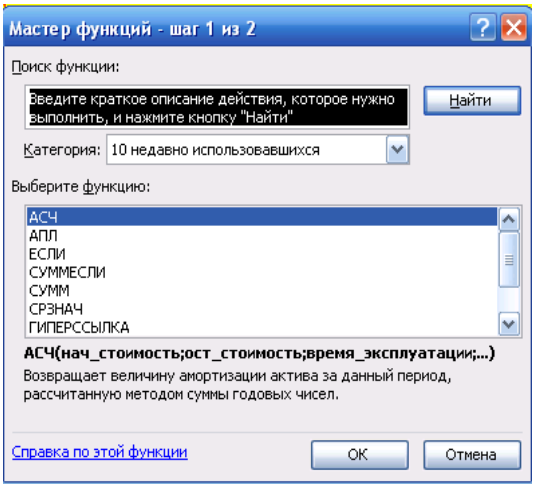


Рис. 5.10. Перше з двох діалогових вікон *Майстра функций*

У секції *Категория (Function Category)* цього діалогового вікна подано список 11 категорій функцій (категорій може бути і більше, якщо використовуються спеціальні функції користувача).

При виборі однієї з категорій у вікні списку *Функция (Function Name)* з'являється перелік її функцій.

У категорії *10 нещодавно використаних (Most Recently Used)* перелічених функцій, які використовувались зовсім недавно. У категорії *Полный алфавитный перечень (All)* включені всі функції з усіх категорій. Цей список доцільно використовувати, якщо відома назва функції, але невідомо, до якої категорії вона належить.

Зверніть увагу на те, що при виборі функції у вікні списку *Функция (Function Name)* внизу діалогового вікна з'являється список її аргументів разом з коротким описом.

Додаткова інформація про діалогове вікно *Мастер функций*.

Наведемо додаткові рекомендації, про які необхідно пам'ятати, працюючи з діалоговим вікном *Мастер функцій*.

У будь-який момент роботи з діалоговим вікном *Мастер функций*, якщо виникне потреба в допомозі, її можна отримати, клацнувши мишею на кнопці *Справка (Help)*.

Якщо користувач вводить нову формулу, *Мастер функций* автоматично починає її зі знака =. Якщо під час виклику *Мастера функций* активна клітинка не була пустою, то її вміст буде видалено.

Користувач може застосовувати *Мастер функций* для підстановки функції у вже створену формулу. Для цього під час редагування формули необхідно встановити курсор у те місце, куди потрібно вставити функцію. Потім для здійснення цієї підстановки необхідно викликати *Мастера функций*.

Перебуваючи в другому діалоговому вікні *Мастера функций*, можна клацнути мишею по кнопці *Назад (Back)*, щоб повернутися до першого діалогового вікна *Мастера функций*, де можна вибрати іншу функцію.

Якщо користувач передумав вводити функцію, то, перебуваючи в будь-якому вікні *Мастера функций*, необхідно клацнути мишею на кнопці *Отмена (Cancel)*.

Кількість полів редагування, які розміщені в другому вікні *Мастера функций*, визначається числом аргументів, які використовує функція. Якщо функція не використовує аргументи, поля редагування не виводяться на екран. Якщо функція використовує змінну

кількість аргументів, наприклад функція *СРЗНАЧ (AVERAGE)*, Excel додає нові поля редагування щоразу, коли вводиться черговий аргумент.

Поле, розміщене праворуч від зони редагування кожного аргумента, показує його поточне значення.

Деякі з функцій мають кілька форм, наприклад функція *ИНДЕКС (INDEX)*. Якщо користувач вибирає одну з таких функцій, Excel виводить на екран додаткове діалогове вікно, яке дає змогу вибрати потрібну форму функції. Таке діалогове вікно позначається як *Шаг 1а* (*Step 1a*).

Якщо користувач бажає тільки ввести список аргументів конкретної функції, потрібно набрати знак рівності **(=)** і назву цієї функції, а потім натиснути комбінацію клавіш <Ctrl+A>. У результаті на екрані з'явиться друге діалогове вікно *Мастера функций*.

Для швидкого пошуку функції в списку *Функция (Function Name)* активізуйте цю секцію першого діалогового вікна *Мастера функций* і введіть першу букву назви функції. Потім прокручуйте список до знаходження потрібної функції. Наприклад, якщо вибрана категорія *Полный алфавитный перечень (All*) і є потреба використати функцію *SIN*, клацніть мишею на вікні списку *Функция (Function Name)* і натисніть на клавіатурі букву <S>. Отже, буде знайдена перша з функцій, імена яких починаються з цієї букви. Це значно звузить коло пошуку.

Якщо користувач бажає використати функцію як аргумент (тобто як вкладену функцію), необхідно клацнути мишею на кнопці, розміщеній ліворуч від поля введення аргументів у другому діалоговому вікні *Мастера функций*. Excel виведе на екран перше діалогове вікно *Мастера функций* і дасть змогу вибрати другу функцію. В рядку заголовка цього вікна з'явиться слово *[Вложеный] [(Nested)]*, що буде нагадувати про те, що користувач створює вкладену функцію. За допомогою *Мастера функций* користувач може створити до семи рівнів вкладених функцій.

Функції роботи з базою даних. До цієї категорії в Excel віднесено 12 функцій. Вони використовуються під час роботи з табличними базами даних (також відомими як списки), що зберігаються в

аркуші робочої таблиці. Всі ці функції в англомовній версії Excel починаються з букви D і мають еквівалентні функції, не пов'язані з роботою над базами даних. Наприклад, функція БДСУММ (DSUM) є спеціальним варіантом функції СУММ (SUM), яка підраховує суму значень бази даних, що відібрані за заданим критерієм.

Таблична база даних – це прямокутний діапазон клітинок, у верхньому рядку якого розміщено назви полів бази даних. Кожний наступний рядок є окремим записом у базі даних.

Зауваження. Для використання функцій роботи з базою даних користувач повинен спочатку визначити спеціальний діапазон критеріїв відбору елементів робочої таблиці. Цей діапазон збігається з тим, що використовується в команді *Данные* та її опціях *Фильтр* і *Расширенный фильтр (Data ⇒ Filter ⇒ Advanced Filter).*

Приклад. На рис. 5.11 відображена таблична база даних зі створеним діапазоном критеріїв у клітинках *A1:B2*. Зверніть увагу, що діапазон критеріїв може бути розміщений у будь-якому місці робочої таблиці (вище або нижче від списку).

	A	B	C	D	E
1	Регіон	Сума продажів			
2	Схід			758	
3					
4	Регіон	Сума продажів			
5	Північ	234			
6	Південь	567			
7	Захід	987			
8	Схід	412			
9	Захід	345			
10	Північ	763			
11	Північ	861			
12	Схід	346			
13	Південь	849			
14	Захід	237			
15	Північ	345			
16					
17					
18					

Рис. 5.11. Таблична база даних з діапазоном критеріїв у клітинках A1:B2

Функція БДСУММ (DSUM) визначає загальний обсяг продажу у конкретному регіоні, який вибрано відповідно до заданого діапазону критеріїв. Наприклад, для підрахунку суми продажу по східному регіону в поле *Регіон* діапазону критеріїв (клітинка *A2*) введіть рядок =”Схід”. Після цього в будь-яку вільну клітинку аркуша робочої таблиці (наприклад, клітинка *D2*) запишіть формулу такого типу:

=БДСУММ(ДиапазонСписка;НазваниеПоля;Критерий)

або

=DSUM(ListRange;FieldName;Criteria)

Для нашого прикладу (рис. 5.12) ця формула матиме вигляд:

перший варіант:

=БДСУММ(A4:B15;B4;A1:B2)

або

=DSUM(A4:B15;B4;A1:B2)

другий варіант:

=БДСУММ(A:B;B4;A1:B2)

або

=DSUM(A:B;B4;A1:B2)

Формула повертає суму значень поля *Сума продажів* тільки тих записів, які задовольняють заданий у діапазоні *Критерий* (Схід). Можна змінити критерій, і формула підрахує новий результат.

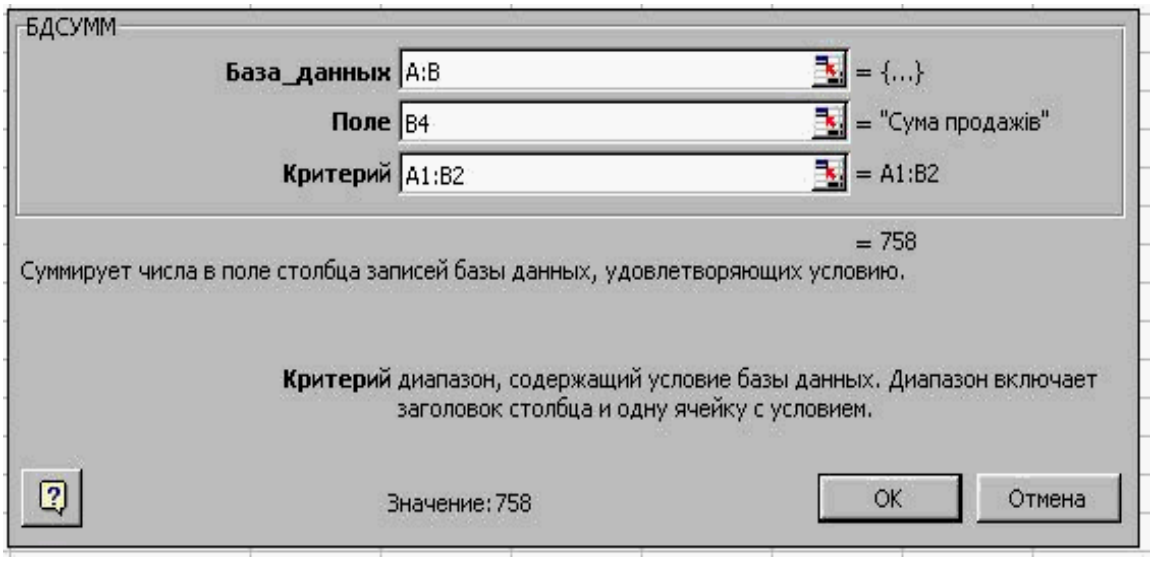


Рис. 5.12. Другий крок виконання функції *БДСУММ*

5.4. Моделювання економічних процесів засобами MS Excel 2013

Пошук рішення. Використання функції *Поиск решения* для вирішення задач виробництва. Розглянемо можливості функції *Поиск решения* на конкретному прикладі.

	A	B	C	D	E	F	G	H
1	Результати випуску продукції в поточному році							
2				=B4*C4		=D4/100*E4	=D4+F4	
3	Назва продукції	Випуск, шт.	Собівартість одиниці продукції, грн.	Витрати на виробництво, грн.	Норма прибутку, %	Прибуток на продукцію, грн.	Сума реалізації, грн	
4	Крісла	450	300,00	135000,00	5	6750,00	141750,00	
5	Стільці	560	200,00	112000,00	7	7840,00	119840,00	
6	Табуретки	200	100,00	20000,00	14	2800,00	22800,00	
7	Столи	180	250,00	45000,00	10	4500,00	49500,00	
8	Полиці	120	80,00	9600,00	9	864,00	10464,00	
9	Карнизи	100	75,00	7500,00	10	750,00	8250,00	
10	Разом	1610		329100,00		23504,00	352604,00	
11								

Рис. 5.13. Початкові дані задачі виробництва

Постановка завдання. За підсумками поточного року виробництво характеризується такими показниками. Припустимо, що на наступний рік є можливість використати у виробництві суму в 500 000 грн.

Тобто завдання полягає у тому, щоб розрахувати кількість виробів кожного виду так, щоб сумарні витрати на їх виробництво не перевищували 500 000 грн.

Створимо нескладну математичну модель, яка буде формалізувати нашу мету і ті обмеження, які маємо у разі її досягнення.

Позначимо кількість кожного виду продукції, що виробляємо, через змінну X.

Тобто:

X_1 – крісла, X_4 – столи,

X_2 – стільці, X_5 – полиці,

X_3 – табуретки, X_6 – карнизи.

Тоді функція цілі матиме такий вигляд:

$300X_1 + 200X_2 + 100X_3 + 250X_4 + 80X_5 + 75X_6 <= 500000$

На шляху досягнення нашої мети є деякі обмеження:

1) виробничі потужності не дозволяють випускати у сумі більше ніж 2 500 одиниць виробів,

тобто:

$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 <= 2500;$

2) виробляються готові вироби, а не комплектуючі до них,

тобто: X_i = ціле;

3) не маємо права порушувати асортимент виробів, тобто виробництво кожного виробу не може дорівнювати нулю:

$X_1, X_2, X_3, X_4, X_5, X_6 >= 1.$

Почнемо вирішувати цю задачу:

1. Для більшої зрозумілості того, що відбувається, зробимо копію нашої таблиці на цей саме аркуш Excel.

	A	B	C	D	E	F	G	H
1	**Результати випуску продукції в поточному році**							
2				**=B4*C4**		**=D4/100*E4**	**=D4+F4**	
3	Назва продукції	Випуск, шт.	Собівартість одиниці продукції, грн.	Витрати на виробництво, грн.	Норма прибутку, %	Прибуток на продукцію, грн.	Сума реалізації, грн	
4	Крісла	450	300,00	135000,00	5	6750,00	141750,00	
5	Стільці	560	200,00	112000,00	7	7840,00	119840,00	
6	Табуретки	200	100,00	20000,00	14	2800,00	22800,00	
7	Столи	180	250,00	45000,00	10	4500,00	49500,00	
8	Полиці	120	80,00	9600,00	9	864,00	10464,00	
9	Карнизи	100	75,00	7500,00	10	750,00	8250,00	
10	Разом	1610		329100,00		23504,00	352604,00	
11								
12								
13	**Планування випуску продукції**							
14								
15	Назва продукції	Випуск, шт.	Собівартість одиниці продукції, грн.	Витрати на виробництво, грн.	Норма прибутку, %	Прибуток на продукцію, грн.	Сума реалізації, грн	
16	Крісла	450	300,00	135000,00	5	6750,00	141750,00	
17	Стільці	560	200,00	112000,00	7	7840,00	119840,00	
18	Табуретки	200	100,00	20000,00	14	2800,00	22800,00	
19	Столи	180	250,00	45000,00	10	4500,00	49500,00	
20	Полиці	120	80,00	9600,00	9	864,00	10464,00	
21	Карнизи	100	75,00	7500,00	10	750,00	8250,00	
22								
23	Разом	1610		329100,00		23504,00	352604,00	
24		**<=2500**		Цільова функція				
25								

2. Починаємо працювати з функцією *Поиск решения*. Встановлюємо табличний курсор у клітинку *D23*, клацаємо мишою по команді *Сервис*, а потім по опції *Поиск решения* – з'явиться вікно *Поиск решения*.

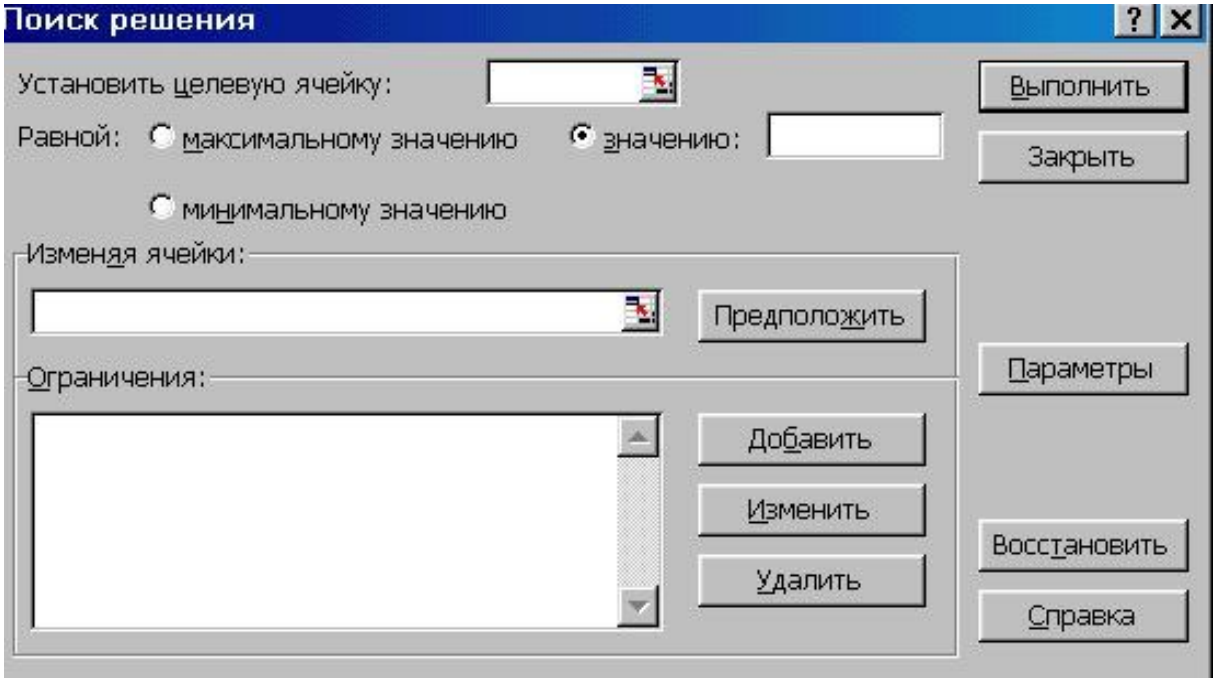


Рис. 5.14. Вікно функції *Пошук розв'язку*

3. Починаємо вводити у відповідні місця цього вікна цільову функцію та обмеження нашої задачі.

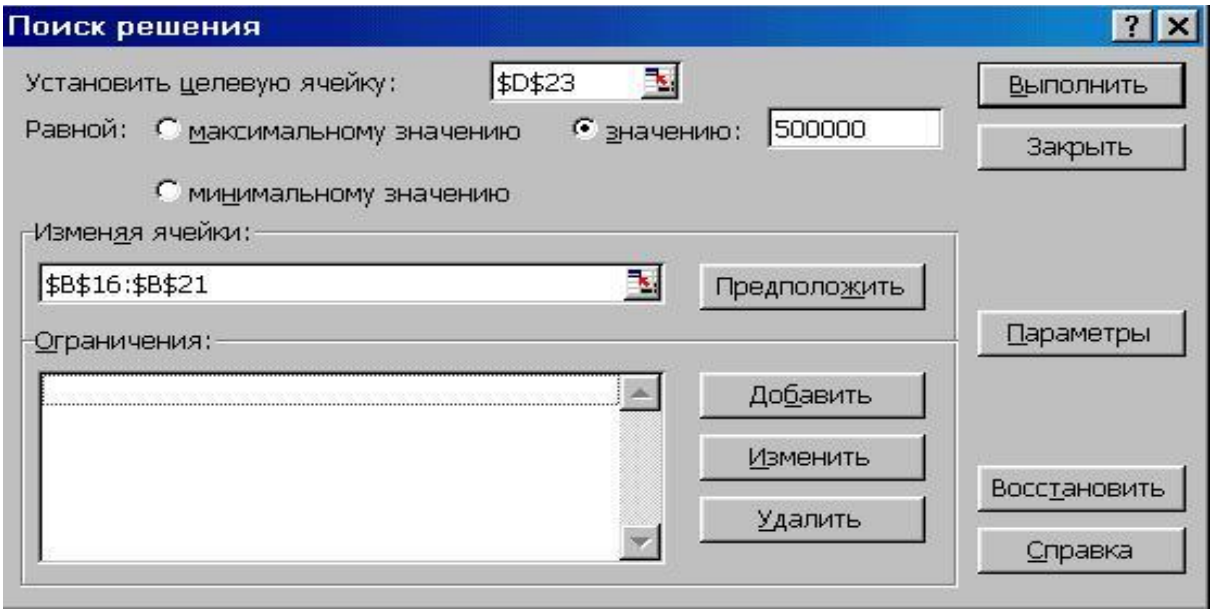


Рис. 5.15. Вибір цільової комірки та змінних комірок

На цьому рисунку можна побачити, що введено адресу клітинки, де буде вираховуватися значення цільової функції ($D$23), а також задано значення (500000), якого ця функція повинна досягнути; крім того, позначено діапазон клітинок, значення в яких можна змінювати під час пошуку рішення ($B$16:$B$21).

Далі будемо вводити обмеження нашої задачі:

1) обмеження на сумарне виробництво продукції (2500);

2) обмеження на асортимент (>=1);

3) обмеження на ціле значення змінних (=ціле).

Для введення кожного обмеження необхідно клацнути мишею по кнопці *Добавить* біля вікна *Ограничения*.

З'явиться вікно *Добавление ограничения*, в яке і будемо вводити перше обмеження. Після введення даних клацнути мишею по кнопці *ОК*.

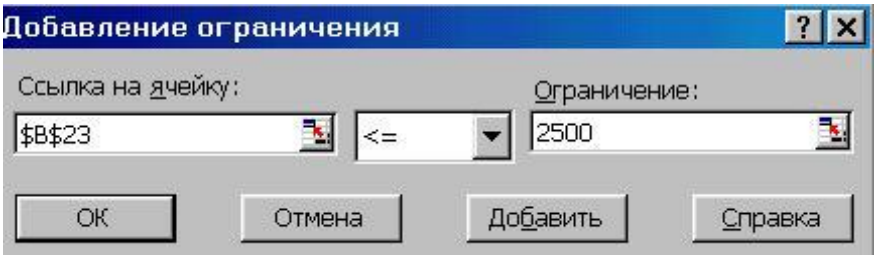


Рис. 5.16. Введення обмежень виробництва продукції

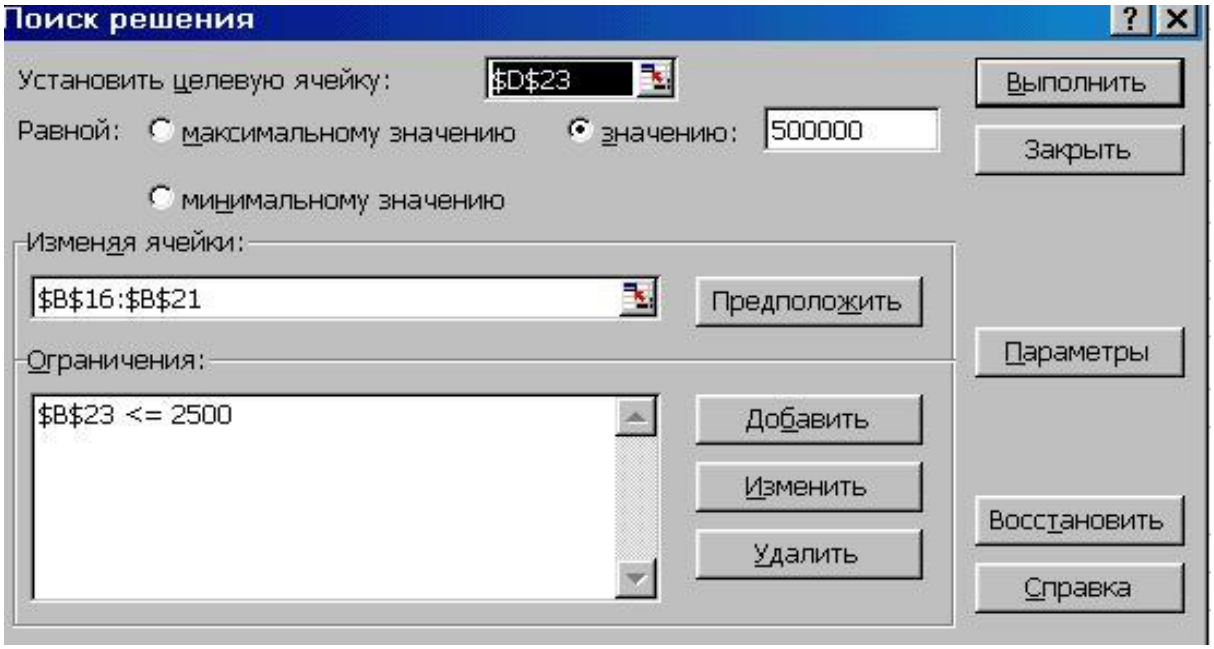


Рис. 5.17. Результат введення обмеження у вікні *Пошук розв'язку*

Далі варто знову клацнути мишею по кнопці *Добавить* і у вікні, що з'явиться, ввести наступне обмеження. Після введення даних клацнути мишею по кнопці *ОК*.

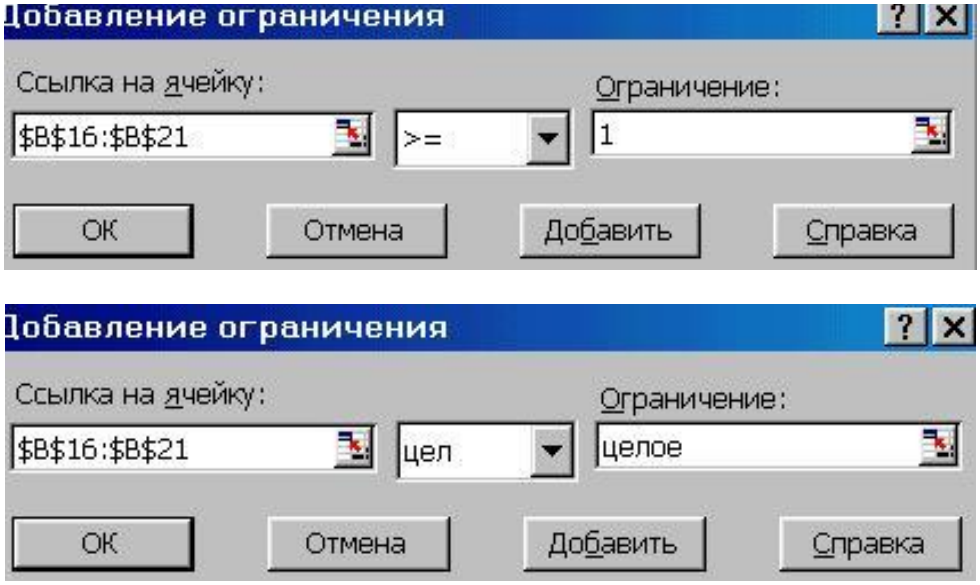


Рис. 5.18. Введення усіх обмежень виробництва

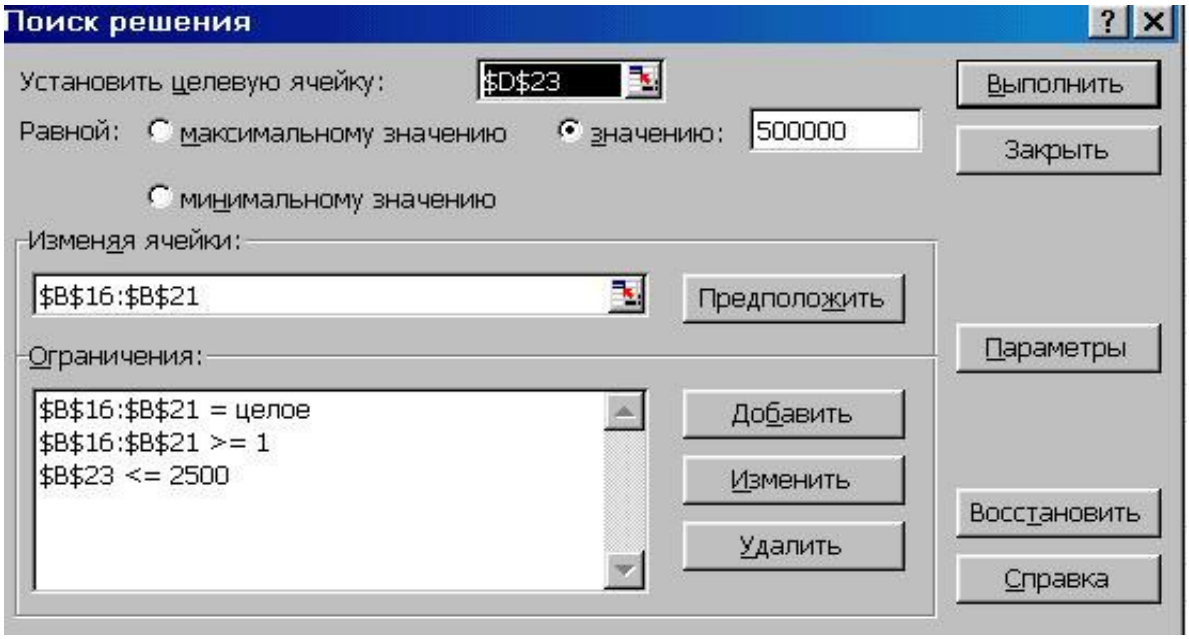


Рис. 5.19. Фінальний вигляд вікна функції *Пошук розв'яку*

Введення обмежень закінчено.

Для того щоб задача була вирішена, необхідно клацнути мишею по кнопці *Выполнить*.

Можливі два варіанти завершення *Поиск решения*: рішення буде знайдено і з'явиться вікно *Результаты поиска решения*, в якому обираємо опцію *Сохранить найденое решение* та обираємо у вікні *Тип отчета* опцію *Результаты*, після чого клацнути мишею по кнопці *ОК*.

Якщо рішення не буде знайдено і тоді у вікні *Результаты поиска решения* буде написано, що рішення не знайдено (у цьому випадку необхідно перевірити правильність введення цільової функції та обмежень або, якщо рішення не знайдено під час правильного введення даних, треба створити іншу математичну модель і розв'язувати задачу за новими даними).

Можна побачити, що у таблиці, де шукали рішення, з'явились значення, які підтверджують, що задача вирішена.

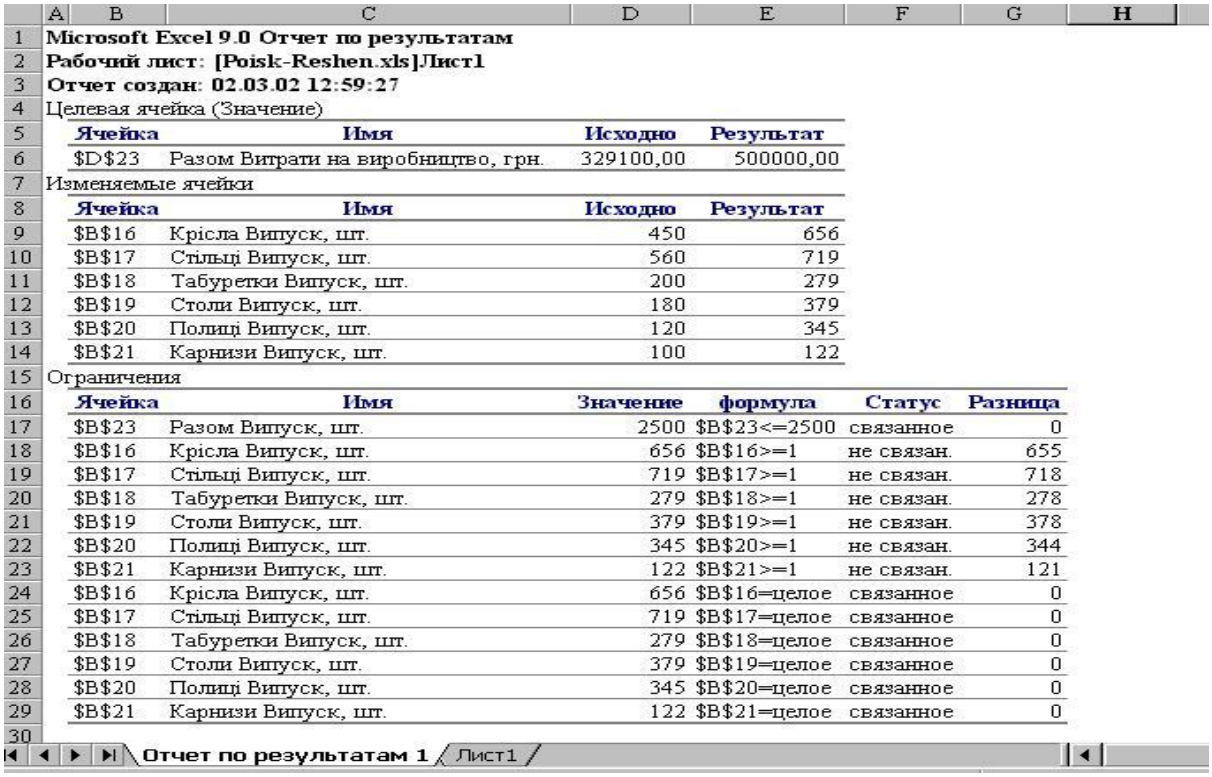
	A	B	C	D	E	F	G	H
1	Результати випуску продукції в поточному році							
2				=B4*C4		=D4/100*E4	=D4+F4	
3	Назва продукції	Випуск, шт.	Собівартість одиниці продукції, грн.	Витрати на виробництво, грн.	Норма прибутку, %	Прибуток на продукцію, грн.	Сума реалізації, грн	
4	Крісла	450	300,00	135000,00	5	6750,00	141750,00	
5	Стільці	560	200,00	112000,00	7	7840,00	119840,00	
6	Табуретки	200	100,00	20000,00	14	2800,00	22800,00	
7	Столи	180	250,00	45000,00	10	4500,00	49500,00	
8	Полиці	120	80,00	9600,00	9	864,00	10464,00	
9	Карнизи	100	75,00	7500,00	10	750,00	8250,00	
10	Разом	1610		329100,00		23504,00	352604,00	
11								
12								
13	Планування випуску продукції							
14								
15	Назва продукції	Випуск, шт.	Собівартість одиниці продукції, грн.	Витрати на виробництво, грн.	Норма прибутку, %	Прибуток на продукцію, грн.	Сума реалізації, грн	
16	Крісла	656	300,00	196800,00	5	9840,00	206640,00	
17	Стільці	719	200,00	143800,00	7	10066,00	153866,00	
18	Табуретки	279	100,00	27900,00	14	3906,00	31806,00	
19	Столи	379	250,00	94750,00	10	9475,00	104225,00	
20	Полиці	345	80,00	27600,00	9	2484,00	30084,00	
21	Карнизи	122	75,00	9150,00	10	915,00	10065,00	
22								
23	Разом	2500		500000,00		36686,00	536686,00	
24		<=2500		Цільова функція				
25								

Отчет по результатам 1 | Лист1

Готово

Рис. 5.20. Розв'язок задачі виробництва

Крім того, з метою можливого подальшого аналізу отриманого рішення нам надано окремий аркуш з назвою *Отчет по результатам 1*, зміст якого такий:



Microsoft Excel 9.0 Отчет по результатам
Рабочий лист: [Poisk-Reshen.xls]Лист1
Отчет создан: 02.03.02 12:59:27

Целевая ячейка (Значение)

Ячейка	Имя	Исходно	Результат
$D$23	Разом Витрати на виробництво, грн.	329100,00	500000,00

Изменяемые ячейки

Ячейка	Имя	Исходно	Результат
$B$16	Крісла Випуск, шт.	450	656
$B$17	Стільці Випуск, шт.	560	719
$B$18	Табуретки Випуск, шт.	200	279
$B$19	Столи Випуск, шт.	180	379
$B$20	Полиці Випуск, шт.	120	345
$B$21	Карнизи Випуск, шт.	100	122

Ограничения

Ячейка	Имя	Значение	формула	Статус	Разница
$B$23	Разом Випуск, шт.	2500	$B$23<=2500	связанное	0
$B$16	Крісла Випуск, шт.	656	$B$16>=1	не связан.	655
$B$17	Стільці Випуск, шт.	719	$B$17>=1	не связан.	718
$B$18	Табуретки Випуск, шт.	279	$B$18>=1	не связан.	278
$B$19	Столи Випуск, шт.	379	$B$19>=1	не связан.	378
$B$20	Полиці Випуск, шт.	345	$B$20>=1	не связан.	344
$B$21	Карнизи Випуск, шт.	122	$B$21>=1	не связан.	121
$B$16	Крісла Випуск, шт.	656	$B$16=целое	связанное	0
$B$17	Стільці Випуск, шт.	719	$B$17=целое	связанное	0
$B$18	Табуретки Випуск, шт.	279	$B$18=целое	связанное	0
$B$19	Столи Випуск, шт.	379	$B$19=целое	связанное	0
$B$20	Полиці Випуск, шт.	345	$B$20=целое	связанное	0
$B$21	Карнизи Випуск, шт.	122	$B$21=целое	связанное	0

Отчет по результатам 1 / Лист1

Рис. 5.21. Звіт за результатами розв'язку задачі засобами функції *Пошук розв'язку*

Використання функції Сценарии для вирішення задач виробництва

Можливості функції *Сценарии* розглянемо на прикладі, що був у *Пошуку розв'язку*.

Тобто за підсумками роботи функції *Поиск решения* розглянемо кілька варіантів зміни фінансових показників підприємства при певних змінах норми прибутку.

Сценарії зміни прибутку при варьюванні норми прибутку

Назва продукції	Випуск, шт.	Собівартість одиниці продукції, грн.	Витрати на виробництво, грн.	Норма прибутку, %	Прибуток на продукцію, грн.	Сума реалізації, грн
Крісла	656	300,00	196800,00	5	9840,00	206640,00
Стільці	719	200,00	143800,00	7	10066,00	153866,00
Табуретки	279	100,00	27900,00	14	3906,00	31806,00
Столи	379	250,00	94750,00	10	9475,00	104225,00
Полиці	345	80,00	27600,00	9	2484,00	30084,00
Карнизи	122	75,00	9150,00	10	915,00	10065,00
Разом	2500		500000.00		36686.00	536686.00

Рис. 5.22. Задача оптимізації виробництва з метою отримання аналізу прибутковості

Постановка завдання

Припустимо, що на наступний рік є можливість змінювати показники прибутковості кожного з виробів.

Тобто завдання полягає у тому, щоб розрахувати кілька варіантів результатів загальної прибутковості нашого виробництва.

Починаємо працювати з функцією *Сценарии*: клацаємо мишею по команді *Сервис*, а потім по опції *Сценарии* – з'явиться вікно *Диспетчер сценариев*.

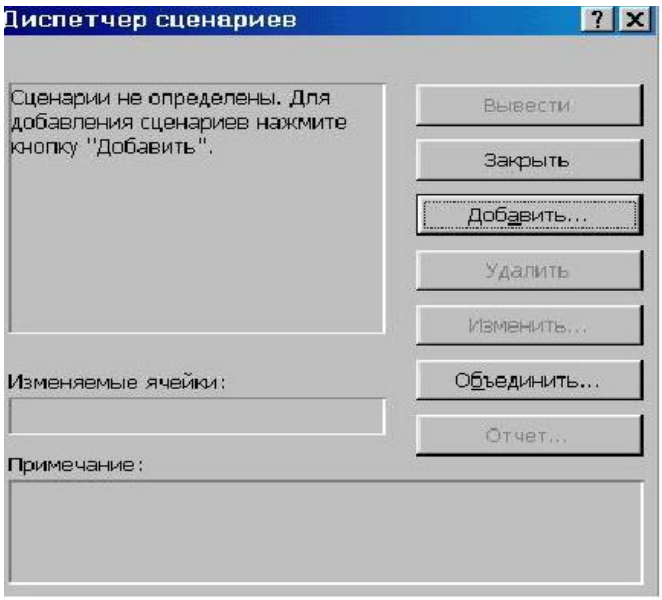


Рис. 5.23. Диспетчер сценаріїв

Почнемо формувати різні сценарії зміни норми прибутку.

Перший сценарій створимо на підставі даних, які є зараз у таблиці, і назвемо цей сценарій *Попередній*.

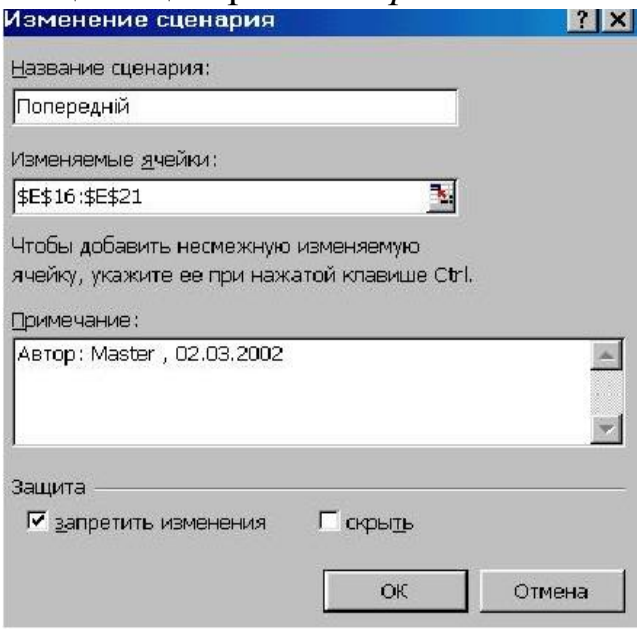


Рис. 5.24. Попередній сценарій

У вікні *Изменение сценария* вводимо назву і діапазон клітинок, значення яких будуть брати участь у цьому варіанті сценарію. Після чого клацнути мишею по кнопці *ОК*.

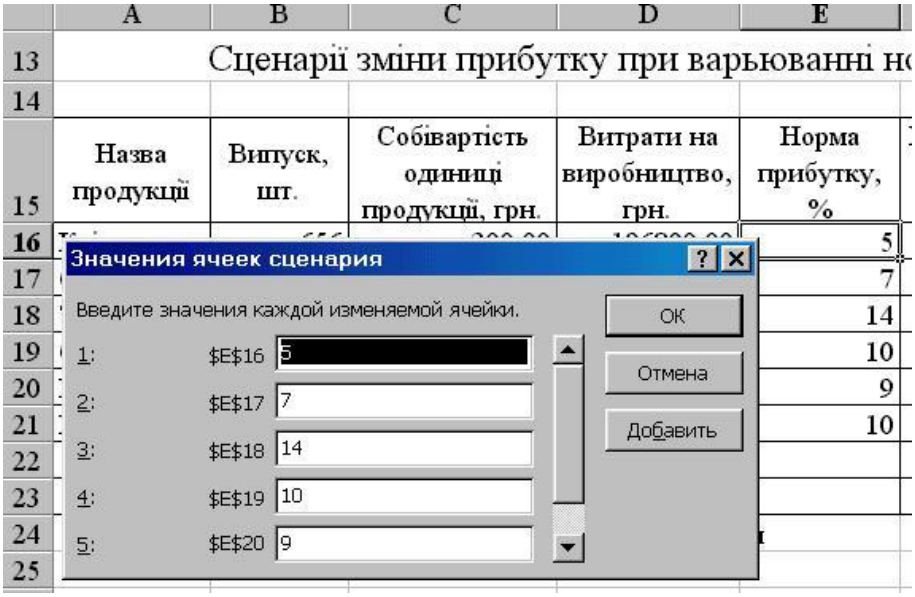


Рис. 5.25. Робота з попереднім сценарієм

У вікні *Значения ячеек сценария* залишаємо ті дані, які зараз є в цих клітинках, тому що цей сценарій назвали *Попередній*. І клацаємо мишею по кнопці *ОК*.

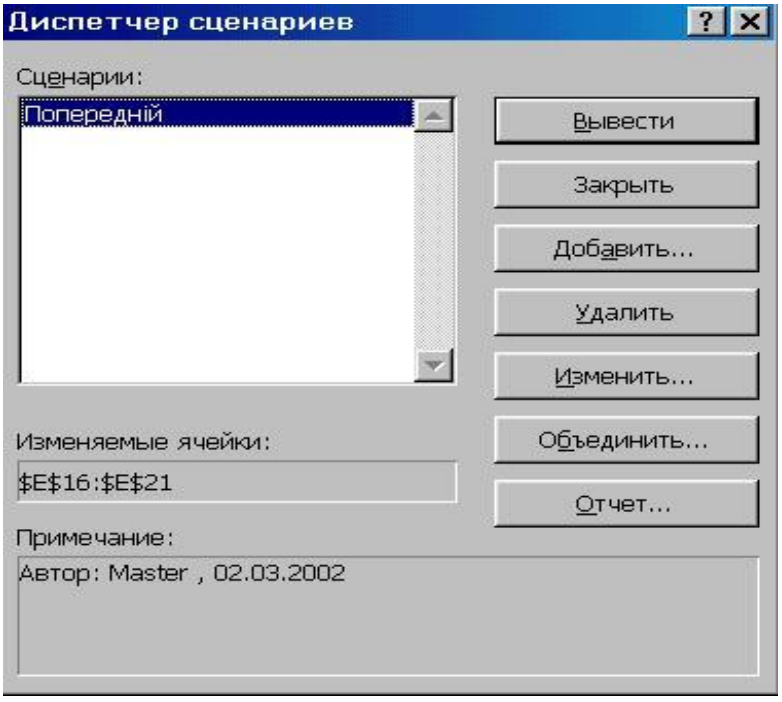


Рис. 5.26. Формування списку сценаріїв

У вікні *Диспетчер сценариев* з'явився сценарій з назвою *Попередній*. Клацаємо мишею по кнопці *Добавить* – для введення наступного сценарію з іменем *Перший варіант*.

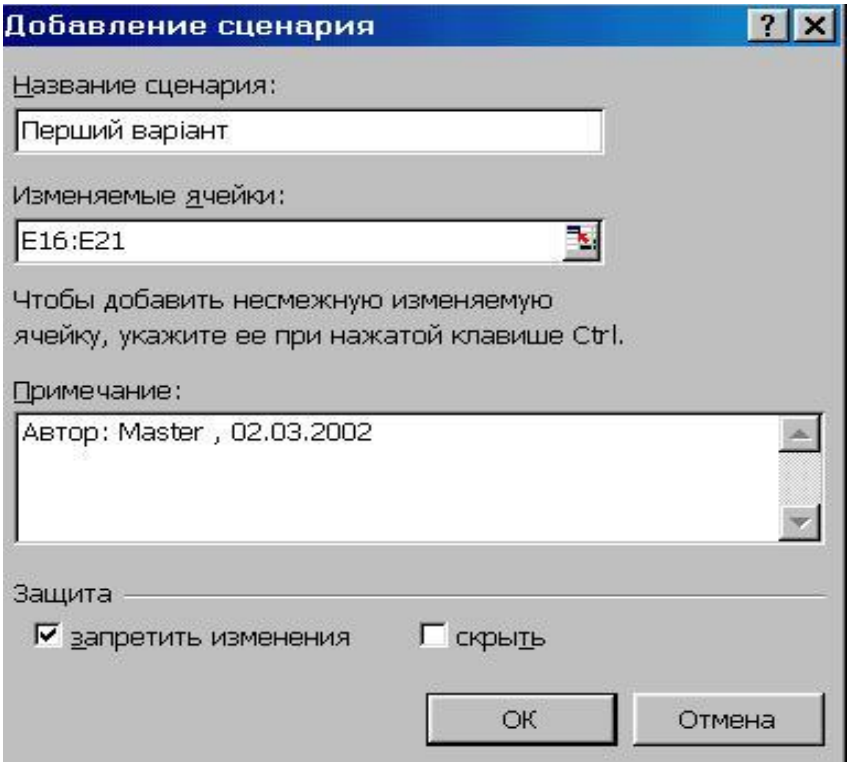


Рис. 5.27. Перший сценарій

Клацаємо по кнопці ОК і вводимо нові значення норми прибутку для кожного виду продукції.

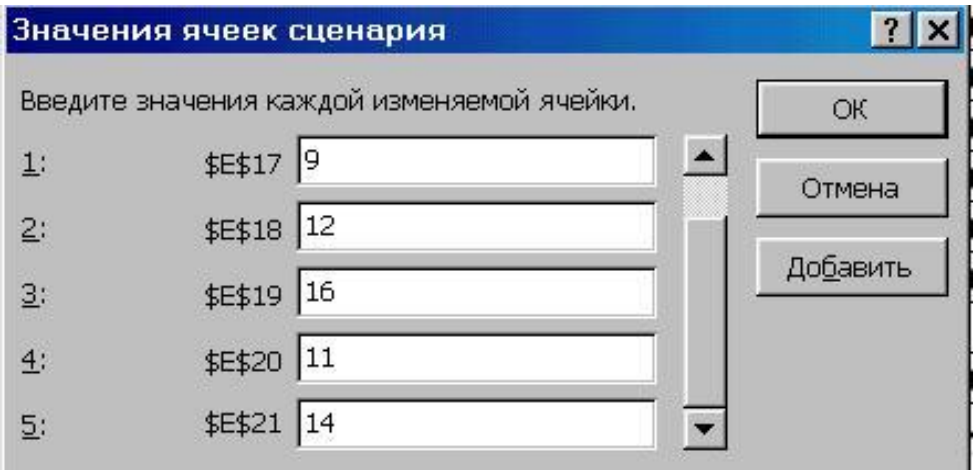


Рис. 5.28. Значення для першого сценарію

У вікні *Диспетчер сценариев* з'являється сценарій з назвою *Перший варіант*.

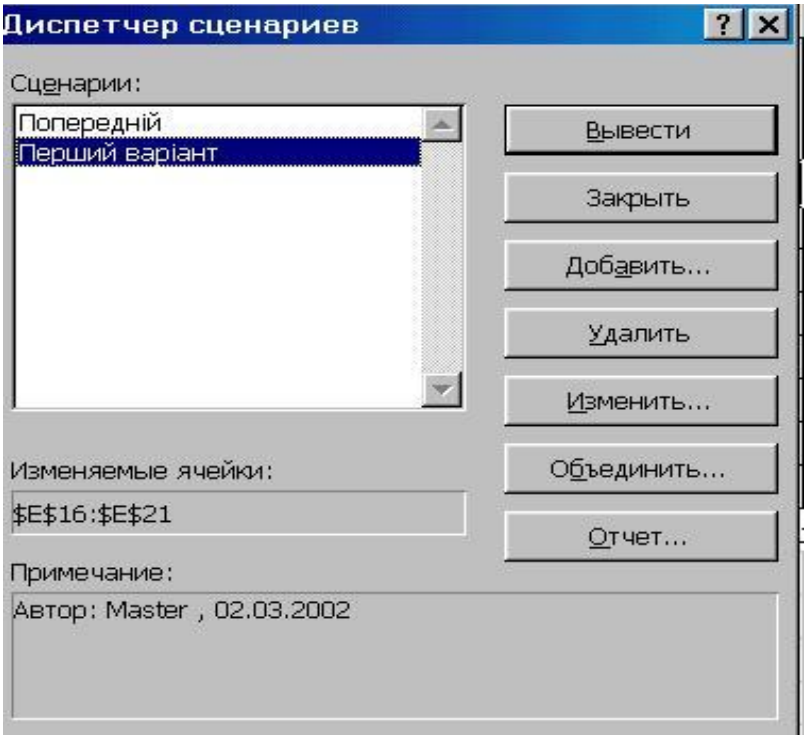


Рис. 5.29. Список сценаріїв

Так само можна додавати інші сценарії.

Для перегляду результатів розрахунків за кожним сценарієм необхідно:

	A	B	C	D	E	F	G
13	Сценарії зміни прибутку при варьюванні норми прибутку						
14							
15	Назва продукції	Випуск, шт.	Собівартість одиниці продукції, грн.	Витрати на виробництво, грн.	Норма прибутку, %	Прибуток на продукцію, грн.	Сума реалізації, грн
16	Крісла				7	13776,00	210576,00
17	Стільці				9	12942,00	156742,00
18	Табуретки				12	3348,00	31248,00
19	Столи				16	15160,00	109910,00
20	Полиці				11	3036,00	30636,00
21	Карнизи				14	1281,00	10431,00
22							
23	Разо					49543,00	549543,00
24							
25							
26							
27							
28							
29							
30							
31							
32							

Рис. 5.30. Виведення результату вибраного сценарію у таблицю

1) у вікні *Диспетчер сценариев* у розділі *Сценарии* потрібно позначити мишею назву того сценарію, результати якого нас цікавлять. Наприклад, сценарій *Перший варіант* – та кликнути по кнопці *Вывести*.

Можна побачити, що в колонці таблиці *Норма прибутку, %* числа змінилися згідно із значенням цього сценарію та по таблиці зроблено автоматичний перерахунок значень у тих клітинках, де у формулах були задіяні адреси клітинок колонки *Норма прибутку, %*.

Крім того, можемо отримати звіт про всі сценарії, які були розраховані. Для цього у вікні *Диспетчер сценариев* необхідно клацнути мишею по кнопці *Отчет*.

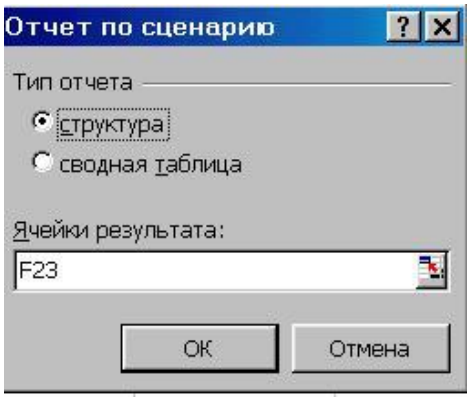


Рис. 5.31. Вибір типу звіту за виконаним сценарієм

У вікні *Отчет по сценарию* обрати форму звіту та клацнути по кнопці *ОК*.

	A	B	C	D	E	F	G	H
1								
2		**Структура сценария**						
3				Текущие значения:	Попередній	Перший варіант		
5		**Изменяемые:**						
6			**$E$16**	7	5	7		
7			**$E$17**	9	7	9		
8			**$E$18**	12	14	12		
9			**$E$19**	16	10	16		
10			**$E$20**	11	9	11		
11			**$E$21**	14	10	14		
12		**Результат:**						
13			**$F$23**	49543,00	36686,00	49543,00		
14		Примечания: столбец "Текущие значения" представляет значения изменяемых ячеек в						
15		момент создания Итогового отчета по Сценарию. Изменяемые ячейки для каждого						
16		сценария выделены серым цветом.						
17								
18								
19								

Отчет по результатам 1 | Структура сценария | Лист1

Рис. 5.32. Звіт за результатами сценарію

Використання функції Підбір параметра.

Можливості функції *Підбір параметра* розглянемо на прикладі фінансових розрахунків під час відкриття вкладів у банках.

Наприклад, у якомусь банку відкрито рахунок на таких умовах:

	A	B	C	D	E	F
1	Розрахунки по вкладах					
2						
3	№ рахунку	Сума вкладу	Термін вкладання, років	Річна % ставка	Накопичена сума	
4	12134	500000	25	5%	1693177,47	
5					=B4*(1+D4)^C4	

Рис. 5.33. Початкові дані фінансових розрахунків

Розглянемо деякі можливості функції *Підбір параметра*:

Задача 1. Клієнт банку хоче знати, яку суму грошей необхідно внести на рахунок, щоб за умови, зазначеної вище, отримати накопичену суму в розмірі 500 000 грн.

Починаємо працювати з функцією *Підбір параметра:* клацаємо мишею по команді *Сервис*, а потім по опції *Подбор параметра* – з'явиться вікно *Подбор параметра*.

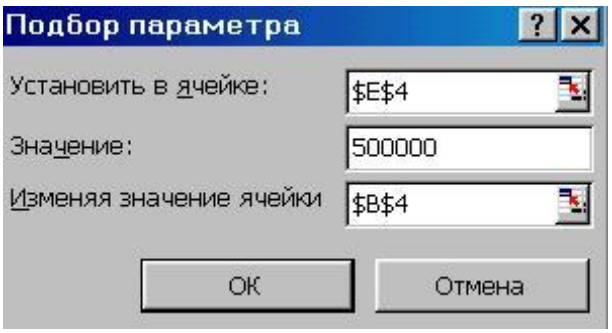


Рис. 5.34. Початкові дані фінансових розрахунків задачі 1

У цьому вікні вводимо потрібні дані:

- розділ *Установить в ячейке* адресу клітинки, в якій буде задано те, чого бажаємо;
- розділ *Значение* заповняється конкретним значенням, що відтворює наше бажання (500000);
- розділ *Изменяя значение ячейки* адреси клітинки, в якій буде підбиратися значення, що дає можливість задовольнити наше бажання за наявних умов.

Після цього клацнути по кнопці *ОК*. Для закриття вікна *Результат подбора параметра* необхідно клацнути мишею по кнопці ОК.

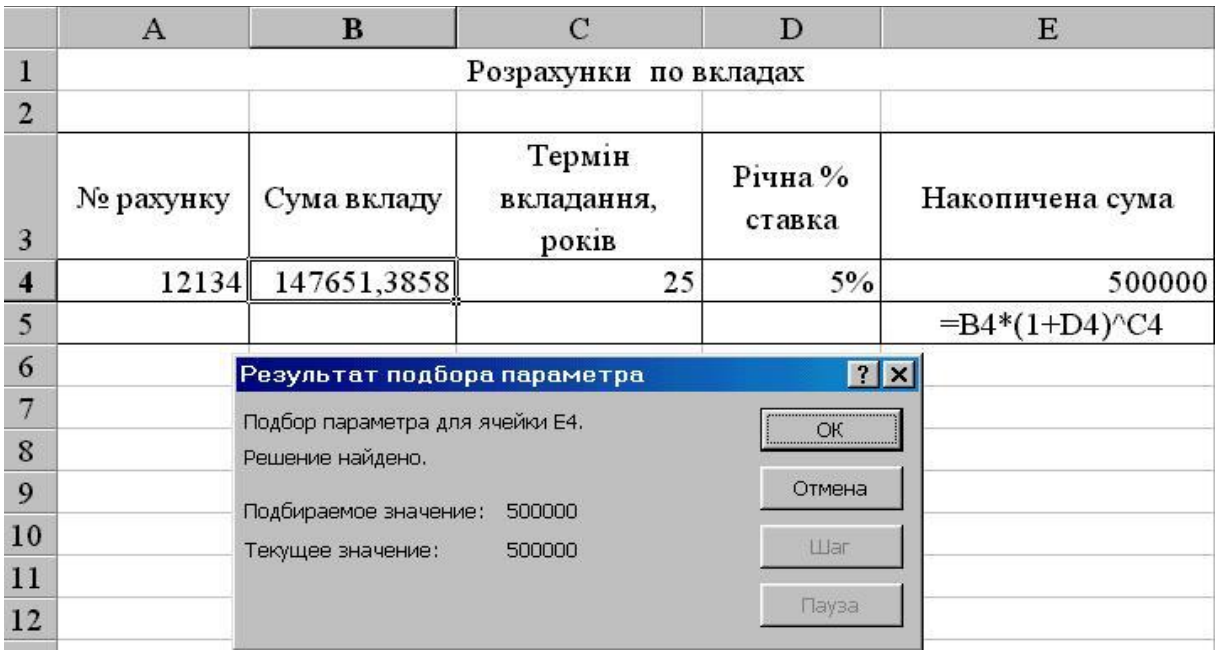


Рис. 5.35. Результат підбору параметра задачі 1

Задача 2. Клієнт банку хоче знати, скільки років необхідно мати рахунок у банку, щоб при початковому внеску 500 000 грн отримати накопичену суму в розмірі 3 500 000 грн.

У цьому вікні вводимо потрібні дані:

- розділ *Установить в ячейке* адресу клітинки, у якій буде задано те, чого бажаємо;
- розділ *Значение* заповняється конкретним значенням, що відтворює наше бажання (500000);
- розділ *Изменяя значение ячейки* адресу клітинки ($C$4), в якій буде підбиратися значення, яке дає можливість задовольнити наше бажання за наявних умов.

Після цього клацнути по кнопці *ОК*.



Рис. 5.36. Початкові дані фінансових розрахунків задачі 2

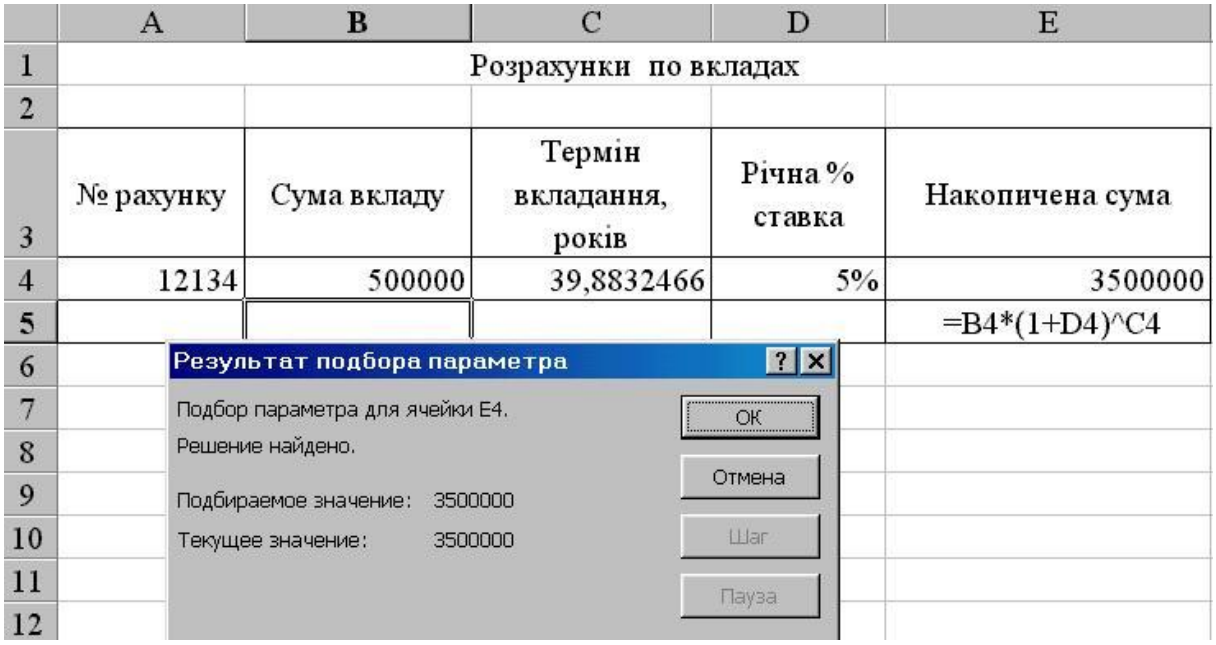


Рис. 5.37. Результат підбору параметра задачі 2

3. Клієнт банку хоче знати, яку річну % ставку необхідно мати, щоб при початковому внеску 500 000 грн, терміні вкладання 25 років отримати накопичену суму в розмірі 3 450 000 грн.

У цьому вікні вводимо потрібні дані:

- розділ *Установить в ячейке* – адресу клітинки, у якій буде задано те, чого бажаємо;
- розділ *Значение* заповнюється конкретним значенням, що відтворює наше бажання (3450000);
- розділ *Изменяя значение ячейки* – адресу клітинки ($D$4), в якій буде підбиратися значення, яке дає можливість задовольнити наше бажання за наявних умов.

Після цього клацнути по кнопці *ОК*.

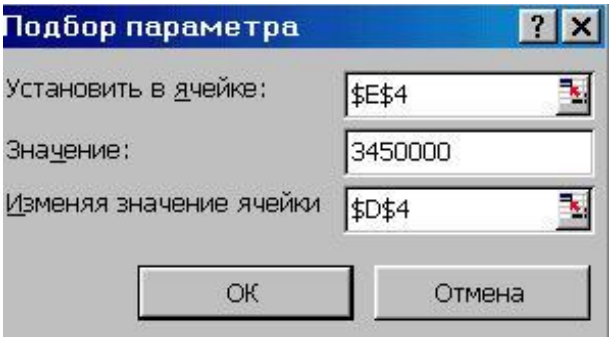


Рис. 5.38. Початкові дані фінансових розрахунків задачі 3

Запитання для самоконтролю

1. Що таке фільтрація даних, які різновиди фільтрів для виконання фільтрації даних існують у базі даних MS Excel 2013?
2. Як створити автофільтр?
3. Як створити розширений фільтр?
4. Яке призначення мають проміжні підсумки?
5. Яка технологія створення проміжних підсумків?
6. Яке призначення мають зведені таблиці і зведені діаграми?
7. Яка технологія створення зведених таблиць і зведених діаграм?
8. Яке призначення функції *Підбір параметру*?
9. Яке призначення функції *Пошук розв'язку*?
10. Для чого потрібні сценарії?

ЛЕКЦІЯ 6
РОБОТА З ТЕКСТАМИ У СЕРЕДОВИЩІ MS WORD 2013

6.1. Електронний документ. Електронний документообіг

На практиці поняття «документ» використовують як у широкому, так і вузькому значенні:

– у широкому значенні документ – це засіб фіксації (закріплення) інформації різними способами на спеціальному матеріалі;

– у вузькому значенні документ – це будь-яке офіційне посвідчення, доказ достовірності дії, предмета, поняття.

Документ повинен відповідати таким конкретним вимогам:

- видається повноважним органом або особою відповідно до їхньої компетенції;
- документ не повинен суперечити чинному законодавству і директивним указівкам керівних органів;
- повинен бути достовірним і відповідати завданням конкретного керівництва, тобто базуватися на фактах і містити конкретні пропозиції або вказівки;
- документ повинен відповідати своєму призначенню й укладатися за встановленою формою;
- документ має бути бездоганно відредагований та оформлений (порушенням в оформленні вважаються неохайність, нерозбірливість тексту, використання прозорого паперу тощо).

Під час укладання документів необхідно дотримуватись основних принципів стилю і мови ділових паперів:

- дотримання норм літературної мови взагалі і норм стилю зокрема;
- уніфікація ділової мови;
- стислість текстів через використання простих речень, введення мовних кліше, зворотів, абревіатур.

Класифікація документів – це поділ їх на класи за найбільш загальними ознаками схожості та відмінності. Мета класифікації

та відповідальності виконавців. У поточній роботі класифікацію документів здійснюють на етапі групування їх справи.

Документи класифікують за такими ознаками: найменування; спосіб фіксації інформації; зміст; вид; складність; місце складання; термін виконання; походження; гласність; стадія виготовлення; термін зберігання та ін.

Стандартність документів найбільшою мірою виявляється у наявності спільних складників – реквізитів (від лат. requisitum – необхідне, потрібне) – вихідних, обов'язкових даних документа, відсутність яких позбавляє діловий папір юридичної сили. Кожний документ містить тільки йому властивий набір реквізитів та порядок їх розташування.

Формуляр документа – сукупність розміщених у встановленій послідовності реквізитів документа.

Формуляр-зразок – це модель побудови форми документа, яка встановлює галузь використання, формати, розміри полів, вимоги до побудови конструкційної сітки та основні реквізити (ДСТУ 2732-04).

Бланк офіційного документа – це стандартний аркуш паперу з відтвореною на ньому постійною інформацією документа й місцем, залишеним для змінної.

Логічні вимоги до офіційного документа:

- використання засобів логічної оцінки матеріалу для зв'язності окремих частин, зазначення черговості здійснення (насамперед, одночасно, після тощо);
- врахування таких елементів, як актуальність фактів та їх своєчасність;
- послідовність викладення тексту та його частин.

Технічні вимоги до офіційного документа:

- для укладання будь-якого документа існує відповідний формат паперу: загальновизнаний формат А4 (1,5 інтервали між рядками), А5 (1 інтервал між рядками);
- колір паперу – білий;
- дотримання полів документа (наприклад, ліве – 30 мм, праве – 10 мм, верхнє – 20 мм, нижнє – 20 мм);
- дотримання державного стандарту.

Поняття «електронний документ». Електронний документ – це документ, інформація в якому зафіксована у вигляді електронних даних, включаючи обов'язкові реквізити документа. Електронний документ може бути створений, переданий, збережений і перетворений електронними засобами у візуальну форму.

Електронний документ характеризується змістом, контекстом та структурою.

Зміст електронного документа – це текстові та графічні частини, що становлять документ.

Контекст електронного документа – інформація про зв'язки задокументованої інформації з фізичними або юридичними особами та іншими документами.

Електронний документ має зовнішню та внутрішню структуру:

- внутрішня структура – це структура змістовної частини документа;
- зовнішня структура – це структура середовища, в якому існує електронний документ (носій інформації, формат файлу тощо).

Електронний документ визначається своїми метаданими. Метадані — це дані, що пов'язані та стосуються електронного документа, містять інформацію про його соціальну визначеність та розуміння змісту, контексту, структури. Метадані електронного документа мають елементи юридичних, ділових, організаційних, процедурних доказів цілісності електронного документа та його автентичності для авторів, користувачів та дослідників. Метадані є обов'язковим елементом процесу зберігання електронного документа.

Обов'язковим реквізитом електронного документа, який використовують для ідентифікації автора та/або підписувача електронного документа іншими суб'єктами електронного документообігу, є кваліфікаційний електронний підпис (КЕП). Накладанням електронного підпису завершується утворення електронного документа.

Оригіналом електронного документа вважається електронний примірник документа з обов'язковими реквізитами, у тому числі з електронним цифровим підписом автора. У разі

надсилання електронного документа кількома адресатам або його зберігання на кількох електронних носіях інформації кожний з електронних примірників вважається оригіналом електронного документа. Якщо автором створюється ідентичний за документарною інформацією та реквізитами електронний документ та документ на папері, кожний з документів є оригіналом і має однакову юридичну силу.

Розглянемо правовий статус електронного документа та його копії. В Україні використання електронних документів та цифрових підписів регулюється на підставі Закону України «Про електронні документи та електронний документообіг» та Закону України «Про електронний цифровий підпис», «Про електронні довірчі послуги».

Юридична сила електронного документа не може бути заперечена виключно через те, що він має електронну форму. Допустимість електронного документа як доказу не може заперечуватися виключно на підставі того, що він має електронну форму.

Електронний документ не може бути використаний як оригінал: свідоцтва про право на спадщину; документа, який, відповідно до законодавства, може бути створений лише в одному оригінальному примірнику, крім випадків існування централізованого сховища оригіналів електронних документів; в інших випадках, передбачених законом.

Отримати кваліфікаційний електронний підпис (електронний цифровий підпис) можна в одному з акредитованих центрів сертифікації ключів (далі – АЦСК). Для оформлення знадобиться паспорт громадянина України та довідка про присвоєння ідентифікаційного номера.

Одним із таких акредитованих центрів сертифікації ключів є Приватбанк. Якщо ви є клієнтом Приватбанку, для отримання електронного підпису скористайтесь розділом системи Приват24 «Усі послуги»/«Бізнес»/«Завантажити сертифікат».

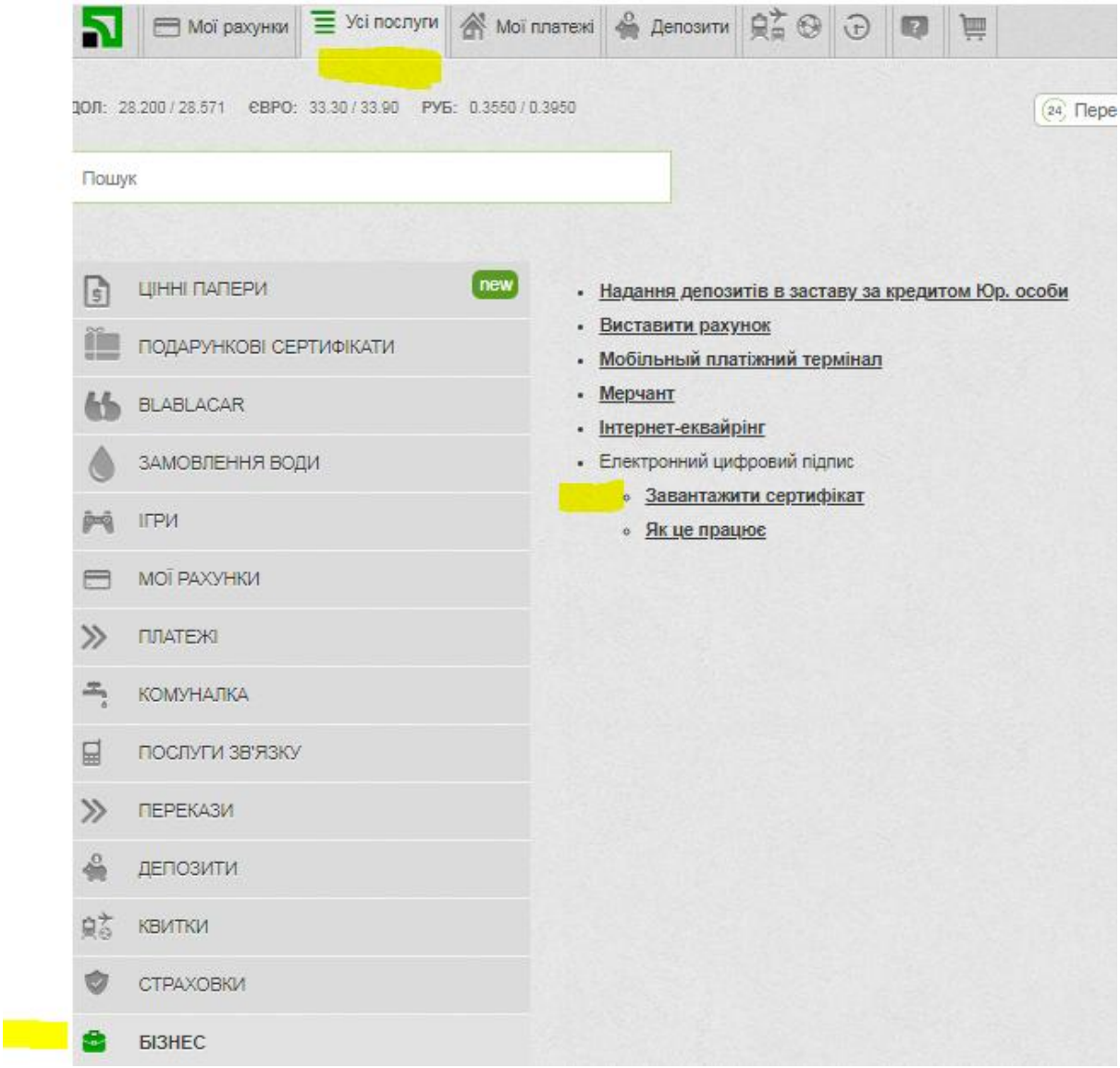


Рис. 6.1. Розділ Бізнес-системи Privat24 для отримання цифрового підпису

Чинною нормативно-правовою базою України щодо організації електронного документообігу є: Закон України «Про електронні документи та електронний документообіг»; Постанова Кабінету Міністрів України «Про затвердження Типового порядку здійснення електронного документообігу в органах виконавчої влади»; наказ Міністерства освіти і науки, молоді та спорту «Про вимоги до форматів даних електронного документообігу в органах державної влади».

Відповідно до ст. 9 Закону України «Про електронні документи та електронний документообіг» електронний документообіг − сукупність процесів створення, оброблення, відправлення,

передавання, одержання, зберігання, використання та знищення електронного документа, які виконуються із застосуванням перевірки цілісності та в разі необхідності з підтвердженням факту одержання таких документів.

Основні принципи електронного документообігу:

- однократна реєстрація документа;
- паралельне виконання різних операцій з метою скорочення часу руху документів і підвищення оперативності їх виконання;
- безперервність руху документа;
- єдина база документарної інформації для централізованого зберігання документів і що виключає дублювання документів;
- ефективно організована система пошуку документа.

До інформаційних технологій у підготовці електронних документів належать текстовий процесор Word, видавнича система Publisher, хмарний сервіс Google Word on-line. Для реалізації електронного документообігу використовують системи https://document.online/, Мегаполіс, Аскод та інші.

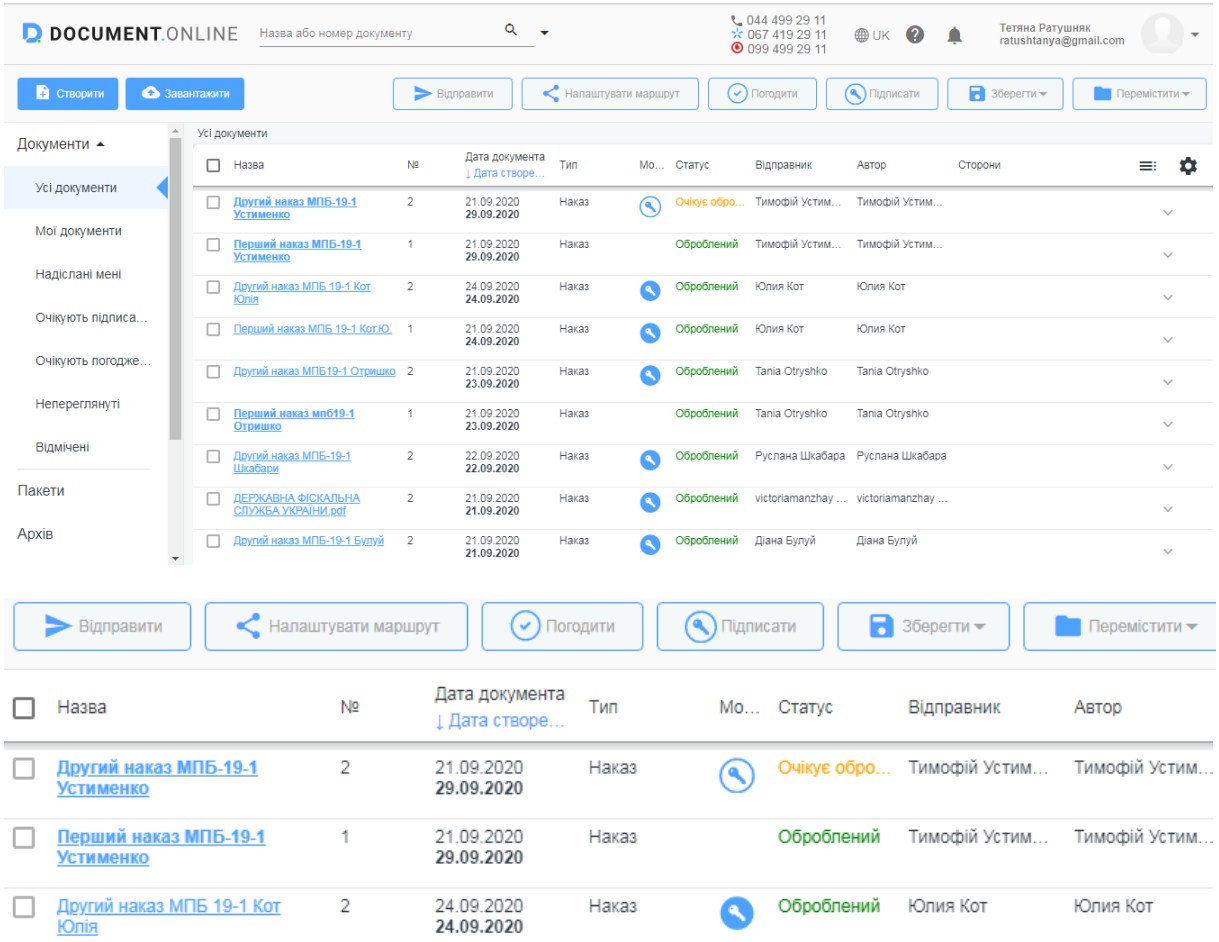


Рис. 6.2. Інтерфейс системи DOCUMENT ONLINE

Запросити користувача зареєструватися у системі можна через пункт меню *Запросити користувача* в обліковому записі. У модальному вікні, що відкриється, введіть e-mail користувача (один або через кому декілька), напишіть короткий супровідний лист та натисніть кнопку *Відправити*.

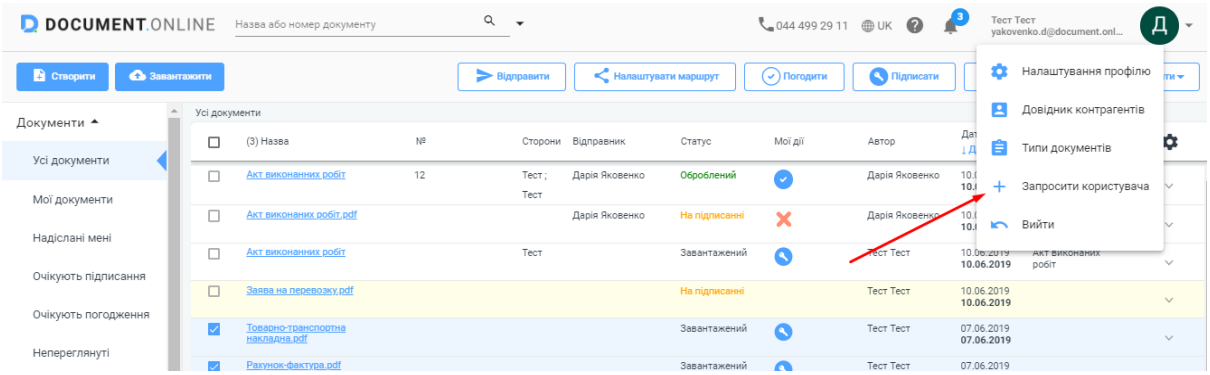


Рис. 6.3. Запрошення на реєстрацію у системі DOCUMENT ONLINE

Для зручної та швидкої роботи з системою можна створити довідник контрагентів. Для цього в меню налаштувань облікового запису оберіть пункт *Довідник контрагентів*. Натисніть кнопку *Створити контрагента*, щоб створити нову картку.

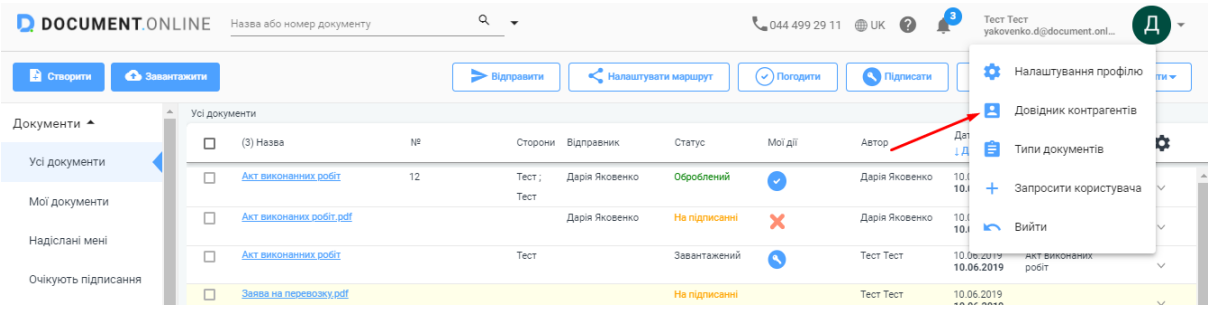


Рис. 6.4. Створення довідника контрагентів у системі DOCUMENT ONLINE

У системі електронного документообігу можна налаштувати список документів, які найчастіше використовують. Для цього потрібно вибрати пункт меню *Типи документів* в обліковому записі. Для відкриття меню натисніть на відповідне позначення для користувача.

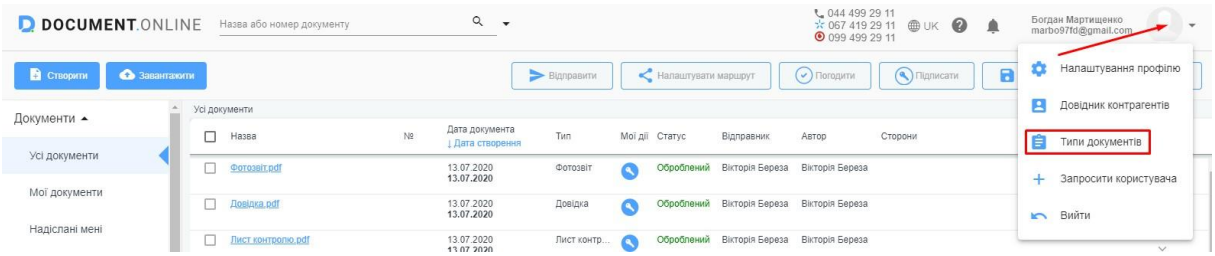


Рис. 6.5. Перехід до налаштування типів документів у системі DOCUMENT ONLINE

У вікні, що відкрилось, виберіть типи потрібних вам документів, поставивши галочку навпроти кожного з них, та натисніть кнопку *Зберегти*.

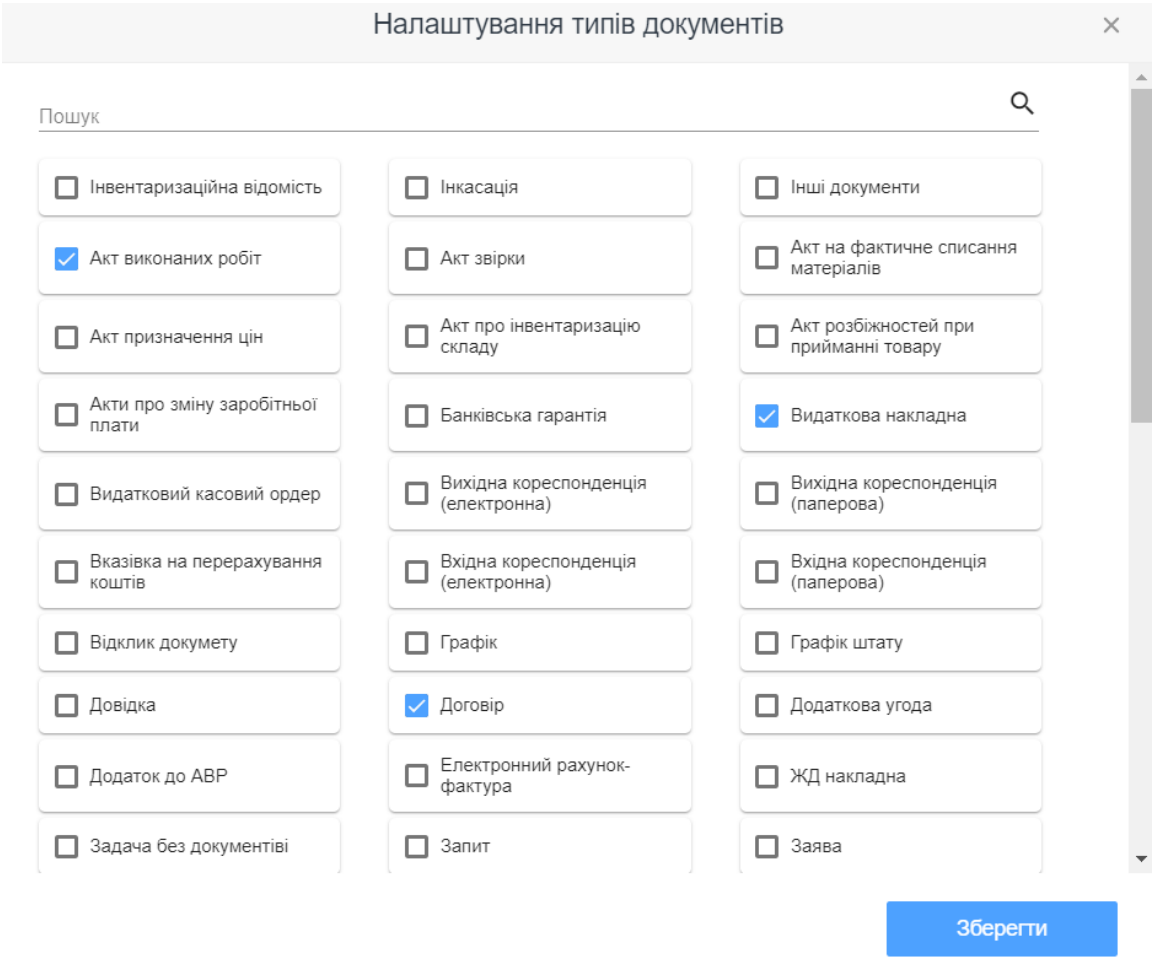


Рис. 6.6. Типи документів у системі DOCUMENT ONLINE

При завантажені документів ці типи документів будуть відображені першими в списку.

Усі документи у системі поділені на розділи, зазначені у таблиці 6.1.

Таблиця 6.1

Розділи документів у системі DOCUMENT ONLINE

Назва розділу	Опис документів
Усі документи	Усі документи, що створені або доступні вам
Мої документи	Документи, що створені вами
Надіслані мені	Документи, що надіслані вам від інших користувачів
Очікують підписання	Документи, що надіслані вам на підписання
Очікують погодження	Документи, що надіслані вам на погодження
Непереглянуті документи	Надіслані вам документи, які ви ще не переглянули
Відмічені	Документи, які ви позначили чекбоксами для виконання масових дій з документами
Архів	Розділ, у якому ви можете зберігати оброблені документи
Кошик	Розділ, у якому ви можете зберігати неактуальні для вас документи
Пакети	Розділ, у якому будуть знаходитись пакети документів, сформовані вами або відправлені вам
Задачі	Розділ, в якому будуть відображатись задачі, що потребують певних дій з вашого боку
Мої розділи +	Будуть знаходитись створені вами розділи

Доступні реквізити документів:

- Назва – назва документа. Вказується автором документа при завантаженні в систему.
- № – номер документа. Вказується автором документа при завантаженні в систему.
- Дата документа – дата, якою виписаний документ. Вказується автором документа при завантаженні в систему. Дата створення – дата створення маршруту погодження до документа в системі.
- Тип – тип документа. Вказується автором документа при завантаженні в систему, обирається зі списку визначених типів документів (Акт виконаних робіт, договір, рахунок-фактура тощо).
- Мої дії – очікувана або вчинена вами дія над документом. Заповнюється системою автоматично.
- Статус – статус документа в системі у межах вашого облікового запису. Заповнюється системою автоматично.

- Відправник – ім’я користувача, який відправив вам документ. Заповнюється для вхідних документів.
- Автор – ім’я користувача, який завантажив документ у систему. Заповнюється автоматично системою.
- Сторони – особи, які вказані у документі як сторони. Вказуються автором документа при завантаженні в систему.

Введення реквізитів документа під час завантаження документа зображено нижче.

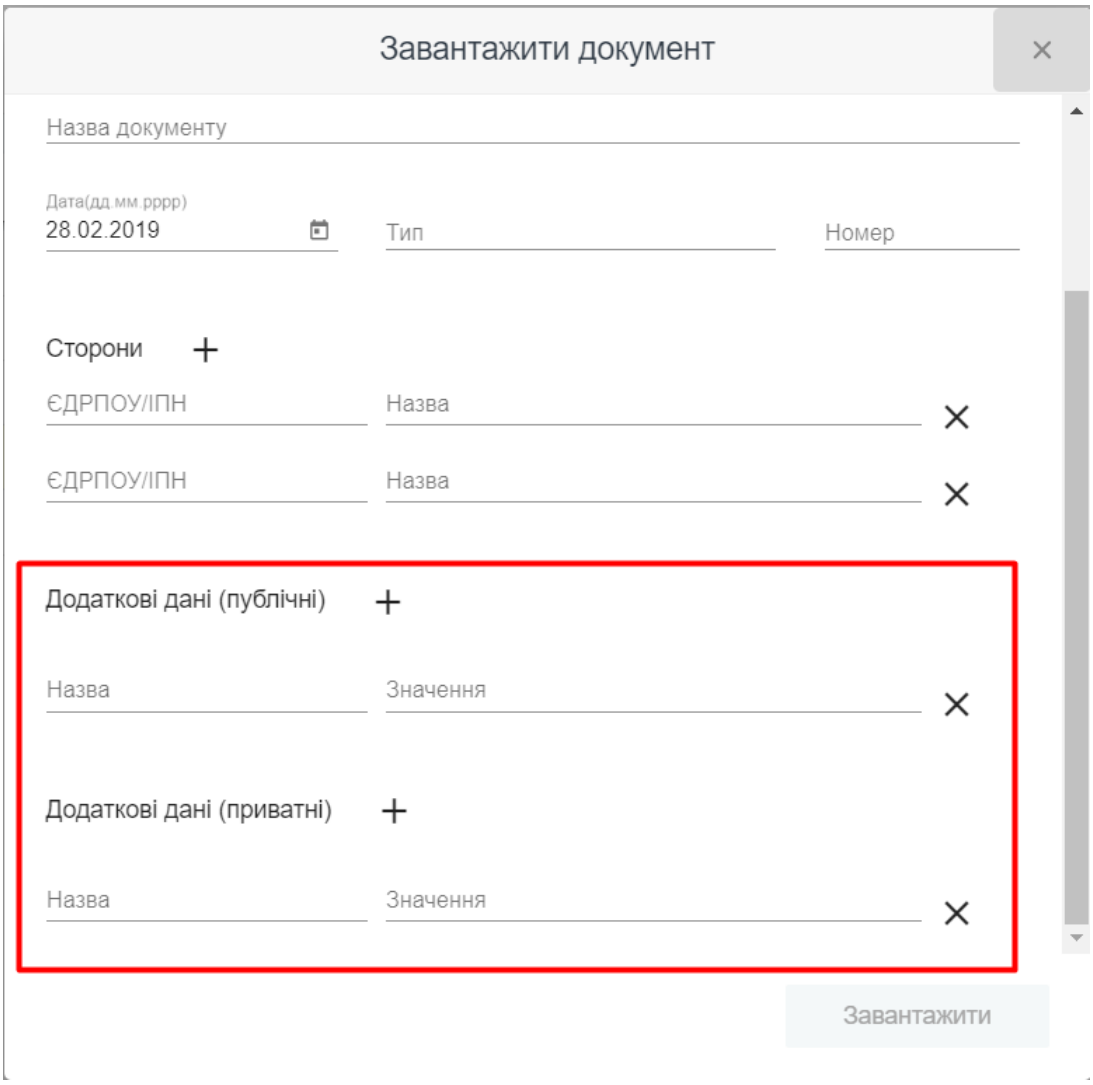
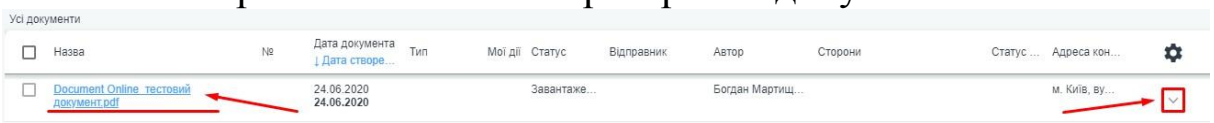


Рис. 6.7. Введення реквізитів документа у системі DOCUMENT ONLINE

Для підписання документа потрібно відкрити необхідний документ. Для цього натисніть на назву документа в робочій області або скористайтеся вікном розкриття документа.



Після відкриття документа ви можете його проглянути.

Для підписання документа з вашої сторони натисніть кнопку *Підписати* на панелі інструментів. У модальному вікні, що відкрилось, оберіть спосіб підписання документа.

Наприклад, обраний вами спосіб підписання – за допомогою файлового носія. Для цього натисніть *Оберіть* та виберіть на комп'ютері папку з необхідним файловим ключем.

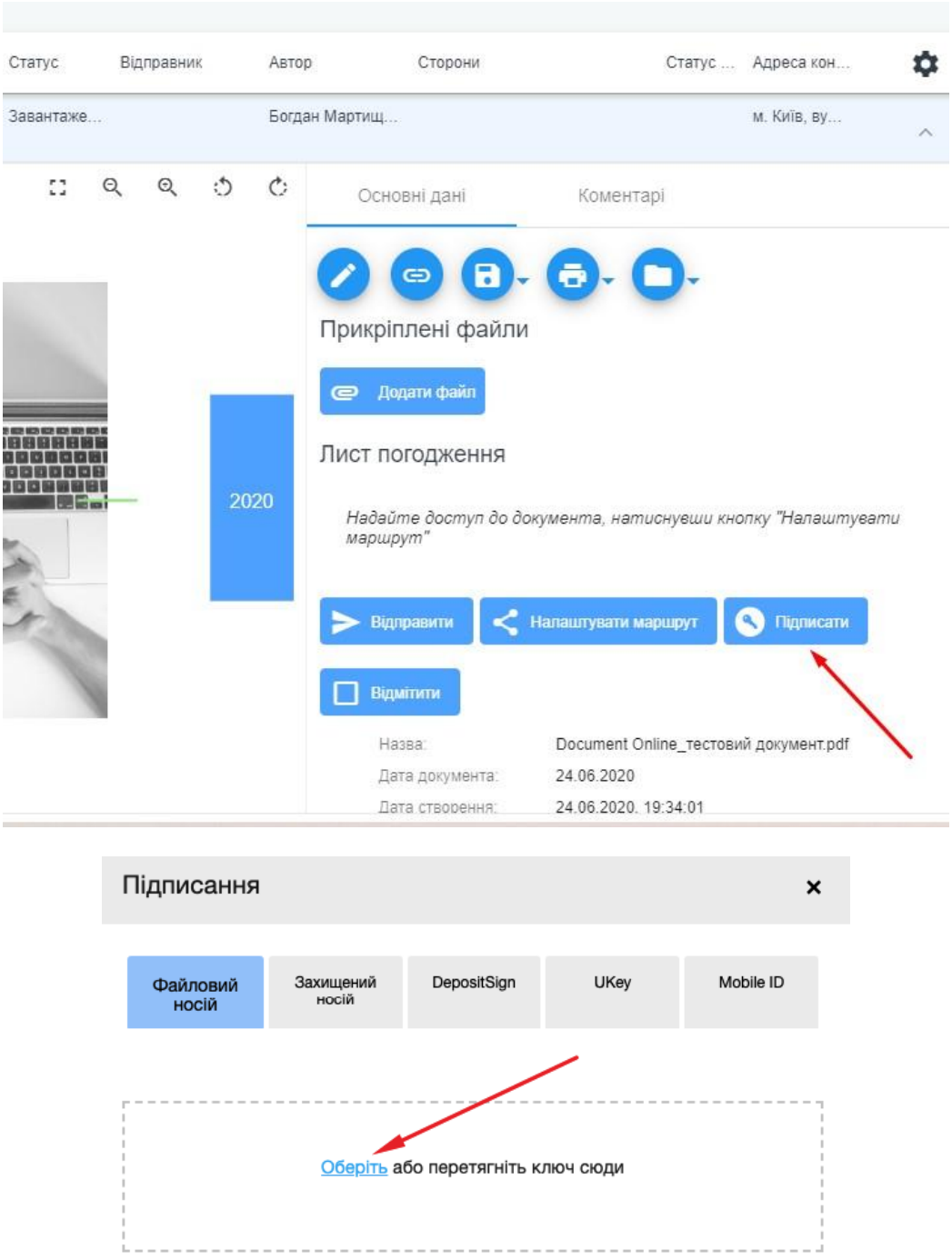


Рис. 6.8. Підготовка до підписання документа у системі DOCUMENT ONLINE

Підтягніть ключ, введіть пароль та натисніть *Додати ключ*. Після цього натисніть *Підписати*.

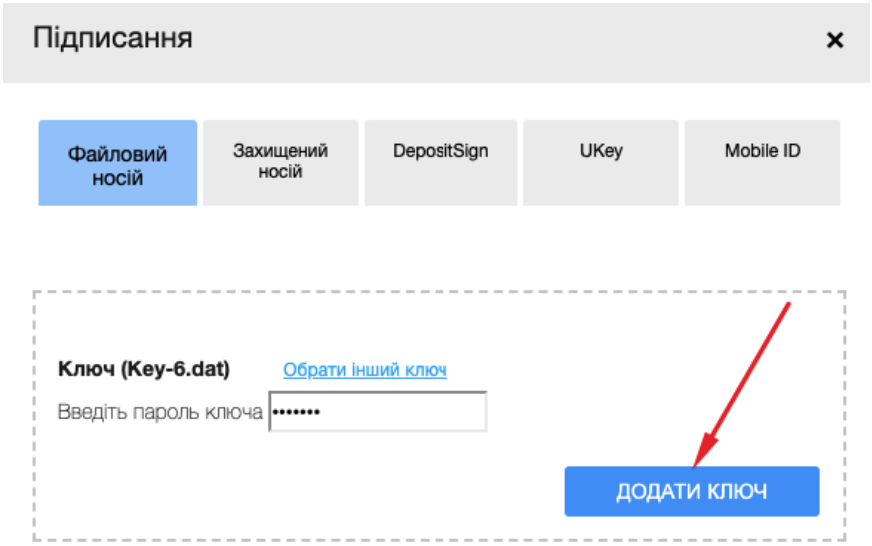
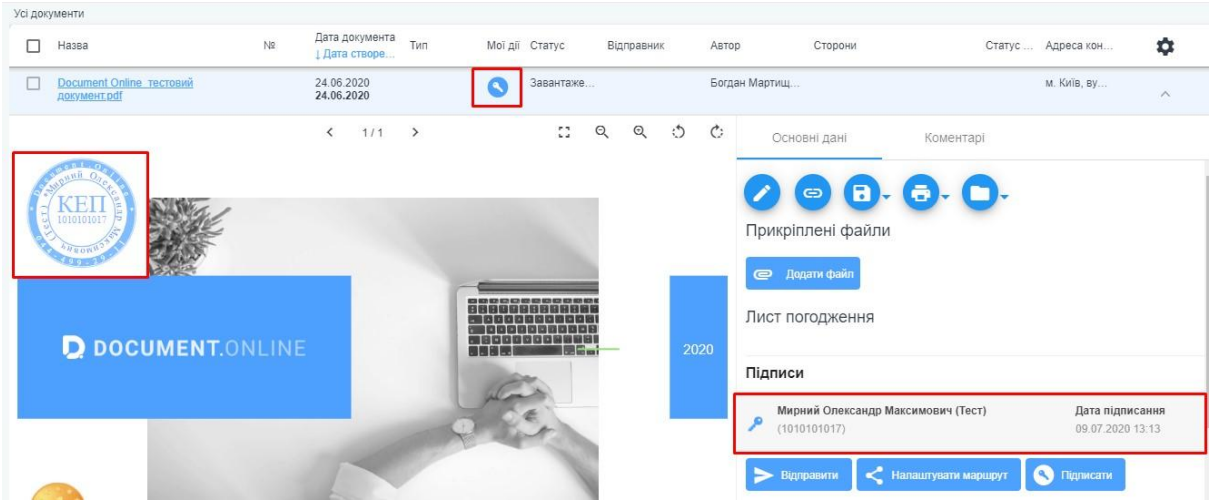


Рис. 6.9. Підписання документа у системі DOCUMENT ONLINE

Підписаний документ відкриється для перегляду. Сам підпис буде вказано на правій бічній панелі, а у колонці *Мої дії* з'явиться піктограма підпису.

Важливо, що відображення КЕП-печатки на підписаному документі доступне лише для документів формату PDF. Цей малюнок печатки не має юридичної сили, це зроблено для візуалізації.



При наступному підписанні документів ваш файловий носій буде автоматично відображатися у вікні *Підписання*. Для підписання достатньо буде натиснути кнопку *Підписати*.

Також ви маєте можливість додавати декілька файлових носіїв.

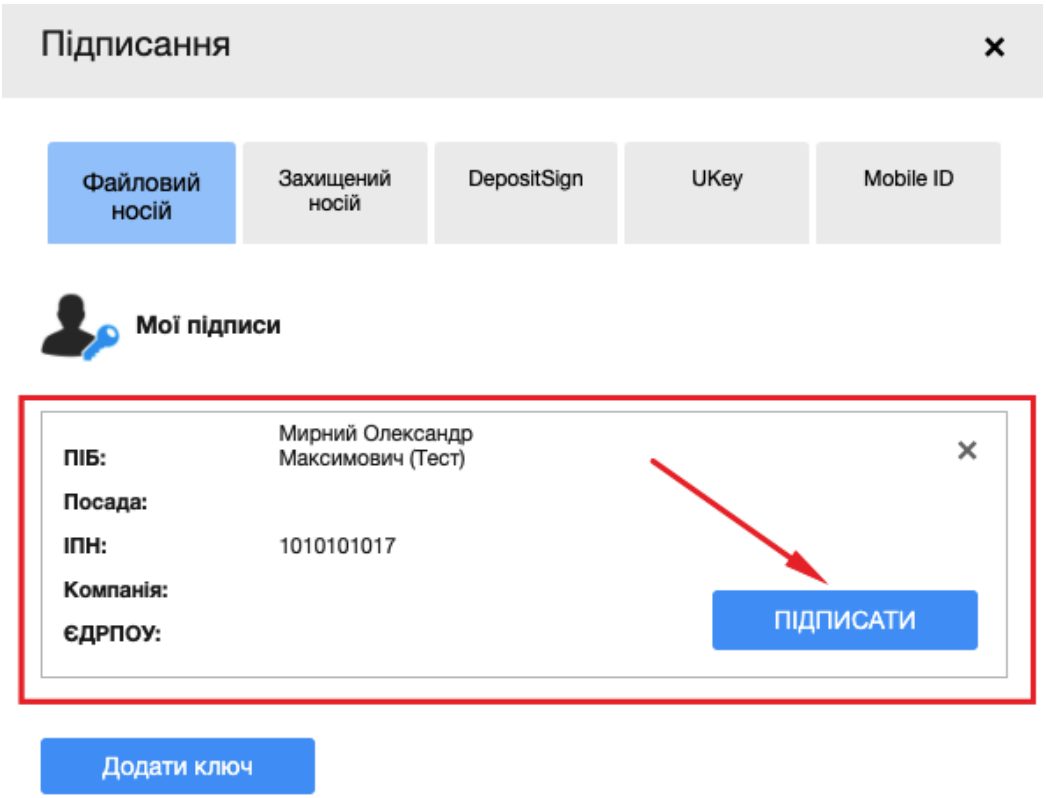


Рис. 6.10. Повторне підписання документа у системі DOCUMENT ONLINE

6.2. Технологія роботи із структурованими документами у текстовому процесорі MS Word 2013

Призначення текстового процесора – підготовка професійно оформлених документів згідно з правилами сучасного діловодства, використовуючи при цьому не тільки набір тексту, а й можливість застосування різних шрифтів, включення у документ таблиць та графічних зображень, перевірку правопису, автоматичне створення змісту та предметних покажчиків, створення складних документів з декількох файлів різних типів та ін. При цьому на екрані користувач може побачити зображення, максимально наближене до того, яким воно буде після друку на папері.

Сьогодні текстовий процесор Microsoft Word 2013 – один з найбільш поширених середовищ створення та редагування документів. Це досить могутній програмний продукт, який надає користувачу широкі можливості якісної підготовки документів.

Він містить потужні бібліотеки стилів і широкий набір засобів форматування і структурування документа. Починаючи з версії MS Word 2013, кожний документ може бути збережений у форматі XML-документа. Щоб зберегти цей документ у форматі XML-документа, потрібно виконати: меню *Файл* → *Сохранить как* → у діалоговому вікні вибрати зі списку Тип документа *Документ XML* (рис. 6.11). Починаючи з версії MS Word 2003, цей редактор може відкривати документи *XML*.

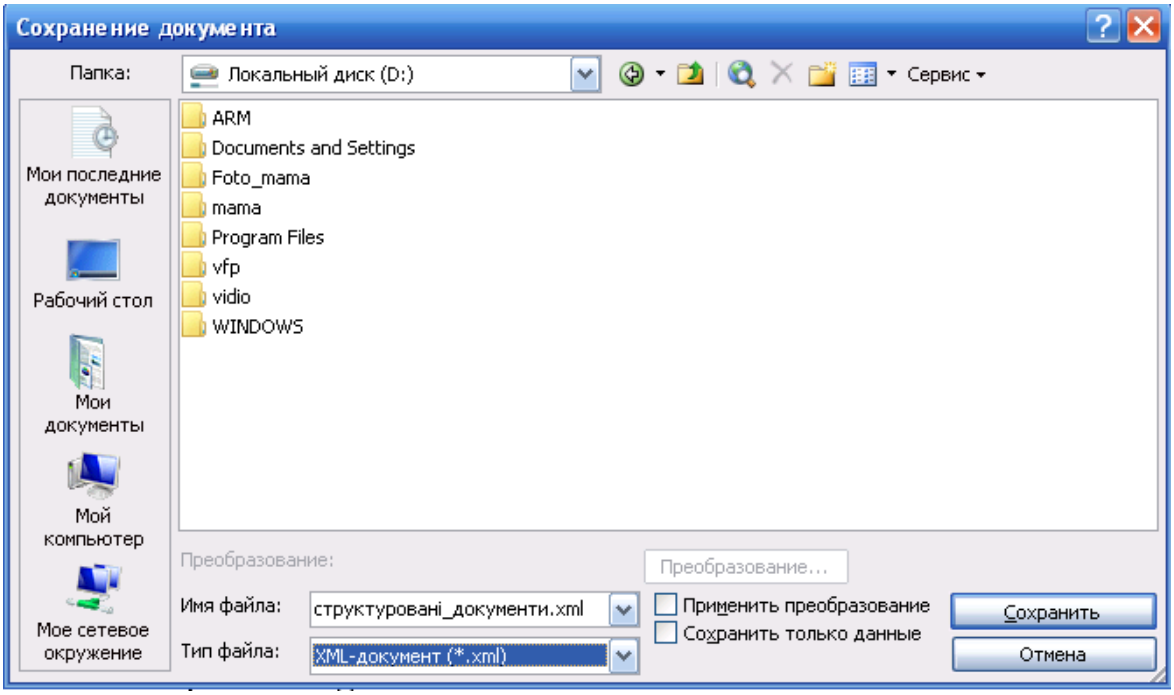


Рис. 6.11. Збереження структурованого документа

Microsoft Word 2013 дозволяє працювати з XML-документами двома способами.

- *З використанням XML-схеми Word.* Створіть документ Word, а потім збережіть його як XML-документ. Word використає свою власну XML-схему за назвою WordML для застосування тегів XML, у яких зберігаються відомості про властивості документа й визначається структура документа – абзаци, заголовки й таблиці. Word також використають XML-теги для зберігання відомостей про форматування й структуру документа відповідно до XML-схеми Word.
- *З використанням будь-якої XML-схеми.* Створіть документ Word або відкрийте його в цьому додатку, приєднайте до документа будь-яку користувальницьку XML-схему, а потім застосуйте

XML-теги до вмісту документа. При збереженні у вигляді XML-документа XML-теги визначають структуру документа відповідно до приєднаної XML-схеми.

При збереженні документа до нього приєднується як схема Word 2013, так і користувальницька схема, щоб зберегти дані відповідно до користувальницької схеми й форматувати текст відповідно до XML-схеми Word. Крім того, відповідно до користувальницької схеми можна зберігати тільки дані документа.

Будь-яке програмне забезпечення, що аналізує дані XML, може також читати й обробляти дані в документі, збереженому у вигляді XML-документа (XML-файл), незалежно від того, чи використалася для структури документа убудована XML-схема Word або приєднана відповідному поставленому завданню користувальницька схема.

Наприклад, якщо користувальницька схема використовується для аналізу даних резюме, XML-теги визначають структуру документа через ім'я, адресу, досвід роботи, дату створення й так далі. При збереженні документ містить як форматований текст, що професійно виглядає після друку документа, так і файл даних, що може бути оброблений будь-якою програмою, призначеною для читання мови XML.

Можна також зберігати дані XML у документі Word (.doc) або у файлі шаблона (.dot). Проте тільки Word зможе прочитати й обробити XML.

Розмітка за допомогою XML-тегів. Якщо до документа приєднана користувальницька XML-схема, в області завдань Структура XML являє собою список елементів, які визначені для схеми. Щоб застосувати XML-теги до документа, вибирають зміст документа, а потім елемент зі списку. Якщо в схемі для елемента визначені атрибути, їх також можна вказати в області завдань Структури XML.

Примітка. До документа може бути приєднано кілька схем. В області завдань *Структури XML* доступний список елементів із всіх приєднаних схем.

Прапорець в області завдань дозволяє побачити XML-теги убудованими у контексті документа.

Якщо структура документа порушує правила схеми, цей фрагмент документа виділяється фіолетовою хвилястою лінією, а в області завдань *Структури XML* відображаються відомості про порушення.

Приєднання й видалення XML-схеми з документа. Виконайте одну з таких дій.

1. У меню *Файл* виберіть команду *Створити* і в області завдань *Створити документ* виберіть *XML-документ*.

2. В області завдань *Структура XML* виберіть команду *Шаблони й надбудови*, а потім перейдіть на вкладку *XML-схема*.

3. У поле *Позначені схеми є приєднаними* встановіть прапорець для кожної XML-схеми, яку потрібно приєднати до документа.

Примітка. У списку відображені всі теми бібліотеки схем. Щоб додати тему в список, натисніть кнопку *Додати схему*.

1. Відкрийте документ у редакторі Word.

2. У меню *Сервіс* виберіть команду *Шаблони й надбудови*, а потім перейдіть на вкладку *XML-схема*.

3. У полі *Позначені схеми є приєднаними* зніміть прапорець для кожної XML-схеми, яку потрібно видалити з документа.

Збереження XML-документа.

1. У меню *Файл* виберіть команду *Зберегти як*.

2. У полі *Тип файлу* виберіть *XML-документ*.

3. У поле *Ім'я файлу* введіть ім'я документа.

4. Якщо необхідно відформатувати документ, доцільно використати файл перетворення Extensible Stylesheet Language (XSLT), встановивши прапорець *Застосувати перетворення*, а потім натиснути кнопку *Перетворення*, щоб знайти XSLT-файл.

5. Натисніть кнопку *Зберегти*.

Перевірка XML. У Microsoft Word можна перевіряти документ XML на відповідність правилам XML-схеми, якщо схема прикріплена до документа. Порушення схеми відображається в області завдань Структура XML.

1. У меню *Сервіс* виберіть команду *Шаблони й надбудови*.

2. На вкладці *XML-схема* виконайте одну з таких дій.

- Щоб перевірка виконувалася, установіть прапорець *Перевірити документ на відповідність приєднаним схемам*.

- Щоб запобігти збереженню документа з невідповідностями в Microsoft Word, зніміть прапорець *Дозволити збереження у форматі XML незалежно від результату перевірки*.

3. Використайте область завдань *Структура XML* для перевірки відповідності документа під час роботи з ним. Значки поруч із елементами означають, що елемент неприпустимий і вказують, що саме порушено.

Примітка. За замовчуванням порушення схеми позначається усередині документа за допомогою пурпурних хвилястих ліній. Щоб виключити ці позначки, на вкладці *XML-схема* діалогового вікна *Шаблони й настроювання* натисніть кнопку *Параметри XML*, а потім установіть прапорець *Сховати порушення схем у документі*. Щоб уникнути помилок перевірки при використанні XML-тегів, в області завдань *Структури XML* встановіть прапорець *Показувати тільки дочірні елементи поточного елемента*.

Завдання подання даних за замовчуванням для XML-документа. При додаванні XML-схеми у бібліотеку схем з нею можна поєднати один або більше файлів перетворення Extensible Stylesheet Language (XSLT). Якщо документ із приєднаною схемою відкривається у Microsoft Word, можна задати спосіб відображення даних, вибравши одне з доступних перетворень XSL або подань даних.

1. У меню *Сервіс* виберіть команду *Шаблони й надбудови*, а потім перейдіть на вкладку *XML-схема*.
2. Натисніть кнопку *Бібліотека схем*.
3. У полі *Виберіть схему* виберіть схему, яку потрібно поєднати з перетворенням XSL.
4. У полі *Використати рішення з* виберіть пункт *Word*.
5. Натисніть кнопку *Додати рішення*, знайдіть XSLT-файл, а потім натисніть кнопку *Відкрити*.
6. У діалоговому вікні *Параметри рішення* виберіть потрібний параметр й уведіть ім'я XSLT у поле *Псевдонім*.

Примітка. Ім'я, уведене як псевдонім, з'явиться у списку доступних подань даних в області завдань *Подання даних XML*.

7. Натисніть кнопку *OK*.
8. Повторіть кроки з 4 по 6 для кожного подання XSL, які потрібно поєднати зі схемою.

9. У полі *Рішення за замовчуванням* виберіть ім'я подання XSL, яке необхідно використати за замовчуванням.

Типи створюваних документів у MS Word 2013.

1. Електронні документи (вебсторінки).
2. Документи для друку.
3. Електронне повідомлення.

Для кожного різновиду документа використовується певний шаблон.

Режими відображення документів:

1) звичайний;
2) режим вебдокумента;
3) режим розмітки;
4) режим структури;
5) попередній перегляд;
6) попередній перегляд вебсторінки.

Найменша структурна одиниця тексту – *символ (друковані і недруковані)*, далі – *слово, речення. абзац, сторінка, розділ.*

Введення тексту в документ. Існує декілька способів введення тексту в документ:

1) безпосередньо з клавіатури;
2) використовуючи буфер обміну;
3) вставка елемента Автотексту та використання Автозаміна;
4) вставка фрагментів інших документів;
5) перетягуванням тексту безпосередньо в межах одного документа чи між різними документами з метою переміщення чи копіювання тексту.

При введенні тексту з клавіатури використовують усі цифрові і символьні клавіші, а також клавішу *Shift* (однократне використання верхнього регістру) і CapsLock (постійне використання верхнього регістру). Клавіша *Enter* використовується тільки для закінчення абзацу. Верстка рядків і сторінок автоматична.

Для примусового утворення нової сторінки потрібно натиснути комбінацію клавіш *Ctrl* + *Enter*, для переходу на новий рядок без утворення абзацу Shift + Enter.

Автотекст – це текст, що зберігається у спеціальному словнику, і використовується для полегшення багаторазового введення повторюваних фраз, виразів, довгих речень. Нижче розглянуті засоби його створення, використання, видалення і запам'ятовування.

Автозаміна – це спеціально створений текст, що має умовне позначення (псевдонім) для полегшення багаторазового введення повторюваних фраз, виразів, довгих речень. Нижче розглянуті засоби його створення, використання, видалення.

Список – це сукупність абзаців, на початку кожного з яких стоїть номер або маркер. *Маркер* – це символ чи рисунок, які використовуються з метою надання більшої виразності абзацам списку, які називають елементами списку. Якщо у ролі маркера виступає символ, то його формат легко змінити, а якщо рисунок – змінити маркер у MS Word 2013 неможливо.

Списки використовують для полегшення сприйняття інформації у документі. У MS Word існує три типи списків – марковані, нумеровані та ієрархічні (багаторівневі). Нижче наведені зразки кожного із типів.

У разі використання кнопок панелі *Форматирование: Маркери*  для маркованого списку або *Нумерація*  для нумерованого будуть використовуватись параметри створення списків, маркери та стилі, які встановлені за замовчуванням у цьому шаблоні, чи ті, які використовувались останніми.

Параметри списків можна настроювати, використовуючи для цього команду меню *Формат* ➔*Список* і, переключившись на потрібну вкладку, діалогового вікна Список, вибрати вбудований стиль списку чи самому створити новий стиль списку, обравши для заміни певний стиль списку, після чого натиснути кнопку *Изменить* і в діалоговому вікні *Изменение …* списка задати необхідні параметри.

Виноски – це символи, що використовуються у тексті у вигляді верхніх індексів, які пов'язані з певним зазвичай роз'яснювальним текстом, який відображається у тексті при наведенні покажчика миші. Розміщується текст виноски зазвичай внизу сторінки, на якій є виноска, у кінці розділу, в якому міститься виноска, чи в кінці всього документа.

Примітка – це різновид виноски, яка відображається на полях документа і спеціальним способом форматується. Примітка може бути текстова чи звукова. Щоб додати примітку, потрібно встановити текстовий курсор у місці її вставки у тексті, а потім виконати: меню *Вставка* ➞*Примечание*. Після цього потрібно створити зміст примітки – ввести текст або (та) додати звукову примітку у вікні звукозапису, натиснувши кнопку *Создать примечание* на панелі *Рецензирование* і обираючи зі списку команду *Звуковое примечание*, яка з'являється автоматично при вставці примітки.

Щоб вилучити примітку потрібно відкрити її контекстне меню і клацнути на команді *Удалить примечание*. При вилученні всіх приміток можна скористатись відповідною командою зі списку біля кнопки *Отклонить примечание и изменения* на панелі *Рецензирование*.

Закладка – це поіменоване місце в документі, яке використовується для швидкого переходу до нього у великому документі, а також може використовуватись як аргумент у деяких функціях.

Щоб створити у тексті *Закладку*, потрібно встановити текстовий курсор у місці її створення у тексті, а потім виконати: меню *Вставка* ➞*Закладка*. Після цього з'явиться діалогове вікно:

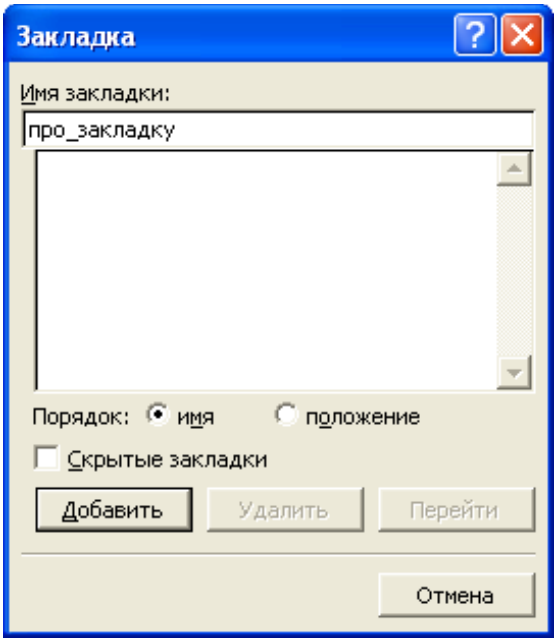


Рис. 6.12. Створення закладки

В полі *Имя* закладки ввести ім’я закладки, яке повинно починатися з букви і не містити пропусків. Потім клацнути на кнопці *Добавить*.

Щоб вилучити закладку, потрібно виконати меню *Вставка* ➞*Закладка*. У діалоговому вікні виділити у списку ім’я закладки і натиснути кнопку *Удалить*.

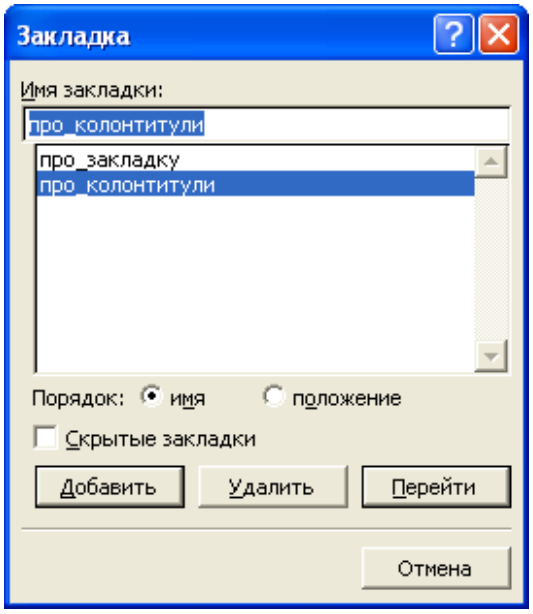


Рис. 6.13. Завершення створення закладки

Колонтитул – це текст чи графічні об’єкти, або і те, і інше, що розташовується вгорі та внизу по відношенню до основного тексту і повторюється на кількох чи всіх сторінках розділу чи всього документа. У MS Word 2013 можна створити на парних сторінках розділу чи всього документа один зміст колонтитула, а на непарних – інший зміст. Щоб створити колонтитули, потрібно виконати: меню *Вид* ➔*Колонтитулы*, після чого відкриється панель *Колонтитулы*, а текстовий курсор встановиться у верхньому колонтитулі.

Кожний розділ може мати власні колонтитули, однакові для всіх сторінок, чи різні для парних і непарних сторінок. У разі наявності різних колонтитулів для парних і непарних сторінок потрібно в діалоговому вікні Параметри страницы, яке доступне після натискання одноіменної кнопки на панелі Колонтитули, вказати це за допомогою перемикача *Различать колонтитулы четных и нечетных страниц*, а також у разі потреби і першої сторінки, якщо, наприклад, вона є титульною.

Щоб вилучити колонтитули, потрібно перейти в них і вилучити текст та інші наявні об'єкти в них.

*Шаблони***.** У шаблонах зберігаються стандартні тексти, панелі інструментів, що настроюються, макроси, сполучення клавіш, стилі й елементи списку автотексту. Найпростіший спосіб створити шаблон – це відкрити документ, що містить елементи, які потрібно використовувати повторно, і зберегти його як шаблон.

Якщо зберегти шаблон у папці *Шаблоны*, то він з'явиться на вкладці *Общие* при виборі команди *Создать* з меню *Файл*. Якщо зберегти шаблон в одній з папок, вкладених у папку *Шаблоны*, наприклад *Записки* або *Письма и факсы***,** то при виборі команди *Создать* він з'явиться на вкладці з відповідною назвою.

Розробка форм. Форма – це документ, що містить такі елементи:

– текст або графічні елементи, що не можуть бути змінені особою, яка заповнює форму. Ці елементи задаються розробником форми і містять у собі питання, списки можливих відповідей, таблиці з даними тощо;

– незаповнені області, у які вводить дані особа, яка заповнює форму. До цих елементів належать поля форми й об'єкти Active X.

Для створення форми необхідно скласти її приблизний вигляд або використати як зразок уже створену форму, крім того, вирішити, якою буде форма: електронною чи друкованою. До переваг електронних форм належить автоматична перевірка введених даних (наприклад, табельного номера співробітника), обновлення залежних полів (наприклад, полів міста й області при введенні поштового індексу) і наявність підказок, що полегшують заповнення форми.

Багато форм, наприклад контракти, складаються в основному з тексту з включеними в нього полями форми, у які вводяться необхідні дані. В інших формах використовуються сітки, що поєднують у собі такі властивості таблиць, як вирівнювання тексту, рамки для виділення полів, що заповнюються, виділення кольором заголовків та інші спеціальні прийоми, що роблять форму більш привабливою і легкою у використанні.

У Word можна створити такі види форм:

– форми, які друкуються, а потім заповнюються на папері;

– електронні форми, які можна поширювати через електронну пошту або мережу;

– потужні електронні форми, що настроюються, утворювані з використанням об'єктів Active X, які надають значно більші можливості для введення даних, вибору відповідей із списку і запису відповідей. Ці форми також можна поширювати через електронну пошту або мережу.

Створення друкованої форми:

– створити новий документ;

– розробити форму або використати як зразок існуючу форму;

– додати текст і графічні елементи. Наприклад, введіть питання і списки можливих відповідей;

– щоб вставити текстове поле для відповіді, треба вказати місце в документі, а потім натиснути кнопку *Текстовое поле* на панелі інструментів *Формы*;

– щоб вставити прапорці зі списком можливих відповідей, наприклад «Так» або «Ні», треба вказати потрібне місце в документі, а потім натиснути кнопку *Флажок* на панелі інструментів *Формы*;

– зберегти форму;

Створення електронної форми. Електронну форму необхідно зберігати і поширювати у вигляді шаблону. Для використання такої форми потрібно створити новий документ з використанням цього шаблону. Шаблон залишається «незаповненою формою» і може бути використаний повторно. Отже, треба виконати такі дії:

– розробити форму або використати як зразок існуючу форму;

– створити новий шаблон. Для цього вибрати команду Создать в меню *Файл*, вибрати *Новый документ* на вкладці *Общие*, встановити перемикач *Новый документ* у положення *Шаблон*, а потім натиснути кнопку *ОК*;

– якщо для створення форми використовується форма-зразок, що не є шаблоном, зберегти створену форму як шаблон;

– додати текст і графічні елементи. Наприклад, ввести питання;

– вставити поля форми в потрібні місця документа. Для цього вказати місце, де повинно знаходитися поле, а потім використати кнопки на панелі інструментів *Формы*;

– щоб вивести панель інструментів *Формы* на екран, необхідно вибрати команду *Панели инструментов* меню *Вид* і встановити прапорець навпроти опції *Формы*;

– двічі клацнути на поле для настроювання його параметрів або натиснути кнопку *Параметры поля формы* на панелі інструментів *Формы*. Щоб отримати інформацію про будь-який параметр, треба натиснути кнопку зі знаком питання, а потім клацнути на цей параметр;

– після створення форми натиснути кнопку *Защита формы* на панелі інструментів *Формы*, щоб дозволити введення даних тільки в полях форми. Захист дозволяє користувачам заповнювати форму, але запобігає випадковим змінам елементів форми та їх розміщення;

– зберегти і закрити шаблон.

Поля форми затінюються, щоб їх легше було знаходити на екрані. Для вмикання або вимикання затінення треба натиснути кнопку *Затенение полей формы* на панелі інструментів *Формы*.

При розробці та зміні форми з неї треба зняти захист. Щоб захистити форму або зняти захист, потрібно натиснути кнопку *Защита формы* на панелі інструментів *Формы*.

Додавання у форму полів форми. Текстове поле призначене для введення різноманітної інформації. Подвійне клацання на текстовому полі форми відкриває вікно *Параметры текстового поля* (рис. 6.14):

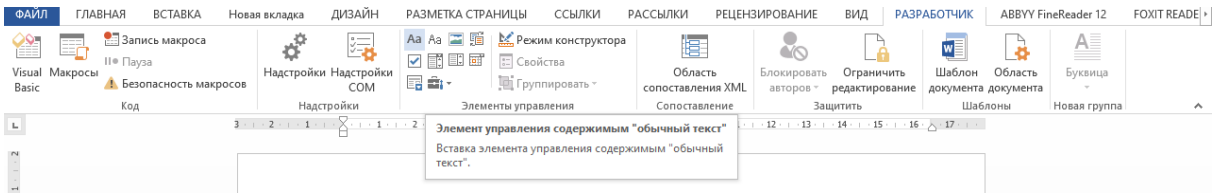


Рис. 6.14. Параметри текстового поля

У полі *Тип* цього діалогового вікна можна встановити такі опції: *Обычный текст, Число, Дата, Текущая дата, Текущее время, Вычисление*.

У полі *Текст по умолчанию* може бути задане значення за замовчуванням, так що користувачам не доведеться вводити найімовірнішу відповідь, крім того випадку, коли їх відповідь відрізняється.

Прапорець. Призначений для додавання прапорця поруч із незалежним параметром, що або встановлюється, або знімається. Прапорці можуть бути також додані до кількох параметрів, якщо вони не виключають один одного.

Поле со списком. Призначено для додавання поля зі списком можливих відповідей, що відкривається, якщо клацнути мишею на полі зі списком. Щоб задати список, необхідно відкрити діалогове вікно *Параметры поля со списком* за допомогою натискання кнопки *Параметры поля формы* на панелі інструментів *Форми* (рис. 6.15).

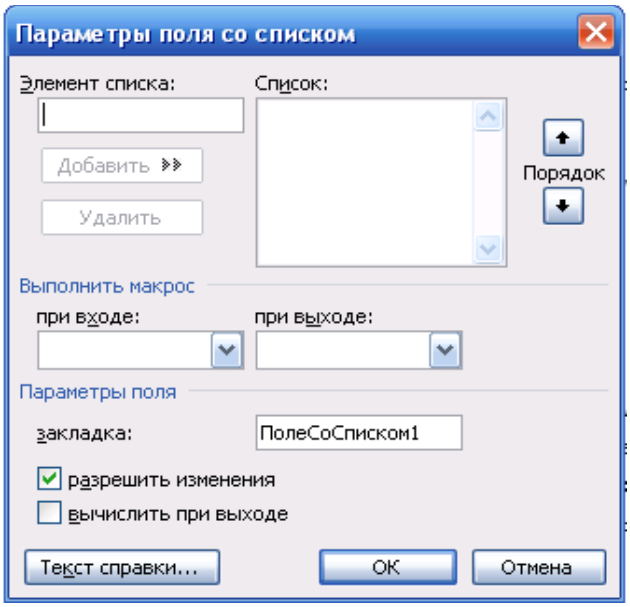


Рис. 6. 15. Вигляд вікна *Параметры поля со списком*

Для створення списку в поле *Элемент списка* необхідно ввести перший елемент списку і натиснути кнопку *Добавить*. Введений елемент списку з'явиться у полі *Список*.

Якщо список достатньо великий, користувач зможе прогортати його для перегляду решти відповідей.

Створення електронної форми з використанням об'єктів Active X:

– розробити форму або використати як зразок існуючу форму;

– створити новий шаблон. Для цього вибрати команду *Создать* в меню *Файл*, вибрати *Новый документ* на вкладці *Общие*, встановити перемикач *Новый документ* у положення *Шаблон*, а потім натиснути кнопку *ОК*;

– якщо для створення форми використовується форма-зразок, що не є шаблоном, треба зберегти створену форму як шаблон;

– додати текст і графічні елементи. Наприклад, ввести питання;

– вставити об'єкти Active X у потрібні місця документа за допомогою панелі інструментів *Элементы управления*;

– щоб настроїти властивості обраного об'єкта, натиснути кнопку *Властивості* на панелі інструментів *Элементы управления*.

Додавання у форму об'єктів Active X

Для додавання в електронну форму або документ об'єктів ActiveX використовується панель інструментів *Элементы управления*.

При виборі об'єкта Word автоматично переключається у режим розробки і вставляє цей об'єкт як такий, що можна перемістити, щоб його було легше помістити в потрібному місці документа. Щоб помістити об'єкт у позицію курсору, потрібно утримувати клавішу SHIFT при натисканні відповідної кнопки панелі інструментів.

Панель інструментів *Элементы управления* має такі кнопки:

– *Флажок.* Дозволяє додати прапорець поряд із незалежним параметром, що або встановлюється, або знімається.

– *Счетчик.* Дає можливість вставити кнопку лічильника, що дозволяє збільшувати або зменшувати числову відповідь на одиницю.

– *Полоса прокрутки.* Дозволяє вставити смугу прокручування поряд із списком, в якому більше елементів, ніж можна вивести на екран.

– *Надпись*. Дає можливість вставити напис.

– *Поле.* Вставляє поле для редагування, у якому відображається введений користувачем текст.

– *Кнопка*. Дозволяє вставити кнопку, що виконує такі дії, як, наприклад, показ додаткових параметрів.

– *Переключатель*. Дозволяє встановити перемикач поруч із групою з двох або більше відповідей, що виключають одна одну.

– *Список*. Дає можливість вставити поле, в якому у вигляді списку будуть виводитися можливі відповіді. Список можна гортати, якщо його розмір перевищує розмір поля.

– *Поле со списком*. Дозволяє вставити текстове поле зі списком. Користувач може або ввести значення в текстовому полі, або вибрати значення зі списку. Поле зі списком також має смугу прокручування.

– *Выключатель*. Дає можливість вставити кнопку-вимикач, що переключає різноманітні стани.

– *Рамка*. Дозволяє вставити рамку, яка буде об'єднувати пов'язані поля і кнопки;

– відкрити шаблон, в якому знаходиться електронна форма;

– зняти захист із форми, натиснувши кнопку *Защита формы* на панелі інструментів *Формы*;

– виділити поле форми, формат якого потрібно змінити;

– вибрати команду *Шрифт* у меню *Формат*;

– вибрати необхідні параметри. Щоб отримати відомості про будь-який параметр, треба натиснути кнопку зі знаком питання, а потім клацнути на цей параметр;

– захистити форму, натиснувши кнопку *Защита формы* на панелі інструментів *Формы*.

Для однакового форматування кількох полів виділити поле, формат якого буде прийнятий для інших полів, натиснути кнопку *Копировать формат*, а потім виділити поле, що треба відформатувати. Щоб застосувати форматування для кількох полів, треба двічі натиснути кнопку *Копировать формат*, виділити по черзі поля, а потім ще раз натиснути цю кнопку.

Запитання для самоконтролю

1. Які системи електронного документообігу вам відомі?
2. Призначення текстового процесора.
3. Типи створюваних документів у MS Word 2013.
4. Використання списків. Скільки типів списків існує у MS Word 2013?
5. Як створити у тексті Закладку?
6. Що таке колонтитул і як його створити у тексті?
7. Що таке шаблон? Як його створити і зберегти?
8. Поясніть створення електронної форми з використанням об’єктів Active X.
9. Збереження й експорт даних XML.
10. Технологія створення діаграм та графіків на основі табличних даних.

ЛЕКЦІЯ 7
ОСНОВИ ОФІСНОГО ПРОГРАМУВАННЯ

7.1. Призначення та основні поняття системи об'єктно-орієнтованого програмування VBA: редактор, процедури та функції, основні конструкції та оператори мови

Об'єктно-орієнтоване програмування. Об'єктно-орієнтоване програмування – це найбільш сучасний стиль у розробці комп'ютерних програм. Цей стиль використовується тому, що сучасне конструювання програм намагається досягти певної специфічної мети. Комп'ютерні програми повинні бути:

- що перевіряються;
- що модернізуються;
- що повторно використовуються;
- що переносяться.

Програми повинні бути такими, що перевіряються, щоб була можливість пересвідчитися, що вони роблять саме те, що мають робити. Програми повинні бути такими, що модернізуються, щоб можна було досить добре зрозуміти програму та внести коректні зміни. Програми повинні бути такими, що повторно використовуються, щоб, лише одного разу написавши процедуру, потім могли повторно використати її за будь-якої потреби. І, нарешті, програми мають бути такими, які можна переносити, тобто можна було перенести їх на різні платформи з мінімальною переробкою.

Сучасна практика програмування шукає досягнення цих цілей в модульності комп'ютерних програм. Модульні комп'ютерні програми під час розробки розбиваються на окремі частини, що іменуються модулями. Кожний модуль виконує специфічні, суворо певні функції перетворення і має доступ тільки до тих даних, які необхідні для цього перетворення. Код модуля, розроблений із суворо певним інтерфейсом для інших модулів програми, дуже легкий для відладки, супроводу і розуміння. Іншим аспектом модульності є його замкненість, що додає упевненість в

тому, що будь-які зміни, які можна внести в код модуля, вплинуть тільки на функції цього модуля і ні на що інше. Отже, ці особливості полегшують відладку, супровід і розуміння коду.

Об'єктно-орієнтоване програмування (далі – ООП) максимально використовує принцип модульності. Програмний об'єкт в ООП називається «контейнером». Контейнер містить дані та код, який знає, як маніпулювати цими даними. Іншими словами, програмний об'єкт – це просто блок, що складається з коду і даних, він може бути представлений як реальний фізичний об'єкт, наприклад у вигляді командної кнопки. Програмний об'єкт володіє специфічною функцією перетворення і містить весь необхідний код і всі необхідні дані для реалізації цієї функції, але його внутрішня робота прихована від стороннього ока.

У більш традиційному програмуванні комп'ютерна програма розроблялася у вигляді функціональних наборів рядків, у яких блоки даних передавалися від модуля до модуля і кожний модуль модифікував або використовував ці дані своїм способом. Модулі не містили даних, а тільки коди для модифікації будь-яких переданих даних. Якщо програма випадково передасть такій процедурі неправильні дані, процедура все одно обробить їх, повернувши вам «сміття», або взагалі викличе аварію вашої системи.

У об'єктно-орієнтованому програмуванні дані та код, який маніпулює цими даними, об'єднані у структурі, званій «об'єкт». Замість того, щоб передавати дані від модуля до модуля для виконання різних обчислень, варто посилати повідомлення об'єкту, що містить ці дані, і, отже, доручити йому зробити обчислення. Передати об'єкту погані дані неможливо, бо дані, з якими об'єкт маніпулює, включені в цей об'єкт. Більш того, не можна викликати або виконати об'єкт, можна тільки послати йому повідомлення з ввічливим проханням виконати одну із задач, що вирішуються ним (звичайно, якщо об'єкт не знайде можливості здійснити необхідну маніпуляцію, то, швидше за все, йому доведеться ввічливо повідомити, що він не може задовольнити запит).

Основні поняття Visual Basic for Application. Мова програмування VBA виникла на основі мови Visual Basic (далі – VB). VBA тісно пов'язаний з мовою VB. У VB і VBA використовуються ідентичні середовища розробки. Основна відмінність мови

програмування VB і мови Visual Basic for Application в тому, що на мові Visual Basic створюються самостійні додатки, а проєкти VBA працюють в середовищі Microsoft Office. Крім того, доцільно зазначити що, програма VBA не компілюється, на відміну від VB, а інтерпретується. Це означає, що переведення програми в машинний код здійснюється у процесі її виконання і проєкти VBA виконуються тільки з допомогою додатка, який підтримує VBA. При компіляції переведення програми в машинний код відбувається до запуску програми, тому скомпільовані додатки працюють швидше і займають менше пам'яті. Проте, по-перше, вони мають меншу гнучкість, і, по-друге, нині комп'ютерна техніка настільки прогресивна, що швидкодія й обсяг пам'яті не є критичними для офісних додатків (помітити різницю у швидкодії програм, написаних на VB і VBA, практично неможливо).

Якщо ж говорити загально, то Visual Basic for Application – це інструмент розробки додатків.

VBA – не просто стандартна макромова додатків Microsoft Office: цей інструмент застосовується для розширення функціональних можливостей додатків, у яких він використовується. Наприклад, можна додати власне меню або функцію до вбудованих засобів Excel, або створити свого майстра, щоб спростити створення презентацій в PowerPoint. Спільно використовуючи VBA з програмами Microsoft Office, можна розв'язувати надто складні задачі. Наприклад, за допомогою VBA можна:

1. Створити макроси, які дозволяють автоматизувати роботу та розширюють функціональні можливості додатка, у який вбудований VBA.

2. Змінити меню будь-якого додатка Microsoft Office.

3. Створити власне діалогове вікно і додати йому потрібний зовнішній вигляд.

4. Керувати іншим додатком Microsoft Office або даними з цього додатка.

5. Об'єднати дані з декількох додатків Microsoft Office в одному документі.

6. Створювати власні функції в Microsoft Excel.

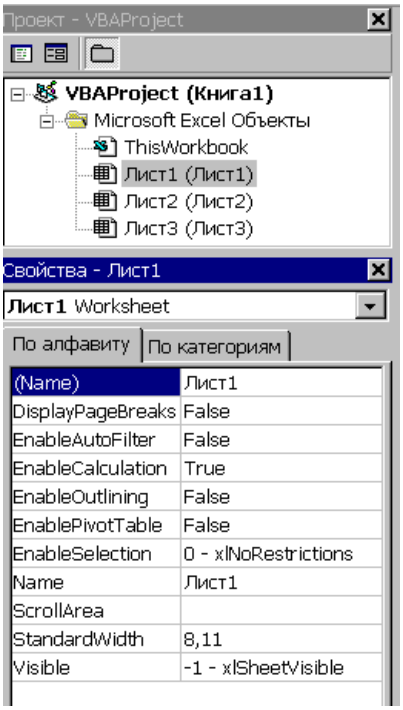
7. Автоматично створювати або змінювати сторінки Web, спільно використовуючи додатки Microsoft Office і VBA.

Структура проєктів VBA. Під час розв'язування задач за допомогою VBA потрібно створити проєкт. Проєкт включає кілька елементів. Розглянемо коротко кожний з цих елементів.

Розробка проєктів і створення додатків. Проєкти VBA виконуються разом з іншими додатками. Додаток, у якому розробляється і виконується проєкт VBA, називається основним. Наприклад, можна створити проєкт VBA, що працює разом з Microsoft Excel. У цьому випадку Excel є основним додатком. Фактично, не використовуючи основний додаток, не можна побудувати додаток VBA.

Середовище розробки VBA. VBA має власне середовище розробки, що називається інтегрованим середовищем розробки, або IDE. VBA IDE – це вікно, що містить меню, інші вікна й елементи, що застосовуються під час створення проєктів VBA.

Редактор VBA активізується командою *Сервіс, Макрос, Редактор Visual Basic* або натисненням кнопки *Редактор Visual Basic* панелі інструментів Visual Basic або комбінації клавіш <лівий Alt> + <F11>. Повернутися з редактора Visual Basic в робочу книгу можна натисненням кнопки *Viev Microsoft Excel* панелі інструментів Standard.



У результаті потрапляємо у середовище розробки додатків редактора VB.

Інтерфейс редактора Visual Basic складається з таких основних компонент:

- вікно Project – VBA Project (Проєкт);
- вікно редагування коду.

Вікно розробки має стандартний вигляд для Windows додатків: рядок меню, панель інструментів і два вікна *Project* (Проєкт) і *Propetis* (Властивість). У вікні *Project* – VBA Project відображається реєстр модулів і форм, які входять до створюваного проєкту. *Модуль* – це сторінка (не плутати з робочим аркушем), де набирається код (програма). Подвійним натисненням на значок модуля можна відкрити відповідний модуль. Значок активного модуля

виділяється сірим кольором. У VBA кожний робочий лист має модуль. Щоб додати його в проєкт, необхідно вибрати команду: *Вставка, Модуль*. У результаті з'являється робочий лист модуля. Для створення програм потрібно скористатись: *Вставка, Процедура*.

Існують два способи запуску програми:

1. Встановити курсор в будь-яке місце програми та натиснути функціональну клавішу *F5*.

2. Натиснути з будь-якого місця програми піктограму *Пуск*. Виникне модальне вікно (модальне – це означає, що поки його не погасили, не можна виконувати ніякі інші дії). Щоб повернутися в модуль, треба натиснути клавішу *ОК*.

Основи мови VBA

Допустимі імена. У VBA користувач визначає імена змінних, функцій, процедур, типів, констант та інших об'єктів. Імена, які вводить користувач, повинні відображати сутність позначуваного об'єкта, щоб зробити програму такою, що легко читається. У VBA існують такі обмеження на імена:

1. Довжина імені не повинна перевищувати 255 символів.

2. В імені не може бути крапок, прогалин і таких символів: %, #, @, $, !, &.

3. Ім'я може містити будь-яку комбінацію букв, цифр, символів, які починаються з букви.

4. Не рекомендують вживати імена, які збігаються з ключовими словами VBA і іменами вбудованих функцій і процедур.

5. Імена повинні бути смисловими.

Типи даних. Типи даних належать до фундаментальних понять будь-якої мови програмування. Дані можуть бути представлені у вигляді змінних або констант.

Константи не можуть змінювати свої значення. Використання констант дозволяє легко читати та виправляти програми. Значення константи задається один раз під час оголошення константи.

Синтаксис: [Public | Private] Const Ім'яКонстанти [As Тип] = Вираз

Таблиця 7.1

Параметр	Опис
Public	Ключове слово, яке використовується на рівні модуля для оголошення констант, доступних всім процедурам у всіх модулях.
Private	Ключове слово, яке використовується на рівні модуля для оголошення констант, доступних тільки всередині модуля, в якому виконується опис.
Ім'я Константи	Ім'я константи, яке задовольняє правилам іменування
Тип	Один з типів даних розглянуто нижче
Вираз	Літерал, інша константа або будь-яка комбінація, яка включає всі арифметичні або логічні оператори, за виключенням оператора Is

Наведемо приклади числової та рядкової константи:

Private Const ПроцСтавка As Single = 0.2

Public Const As String = "abc".

Змінні використовуються для тимчасового зберігання даних в оперативній пам'яті, тобто вони ідентифікують області в пам'яті, де зберігається деяка інформація. Після того, як змінна оголошена, вона вказує на одну і ту саму область пам'яті до тих пір, поки вона не буде знищена. VBA дозволяє оголосити змінні двома способами: явно і неявно.

Явне оголошення змінної означає, що змінна повинна бути оголошена раніше, ніж її буде використано. Явно змінну можна оголосити одним з таких способів:

Dim оголошення _змінної [As тип_даних] – для оголошення локальної змінної – видимої тільки у цій процедурі, у разі виходу з процедури її значення буде втрачено.

Private оголошення _змінної [As тип_даних] – для оголошення локальної змінної – видимої у всіх процедурах одного модуля, до якого належать ці процедури, але опис такої змінної потрібно зробити у розділі *Declarations*.

Static оголошення_змінної [As тип_даних] – для оголошення локальної змінної – видимої тільки у цій процедурі, але після виходу з цієї процедури її значення не втратиться.

Public оголошення _змінної [As тип_даних] – для оголошення глобальної змінної – видимої у всіх процедурах усіх модулів. Вона може бути оголошена у стандартному модулі.

У процедурах обробки подій об’єктів форми можуть оголошуватись тільки локальні змінні, а використовуватись можуть і глобальні змінні цього проєкту.

Приклади: Dim Cost As Currency;

Public a, b, KrokZm As Single

Неявне оголошення змінної здійснюється включенням у кінець її імені спеціального символу, який встановлює тип змінної. У цьому випадку змінну нема потреби оголошувати перед тим, як її використовувати. Допустиме використання таких символів для визначення типів:

Таблиця 7.2

Суфікс	**Тип змінної**	**Приклади**
%	Integer	x%
&	Long	a&
!	Single	b!
#	Double	t#
@	Currency	Suma@
$	String	Name$

Тип даних визначає множину допустимих значень, які може приймати вказана змінна.

У VBA використовуються такі типи даних:

Таблиця 7.3

Тип змінної	**Розмір пам’яті, байти**	**Діапазон значень**
Integer	2	-32768 … 3267
Long	4	-2147483648 … 2147483647
Single	4	-3.402823E38 … 3.402823E38
Double	8	-1.79769313486232E308…1.79769313486232E308
Currency	8	-922337203685477.5808…922337203685477.5808
Boolean	2	True або False
Date	8	01.01.100 … 31.12.319999
String	До 255 символів (1 байт на 1 символ)	
Variant	не менше 16 байт	

Області видимості змінних

1. Змінні рівня процедури розпізнаються тільки у процедурі, у якій вони описані за допомогою інструкції Dim і Static. Такі змінні називаються локальними.

2. Змінні рівня модуля використовуються тільки у модулі, в якому вони описані перед процедурою за допомогою інструкції Dim або Private, але не в інших модулях такого проєкту.

3. Змінні рівня модуля, які описані за допомогою інструкції Public, є доступними для всіх процедур проєкту. Такі змінні називаються відкритими.

Час життя змінної. Змінна Private зберігає своє значення доти, поки виконується процедура, у якій вона описана. Змінні, які описані як Static, зберігають своє значення після виходу з процедури, поки працює програма.

Масиви. Масивом називається впорядкована послідовність величин, які позначені одним іменем. Впорядкованість полягає в тому, що елементи масиву розміщуються в послідовних комірках пам'яті. Щоб звернутись до елемента масиву, треба вказати ім'я масиву та індекс елемента масиву. Елементи масиву мають один і той самий тип.

Масиви бувають статичні, динамічні, одновимірні та багатовимірні.

Статичні масиви – це масиви, які мають постійний або фіксований розмір.

Синтаксис:

Dim Ім'яЗмінної(індекси) [As Тип] [,Ім'яЗмінної(індекси) [As Тип]] …

Таблиця 7.4

Параметр	**Опис**
Ім'я змінної	Ім'я змінної, яке задовольняє стандартні правила іменування змінних
Індекси	Розмірності масиву; кількість розмірностей може бути до 60 Якщо нижній індекс не заданий явно, нижня межа масиву визначається інструкцією Option Base. Якщо відсутня інструкція Option Base, нижня межа масиву дорівнює нулю за замовчуванням
Тип	Тип даних масиву

Приклади

Dim A(12) As Integer – одномірний масив цілих чисел. Розмір – 12 елементів. Причому за замовчуванням перший елемент масиву – A(0), а останній – A(11). У цьому випадку 0 – базовий індекс. Його можна змінити, написавши інструкцію Option Base 1. Тоді першим елементом масиву буде A(1).

Dim B(3, 3) As Single – двовимірний масив (матриця) дійсних чисел.

Динамічні масиви

Іноді у процесі виконання програми необхідно змінити розмір масиву. У цьому випадку масив оголошують як динамічний. Для цього при оголошенні не треба вказувати розмірність, наприклад:

Dim R() As Single

Потім у програмі треба обчислити необхідний розмір масиву в деякій змінній, наприклад *n*, і змінити розмір динамічного масиву за допомогою інструкції ReDim.

Синтаксис: ReDim [Ім'яЗмінної (Індекси) [As Тип] [,Ім'яЗмінної (Індекси) [As Тип]] …

Таблиця 7.5

Параметр	Опис
Ім'я змінної	Ім'я змінної, яке задовольняє стандартні правила іменування змінних
Індекси	Розмірності змінних масиву; допускається до 60 розмірностей. Цей параметр має синтаксис: [Нижній To] Верхній [,[Нижній To] Верхній]] … Якщо нижній індекс не задано явно, то нижня межа визначається інструкцією Option Base. Якщо відсутня інструкція Option Base, то нижня межа масиву дорівнює нулю
Тип	Тип даних масиву

Приклад 1. ReDim R(1 to 10). У цьому прикладі встановлюються межі динамічного масиву від 1 до 10.

Приклад 2

```
Dim B() As String
N = 2
M = 2
```

ReDim B(N, M)
N = 3
M = 3
ReDim(N, M)

У цьому прикладі варто застосовувати інструкцію ReDim.

Операції VBA. У програмах на VBA можна використовувати стандартний набір операцій над даними. Існує три типи операцій: математичні (виконуються над числами і їх результатом є числа), відношення (виконуються не лише над числами і їх результатом є логічні значення, наприклад x < y), логічні (застосовуються для логічних виразів і їх результатом є логічні вирази, наприклад Not x And y).

Математичні операції: +, -, *, /, \ (цілочисельне ділення), Mod (залишок від ділення), ^ (піднесення до степеня).

Операції відношення: <, >, <=, >=, <> (нерівно), =.

Логічні операції: виконуються з даними логічного типу (boolean).

Таблиця 7.6

Операція	**Опис**
[Операнд 1] And [Операнд 2]	Логічне множення (кон'юнкція)
[Операнд 1] Or [Операнд 2]	Логічне додавання (диз'юнкція)
[Операнд 1] Xor [Операнд 2]	Виключаюче Or, тобто повертає True тільки тоді, коли один операнд повертає True
Not [Операнд]	Логічне заперечення (працює тільки з одним операндом)
[Операнд 1] Equ [Операнд 2]	Логічна еквівалентність

Нехай маємо два вирази: X = (a > b) X = 3 > 2 true

Y = (c = d) Y = 4 = 5 false

Результат обчислення логічної операції залежно від значень операндів X і Y наведені у таблиці.

Таблиця 7.7

X	Y	And	Or	Xor	Eqv
True	true	true	true	false	true
True	false	false	true	true	false
False	true	false	true	true	false
false	false	false	false	false	true

Інші операції. [Рядок1] & [Рядок2] – зчеплення рядків. Для зчеплення рядків допустиме використання операції [Рядок1] + [Рядок2], але рекомендується застосовувати знак & (амперсанд).

Приклад

S1 = ”Hello”

S2 = ”Word”

S = S1 & S2

*Пріоритети (старшинство) операторів***:** арифметичні > порівняльні > логічні.

1. Арифметичні: виклик функції та дужки, ^, -(зміна знака), [*; /], \, mod, [+; -].
2. Порівняльні: однакове старшинство для всіх.
3. Логічні: Not, And, Or, Xor, Equ.

Вбудовані функції. У VBA є великий набір вбудованих функцій і процедур, використання яких суттєво спрощує програмування. Ці функції можна розподілити на такі основні категорії: математичні функції, функції перевірки типів, функції перетворення форматів, функції обробки рядків, функції дати й часу.

Математичні функції дозволяють проводити будь-які обчислення.

Таблиця 7.8

Позначення функції		**Дія**
у математиці	**у програмі VB**	
|x|	Abs(X)	Обчислює абсолютне значення виразу X
*cos*x	Cos(X)	Обчислює косинус кута X, вираженого в радіанах
*sin*x	Sin(X)	Обчислює синус кута X, вираженого в радіанах
*tg*x	Tan(X)	Обчислює тангенс кута X, вираженого в радіанах
e^x	Exp(X)	Обчислює константу e в степені X
lnx	Log X)	Обчислює логарифм виразу X
Знак числа x	Sgn(X)	Повертає – 1, якщо X<0; 0, якщо X=0; 1, якщо X> 0
$\sqrt{x}$	Sqr(X)	Обчислює квадратний корінь з виразу X
-	Str(X)	Перетворює числове значення X на рядкову величину
-	Val(X)	Перетворює рядкову величину X на числове значення
-	Rnd(X)	Генерує випадкове число між 0 та 1
	Rnd (X)	Генерує випадкове число між 0 та 1
	Atn (X)	Повертає арктангенс X. Кут X виражається в радіанах

Таблиця 7.9

Функції обробки рядків

Функція	Дія	Приклад
Ucase	Переводить усі символи рядка у верхній регістр	Ucase (”Ivanenko”) =IVANENKO
Lease	Переводить усі символи рядка в нижній регістр	Lease (”Ivanenko”) =ivanenko
Len	Визначає довжину рядка	Len (”Ivanenko”) = 8
Right	Повертає задане число символів, починаючи від кінця рядка	Right (”Ivanenko”, 2) = ”ko”
Left	Повертає задане число символів, починаючи від початку рядка	Left (”Ivanenko”, 4) = ”Ivan”
Mid	Повертає задане число символів, починаючи від указаної позиції	Mid (”Ivanenko”, 2, 3) = ”van”
String	Повторює вказаний символ установлену кількість разів	String (4, ”$”) = $$$$
Asc	Повертає ASCII-код для вказаного символу	Asc(”R”) = 82
Chr	Повертає символ для вказаного ASCII-коду	Chr(114) = r

Функції вибору. Крім перерахованих функцій, у VBA існують також функції вибору, які дублюють деякі аспекти операторів умовного переходу.

1. iif – повертає одну з двох альтернатив.

Синтаксис: iif(Expr, Truepart, Falsepart), де Truepart – значення або вираз, який повертається, якщо Expr має тип True, Falsepart – значення або вираз, який повертається, якщо Expr має тип False.

Приклад: a = iif(b > c, b, c)

2. Choose – повертає значення, вибране зі списку параметрів.

Синтаксис: Choose(Index, Choice1 [,Choice2, … [,ChoiceN]]), де Index – числовий вираз або поле, значенням якого є число, яке лежить між 1 і числом елементів списку, Choice – вираз типу Variant, який містить один з елементів списку.

Приклад: N=3

a = Choose(n, ”1”,”2”,”3”,”4”)

У результаті виконання в a буде міститься ”3”.

Перенесення рядка. Для продовження рядка інструкцій VBA використовується конструкція <Прогалина> + <Знак підкреслення> в кінці рядка, яка забезпечує, що наступний рядок є продовженням попереднього. При цьому потрібно пам'ятати, що сам рядок не може складатись більше ніж з 1024 символів.

Приклад: if a > b then _
c = a else _
c = b

Коментарі. Текст, що міститься у програмі за символом (‘) або оператором Rem до кінця рядка праворуч, ігнорується компілятором і є коментарем. Оператор Rem може бути, як і будь-який інший оператор VBA, єдиним у рядку, а символ (‘) може стояти у будь-якому місці рядка, тобто стояти як на початку тексту, так і бути другим оператором VBA.

Розміщення декількох операторів в одному рядку. Використання знака (:) дозволяє розмістити декілька операторів в одному рядку. Відтак наступні дві конструкції еквівалентні:

x = 1
x = x + 1
та
x = 1: x = x + 1

Оператор присвоєння. Оператор присвоєння завжди включає знак (=).

Синтаксис:

Змінна (або *Константа*) = *Вираз*

Оператор присвоєння виконує вираз, який задано в правій частині, та присвоює його значення (значення виразу правої частини) змінній або константі, які вказані в лівій частині.

Приклад

x= 2
x = x + 2

Змінній x буде присвоєно значення 4.

Оператори умовного переходу. Оператори умовного переходу задають виконанням певної кількості повтору команд залежно від значення виразу.

Синтаксис: 1. *if Умова then* [*інструкція*] [*Else Інструкція_else*]

2. *if Умова1 then*

[*Інструкція*1]

[*elseif Умова*2]

[*Інструкція*2]

...

[*else*]

[*Інструкція_else*]

endif

Приклад 1. Прочитати з робочого аркуша 1 Excel число і піднести його до квадрата, якщо воно від'ємне; піднести його до куба, якщо воно додатне.

```
Public Sub tyr()
x = Worksheets("Лист1").Cells(1, 1)
If (x < 0) Then y = x ^ 2 Else y = x ^ 3
MsgBox Str(y)
End Sub
```

Приклад 2

Знайти значення x:

$$x = \begin{cases} y + (y + \cos(y))^{(1/3)} & y < -3 \\ \arccos(0.2y + 0.23) & -3 < y < 3 \\ \operatorname{arcctg}(y - e^{y}) & 3 < y < 5 \end{cases}$$

```
Public Sub tyr()
x = Worksheets("Лист1").Cells(1, 1)
If (x = -3) Or (x = 3) Or (x >= 5) Then
MsgBox "FNO"
ElseIf (x < 3) Then
y = x + Sgn(x + Cos(x)) * Abs(x + Cos(x)) ^ (1 / 3)
ElseIf ((x > -3) And (x < 3)) Then
y = Atn(0.2 * x + 0.23) / Sqr(1 – (0.2 * x + 0.23) ^ 2)
Else
y = 3.1416 / 2 – Atn(y – Exp(y))
End If
MsgBox Str(y)
End Sub
```

Оператор вибору. Якщо потрібно перевірити декілька умов, то використання оператора *if* може стати занадто складним. У таких випадках застосовують оператор вибору, який дозволяє виконати одну з декількох інструкцій залежно від значення виразу.

Синтаксис: *select case*

[*case СписокВиразів*_1]

[*Інструкція*_1]

...

[*case списокВир*азів_*n*]

[*Інструкція_n*]

[*case else*]

[*Інструкція_else*]

end select

СписокВиразів – список, що розділяється та складається з однієї або декількох форм вигляду: *Вираз, Вираз To Вираз, Is* ОператорПорівняння *Вираз*, де *Вираз* – це будь-який числовий чи рядковий вираз. Ключове слово *To* задає діапазон значень. Якщо використовують ключове слово *To*, перед ним повинно знаходитись менше значення. Ключове слово *Is* з операторами порівняння задає діапазон значень. Якщо ключове слово *Is* не вказане, воно вставляється за замовчуванням.

Інструкції – це одна або декілька інструкцій, які виконуються у тому випадку, якщо вираз збігається з деяким значенням зі списку від *СписокВиразів*-1 до *СписокВиразів-n*.

Інструкції_Else – одна або декілька інструкцій, які виконуються у тому випадку, якщо вираз не збігається ні з одним із пропозиції *Case*.

Приклад 1. Залежно від величини введеного числа відображається повідомлення, яке вказує на величину числа або діапазон, якому воно належить.

```
Public Sub iui()
x = InputBox(”Введіть  ціле число”)
Select Case x
Case 1
MsgBox ”Число дорівнює 1”
```

```
Case 2, 3
MsgBox ”Число дорівнює 2 або 3”
Case 4 To 6
MsgBox ”Число від 2 до 6”
Case Is >= 7
MsgBox ”Число не менше”
End Select
```

Приклад 2. Залежно від вибраного значення n присвоїти значення змінній x.

```
Public Sub qq()
n = 4
Select Case n
Case 1
x = 1
Case 2 To 5
x = 2
Case Else
x = 3
End Select
MsgBox x
End Sub
```

Оператор For – Next (оператор циклу). Оператор *For – Next* виконує певну кількість команд інструкцій вказану кількість разів.

Синтаксис:

For Лічильник = Початок To Кінець [*Step Крок*]
[*Інструкції*]
[*Exit For*]
[*Інструкції*]
Next [*Лічильник*]

Оператор *For – Next* повторює виконанням певної кількості команд інструкцій з вказаним *Кроком* доти, поки *Лічильник* змінюється від *Початку* до *Кінця*. Якщо *Крок* не вказаний, за замовчуванням він вважається рівним 1. Альтернативний вихід з циклу надає інструкція *Exit For*.

Приклад 1

Знайти суму елементів одномірного масиву, введених з робочого листа Excel.

```
Public Sub rtr()
Dim a(100)
n = InputBox(”Введіть кількість елементів масиву”)
For i = 1 To n
a(i) = Worksheets(”Лист1”).Cells(i, 1)
Next i
s = 0
For i = 1 To n
s = s + a(i)
Next i
MsgBox Str(s)
End Sub
```

Оператор While-Wend (оператор циклу). Використовується рідко.

Оператор *While-Wend* виконує послідовність інструкцій, поки задана умова має значення *True*.

Синтаксис:

While Умова

[*Інструкції*]

Wend

Оператор *While-Wend*, на відміну від *For – Next*, працює не задану кількість раз, а поки виконується умова.

Приклад. Використання оператору *While-Wend*

```
Public Su1b rtr()
Dim a(100)
n = Val(InputBox(”Введіть кількість елементів масиву”))
i = 0
While i <= n
i = i + 1
a(i) = Worksheets(”Лист1”).Cells(i, 1)
Wend
s = 0
i = 0
```

```
While i <= n
i = i + 1
s = s + a(i)
Wend
MsgBox Str(s)
End Sub
```

Оператор Do – Loop. Оператор *Do – Loop* повторює виконання набору інструкцій, поки умова має значення *True* (випадок *While*) або поки умова не набуде значення *True* (випадок *Until*).

Синтаксис:

Do [*While* | *Until*] *Умова* або *Do*
[*Інструкції*] [*Інструкції*]
[*Exit Do*] [*Exit Do*]

[*Інструкції*] [*Інструкції*]
[*Exit Do*] *Loop*[(*While* | *Until*) *Умова*]
Loop

У будь-якому місці структури *Do – Loop* може бути розміщена будь-яка кількість інструкцій *Exit Do*, які забезпечують альтернативні можливості виходу з циклу *Do – Loop*.

Приклад 1. Використання циклу *Do – Loop*.

```
Public Sub rtr()
Dim a(100)
n = Val(InputBox(”Введіть кількість елементів масиву”))
i = 0
Do While i <= n
i = i + 1
a(i) = Worksheets(”Лист1”).Cells(i, 1)
Loop
s = 0
i = 0
Do While i <= n
i = i + 1
s = s + a(i)
Loop
MsgBox Str(s)
End Sub
```

Приклад 2. Використання циклу *Do – Loop*.

```
Public Sub rtr()
Dim a(100)
n = Val(InputBox(”Введіть кількість елементів масиву”))
i = 0
Do
i = i + 1
a(i) = Worksheets(”Лист1”).Cells(i, 1)
Loop While i <= n
s = 0
i = 0
Do
i = i + 1
s = s + a(i)
Loop While i <= n
MsgBox Str(s)
End Sub
```

Типи процедур. У VBA використовуються процедури двох типів: Sub і Function. Процедури типу Sub можуть повертати як одне, так і декілька значень, а процедури типу Function завжди повертають тільки одне значення. Наприклад, функція $y = x^2$ набуде значення x, а повертає значення y. Будь-яка програма у VBA є процедурою типу Sub або Function.

Процедури. Процедура є самостійною частиною коду, яка має ім’я і може мати параметри, змінювати їх значення і виконувати послідовність інструкцій.

Синтаксис:

[*Private | Public*] [*Static*] *Sub Ім’я* [*СписокПараметрів*]
[*Інструкції*]
[*Exit Sub*]
[*Інструкції*]
End Sub

Таблиця 7.10

Параметр	Опис
Public	Вказує, що процедура Sub доступна для усіх інших процедур в усіх модулях.
Private	Вказує, що процедура Sub доступна для інших процедур тільки того модуля, де вона описана.
Static	Вказує, що локальні змінні процедури Sub зберігаються в проміжках часу між викликами цієї процедури.
Ім'я	Ім'я процедури Sub, яке задовольняє стандартним правилам іменування змінних.
СписокПараметрів	Список параметрів, значення яких передаються в процедуру або повертаються з процедури при її виклику. Розділителем у списку параметрів є кома.
Інструкції	Будь-яка група інструкцій, яка виконується у процедурі Sub

Виклик процедури. Виклик процедури Sub з іншої процедури можна здійснити декількома способами.

Перший спосіб виклику процедури Sub.

Ім'яПроцедури СписокФактичнихПараметрів

Таблиця 7.11

Параметри	Опис
Ім'яПроцедури	Ім'я процедури, що викликається.
СписокФактичних-Параметрів	Список параметрів, які передаються процедурі. Цей список повинен відповідати за кількістю та типом списку параметрів, які задаються у процедурі при її визначенні

Якщо необхідно використати декілька процедур з однаковими назвами, у разі їх виклику через крапку необхідно вказати ім'я модуля, де вони розміщені. А саме:

Ім'яМодуля. Ім'яПроцедури СписокФактичнихПараметрів

Другий спосіб виклику процедури Sub.

Цей спосіб здійснюється за *допомогою інструкції Call.*

Call Ім'яПроцедури (*СписокФактичнихПараметрів*)

Зверніть увагу на те, що в цьому випадку список фактичних параметрів береться в дужки. У першому способі дужки не використовувались.

Пояснення: *формальні параметри* – це такі параметри, які вказані у заголовку процедури; *фактичні параметри* – це параметри, вказані в операторі виклику процедури.

Вбудовані діалогові вікна. У проєктах VBA часто зустрічаються два типи діалогових вікон:

1. Вікна повідомлень.
2. Вікна вводу (діалогу).

Вони вбудовані у VBA, і якщо їх можливостей достатньо, то можна обійтись без проєктування діалогових вікон. Вікно повідомлень MsgBox виводить найпростіші повідомлення для користувача, а вікно вводу InputBox забезпечує введення інформації.

Процедура MsgBox. MsgBox – це скринька з повідомленнями (скорочення від Message Box). Ця процедура виводить на екран діалогове вікно, яке містить повідомлення, встановлює режим очікування натиснення кнопки користувачем.

Синтаксис:

MsgBox (*Prompt*, *Buttons*, *Title*)

Prompt – рядковий вираз, який відображає повідомлення в діалоговому вікні.

Buttons – числовий вираз, або константа, еквівалентна числовому виразу, що являє собою суму значень, які вказують число і тип кнопок, що відображаються в діалоговому вікні, основну кнопку і модальність вікна повідомлень. Значення за замовчуванням цього параметру дорівнюють нулю.

Title – вираз, який відображається у рядку заголовка діалогового вікна. Якщо цей параметр опущений, то в рядок заголовка поміщається ім’я додатка.

Значення параметра Buttons процедури MsgBox визначають інформаційні значки у діалоговому вікні

Значення параметра Buttons процедури MsgBox визначають основну кнопку в діалоговому вікні повідомлень, тобто активну кнопку

Активна кнопка буде працювати, якщо натиснемо клавішу *Enter* в діалоговому вікні.

Приклади *використання процедури MsgBox*

1. Вивести вікно повідомлень, що містить повідомлення *Hello Word*, в заголовку якого вказати своє ім'я.

```
Public Sub MyFirstProgram()
MsgBox ”Hello Word”,. ”Ольга”
End Sub
```

2. Вивести вікно повідомлень, яке містить кнопки *Да* і *Нет*, а також повідомлення *Hello Word*.

```
Public Sub MyFirstProgram()
MsgBox ”Hello Word”, vbYesNo + vbCritical
End Sub
```

1. Вивести вікно повідомлень, яке містить лише ім'я в рядку заголовка діалогового вікна.

```
Public Sub MyFirstProgram()
MsgBox ” ”,. ”Ольга”
End Sub
```

Отже, у процедурі MsgBox є обов'язковим лише перший параметр, інші необов'язкові.

Процедура MsgBox зручна для виведення тієї чи іншої інформації. Однак, якщо необхідно дізнатись, який вибір зробив користувач при натисканні кнопок з діалогового вікна, то процедуру MsgBox треба використовувати як функцію. У цьому випадку значення, яке повертає MsgBox, треба присвоїти якійсь змінній, а її параметри потрібно взяти в дужки. Отже, MsgBox може бути використана як функція і тоді вона повертає значення коду натиснутої кнопки.

Таблиця кодів кнопок

Приклад. Записати у вікно повідомлень один з кодів натиснутих кнопок: *Да, Нет, Отмена*.

Створимо вікно повідомлень, у якому будуть 3 кнопки: *Да, Нет, Отмена*, й активною другою кнопкою

```
Public Sub MyFirstProgram()
MsgBox ””, vbYesNoCancel + vbDefaultButton2
End Sub
```

Використаємо MsgBox як функцію, яка повертатиме код натиснутої кнопки з діалогового вікна

```
Public Sub MyFirstProgram()
a = MsgBox(””, vbYesNoCancel + vbDefaultButton2)
If a = 6 Then MsgBox ”6”
If a = 7 Then MsgBox ”7”
If a = 2 Then MsgBox ”2”
End Sub
```

Приклад. Розглянемо наступну програму, яка записує на робочий лист Excel Лист1 в комірку (2, 2) текст ”Hello Word”.

```
Public Sub MySecondProgram()
Worksheets(”Лист1”).Cells(2, 2) = ”Hello Word”
End Sub
```

Приклад. Прочитати з листа Excel з комірки (3, 3) число і записати його в MsgBox.

```
Public Sub MyNextProgram()
a = Worksheets(”Лист1”).Cells(3, 3)
MsgBox Str(a)
End Sub
```

Примітка. Оскільки перший параметр в MsgBox має тип рядковий, щоб побачити число, яке записане на робочому листі в комірці (3, 3), необхідно перетворити його в рядок. Для цього використовуємо функцію Str, яка детально буде описана нижче.

Приклад. Прочитати з листа Excel з комірок (3, 3), (4, 3), (5, 3) числа і записати їх в MsgBox.

```
Public Sub MyNextProgram()
a = Worksheets(”Лист1”).Cells(3, 3)
b = Worksheets(”Лист1”).Cells(3, 4)
c = Worksheets(”Лист1”).Cells(3, 5)
MsgBox Str(a) + Chr(13) + Str(b) + Chr(13) + Str(c) + Chr(13)
End Sub
```

Примітка. Для того щоб кожне число у вікні повідомлень MsgBox записувалось з нового рядка, було використано функцію Chr(13), яка діє аналогічно клавіші *ENTER*.

Параметри процедури Sub пишуться без дужок, а процедури Function – у дужках.

Функція InputBox. Функція Inputbox виводить на екран діалогове вікно, що містить повідомлення і поле вводу, дві кнопки *OK* і *Cancel*. Встановлює режим очікування вводу тексту користувачем і натиснення кнопки, а потім повертає значення типу String по натисненню кнопки *OK*, яке містить текст, введений в поле вводу. Отже, Inputbox завжди повертає текст. Щоб зробити з нього число, необхідно скористатись функціями str() або cstr(), якщо необхідно вивести дату – cdata().

Синтаксис

Inputbox(*Prompt*, *Title*, *Default*, *xPos*, *yPos*, *Helpfile*, *Context*)

Promp – рядок, який відображається як повідомлення в діалоговому вікні. Рядковий вираз *Promp* може містити декілька рядків. Для розділу рядків допускається використання символу повернення каретки Chr(13), символу переводу рядка Chr(10), або комбінації цих символів Chr(13) & Chr(10), де & – конкатенація (зчеплення) рядків. (Замість знаку & допускається використання знаку +.)

Title – рядок, який відображається в рядку заголовка діалогового вікна. Якщо цей параметр опущений, то в рядок заголовка поміщається ім’я додатка.

Default – рядок (підказка), який відображається в полі вводу. Він буде використаний за замовчуванням, якщо користувач не введе інший рядок. Якщо цей параметр опущений, то поле вводу відображається пустим.

xPos, *yPos* – координати вікна повідомлень на екрані.

Helpfile і *Context* – довідка.

Перші три параметри є обов’язковими.

Приклад

```
Public Sub MyNextProgram()
a = InputBox(”Введіть значення”, ”Ольга”, 10)
End Sub
```

Оператор виведення даних Print. Дія цього оператора полягає у виведенні результатів роботи програми у форму. Синтаксис:

[Form n]. *Print* [список елементів виведення],

де до списку елементів виведення можуть входити: константи, змінні, арифметичні вирази, послідовність символів у лапках, функції *Tab, Spc, String*. Елементи списку відокремлюються комою (,) або крапкою з комою (;). У випадку розділення комою наступна інформація друкується з нової зони (одна зона 14 позицій). У випадку розділення крапкою з комою наступна інформація друкується безпосередньо після попередньої. Якщо елементом списку є вираз, то спочатку система обчислює значення виразу, а потім друкує результат. Функція *Tab (n)* – визначає номер позиції, з якої відбувається друк. Функція *Spc (n)* – визначає кількість позицій, які треба пропустити. Функція *String (n, ”символ”)* – визначає кількість символів, які друкуються.

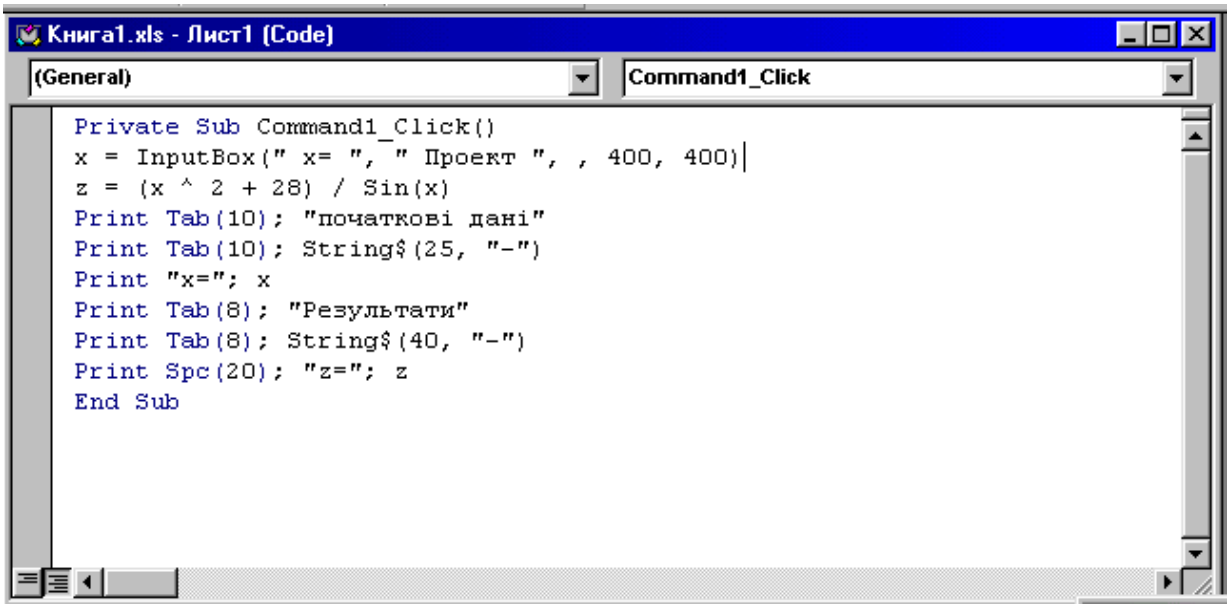


Рис. 7.1. Приклад процедури

Об’єктна модель Ms Excel являє собою ієрархію об’єктів, підпорядкованих об’єкту Application, який відповідає самому додатку Ms Excel. Більшість з цих об’єктів зібрані у бібліотеці об’єктів Ms Excel, але деякі входять до складу бібліотеки об’єктів Ms Office.

Основні об’єкти та сімейства:

Таблиця 7.12

Об’єкт	Сімейство	Опис
Application		Активний додаток Ms Excel
WorkBook	WorkBooks	Сімейство об’єктів, кожний з яких являє собою робочу книгу
Worksheet	Worksheets	Сімейство об’єктів, кожний з яких являє собою робочий аркуш цієї робочої книги
Chart	Charts	Сімейство об’єктів, кожний з яких являє собою діаграму у цій робочій книзі
Name	Names	Сімейство об’єктів, кожний з яких є ім’ям об’єкта, наприклад ім’ям робочого аркуша
Style	Styles	Сімейство об’єктів, кожний з яких є вбудованим стилем або стилем користувача, який використовується у робочій книзі
Window	Windows	Сімейство об’єктів, кожний з яких є вікном робочої книги
WorksheetFunction		Об’єкт, який являє собою контейнер для функцій робочого аркуша
Range		Діапазон комірок робочого аркуша. Приклади: Range(”A:G”) – діапазон комірок вказаного діапазону стовпців; Range(”2:2”) – діапазон комірок другого рядка; Range(”A2”) – комірка A2

Діапазон комірок може задаватись не тільки як об’єкт Range, а й з використанням функцій робочого аркуша (об’єкта Worksheet) Rows та Columns. Наприклад: Rows(4); Columns(3).

Альтернативним способом звернення до комірки робочого аркуша є властивість Cells робочого аркуша (об’єкта Worksheet). Наприклад, комірку A5 робочого аркуша можна описати як Range(”A5”) або Cells(1, 5) (комірка першого стовпця і п’ятого рядка).

Крім того, можливі такі способи опису діапазону комірок A2:C7:

1) Range(”A2:C7”);
2) Range(Cells(1, 2), Cells(3, 7).

Повне та неявне посилання на об’єкт. Повне посилання на об’єкт складається з переліку імен вкладених один в одного об’єктів. Наприклад, повне посилання на комірку А3 робочого аркуша з ім’ям *Аналіз* робочої книги *Облік* має вигляд:

Application. WorkBooks(*Облік)*.Worksheets(*Аналіз)*. Range(*А3)*

Кожного разу застосовувати таке посилання досить обтяжливо і не завжди в цьому є потреба. Натомість у багатьох випадках застосовують неявне посилання на об’єкт. Якщо посилання використовується у програмі, яка буде виконуватись у середовищі Ms Excel, то достатньо застосувати відносне посилання у вигляді

WorkBooks(*Облік)*.Worksheets(*Аналіз)*. Range(*А3)*.

Якщо вказана книга є активною, то посилання можна ще скоротити

Worksheets(*Аналіз)*. Range(*А3)*.

Якщо і вказаний робочий аркуш є активним, то посилання можна скоротити ще: Range(*А3)*.

При роботі з властивостями та методами одного й того ж об’єкта потрібно повторювати при кожному зверненні одне й те саме ім’я об’єкта. Між тим існує оператор with, який дозволяє спростити написання програми й водночас покращити читабельність тексту програми. Наприклад, замість коду

```
Range(”A3”).Value=32
Range(”A3”).Font.Italic=True
Range(”A3”).Font.Color=vbRed
```

можна записати

```
with Range(”A3”)
.Value=32
.Font.Italic=True
.Font.Color=vbRed
end with
```

або ще так

```
with Range(”A3”)
.Value=32
with Font
.Italic=True
.Color=vbRed
end with
end with
```

Цей приклад показує як виділити клітинку та помістити туди число, або формулу, або діапазон

```
Private Sub CommandButton1_Click()
‘ Вказуємо робочий файл
With Application.Workbooks.Item(”M4_1_1.xls”)
‘ Надаємо значення клітинці A2
Range(”A2”) = 2+RND
‘ Надаємо значення клітинці A3
Range(”A3”) = 3+RND
‘ Задаємо у клітинку A4 формулу
Range(”A4”) = ”=A2+A3” ‘ формула
‘ Надаємо значення діапазону
Range(”b2:b5”) = 2+RND ‘ діапазон
MsgBox ”Перейти на Лист2? ”
‘ Відкриваємо ”Лист2”
Worksheets(”Лист2”).Activate
‘ Надаємо нове значення клітинціA2 цього листа
Worksheets(”Лист2”).Range(”A3”) = 5+RND
MsgBox ”Повернутися на Лист1?”
‘ Відкриваємо ”Лист1”
Worksheets(”Лист1”).Activate
‘ Закінчуємо роботу з блоком With
End With
End Sub
```

7.2. Створення макросів, функцій користувача, форм з елементами управління у додатках Microsoft Office 2013

Макроси. Автоматизація роботи в Microsoft Excel 2013. Перед тим, як перейти до програмування на мові Visual Basic, ознайомимося з найпростішими прикладами використання можливостей Visual Basic. Таким прикладом є макроси, для створення яких мову Visual Basic знати необов'язково. Макроси використовують досить часто, вони дозволяють як спростити процес обробки документів, так і автоматизувати цей процес.

Макрос – це послідовність команд і функцій, яка зберігається у модулі Visual Basic.

Над макросами можна виконувати такі операції:

– запис або створення макросів;

– запуск макросів;

– керування макросами;

– забезпечення безпеки макросів.

Макроси часто використовують з такою метою:

– для прискорення часто виконуваних операцій редагування або форматування;

– для об'єднання складних команд;

– для спрощення доступу до параметрів у діалогових вікнах;

– для автоматизації складних послідовних дій у завданнях на редагування.

Автоматизація виконання завдань за допомогою макросів MS Word 2013. Якщо у Word необхідно багаторазово виконувати яке-небудь завдання, то найкраще це зробити за допомогою макроса. Замість того, щоб вручну робити повторювані дії, на які потрібно багато часу, можна створити та запускати один макрос, який буде виконувати ці дії.

Макроси можна створювати двома способами: за допомогою діалогового вікна *Запись макроса* й у редакторі *Visual Basic*. Перший спосіб дозволяє швидко створювати макроси при мінімумі зусиль. При цьому макрос записується у вигляді послідовності інструкцій мовою програмування Visual Basic для додатків.

Щоб змінити інструкції, макрос треба відкрити за допомогою редактора *Visual Basic*. У редакторі *Visual Basic* можна написати потужні й досить складні макроси, що не можуть бути записані першим способом.

Після присвоєння макросу кнопки панелі інструментів, команди меню або сполучення клавіш для його виконання буде достатньо вибрати команду в меню, натиснути кнопку на панелі інструментів або натиснути сполучення клавіш. Щоб запустити макрос, можна також вибрати команду *Макрос* у меню *Сервис*, команду *Макросы*, а потім – ім'я макроса, який потрібно виконати.

Макроси зберігаються в шаблонах або документах. За згодою макроси зберігаються у шаблоні *Обычный*, щоб вони були доступні всім документам. Якщо макрос, збережений у шаблоні *Обычный*, використовується тільки в документах визначеного типу, можна скопіювати цей макрос у шаблон цього типу і видалити його із шаблона *Обычный*. Для копіювання, видалення або перейменування макросів використовується діалогове вікно *Организатор*. Щоб відкрити діалогове вікно *Организатор*, треба вибрати команду *Макрос* у меню *Сервис*, а потім – команду *Макросы* і натиснути кнопку *Организатор*.

Для створення макроса достатньо активізувати режим запису і записати послідовність дій. Після цього макрос можна виконувати щоразу, коли треба виконати такий самий набір дій знову. Наприклад, якщо часто виникає необхідність вставити в документи спеціально відформатовану таблицю, то можна записати макрос, що вставляє таблицю з потрібними межами, розміром і числом рядків та стовпчиків.

При записі макроса команди та параметри можна вибирати за допомогою миші, але рухи миші у вікні документа не записуються. Наприклад, мишу не можна використовувати для виділення, копіювання, вставки й перетягування елементів у вікні документа. Для запису цих дій треба використовувати клавіатуру. Під час запису макроса можна тимчасово призупинити запис, а пізніше продовжити його з того місця, де була пауза.

Інший спосіб створити макрос – ввести його код на мові Visual Basic.

Запис макроса робиться так:

– вибрати команду *Макрос* у меню *Сервис*, потім – команду *Начать запись*;

– у поле *Имя макроса* ввести ім'я нового макроса;

– вибрати в полі *Макрос доступен для* шаблон або документ, у якому буде зберігатися макрос. За згодою макрос зберігається у шаблоні *Обычный*;

– можна ввести стислий опис макроса в поле *Описание*;

– для присвоєння макросу кнопки на панелі інструментів або команди меню натиснути кнопку *Панели*. Вибрати записуваний макрос у списку *Команды* і перетягнути його на панель інструментів або в меню. Натиснути кнопку *Закрыть* для запису макроса;

– щоб присвоїти макросу сполучення клавіш, треба натиснути кнопку *Клавишам***.** Вибрати записуваний макрос у списку *Команды*, ввести сполучення клавіш у поле *Новое сочетание клавиш*, а потім натиснути кнопку *Закрыть* для запису макроса;

– натиснути кнопку OK, щоб почати запис макроса. Під час запису Word виконує такі дії:

– виводить поруч із покажчиком символ касети;

– виділяє напівжирним шрифтом індикатор ЗАП (REC) у рядку стану;

– відображає на екрані панель інструментів запису макроса;

– виконує дії, які треба включити в макрос. При цьому дії, зроблені у вікні документа за допомогою миші, не записуються. Для запису таких дій, як виділення тексту або переміщення курсору, необхідно використовувати клавіатуру. Миша може бути використана для вибору потрібних команд і параметрів;

– щоб закінчити запис макроса, натиснути кнопку *Остановить запись* на панелі інструментів запису макроса.

Якщо новому макросу буде присвоєне ім'я макроса, що вже існує, то останній буде замінений. Щоб переглянути список вбудованих макросів Word, вибрати команду *Макрос* у меню *Сервис*, а потім – команду *Макросы***.** Вибрати *Команду Word* зі списку *Макросы из….*

Перед записом макроса заплануйте заздалегідь кроки та команди, які він повинен виконати. Якщо під час запису буде допущена помилка, то в макросі також запишуться усі виправлення. Непотрібні інструкції можна буде знищити пізніше шляхом редагування макроса.

Постарайтеся угадати всі повідомлення, що може вивести Word. Наприклад, якщо в макрос входить команда закриття документа, то Word запропонує зберегти внесені в документ зміни. Щоб це повідомлення не виводилося, під час запису макроса треба зберегти документ перш ніж закрити.

Перед тим як використовувати макрос, записаний в іншому документі, треба переконатися, що він не залежить від змісту цього документа.

Припинення і поновлення запису макроса:

– щоб припинити запис макроса, треба натиснути кнопку *Пауза* на панелі інструментів запису макроса;

– виконати дії, що не повинні бути записані;

– для поновлення запису натиснути кнопку *Возобновить запись*.

Написання макроса на мові Visual Basic:

– вибрати команду *Макрос* у меню *Сервис*, а потім – команду *Макросы*;

– вибрати у списку *Макросы из* шаблон або документ, у якому буде зберігатися макрос;

– у поле *Имя макроса* ввести ім'я нового макроса;

– натиснути кнопку *Создать*, щоб викликати редактор *Visual Basic*.

Видалення макроса:

– вибрати команду *Макрос* у меню *Сервис*, а потім – команду *Макросы*;

– вибрати ім'я макроса, що потрібно видалити, у списку *Имя*;

– якщо імені потрібного макроса немає в списку, вибрати у полі *Макросы из* інший список макросів;

– натиснути кнопку *Удалить*.

Щоб видалити відразу кілька макросів, треба виділити їх у списку *Имя*, утримуючи натиснутою клавішу CTRL, а потім натиснути кнопку *Удалить*.

Виконання макроса:

– вибрати команду *Макрос* у меню *Сервис*, а потім – команду *Макросы*;

– вибрати ім'я макроса, що потрібно виконати, у списку *Имя*;

– якщо потрібного макроса немає в списку, вибрати у полі *Макросы из* інший список макросів;

– натиснути кнопку *Выполнить*.

Зміна макроса:

– вибрати команду *Макрос* у меню *Сервис*, а потім – команду *Макросы*;

– вибрати ім'я макроса, що потрібно відредагувати, у списку *Имя*;

– якщо імені потрібного макроса немає в списку, вибрати у полі *Макросы из* інший список макросів;

– натиснути кнопку *Изменить*.

Описані дії дозволяють редагувати текст макроса. Відкривається вікно з описом макроса в редакторі Visual Basic, за допомогою якого можна внести необхідні виправлення, видалити непотрібні дії або додати інструкції, що не можна записати звичайним способом.

Копіювання макроса

Для копіювання макроса в інший шаблон або документ використовується діалогове вікно *Организатор*:

– вибрати команду *Макрос* у меню *Сервис*, а потім – команду *Макросы*;

– натиснути кнопку *Организатор*;

– вибрати потрібні макроси в одному зі списків, а потім натиснути кнопку *Копировать*.

Макроси, що використовуються в активному документі, виводяться у списку ліворуч, а стилі в шаблоні *Обычный* – у списку праворуч.

Перейменування макросів:

– вибрати команду *Шаблоны и надстройки* в меню *Сервис*;

– натиснути кнопку *Организатор*;

– вибрати вкладку *Макросы*;

– у лівому списку вибрати елемент, що треба перейменувати, а потім натиснути кнопку *Переименовать*;

– у діалогове вікно *Переименование* ввести нове ім'я елемента;

– натиснути кнопку *ОК*, а потім – кнопку *Закрыть*.

Автоматизація виконання завдань за допомогою макросів MS Excel 2013.

Досить часто при роботі з Microsoft Excel 2013 зустрічаються ситуації, коли потрібне виконання деяких рутинних дій, що повторюються. Тобто іноді доводиться у деяких клітинках форматувати певним чином текст, будувати масиви, вводити складні формули тощо. Такі процеси можна автоматизувати за допомогою макроса.

Перед тим як написати макрос, необхідно спланувати кроки та команди, що він буде виконувати. Якщо при записі макроса була допущена помилка, її виправлення буде також записано. Щоразу при записі макроса, він зберігається у новому модулі, приєднаному до книги. Редактор *Visual Basic* дозволяє змінювати макроси, а також копіювати їх або з одного модуля в інший, або між різноманітними книгами. Крім того, можна перейменовувати модулі, у яких зберігаються макроси, або перейменовувати самі макроси.

Запис макросів. При записі макроса Microsoft Excel 2013 зберігає інформацію про кожний крок виконання послідовності команд. Наступний запуск макроса викликає повторення (відтворення) команд. Якщо при записі макроса була допущена помилка, зроблені виправлення також будуть записані. Кожний макрос зберігається в окремому модулі, прикріпленому до книги. Запис макросів відбувається за допомогою редактора *Visual Basic*.

Запуск макросів. Макроси запускаються різними способами. Макрос можна запустити, вибравши його зі списку в діалоговому вікні *Макрос*. Можна зробити так, щоб макрос запускався після кожного натискання визначеної кнопки або сполучення клавіш, для цього необхідно призначити макросові кнопку на панелі інструментів, сполучення клавіш або графічний об'єкт на листі.

Керування макросами. Після запису макроса код програми можна подивитися в редакторі *Visual Basic* для виправлення помилок і внесення необхідних змін у макрос. Наприклад, якщо макрос, що переносить текст за словами, виділяє текст напівжирним

шрифтом, то можна записати інший макрос, що виділяє текст напівжирним шрифтом, і скопіювати інструкції з цього нового макроса в старий макрос, що переносить текст за словами.

Редактор *Visual Basic* розроблений для спрощення написання і редагування програм користувачами-початківцями та надає велику оперативну довідку. Для внесення найпростіших змін у макроси нема потреби вивчати мову програмування. Редактор *Visual Basic* дозволяє редагувати макроси, копіювати їх з одного модуля в інший, копіювати з інших книг, перейменовувати модулі, що зберігають макроси, й перейменовувати макроси.

Безпека макросів. Microsoft Excel 2013 забезпечує захист від комп'ютерних вірусів, що можуть передаватися макросами. Якщо макроси використовуються разом з іншими користувачами, вони можуть бути позначені цифровим підписом – електронним підписом, що підтверджує дійсність макроса або документа. Наявність цифрового підпису підтверджує, що макрос або документ був отриманий від власника підпису і не був змінений. Отже, користувачі можуть перевірити надійність джерела макросів. Щоразу при відкритті книги, що містить макроси, можна перевірити джерело макросів.

Створення функції користувача. У загальному вигляді функція користувача має такий вигляд:

Синтаксис:

Function Ім'яФункції (*СписокПараметрів*)

[*Інструкції*]

End Function

СписокПараметрів – це список параметрів, від яких залежить функція. Розділителем у списку параметрів є кома.

Інструкції – це послідовність інструкцій, які виконуються при знаходженні значення функції. У сукупності вони становлять тіло функції. Важливою особливістю функції користувача є те, що носієм значення, що повертає функція, є ім'я функції, тобто обов'язково результат обчислення повинен бути присвоєний імені функції. Тому серед інструкцій має бути присутнім оператор, який присвоює імені функції значення якогось виразу.

Для створення функцій користувача потрібно виконати дії:

- якщо в проєкті немає модуля, то створити його можна, виконавши команду меню редактора *VB Вставка/Модуль*;
- виконати команду меню редактора *VB Вставка/Процедура*;
- у діалоговому вікні, що відкрилося, *Вставка процедури* встановити перемикач *Функція*;
- у вікні *Ім'я* ввести ім'я функції;
- встановити відповідний перемикач *Область визначення;*
- клацнути на *OK*. Після виконання цих дій у вікні редагування модуля з'явиться автоматично заголовок і закінчення, між якими потрібно помістити код тіла функції;
- ввести список параметрів функції, їх типів даних, а також вказати тип повернутого функцією значення;
- використовуючи команду меню *Вид/Перегляд об'єктів* або натиснувши клавішу F2, викликати вікно *Перегляд об'єктів;*
- розкрити список верхнього лівого вікна (Список проєктів) і вибрати з нього *VBA Project*. У вікні *Класи* відобразяться елементи поточного проєкту;
- вибрати в цьому вікні модуль, в якому створена функція, – у вікні *Компоненти* відобразяться елементи, які містяться у цьому модулі;
- виділити у вікні *Компоненти* елемент з ім'ям створеної функції та включити контекстне меню;
- виконати команду контекстного меню *Властивості* – відкриється вікно *Параметри компонента*.
- у полі *Опис* цього вікна ввести текст короткого опису функції; якщо необхідно, то вказати файл довідки й ідентифікатор.

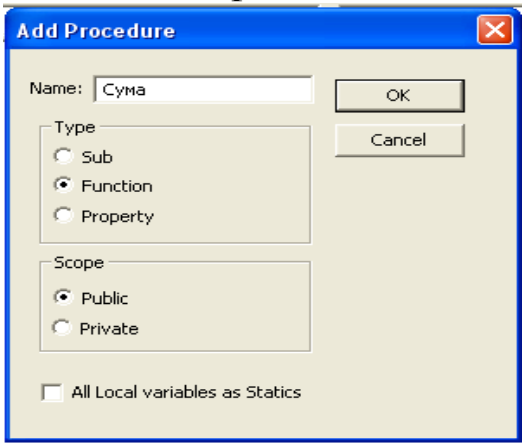


Рис. 7.2. Послідовність створення процедури. Крок 1

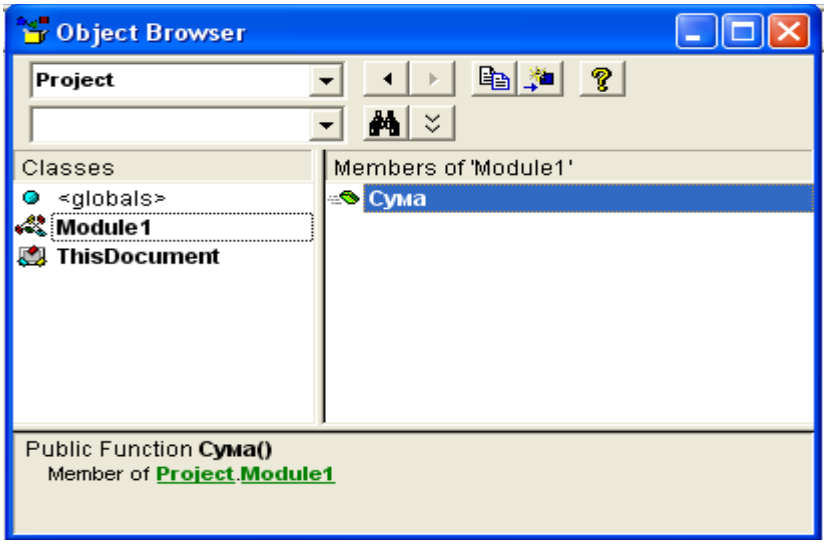


Рис. 7.3. Послідовність створення процедури. Крок 2

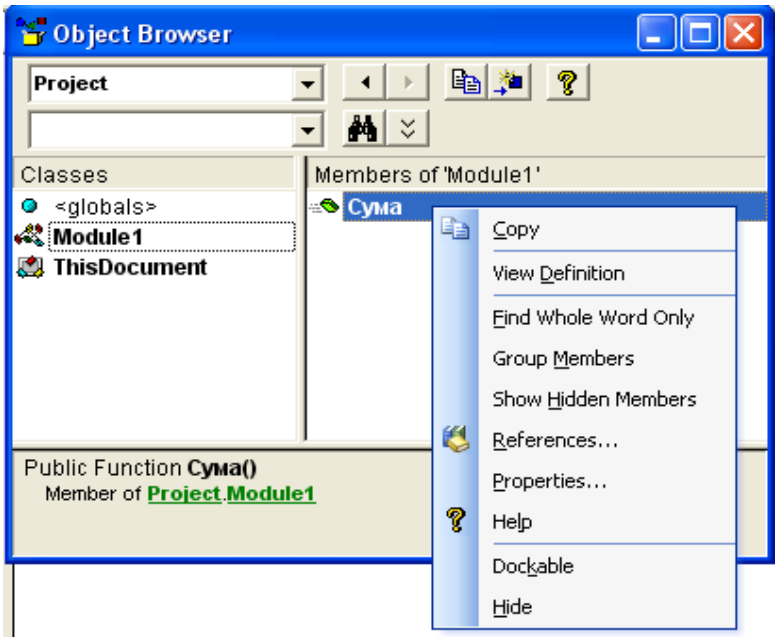


Рис. 7.4. Послідовність створення процедури. Крок 3

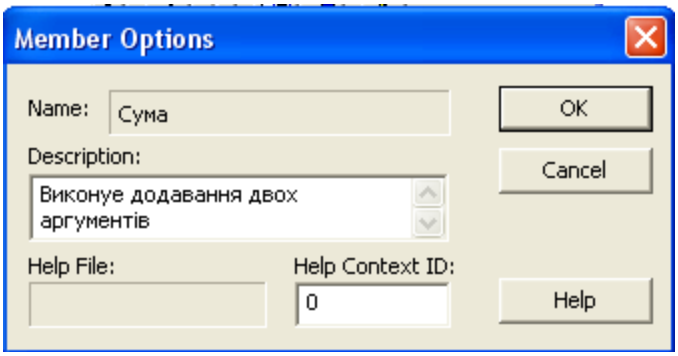
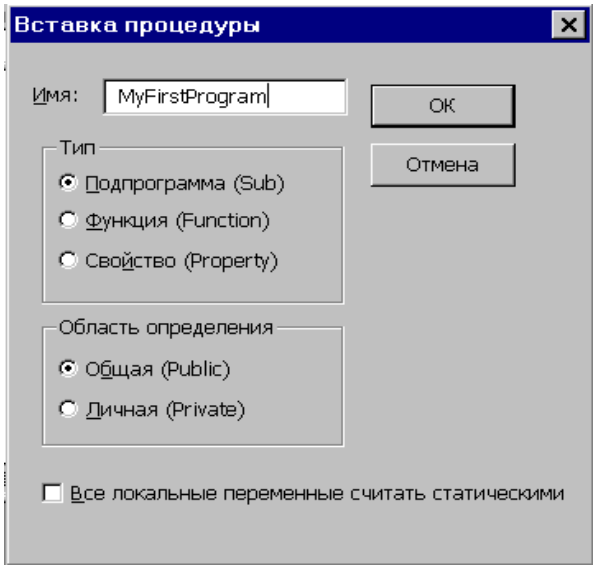


Рис. 7.5. Послідовність створення процедури. Крок 4

У VBA кожний робочий лист має модуль. Щоб додати його в проєкт, необхідно вибрати команду: *Вставка, Модуль*. У результаті з'являється робочий лист модуля. Для створення програм потрібно скористатись: *Вставка, Процедура*.



Даємо ім'я процедури: MyFirstProgram, вказуємо тип процедури, наприклад Sub, натискуємо *ОК*.

На листі отримуємо:

Public Sub MyFirstProgram()

[інструкції]

End Sub

Наша перша програма буде виводити на екран текст Hello Word. Для цього напишемо таку інструкцію: MsgBox "Hello Word".

Існують два способи запуску програми:

1. Встановити курсор у будь-яке місце програми та натиснути функціональну клавішу *F5*.

2. Натиснути з будь-якого місця програми піктограму пуск.



Виникне модальне вікно (модальне – це означає, що поки його не закрили, не можна виконувати ніякі інші дії). Щоб повернутися у модуль, треба натиснути клавішу *ОК*.

Вікно конструктора форми (Form). У вікні конструктора форми візуально конструюються елементи додатка. Для точного розміщення об'єктів у формі слугує сітка. При виконанні додатка вона не відображається. За необхідності її можна відключити, скориставшись параметрами діалогового вікна *Options*, що відкривається командою *Options* (Параметри) з меню *View* (Вид).

Розмір форми у вікні можна змінювати, використовуючи маркери виділення форми та мишу. Для зміни розміру форми необхідно встановити покажчик миші на маркер розміру і, коли він набуде вигляду двонапрямленої стрілки, пересувати його до отримання потрібного розміру.

Панель об'єктів управління (General).



Рис. 7.6. Панель об'єктів управління (General)

Елементи управління додатком під час проєктування форми вибираються з об'єктів панелі управління.

Об'єкт – елемент інтерфейсу, який створюється в екранній формі VB за допомогою засобів панелі управління.

Об'єкт характеризується *властивостями*, *методами* та *подіями*.

Подія – це характеристика об'єкта, що описує зовнішній вплив, на який реагує об'єкт при застосуванні додатка. Події – це, наприклад, завантаження форми, натиснення клавіші на клавіатурі, клацання миші, подвійне клацання миші на об'єкті тощо. Кожна подія має своє ім'я.

Є дві події, найбільш поширені при використанні миші:

Click – подія, яка виникає, коли користувач клацає кнопкою миші на об'єкті;

Dbclick – подвійне клацання на елемент, керування.

Ще одна характеристика об'єкта – *метод*.

Метод – це дія, яка може здійснюватися над об'єктом. Метод – це також процедура.

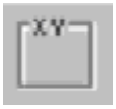
Елементи панелі управління:

1.  *Pointer (Покажчик)* – для позиціювання покажчика миші.

2.  *PictureBox (Графічне вікно)* – розміщує у формі графічне вікно, призначене для об'єднання елементів у групи, вставки в нього графічних об'єктів, тексту, анімації та ін.

3.  *Label (Напис)* – розміщує у формі текстові об'єкти-написи, редагувати які під час виконання додатка неможливо.

4.  *TextBox (Текстове поле)* – розміщує у формі поле для введення текстової інформації, чисел, дат.

5.  *Frame (Рамка)* – створює у формі рамку із заголовком для логічного групування об'єктів.

6.  *CommandButton (Командна кнопка)* – розміщує у формі кнопки управління для ініціювання дій, виконання команд тощо.

7.  *CheckBox (Прапорець)* – розміщує у формі індикаторний перемикач, призначений для формування умов виконання чи певних налаштувань, значення якого (Value) при встановленому прапорці – 1; за відсутності прапорця – 0; позначений, але неактивний стан – 2.

8.  *OptionButton (Перемикач)* – створює у формі альтернативний перемикач для вибору налаштування чи режиму виконання програми.

9.  *ComboBox (Поле зі списком)* – створює у формі об'єкт, що містить одночасно поле введення тексту і список значень, введений текст додається до елементів списку.

10.  *ListBox (Список)* – створює у формі список для вибору одного чи декількох значень із запропонованого списку.

11.  *HScrollBar (Горизонтальна смуга прокручування)* – розміщує у формі горизонтальну смугу прокручування.

12.  *VScrollBar(Вертикальна смуга прокручування)* – розміщує у формі вертикальну смугу прокручування.

13.  *Timer (Таймер)* – розміщує у формі таймер.

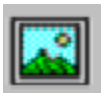
14.  *DriveListBox (Список пристроїв)* – створює у формі список пристроїв.

15.  *DirListBox (Список папок)* – створює у формі деревоподібний список папок.

16.  *FileListBox (Список файлів)* – створює у формі список файлів.

17.  *Shape (Фігура)* – створює у формі геометричні фігури: прямокутник, квадрат, коло, еліпс, прямокутник і квадрат із заокругленими кутами.

18.  *Line (Лінія)* – створює лінії.

19.  *Image (Зображення)* – створює у формі поля для відображення графічних зображень.

20.  *Data (Дані)* – створює елементи управління даними в базі даних, для переміщення по записах і відображення результатів навігації.

21.  *OLE* – створює у формі поля для розміщення об’єктів за допомогою OLE-технологій.

Об’єкти управління на формі створюються таким чином:

1 спосіб. Потрібно вибрати потрібний елемент на панелі елементів управління і клацнути на ньому. У межах форми курсор миші прийме вигляд хрестика.

«Намалювати» на формі об’єкт, утримуючи натиснутою ліву кнопку миші.

2 спосіб. Подвійним клацанням на значку об’єкта на панелі елементів управління. Об’єкт з’явиться у формі, у місці та з розмірами, визначеними за замовчуванням.

Вікно властивостей форми (Properties: Form1) та її об’єктів. Вікно властивостей *Properties* призначене для відображення і задання властивостей форми, а також розміщених у ній об’єктів.

За допомогою вкладок *Alphabetic (За алфавітом)* і *Categorized (По категоріях)* властивості об’єкта можна переглянути за абеткою чи по групах (категоріях) відповідно. У цьому

вікні можна проглянути властивості не тільки вибраного об'єкта: вибрати інші можна зі списку, розміщеного під заголовком вікна. Списки властивостей складаються з двох стовпців: у першому з них вказано назву властивості, у другому – її значення.

Для зміни значень передбачені різні способи залежно від різновиду властивості: введення нових значень може відбуватись як у текстовому полі (наприклад назва форми – у полі *Name*), вибором зі списку (наприклад колір підкладки форми – у списку *BackColor*), а іноді й з діалогового вікна (наприклад шрифт – *Font*).

Після виділення властивості у нижній частині вікна з'являється пояснення призначення обраної властивості.

Якщо вибрати одразу кілька об'єктів, то у вікні властивостей відобразяться загальні для цих об'єктів властивості.

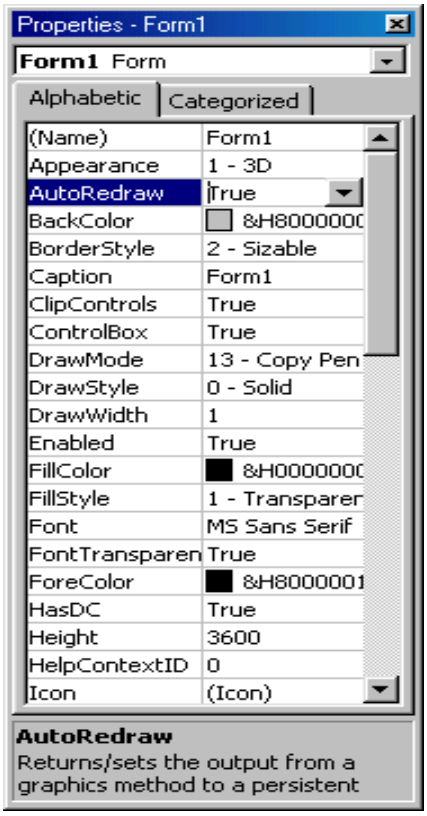


Рис. 7.7. Властивості форми

Наприклад, усі без винятку об'єкти форми мають властивість *Name* (Ім'я), яка використовується при написанні програмних кодів. Ім'я об'єкта повинне бути унікальним у формі. Розміри об'єкта визначаються властивостями *Height* (Висота) і *Width* (Ширина). Задання положення об'єкта у формі визначають властивості *Left* (відстань від лівого краю форми) і *Top* (відстань від верхнього краю форми).

Розглянемо основні групи властивостей, згруповані за типом на вкладці *Categorized* вікна *Properties*.

У групі оформлення *Appearance* містяться властивості об'єкта, які задають атрибути його зовнішнього вигляду. Основні властивості цієї групи такі:

Caption – задає текст в рядку заголовка об'єкта;

BorderStyle – задає стиль рамки об'єкта;

Palette – встановлює колірну палітру;

Picture – призначає значок, картинку для об'єкта. Можна задати фонове графічне зображення.

Властивості об'єктів, що характеризують їх поведінку, зібрані в групі *Behavior***.** Основні властивості цієї групи такі:

Causes Validation – встановлює ознаку перевірки умови достовірності даних при виході з об'єкта;

Enabled – дозволяє або забороняє доступ до об'єкта;

MaxLength – встановлює максимальну довжину даних в об'єкті;

Visible – встановлює видимість об'єкта.

Група *Font* має одну властивість – *Font*, що викликає діалогове вікно для задання шрифту тексту об'єкта, його розміру, стилів накреслювання.

Властивості групи *Misc* задають загальні атрибути об'єкта. Основні властивості цієї групи такі:

Name – задає ім'я об'єкта;

Text – встановлює текст у полі за умовчанням;

Index – задає унікальний індекс об'єкта.

Група *Position* слугує для позиціювання об'єктів у системі координат форми та встановлення їх розмірів. Основні властивості цієї групи такі:

Left – задає положення об'єкта по горизонтальній осі від лівого краю форми;

Top – задає положення об'єкта по вертикальній осі від його верхнього краю до верхньої сторони форми;

Width – задає горизонтальний розмір (ширину) об'єкта,

Height – задає вертикальний розмір (висоту) об'єкта;

Властивості групи *Scale* встановлюють максимальні масштаби об'єктів у системі координат форми. Основні властивості цієї групи такі:

ScaleLeft – задає максимальне положення об'єкта по горизонтальній осі.

ScaleTop – задає максимальне положення об'єкта по вертикальній осі.

Scale Width – задає максимальний горизонтальний розмір (максимальна ширина).

ScaleHeight – задає максимальний вертикальний розмір (максимальна висота).

Вікно *Редактор коду*. Властивості об'єктів допускається змінювати при виконанні додатка, написавши відповідні коди в процедурах, що створюються за допомогою редактора коду.

З елементом управління, як правило, пов'язують певні події. Для того щоб додаток працював (як нам потрібно), необхідно запрограмувати процедури обробки певних подій. Це здійснюється у вікні редактора коду. Вікно редактора коду можна активізувати подвійним кліком на об'єкті управління.

Розташування форми на екрані при виконанні визначається властивостями *Left* і *Top*, що вказують відстані від лівого і верхнього краю відповідно. Крім цього, для задання розташування форми під час виконання додатка використовується властивість *StartUpPosition*, яка може мати такі значення:

Manual – положення форми задають властивості *Left і Top*;

CenterOwner – форма розташовується у центрі робочого стола Windows;

CenterScreen – форма розташовується у центрі екрана;

WindowsDefault – положення форми задається системою, виходячи з кількості відкритих вікон і їх розташування.

Форма, як і будь-яке інше вікно *Windows*, може містити з правого боку заголовка кнопки управління її розмірами. Статус цих кнопок визначають властивості *MaxButton і MinButton*. Властивість *MaxButton* визначає, чи буде кнопка максимізації активною при запуску форми на виконання, а властивість *MinButton*,

відповідно, визначає активність кнопки мінімізації форми. Властивість *ControlBox* форми визначає, чи буде під час виконання форми в рядку заголовка розташовуватися кнопка виклику системного меню, в якому присутні команди *переміщення, зміни розміру, закриття форми*.

Запитання для самоконтролю

1. Що таке VBA?
2. Назвіть основні об'єкти VBA.
3. Які типи модулів розрізняють у VBA?
4. Що розуміється під *властивістю* і *методом об'єкта?*
5. Що є *подією* і *повідомленням?* Що може бути відправником повідомлення? Які об'єкти можуть виступати як одержувач повідомлення?
6. Що є подіями GotFocus і LostFocus, який синтаксис процедур їх обробки?
7. Назвіть основні події форми. У яких випадках вони відбуваються?
8. Назвіть основні події миші. У яких випадках вони відбуваються?
9. Назвіть основні події клавіатури. У яких випадках вони відбуваються?
10. Які символи можна застосовувати в іменах змінних?
11. Чи можна ім'я змінної починати з цифри або з символу підкреслення?
12. Які існують угоди зі стилю імен?
13. Які базові типи даних підтримує VBA?
14. Напишіть оператора, що чітко декларує рядкову змінну для зберігання поштової адреси.
15. Який оператор має бути присутній у розділі глобальних оголошень, щоб унеможливити помилки у разі появи неявно оголошених змінних?
16. Напишіть оператора, який декларує одновимірний масив із 7 елементів цілого і базовий індекс, що має, – 3.

17. Що таке динамічні масиви і як вони декларуються?

18. Яка відмінність між змінною і константою? Які типи констант підтримує VBA?

19. Що є переліками, для чого вони застосовуються?

20. Що є призначеним для користувача типом даних, як він визначається і оголошується?

21. Яка відмінність між лінійним і блоковим оператором If?

22. Як визначається цикл з відомим числом повторень?

23. Для яких цілей слугує цикл For Each, який його синтаксис?

24. Які цикли з умовою підтримує VBA?

25. У чому відмінність циклів з передумовою й післяумовою?

26. Що таке функція і як вона визначається?

27. Як проводиться звернення до функції?

28. Які типи вбудованих функцій є у VBA?

29. Як створюються функції користувача?

30. Що таке аргументи функції та значення функції?

31. Що таке формальні та фактичні параметри функції? У чому полягає між ними різниця?

32. Де і як вказуються типи параметрів, функції, що передаються?

33. Значення яких типів можуть повертати функції користувача?

34. Яку структуру має директива, яка оголошує початок функції користувача?

35. Який синтаксис має інструкція виклику певної функції в робочих листках MS Excel 2013? Який синтаксис має інструкція виклику певної функції у програмних кодах процедур обробки певних дій елементів керування MS Excel 2013?

36. Яким чином визначається значення, яке повертає функція?

37. Яке повідомлення показується у комірці робочого листка, якщо допущена помилка в написанні назви функції, що використовується при обчисленні значення цієї комірки?

ЛЕКЦІЯ 8
ПРОГРАМУВАННЯ ТИПОВИХ АЛГОРИТМІВ У СЕРЕДОВИЩІ MS OFFICE 2013

8.1. Приклад розроблення форми засобами VBA *Анкета студента*

Етапи розробки додатка:

Розробка інтерфейсу користувача

Розробимо форму з назвою *Анкета*.

1. Завантажимо VBA. Створимо форму Insert – UserForm.

2. Поміняємо колір фону форми(BackColor) і дамо назву *Анкета* (Caption).

3. Вставимо у форму текстове поле (об'єкт типу Label) з текстом "*Анкета студента*".

Задамо значення властивості AutoSize цього об'єкта True і властивості Font (шрифт) цього поля такі:

Font : Times New Roman Cyr (або MS Sans Serif).

Font style: Bold; : 14.

Задайте колір підпису, вказавши значення властивості ForeColor на рис. 8.1, рис. 8.2.

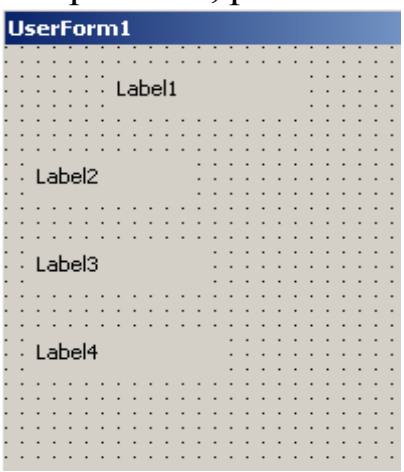


Рис. 8.1

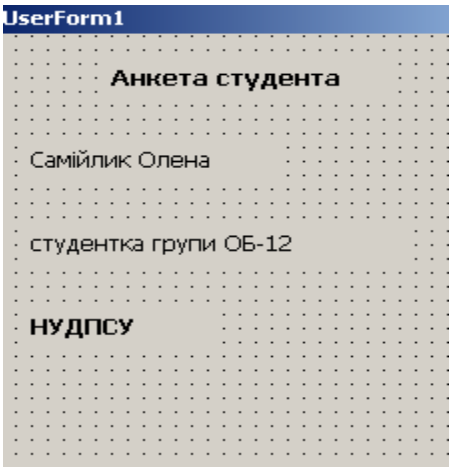


Рис. 8.2

4. Аналогічно вставляємо у форму ще декілька текстових полів із біографічними даними.

5. Вставимо у форму об'єкт типу Image (малюнок) художню і портретну фотографії.

Для цього клацніть один раз лівою клавішею миші по піктограмі Image – палітри компонентів і, наприклад, у нижньому правому кутку форми обведіть контур майбутнього зображення (фотографії). У разі потреби поміняйте розмір форми або вставленого об'єкта для якнайкращого розташування на ній створених раніше об'єктів. Міняти розміри об'єкта можна методом їх «розтягання» за маркери (чорні габаритні квадратики). За замовчуванням цьому об'єкту буде привласнено стандартне ім'я Imagel.

Нехай портретна фотографія має назву Imagel, а художня фотографія Image2.

6. Задамо значення False властивості Visible для обох зображень.

7. Вставимо у форму кнопки для почергового відображення фотографій – два об'єкти типу CommandButton з іменами Commandl і Command2.

Піктограма об'єкта типу CommandButton (кнопка) знаходиться на палітрі компонентів Visual Basic. Зробіть підписи на кнопках (властивість *Caption*) "Портретна фотографія" і "Художня фотографія" відповідно. Виберіть найкращий, на вашу думку, шрифт для підписів кнопок. Якщо використовуєте картинки зі стандартної бібліотеки Clipart, виберіть для кнопок цікаві підписи. Один із варіантів розташування кнопок зображений на рис. 8.3, рис. 8.4.



Рис. 8.3



Рис. 8.4

8. Запрограмуйте кнопку "Портретна фотографія" так, щоб після натискання на неї у формі з'являлася портретна фотографія.

Для програмування кнопки *Commandl* необхідно двічі клацнути на ній лівою клавішею миші. У результаті активізується вікно тексту програми із заготівкою процедури CommandlClick, яка оброблятиме подію клацання на кнопці *Command 1*:

```
Private Sub Commandl_Click()
...
End Sub
```

У заготівку необхідно вставити текст програми реакції на цю подію. Процедура матиме такий вигляд:

```
Private Sub Commandl_Click()
Imagel.Visible = True
‘Портретна фотографія стає видимою
Image2.Visible = False
‘Художня фотографія стає невидимою
End Sub
```

За допомогою цієї процедури властивість видимості об’єкта Imagel включаємо, і цією ж властивість об’єкт Image2 вимикаємо. Для кнопки *Художня фотографія* дії будуть протилежні. Зверніть увагу на використання складених імен типу Imagel.Visible, у яких ім’я об’єкта від його властивості відділяється крапкою. Такі складені імена дають доступ до значення конкретної властивості об’єкта. Після введення з клавіатури точки Visual Basic пропонує програмісту список властивостей, методів і подій для даного об’єкта. Вибір потрібної властивості здійснюється клавішами зі стрілками, а підтвердження – пропуском. Крім того, ім’я потрібної властивості або методу можна безпосередньо набрати на клавіатурі.

9. Запрограмуйте кнопку *Художня фотографія* відповідно до її призначення (пункт 8).

Текст процедури для цієї кнопки матиме вигляд:

```
Private Sub Command2_Click()
Imagel.Visible = False
‘Портретна фотографія стає невидимою
Image2.Visible = True
‘Художня фотографія стає видимою
End Sub
```

10. Запускаємо програму і переконуємося, що кнопки виконують свої функції. Закрийте вікно програми *Анкета студента* і виходьте з VBA.

8.2. Автоматизований розрахунок обміну валют

Об'єкти типу TextBox використовують для введення у рядки символів із клавіатури. Крім відомих уже властивостей, поля редагування TextBox володіють такими властивостями:

Таблиця 8.1

Властивість	Опис властивості	Приклади значень
PasswordChar	Символ для введення пароля	Порожній рядок (пряме відображення тексту) * (текст відображатиметься зірочками)
ToolTipText	Текст підказки, який з'являється при наведенні курсора миші	«Введіть суму» (довільний рядок символів)
Text	Текст у полі редагування	«0,0001» (довільний рядок символів)

Об'єкти типу OptionButton призначені для створення у формі способу для вибору однієї альтернативної можливості з декількох. Розглянемо такі властивості перемикачів:

Таблиця 8.2

Властивість	Опис властивості	Приклади значень
Value	Стан перемикача	True (вибраний), False (не вибраний)
Tablndex	Порядок вибору об'єкта клавішею	Tab0 (перший), 4 (п'ятий)
TabStop	Доступ до цього об'єкта табулятором	True (буде доступним), False (не доступний)

1. Відкрийте редактор VBA і створіть форму Insert – UserForm.

2. Відмовтеся від можливості міняти розміри вікна програми, вказавши значення властивості форми BorderStyle – Fixed Dialog.

Задаючи це значення, виконайте програму і переконайтеся, що не можна поміняти розмір форми. Зверніть увагу на відсутність у вікні кнопок мінімізації й максимізації. Закінчіть роботу програми.

3. Вставте у форму два об’єкти типу OptionButton (перемикачі), як показано на рис. 6.

Для цього двічі клацніть на піктограмі об’єкта типу OptionButton (перемикач) на палітрі компонентів і помістіть його в потрібному місці форми. Повторіть ці дії, щоб вставити другий перемикач.

4. Задайте початкове значення правого перемикача як активне.

Для цього клацніть по правому перемикачі його властивості та значення Value (контроль вибору) задайте True.

5. Вставте у форму два поля редагування – об’єкти Textl і Text2.

Для цього клацніть на піктограмі об’єкта типу TextBox (поле редагування) палітри компонентів, а потім обведіть контур цього об’єкта на формі. Вставте другий об’єкт.

Запустіть програму і поекспериментуйте зі вставленими об’єктами: клацніть в полі редагування, введіть будь-яке число, видаліть його. Закрийте вікно програми.

6. Розташуйте у формі два текстові поля – об’єкти Labell і Label2.

7. Вставте у форму два поля редагування – об’єкти Text3 і Text4.

8. Вставте у форму дві кнопки – об’єкти типу CommandButton.

9. Вставте у форму ще два текстові поля – об’єкти Label3 і Label4 (рис. 8.5).

10. Збережіть створену форму.

11. Змініть назву форми з ”Form1” на ”Обмін валюти”.

Для цього змініть значення властивості Caption форми. Клацніть на формі і на рядку Caption у вікні Properties введіть назву форми без лапок. *Зверніть увагу на те, що для об’єктів деяких типів (зокрема Label. Form. CheckBox і ін.) значення властивостей Caption і Name збігаються.*

12. Поміняйте підписи Caption до об’єктів типу OptionButton, Label і CommandButton так, як зображено на рис. 8.6.

Для цього по черзі вибирайте об’єкти (клацайте на них) і міняйте значення властивості *Caption*.

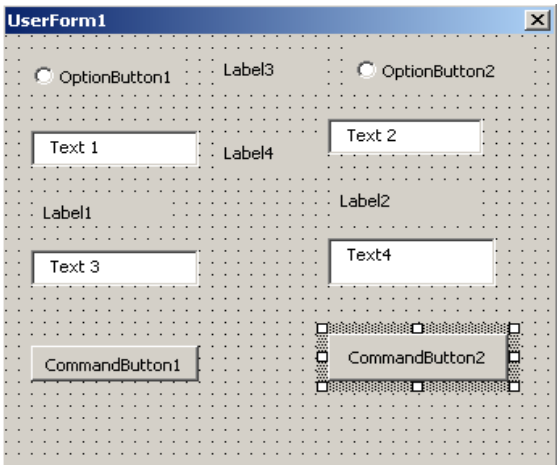


Рис. 8.5

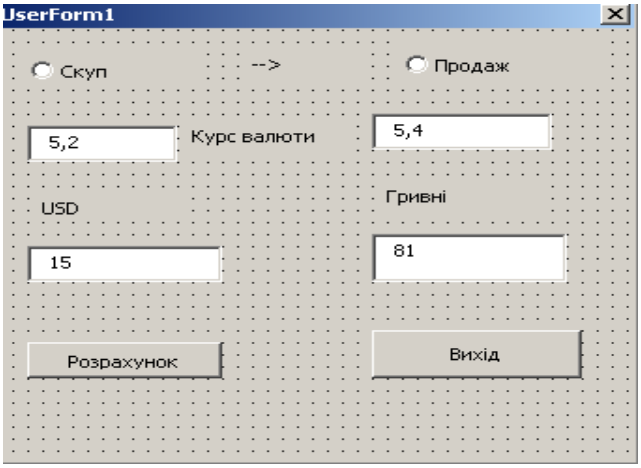


Рис. 8.6

13. Задайте однакові розміри для всіх текстових полів, полів редагування і кнопок і вирівняйте їх на формі.

Для цього синхронно виділіть п'ять об'єктів лівого стовпця одним із таких способів:

– утримуючи клавішу Shift натиснутою, по черзі активізуйте об'єкти, клацаючи на них лівою клавішею миші;

– обведіть навколо цих об'єктів контур при натиснутій лівій клавіші миші.

У вікні властивостей задайте загальні для цих об'єктів значення властивостей Width (ширина), Height (висота) і Left (відступ від лівої межі вікна) в твіпах. Зверніть увагу, що подвійне клацання на назві загальної властивості приводить до надання йому значення властивості першого виділеного об'єкта створеної групи. Можете поміняти стиль, колір або розмір шрифту синхронно для усіх виділених об'єктів (властивості Font, ForeColor, BackColor). Зніміть виділення, клацнувши на вільному місці форми. Аналогічно виконайте вирівнювання правого стовпця об'єктів. Вирівняйте вставлені поля попарно в горизонтальному напрямі. Для цього змінюйте властивість Top (відступ від верхньої межі вікна в пікселях) для відповідних груп об'єктів. Збережіть роботу (Save All).

14. Задайте значення курсів купівлі-продажу валюти, а також кількість валюти, яку обмінний пункт купує або продає.

Для цього введіть потрібне число (наприклад, 8,20) як значення властивості Text об'єкта Textl. Повторіть це для об'єкта Text2 (значення 9,40) і Text3 (наприклад, 15).

15. Очистіть поле редагування Text4.

Для цього видаліть значення властивості Text об'єкта Text4. Не переплутайте значення властивостей Name і Text цих об'єктів.

16. Заблокуйте можливість введення даних для поля Text4, оскільки в це поле виводитиметься результат. Для цього задайте значення його властивості Enabled (доступність) – False.

Змінити значення певної властивості можна, двічі клацнувши на ньому лівою клавішею миші. Виконайте програму і переконайтеся, що в полі об'єкта Text 4 не можна ввести або відредагувати дані.

17. Запрограмуйте перемикачі так, щоб напрям стрілки указував на вигляд операції: купівля або продаж. Зробіть активним поле Text3.

Клацніть двічі на правому перемикачі Option2 (Продаж). Отримаєте заготівку процедури Option2_Click. У тілі цієї процедури опишіть дії, які будуть виконані в результаті клацання на правому перемикачі Option2:

```
Private Sub Option2_Click()
Label3.Caption = "=>"
' Міняємо напрям стрілки
Text3.SetFocus ' Активізуємо поле Text3
End Sub
```

Аналогічно запрограмуйте подію Click для лівого перемикача Optionl, враховуючи, що стрілка повинна показувати на ліве поле ('<=').

Фрагмент програмного коду створеної процедури запишіть у звіт.

18. Запустіть програму і переконайтеся, що перемикач виконує надані йому функції.

19. Запрограмуйте кнопку *Вихід*.

Скористайтеся процедурою закінчення роботи програми End:

```
Private Sub Command2_Click()
End 'Закінчуємо роботу програми
End Sub
```

20. Запрограмуйте кнопку *Обчислити*.

Текст процедури цієї кнопки передбачатиме перевірку стану одного з перемикачів (включений чи ні). Перемикачі створені так, що вони завжди знаходитимуться у протилежному стані.

```
Private Sub Commandl_Click()
If Optionl.Value = True Then
Text4.Text  = Text3.Text  * Textl.Text
‘USD * курс покупки
Else
Text4.Text  = Text3.Text  * Text2.Text
‘USD * курс продажу
End If
End Sub
```

21. Виконайте програму і проекспериментуйте з різними грошовими сумами та операціями купівлі або продажу. Закрийте вікно програми *Обмін валюти*.

Для переривання роботи програми у разі неправильного введення вхідних даних виконайте пункт головного меню *Run => End* або клацніть на кнопці ■ End панелі інструментів головного вікна Visual Basic.

Зауваження. Звертайте увагу на використання коми або крапки у вхідних даних. У числах, що належать до курсу валют, сум гривень або доларів для десяткової крапки, використовуйте символ, передбачений операційною системою вашого комп’ютера (Start (Пуск) => Settings (Настройка) =* Control Panel (Панель управління) => Regional Settings (Мова і стандарти) => закладка Number (Числа), рядок Decimal symbol (Роздільник цілої і дробової частин числа)).

23. Поміняйте розміри і кольори символів (зокрема, об’єктів Label3 і Text3), розташування об’єктів, фон форми (властивість BackColor) так, щоб форма виглядала якнайкраще.

24. Забезпечте появу підказки *Введіть суму в доларах* після переміщення покажчика миші до поля *Text3*.

Виберіть об’єкт Text3 і як значення властивості ToolTipText введіть текст підказки. Збережіть роботу, запустіть програму і переконайтеся у появі підказки.

25. Поміняйте вид стрілки з => на ⇨, а <= на ⇦. Для цього виберіть об’єкт Label3 і як значення властивості *Caption* введіть російську букву р, після чого, активізувавши властивість *Font*,

виберіть назву шрифту Wingdings. Двічі клацніть на правому перемикачі та в його процедурі введіть російську букву *р* замість =>. У процедурі для лівого перемикача символи <= замінить буквою *п*. Запустіть програму і переконайтеся, що стрілка змінила свій вигляд.

Запитання для самоконтролю

1. Які базові типи даних підтримує VBA?
2. Напишіть оператора, що явно декларує рядкову змінну для зберігання поштової адреси.
3. Який оператор повинен бути присутнім у розділі глобальних оголошень, щоб виключити помилки у разі появи неявно оголошених змінних?
4. Напишіть оператора, що декларує одновимірний масив з 7 елементів цілого і базовий індекс, що має – 3.
5. Що таке динамічні масиви і як вони декларуються?
6. Яка відмінність між змінною і константою? Які типи констант підтримує VBA?
7. Що є переліками, для чого вони застосовуються?
8. Що призначене для користувача типом даних, як він визначається та оголошується?
9. Яка відмінність між лінійним і блоковим оператором If ?
10. Як визначається цикл з відомим числом повторень?
11. Для яких цілей слугує цикл For Each, який його синтаксис?
12. Які цикли з умовою підтримує VBA?
13. У чому відмінність циклів із передумовою і умовою поста?
14. Що таке функція і як вона визначається?
15. Як проводиться звернення до функції?
16. Які типи вбудованих функцій є в VBA?
17. Як створюються функції користувача?
18. Що таке аргументи функції та значення функції?
19. Що таке формальні та фактичні параметри функції? У чому полягає між ними різниця?

20. Де і як вказуються типи параметрів, що передаються функції?

21. Значення яких типів можуть повертати функції користувача?

22. Яку структуру має директива, яка оголошує початок функції користувача?

23. Який синтаксис має інструкція виклику певної функції в робочих листках MS Excel 2013? Який синтаксис має інструкція виклику певної функції в програмних кодах процедур обробки певних дій елементів керування MS Excel 2013?

24. Яким чином визначається значення, яке повертає функція?

25. Яке повідомлення показується у комірці робочого листка, якщо допущена помилка в написанні назви функції, що використовується при обчисленні значення цієї комірки?

ЛЕКЦІЯ 9
ОСНОВИ ВЕБДИЗАЙНУ

9.1. Етапи проєктування та розробки вебсайту

Вебсайти створюються для різних цілей. Але оскільки інтернет став середовищем для економічної діяльності, то сайти в основному створюються для ведення бізнесу, тобто здобуття прибутку.

Відомо, що сьогодні існує два основні напрями використання інтернет у бізнесі: інтернет як засіб комунікації, джерело довідкової інформації, засіб реклами і маркетингу для ведення бізнесу (у господарській діяльності) поза електронними мережами і інтернет як інструмент ведення електронного бізнесу, заснованого на принципах мережевої економіки.

У першому випадку сайти створюються для формування іміджу компанії, ведення ефективної реклами і маркетингових досліджень із метою пошуку нових покупців і збільшення обсягу продажу, а також для інформаційної і сервісної підтримки клієнтів. Крім того, сайти можуть бути використані як інформаційні канали обміну інформації з партнерами. Сайти також використовуються як засіб підтримки очного і заочного навчального процесу і підвищення кваліфікації з певного курсу.

У другому випадку – сайти виконують усі вищезазначені функції і, крім того, є інструментом ведення електронного або мережевого бізнесу. В цьому випадку сайти виконують функції торговельних майданчиків, до яких відносять інтернет-біржі, аукціони і. т.д.

Сайти використовують для надання фінансових послуг (онлайнові платіжні системи, змінні пункти та ін.) і так далі. Крім того, сайти необхідні під час дистанційного навчання, яке є однією із форм здобування вищої освіти. Отже, ведення електронного бізнесу (електронній комерції) без сайту неможливе.

Вебсайт – це набір вебсторінок і файлів, пов’язаних між собою гіперпосиланнями. Вебсторінки або гіпертекстові документи

є текстом, у якому містяться спеціальні команди – теги (tags). Ці теги забезпечують форматування елементів сторінки і дозволяють розміщувати на ній графічні об'єкти, малюнки, гіперпосилання тощо.

Вебсторінки створюються за допомогою спеціальної мови HTML. HTML або Hyper Text Markup Language є мовою розмітки гіпертексту, розмітка здійснюється за допомогою тегів. Сьогодні, крім HTML, використовують також інші мови розмітки: WML, XML.

Для створення вебсайту компанії необхідно:

- визначити мету створення сайту;
- розробити технічне завдання (далі – ТЗ);
- зареєструвати домен сайту у певній зоні (com, ru, ua, net і так далі);
- розробити сайт;
- розмістити сайт на хортингу;
- зареєструвати в пошукових системах і тематичних каталогах;
- виконати пошукову оптимізацію сайту;
- здійснювати постійну підтримку сайту.

Визначення мети створення сайту. Сайти створюються для різних цілей, наприклад, для ведення електронного бізнесу, для підтримки навчального процесу, для надання інформації, надання фінансових послуг тощо.

Розробка технічного завдання. У технічному завданні враховуються усі етапи розробки і супроводу сайту, цілі і призначення сайту, його дизайн, методи навігації, вказується мова розмітки сторінок тощо.

Зазвичай сайт повинен включати:

1. Інформацію про компанії, реквізити: поштова адреса, телефон, адреса електронної пошти.
2. Каталог пропонованої продукції або послуг.
3. Інформаційний розділ (новини, статті, аналітичні огляди з тематики пропонованої продукції або послуг).
4. Гостьову книгу.
5. Способи оплати.
6. Лічильники кількості відвідувачів (лічильники рейтингів).

Розробка сайту – найважливіший етап створення сайту. Розробка сайту – це складний і трудомісткий процес. Під час розробки сайту необхідно приділяти велику увагу вмісту, структурі і дизайну (графічному оформленню) вебсторінок, а також структурі вебсайту і методам навігації по вебвузлу.

Головне на сайті – це його вміст або контент, структурованість інформації, навігація, а потім графічне оформлення або дизайн сайту. Дизайн сайту визначається вмістом матеріалів, які на ньому публікуватимуться.

Для розробки сайту використовують різні засоби: конструктори сайтів (дизайнери), WebCoder 1.6.0.0, професійні застосування: Macromedia HomeSite Plus v5.1 for Windows XP, Macromedia Dreamweaver, Microsoft FrontPage, Microsoft Publisher тощо.

У редакторові FrontPage існують майстри, які дозволяють легко створювати вебсайт, і шаблони вебсторінок і вебвузлів, розмітку (структуру), що містять, дизайн, необхідний для швидкого створення як вебсторінок, так і вебсайтів. За допомогою FrontPage можна також створювати вебсторінки і вебсайти на базі порожньої сторінки і порожнього вебвузла.

Програма MS Publisher допомагає працівникам малого бізнесу створювати і розповсюджувати професійно виконані рекламні матеріали і документи для ділової переписки. Publisher – це програма, що дозволяє створювати рекламні матеріали для друку, публікації в інтернеті та розсилання електронною поштою.

Publisher містить розширений набір типів особистих публікацій, таких як особисті бланки, професійно макетовані візитні картки і запрошення.

Під час створення сайту необхідно оптимізувати його для пошукових систем, оскільки цільовий відвідувач приходить на сайти з пошукових систем, тому необхідно прагнути до високого рейтингу в пошукових системах.

Особливу увагу необхідно приділяти таким метатегам, як Title (заголовок), Keywords (ключові слова) і Description (опис), а також розташуванню ключових слів у тексті вебсторінок.

Службу інтернет, яка управляє передаванням таких документів, називають World Wide Web (Web, WWW). Цим терміном або *простором WEB* називають сукупність вебдокументів, між якими існують гіпертекстові зв'язки.

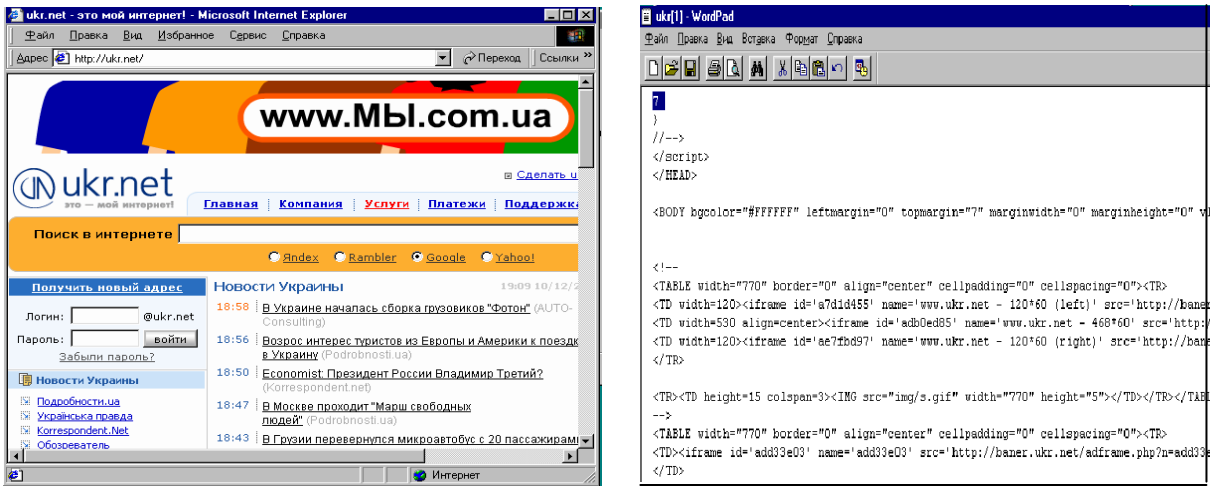
Поняття «гіпертекст» має досить давню історію. Воно виникло на стику філософії та комп'ютерних технологій ще у середині XX ст. Теоретичною основою гіпертексту стали сформульовані В. Бушем (1945 р.) принципи нелінійного письма, які він докладно висвітлив у роботі «Як ми могли б мислити». У 60-х роках Т. Нельсон і Д. Енгельбарт, застосовуючи комп'ютерні технології опрацювання інформації, розробили нову технологію, яку назвали «гіпертекст». У гіпертексті є посилання на інші сторінки або документи. У більшості *браузерів* (програм для завантаження та перегляду вебдокументів) ці посилання підкреслюються або виділяються іншим кольором. Якщо клацнути лівою кнопкою миші на гіперпосиланні, то програма перегляду гіпертексту завантажить документ, на який вказує посилання. Таким чином сторінка набуває властивості інтерактивності. У деякому розумінні завдяки гіпертексту інформація у WWW організована за тим самим принципом, що і в людському мозку.

Окремі документи, які становить простір веб, називають *вебсторінками*. Вебсторінки зберігаються на жорстких дисках *вебсерверів*. Вебсервери – це спеціалізовані комп'ютери з відповідним програмним забезпеченням, яке дає можливість доступу користувачів до вебсторінок; це програмне забезпечення також називається вебсервером.

Групу сторінок, присвячену певній темі та розміщену в певному каталозі вебсервера, називають *вебвузлом* або *вебсайтом*. Один фізичний вебсервер може містити кілька вебсайтів.

9.2. Структура вебсторінки та її об'єкти

Вебсторінки мають вигляд звичайних текстових документів, в які введено вказівки, форматування. Принцип роботи браузера полягає в інтерпретації цих вказівок. При відображенні таких документів браузером самі вказівки не відображаються, проте впливають на спосіб відображення решти частини документа. Зазначені вказівки називають *дескрипторами* або *тегами*. З їх допомогою текст можна робити кольоровим, використовувати шрифти різного розміру, вбудовувати мультиплікацію, відеофрагменти тощо. Формат дескрипторів задається в описі спеціальної *мови розмітки*. Вона називається *мовою розмітки гіпертексту-HTML {HyperText Markup Language)*. Наприклад, вигляд головної сторінки сайту Ukr.Net та її код зображено на рис. 9.1.



а) б)

Рис. 9.1. Головна сторінка сайту Ukr.Net (а) та її код (б)

Документи, розмічені за допомогою цієї мови, називають *HTML-документами*. HTML-документи мають розширення.htm або.html.

Іноді процес розробки вебдокументів засобами мови HTML називають вебпрограмуванням. Проте потрібно розуміти, що HTML не є мовою програмування у звичайному розумінні, а є мовою розмітки (опису). Термін вебпрограмування мовою HTML має історичне походження. Під терміном вебпрограмування будемо розуміти процес розробки вебдокументів (і не лише засобами HTML).

Мова HTML розроблена спеціально для веб. Її популярність забезпечують зокрема такі властивості:

– документ, створений за допомогою деякої програми, наприклад текстового редактора, часто важко (а іноді і неможливо) використовувати в іншій програмі; HTML у цьому відношенні є універсальною;

– HTML – це відкритий стандарт;

– HTML не є власністю якої-небудь фірми;

– можливість використання гіпертексту;

– HTML підтримує мультимедіа.

У мові HTML використовуються поняття *контейнера*. Контейнер – це дескрипторна пара, яка складається з початкового і кінцевого *дескрипторів (тегів)*. Початковий дескриптор має вигляд <TAG>, де TAG ім'я певного HTML-дескриптора. Кінцевий дескриптор має вигляд </TAG>. Наприклад,

<B> Програми для перегляду вебсторінок називаються браузерами </B>.

Контейнер <B></B> дає вказівку браузеру відображати текст «Програми для перегляду вебсторінок називаються браузерами» напівжирним шрифтом. Тобто контейнери впливають на частину документа, розміщену між ними. Зазначимо, що контейнери можуть бути вкладені.

Одиночний дескриптор, звичайно, має самостійне завдання, не пов'язане з конкретним текстом. Наприклад, дескриптор <HR> (від слів Horisontal Line) слугує для створення і відображення горизонтальної лінії.

Дію дескриптора можна дещо змінити, якщо задати певні *атрибути {параметри)*. Атрибути – це додаткові ключові слова, які відокремлюються від ключового слова, що визначає дескриптор, пропуском і розміщуються до символу «>». У контейнерах атрибути додаються тільки до початкового дескриптора. Атрибути задаються своїми значеннями. Наприклад,

<H1 ALIGN=»LEFT»> Транспортний рівень моделі TCP/IP</H1>

– це вказівка браузеру відобразити текст «Транспортний рівень моделі TCP/ IP» у вигляді заголовка та вирівняти його по лівому краю.

HTML-код вебсторінки може містити коментарі, тобто деякий текст, який не відображається браузером і слугує для пояснення призначення сторінки або частини її коду. Текст, що повинен слугувати коментарем, необхідно помістити між символами «<!--» і «→». Наприклад,

<HR> <! – створюємо горизонтальну лінію →

Коментар можна поставити у будь-яке місце коду сторінки, де дозволяються пропуски.

Усі HTML-документи мають однакову загальну структуру. Праворуч показано відображення цього коду браузером.

```
<HTML>
<HEAD>
<TITLE>
Назва WEB-сторінки
</TITLE>
</HEAD>
<BODY>
”Тіло” – це вміст WEB-сторінки:
текст, графіка, гіперпосилання
</BODY>
</HTML>
```

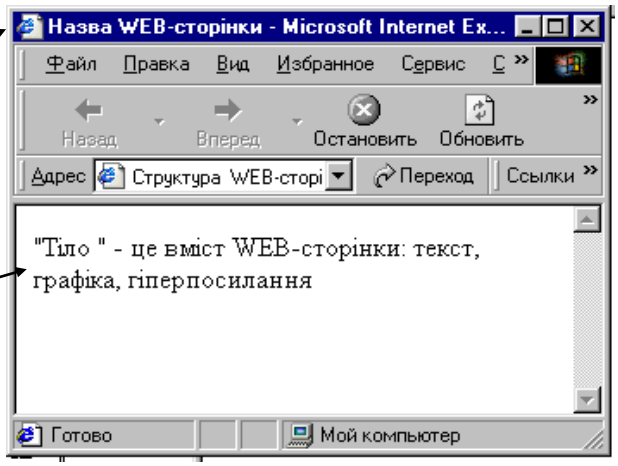


Рис. 9.2. Структура HTML-документа

Контейнер <HTML></HTML> є ознакою того, що даний файл містить документ HTML. У HTML-документах є «голова» (заголовок) і «тіло» (основна частина).

Розділ <HEAD></HEAD> містить дескриптори, які описують документ в цілому. Зокрема, тут вказується назва документа.

Контейнер <TITLE></TITLE> служить для визначення назви документа. Текст, включений в нього, відображається у верхній частині вікна браузера. Назва сторінки – це один з елементів, які мають важливе значення для привернення уваги «відвідувачів». Вона повинна відображати зміст сторінки. Системи пошуку орієнтуються саме на назву сторінки, тому сторінка із змістовною назвою має більше шансів бути вибраною у процесі проведення пошуку інформації з конкретної теми.

Контейнер <B0DY></B0DY> задає основну частину документа – його «тіло». Інформація, розміщена між дескрипторами <BODY> та </BODY>, відображається в області документа.

Крім контейнера <TITLE>…</TITLE> у заголовок документа (контейнер <HEAD>…. </HEAD>) часто включаються дескриптори <BASE>, <META>.

Дескриптор <BASE> використовується для вказування повної URL адреси документа. Це дозволяє підтримувати відносні гіперпосилання в робочому стані при переміщенні документа в інший каталог і навіть на інший комп'ютер. Адреса URL задається за допомогою атрибута HREF, наприклад,

<BASE HREF="http://www.ia.net/~rmeegan">

Докладніше цей дескриптор описано під час розгляду гіперпосилань.

Дескриптор <META> дозволяє автору документа описати інформацію або виконати дії, які ще не підтримуються офіційною версією HTML. Він найчастіше використовується завдяки його властивості Keywords (ключові слова). У багатьох пошукових системах використовуються слова, визначені в цьому елементі для складання вказівника документа. Наприклад,

<META NAME="Keywords" CONTENT ="внз, університет, фізика, математика, економічна інформатика">

<META NAME="Description" CONTENT="Браузера фізико-метематичного факультету">

Цей дескриптор дозволяє також задати метаінформацію про кодову сторінку, яка використовується для кодування літер кирилиці:

<META content="text/html; charset=Windows-1251" http-equiv=Content-Type>

Контейнер <BODY> має велику кількість атрибутів, важливих для визначення загального виду документа. Основні з цих атрибутів задано в таблиці 9.1.

Таблиця 9.1

Атрибут	Призначення
ALINK	Визначає колір, яким виділяється активне гіперпосилання
BACKGROUND	Вказує URL зображення, яке потрібно використовувати як фон документа
BGCOLOR	Визначає колір фону документа
LINK	Визначає колір гіперпосилання, яке не було «відвідане»
TEXT	Визначає колір тексту документа
VLINK	Визначає колір гіперпосилання, яке було «відвідане»

Створити просту сторінку можна, наприклад, так:

1. Запустити текстовий редактор *Блокнот*.
2. Ввести текст HTML-документа.
3. Зберегти під деяким іменем із розширенням.htm (наприклад, Структура вебсторінки.htm). При збереженні, у вікні *Сохранение документа* потрібно поміняти тип файл з *Текстовые документы* на *Все файлы*. Інакше редактор у кінець назви файлу автоматично додасть розширення.txt.
4. Запустити програму Internet Explorer.
5. За допомогою меню *Файл -> Открыть* відкрити створений файл (Структура вебсторінки.htm).

У наступному прикладі наводиться код сторінки з синім фоном:

```
<HTML>
<HEAD>
<TITLE> Сторінка з фоном </TITLE>
</HEAD>
<BODY BGCOLOR=”blue”>
Білий текст на синьому фоні
</BODY>
</HTML>
```

9.3. Використання фреймів

Мова HTML дозволяє поділити вікно браузера на окремі секції, які називаються фреймами (рамки). Вміст кожного фрейму знаходиться в окремому HTML-файлі. Кожний фрейм можна переглядати або змінювати не залежно від решти фреймів. Браузер, який «розпізнає» фрейми, завантажує різні сторінки у різні секції свого вікна.

Наприклад, можна побудувати сторінку так, що фірмовий знак буде зафіксовано у верхній частині вікна, збоку розміщено панель навігації сторінками сервера, а в решту вікна будуть завантажуватися різні сторінки, які користувач зможе переглядати звичайним чином. Коли користувач клацає кнопкою миші на гіперпосиланні – змінюється лише частина вікна, а система навігації і фірмовий знак залишаються нерухомими.

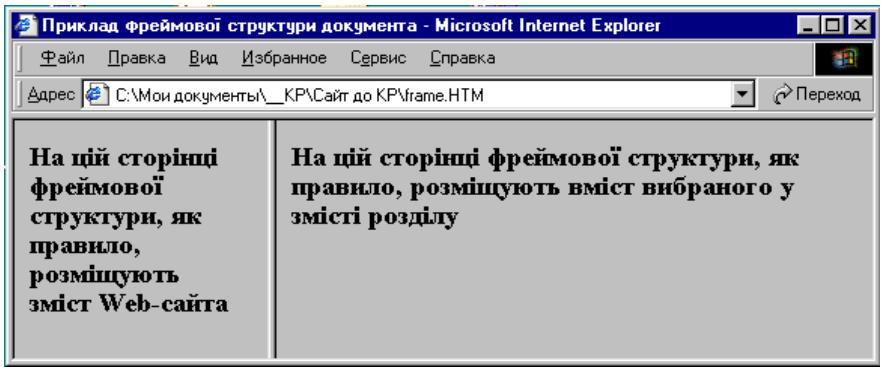


Рис. 9.3. Поділ вікна на два фрейми

Код цієї сторінки наведено нижче:

```
<HTML>
<HEAD>
<TITLE>Приклад фреймової структури документа
</TITLE></HEAD>
<frameset cols=”30%,70%”>
<frame src=”a1.htm”>
<frame src=”a2.htm”>
</frameset>
</BODY>
</HTML>
```

Файл a1.htm

```
<HTML>
<HEAD>
<TITLE>Приклад фреймової структури документа</TITLE>
</HEAD>
<BODY bgcolor=”silver” text=”black” link=”#ff0000”>
<h3> На цій сторінці фреймової структури, як правило, розміщують зміст вебсайту</h3>
</BODY>
</HTML>
```

Файл a2.htm

```
<HTML>
<HEAD>
<TITLE>Приклад фреймової структури документа</TITLE>
</HEAD>
<BODY bgcolor=”silver” text=”black” link=”#ff0000”>
<h3> На цій сторінці фреймової структури, як правило, розміщують вміст вибраного у змісті розділу</h3>
</BODY>
</HTML>
```

9.4. Створення списків, таблиць

Для створення списків використовують теги, що задають тип списку. Нумерований список починається тегом <OL> і завершується тегом </OL>, наприклад:

```
<OL>
<LI> Один       1. Один
<LI> Два        2. Два
<LI> Три        3. Три
<LI> Чотири     4. Чотири
<LI> П’ять      5. П’ять
</OL>
```

Тег <OL> може мати такі параметри:

TYPE – вид лічильника:

A – великі латинські букви

a – малі латинські букви

I – великі римські цифри

i – малі римські цифри

1 – звичайні цифри

START – встановлює число, з якого буде починатися відлік, наприклад:

```
<OL TYPE=”I”
START=”8”>
<LI> Вісім
<LI> Дев’ять
<LI> Десять
<LI> Одинадцять
<LI> Дванадцять
</OL>
```

VIII. Вісім
IX. Дев’ять
X. Десять
XI. Одинадцять
XII. Дванадцять

Неномерований список починається тегом <UL> і завершується тегом </UL>. Кожний елемент списку починається з тегу <LI>, наприклад:

```
<UL>
<LI> Один
<LI> Два
<LI> Три
<LI> Чотири
<LI> П’ять
</UL>
```

- Один
- Два
- Три
- Чотири
- П’ять

Тег <UL> може включати параметр *TYPE* із значеннями disc, circle, square

```
<UL TYPE=disc><LI> disc </UL>
<UL TYPE=circle><LI> circle </UL>
<UL TYPE=square><LI> square </UL>
```

- disc
- circle
- square

Дані значення – це зовнішній вид маркера, який за замовчуванням ставиться у вигляді диску, тобто disc.

Списки визначень мають такий вигляд:

```
<DL>
<DT> Термін                          Термін
<DD> Визначення терміну              Визначення терміну
<DD> Визначення терміну              Визначення терміну
<DD> Визначення терміну              Визначення терміну
<DD> Визначення терміну              Визначення терміну
</DL>
```

Таблиці у вебдокументах використовуються не тільки для розміщення табличних даних, а й для вставки зображень і посилань, для раціонального компонування вебсторінки.

Елементи таблиць. Таблиці будуються за принципом вкладання і вводяться на вебсторінку за допомогою ряду елементів. Кожна таблиця починається тегом <TABLE> і закінчується тегом </TABLE>. Створювана таблиця ніби розгортається по рядках, а рядки заповнюються комірками. При цьому всередину тегів <TABLE> …</TABLE> можуть вставлятися такі елементи:

TR – елемент створення рядка;

TD – елемент, що визначає вміст комірки даних;

TH – елемент, що визначає комірку заголовка.

Наприклад, для створення таблиці 3*2 використовується такий шаблон:

```
<TABLE>
<TR><TD>…. </TD><TD> … </TD></TR>
<TR><TD>…. </TD><TD> … </TD></TR>
<TR><TD>…. </TD><TD> … </TD></TR>
</TABLE>,
```

де крапками позначений вміст кожної комірки.

Приклад таблиці:

Комірка 1	Комірка 2
Комірка 3	Комірка 4

Примітка. У цьому прикладі створена таблиця з фіксованою шириною (WIDTH=”200” пікселів), але краще використовувати проценти, бо у цьому випадку розмір таблиці буде змінюватися залежно від розміру вікна.

Така таблиця реалізується таким кодом:

```
<TABLE BORDER=”2” WIDTH=”200”
BGCOLOR=”#00FF00”>
<TR>
<TD>Комірка 1</TD>
<TD> Комірка 2</TD>
</TR>
<TR>
<TD> Комірка 3</TD>
<TD> Комірка 4</TD>
</TR>
</TABLE>
```

Таблиця починається тегом <TABLE> і завершується тегом </TABLE>. Тег <TABLE> може включати такі атрибути:

Таблиця 9.2

WIDTH=”n”	Визначає ширину таблиці в пікселях або процентах, за замовчуванням ширина таблиці визначається вмістом клітинок
BORDER=”n”	Встановлює товщину рамки. За замовчуванням n=0 – таблиця відображається без рамки
BORDERCOLOR=”#FFFFFF”	Встановлює колір обрамлення, де #FFFFFF (білий) – шістнадцяткове значення кольору
BGCOLOR=”#FFFFFF”	Встановлює колір фону для всієї таблиці, де #FFFFFF – шістнадцяткове значення кольору
BACKGROUND=”image.gif”	Заповнює фон таблиці зображенням
CELLSPACING=”n”	Визначає відстань між рамками клітинок таблиці в пікселях

Продовження таблиці 9.2

ALIGN=LEFT	Визначає положення таблиці в документі. За замовчуванням таблиця притиснута до лівого краю сторінки. Допустимі значення атрибута: LEFT (зліва), CENTER (по центру сторінки), RIGHT (справа)
FRAME=”значення”	Управляє зовнішнім обрамленням таблиці і може приймати такі значення: VOID – обрамлення немає (значення за замовчуванням). ABOVE – межа тільки зверху. BELOW – межа тільки знизу. HSIDES – межа тільки зверху і знизу. VSIDES – межа тільки зліва і справа. LHS – тільки ліва межа. RHS – тільки права межа. BOX – малюються усі чотири сторони. BORDER – також усі чотири сторони.
RULES=”n”	Управляє лініями, що розділяють клітинки таблиці. Можливі такі значення (n): NONE – немає ліній (значення за замовчуванням). GROUPS – лінії будуть тільки між групами рядів. ROWS – тільки між рядами. COLS – тільки між колонками. ALL – між усіма рядами і колонками.

Таблиця може включати заголовок, який розміщується між тегами <CAPTION></CAPTION>. Він має бути безпосередньо після тегу <TITLE>. До заголовка можна застосувати атрибут ALIGN, що визначає його положення щодо таблиці:

TOP – значення за замовчуванням, заголовок над таблицею по центру.

LEFT – заголовок над таблицею зліва.

RIGHT – заголовок над таблицею справа.

BOTTOM – заголовок під таблицею по центру.

Рядки таблиці починаються тегом <TR> і завершуються тегом </TR>, а кожна клітинка таблиці починається тегом <TD> і завершується </TD>. Дані теги можуть мати такі атрибути:

Таблиця 9.3

Атрибути, що можуть застосовуватися до рядків і клітинок	
ALIGN=LEFT	Встановлює горизонтальне вирівнювання тексту в клітинках рядка. Може приймати значення LEFT (вирівнювання вліво), CENTER (вирівнювання по центру), RIGHT (вправо)
VALIGN=CENTER	Встановлює вертикальне вирівнювання тексту в клітинках рядка. Допустимі значення: TOP (вирівнювання по верхньому краю), CENTER (вирівнювання по центру – це значення приймається за замовчуванням), BOTTOM (по нижньому краю)
BGCOLOR=”#FFFFFF”	Встановлює колір фону рядка або клітинки
BACKGROUND=”image.gif”	Заповнює фон рядка або клітинки зображенням
Атрибути, що можуть застосовуватися тільки до клітинок	
WIDTH=”n”	Визначає ширину клітинки в n пікселях
HEIGHT=”n”	Визначає висоту клітинки в n пікселях
COLSPAN=”n”	Розтягує комірки по горизонталі. Наприклад, <TD COLSPAN=”2”> означає, що комірка буде розтягнута на 2 колонки
ROWSPAN=”n”	Розтягує клітинки по рядку. Наприклад, <TD ROWSPAN=”2” означає, що клітинка буде розтягнута на два рядки таблиці
NOWRAP	Цей атрибут показує, що текст буде розміщено в один рядок
BACKGROUND=”image.gif”	Заповнює фон клітинки зображенням

Крім того, будь-яка клітинка таблиці може бути визначена не тегами <TD></TD>, а тегами <TH></TH> – Table Header (заголовок таблиці). Це звичайна клітинка, але текст між цими тегами буде виділений напівжирним шрифтом і відцентрований.

Якщо клітинка порожня, то навколо неї рамка не малюється. Якщо рамка потрібна навколо порожньої клітинки, то в неї треба ввести символьний об’єкт   (non-breaking space – нерозривний пропуск).

Теги, що встановлюють шрифт (<B>, <I>, <FONT SIZE=”n”, FONT COLOR=”#FFFFFF”), необхідно повторювати для кожної клітинки.

9.5. Оформлення тексту, створення заголовків. Графічні об’єкти

Форматування тексту передбачає створення заголовків, абзаців, вирівнювання тексту, вибір стилів шрифту (напівжирний, курсив, підкреслений), зміну розміру шрифту.

У HTML існує шість рівнів заголовків і кожному з них відповідає свій розмір шрифту. За допомогою заголовків різних рівнів можна створювати чітку ієрархічну структуру документа. Заголовку I рівня відповідає найбільший розмір шрифту. Синтаксис HTML-коду, що створює заголовки різного рівня та відображення цього коду браузерам, зображено на рис. 9.4.

<H1> Заголовок першого рівня </H1>
<H2> Заголовок другого рівня </H2>
<H3> Заголовок третього рівня </H3>
<H4> Заголовок четвертого рівня </H4>
<H5> Заголовок п’ятого рівня </H5>
<H6> Заголовок шостого рівня </H6>

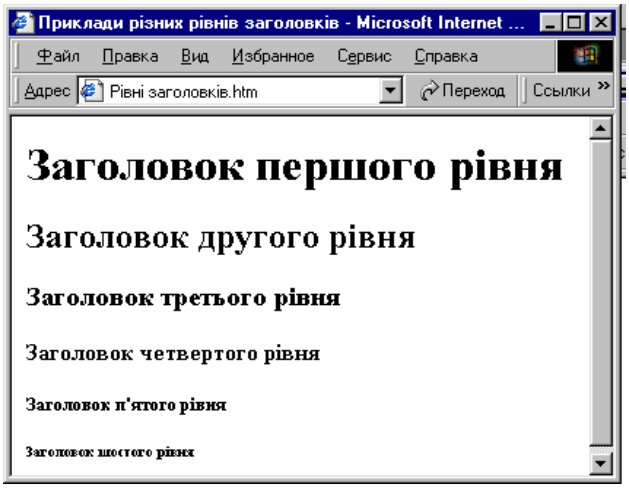


Рис. 9.4. Заголовки різного рівня

Для кращого сприйняття сторінки користувачем не потрібно пропускати рівні заголовків.

Для відображення тексту в HTML немає спеціального дескриптора. Текст вводиться у будь-якому місці між дескрипторами <BODY></BODY>. У текст не потрібно вставляти ні порожні рядки, ні символи переведення рядка, оскільки при відображенні сторінки браузером вони не враховуються. Текст автоматично розміщується за шириною вікна браузера. Якщо необхідно на вебсторінці помістити порожній рядок або забезпечити в потрібному місці перехід до наступного рядка, потрібно використовувати спеціальні дескриптори:

– дескриптор розриву абзацу <P></P> та

– дескриптор розриву рядка <BR>.

Дескриптор розриву абзацу дає команду браузеру закінчити поточний абзац і помістити порожній рядок перед наступним абзацом. Дескриптор розриву рядка вказує про закінчення поточного рядка і переходу до наступного. Для дескриптора розриву абзацу використовувати закриваючий дескриптор </P> не обов’язково.

Використання зазначених дескрипторів розглянуто на прикладі:

<HTML>

<HEAD>

<TITLE>

Приклад використання дескрипторів розриву рядка та абзацу

</TITLE>

</HEAD>

<BODY>

Дескриптор розриву абзацу дає команду браузеру вставити порожній рядок перед початком наступного абзацу.

<P>

Дескриптор розриву рядка дає команду браузеру
 перейти у наступний рядок.

</P>

</BODY>

</HTML>

Відображення сторінки з таким кодом зображено на рис. 9.5.

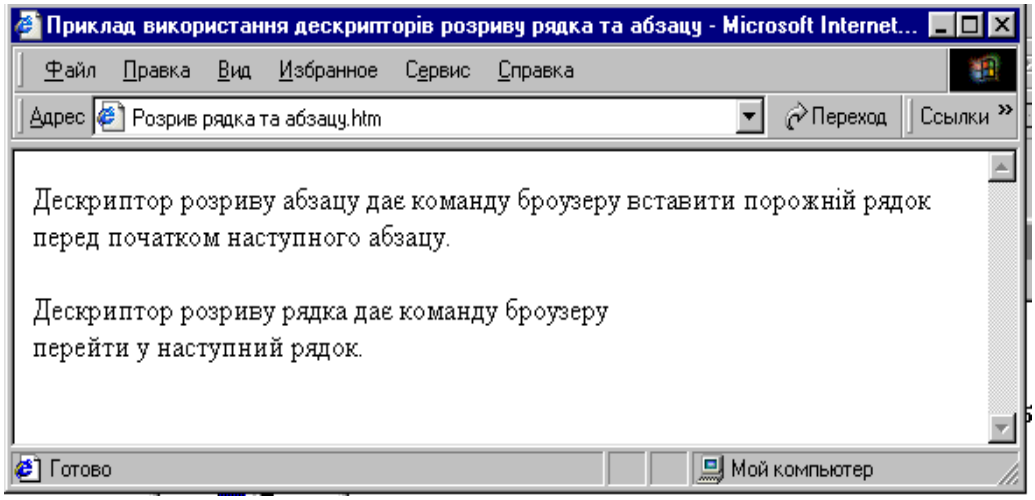


Рис. 9.5. Абзаци та пусті рядки

Текст абзацу чи заголовка може бути вирівняний по лівому краю, по правому краю або по центру вікна браузера. Для вирівнювання тексту використовується параметр ALIGN=вирівнювання, де вирівнювання може приймати значення: LEFT для вирівнювання по лівому краю, RIGHT для вирівнювання по правому краю, CENTER для вирівнювання по центру. Найчастіше параметр ALIGN використовується у дескрипторах розриву абзацу та визначення заголовка.

Наприклад:

```
<HTML>
<HEAD>
<TITLE> Приклад вирівнювання тексту </TITLE>
</HEAD>
<BODY>
<H2 ALIGN=CENTER>Цей заголовок буде розміщено посередині</H2>
<!-- Наступний абзац буде вирівняно по правому краю
<P ALIGN=RIGHT> Цей абзац вирівняно по правому краю. Вивчення HTML невдовзі стане такою ж необхідністю, як вміння ввімкнути комп’ютер, скопіювати файл чи скористатися текстовим редактором. Якщо користувач вміє набирати текст, то зможе освоїти і використання HTML.
</P>
</BODY>
</HTML>
```

Відображення сторінки з таким кодом зображено на рис. 9.6.

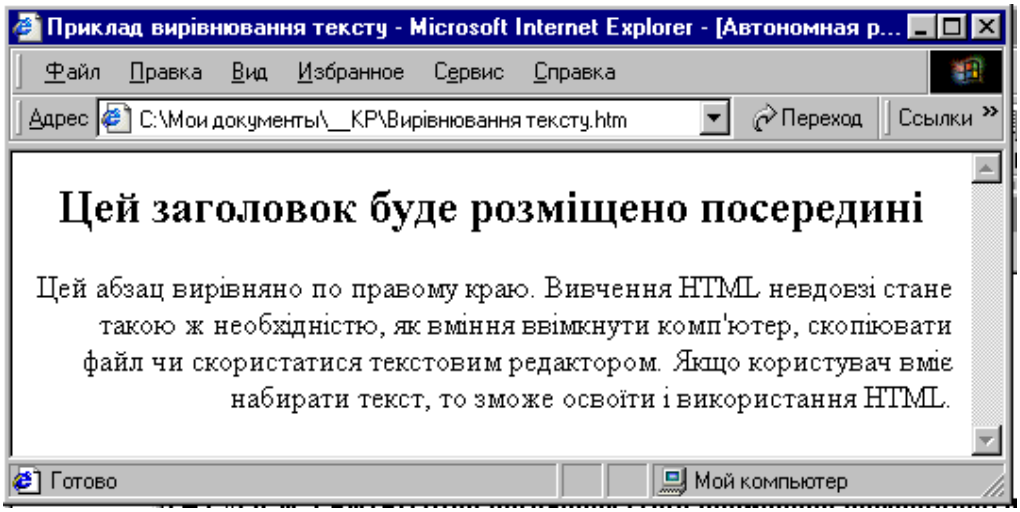


Рис. 9.6. Вирівнювання абзацу по центру та по правому краю

У HTML є контейнер, в який можна помістити заздалегідь відформатований текст. Контейнер <PRE>…</PRE> використовується для відображення відформатованого текстовим редактором блоку тексту, який складно було б прочитати, якщо б браузер переформатував його (наприклад, блоки програмного коду мовою C++). Цей контейнер дозволяє зберегти розбиття тексту на рядки (символи переходу на новий рядок, отримані в результаті натискування клавіші *Enter*, інтерпретуються як розриви рядка), пропуски всередині тексту інтерпретуються у точній відповідності з їх розміщенням у текстовому редакторі. Недоліком такого підходу є втрата гнучкості, а також те, що браузери, зазвичай, виділяють відформатований заздалегідь текст із допомогою моноширинного шрифту. Це призводить до того, що довгі фрагменти тексту зливаються. У наступному прикладі створюється сторінка із заздалегідь відформатованим текстом:

<HTML>

<HEAD>

<TITLE> Приклад заздалегідь відформатованого тексту </TITLE>

</HEAD>

<BODY>

```
<H1 ALIGN=CENTER> Книги про HTML </H1>
<PRE>
НАЗВА                АВТОР      ВИДАВНИЦТВО    РІК
Использование HTML   М. Браун   «ВИЛЬЯМС»      2000
</PRE>
</B0DY>
</HTML>
```

Відображення документа з таким кодом у браузера зображено на рис. 9.7.

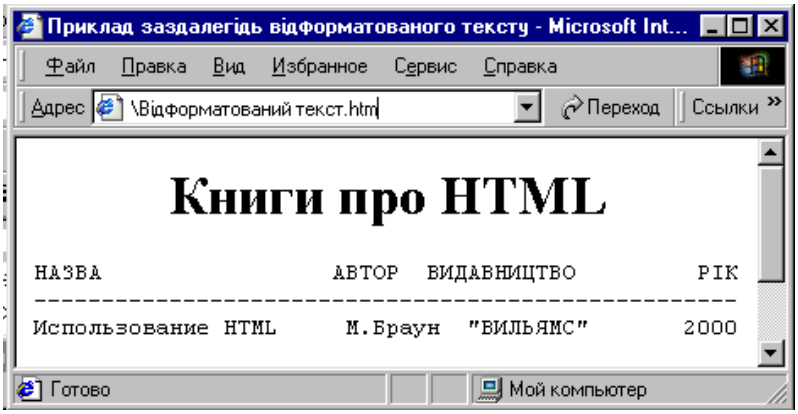


Рис. 9.7. Використання відформатованого тексту

Мова HTML надає можливість вибирати для написання фрагментів тексту різні стилі: напівжирний, курсив, підкреслення тощо. Для цього в HTML існують спеціальні дескриптори, які називають дескрипторами стилів. Усі дескриптори стилів є контейнерами. Основні дескриптори стилів зображено у таблиці 9.4.

Таблиця 9.4

Основні дескриптори стилів

Дескриптор	Стиль
<B>	Напівжирний шрифт
<I>	Курсив
<U>	Підкреслення
<SUB>	Підрядковий текст
<SUP>	Надрядковий текст
<STRIKE>	Перекреслення

Використання дескрипторів стилів розглянуто в такому прикладі:

<HTML>

<HEAD>

<TITLE> Приклади виділення тексту дескрипторами стилів </TITLE>

</HEAD>

<BODY>

Приклад виділення тексту <B> напівжирним шрифтом </B>

Можна також виділити текст <I>курсивом</I>

Особливо важливі місця можна <U> підкреслити </U>

Можна створити ^{надрядковий}

або _{підрядковий} текст

Крім того, текст можна <STRIKE>перекреслити</STRIKE>.

</BODY>

</HTML>

Відображення сторінки з таким кодом зображено на рис. 9.8.

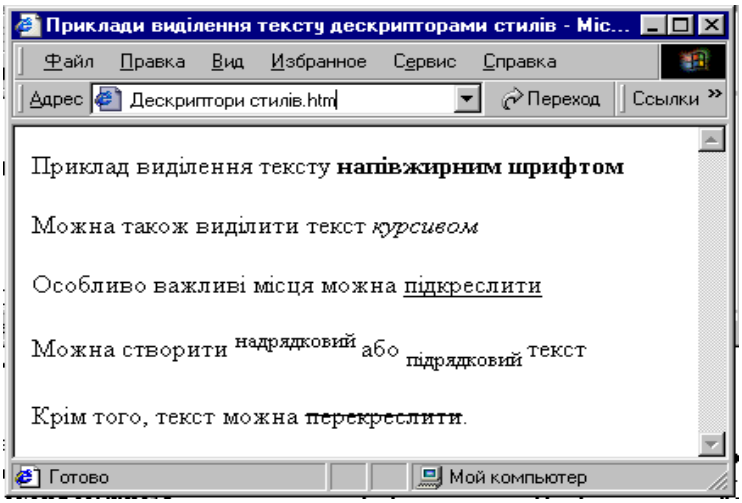


Рис. 9.8. Стилі форматування

Для вибору шрифту, його розміру та кольору в HTML слугує контейнер <FONT>…. </FONT>.

Розмір шрифту можна визначити за допомогою атрибута SIZE дескриптора <FONT>:

<FONT SIZE=”значення”>

Атрибут SIZE дозволяє вказувати абсолютний (значення від 1 до 7) та відносний (значення від –4 до +4) розміри шрифту щодо розміру основного шрифту, що використовується в основній частині сторінки.

Для зміни розміру основного шрифту використовується елемент <BASEFONT SIZE=значення>. Значенням атрибута SIZE може бути число від 1 до 7. Для браузера стандартним є розмір шрифту 3.

Атрибут FACE дескриптора <F0NT> дозволяє вказати браузеру шрифт, яким необхідно відображати документ. Якщо розробник сторінки не знає, які шрифти можуть бути на комп'ютері користувача, то в атрибуті FACE можна вказати через кому декілька назв шрифтів, наприклад:

<FONT FACE="Verdana", "Arial", "Helvetica">

Браузер переглядає список шрифтів зліва направо і використовує перший, який встановлено на комп'ютері користувача. Якщо браузер не знаходить на комп'ютері користувача жодного шрифту, що вказані в атрибуті FACE, документ відображається шрифтом, який задано в браузері користувача за замовчуванням.

За допомогою дескриптора <FONT> можна також змінити колір шрифту. Для цього використовується атрибут COLOR, наприклад, код

<FONT COLOR="GREEN">текст</FONT>

задає вказівку браузеру відобразити слово «текст» зеленим кольором. Детальніше про використання кольору у вебдокументах розглянуто в розділі вебграфіка.

Крім дескрипторів у мові HTML існують також спеціальні символи форматування. Одним із таких символів є нероздільний пропуск – . Використання цього символу в коді HTML-файлу забезпечує появу на вебсторінці одного пропуску. Наприклад, за кодом:

розмір   шрифту

браузер між словами «розмір» і «шрифту» містить 6 пропусків*. У таблиці 9.5 описано коди деяких спеціальних символів.

Таблиця 9.5

Коди спеціальних символів

Коди	Символи
&copy, &reg	©, ®
&frac14, &frac12,	1/4, 1/2, 3/4
&pi	π
&1e, &де	<,>
&amp	&

*Графічні об'єкти***.** Важливу роль в оформленні вебсторінок відіграють графічні зображення. Включення їх у HTML-документи дозволяє надати сторінці «настрою» (ліричного, гумористичного, ділового), подати інформацію, яку неможливо отримати з тексту, розбити сторінку на «теми», що дозволить користувачеві краще орієнтуватися у матеріалі. Для того щоб сторінка подобалась відвідувачам, необхідно досягти оптимального співвідношення форми (дизайну) і змісту.

Швидкість появи зображення на екрані залежить від розміру файлу зображення. Графіка, яка повільно завантажується, може «відлякати» відвідувачів сторінок. Для зменшення розміру файлу необхідно використовувати зображення типу JPEG або GIF. Конвертацію графіки у ці формати можна здійснити за допомогою редакторів растрових зображень. Допустимими зображеннями є такі, розмір яких не перевищує 40–50 Кбайт.

На HTML-сторінках графіка можна розміщувати різними способами.

Вбудовані зображення – це графічні зображення, які завжди залишаються в одному і тому самому місці сторінки (справа від тексту чи іншого об'єкта, що безпосередньо передує зображенню) і не обрамлюються текстом. Для того щоб помістити на вебсторінці вбудоване зображення, використовують елемент <IMG SRC=URL_зображення>. Дескриптор <IMG> є одиночним. В атрибуті SRC (від слова source) необхідно вказати повну або відносну URL-адресу файлу зображення. Наприклад код:

<P>Це зображення нової моделі модема

<IMG SRC="modem8.jpg">цей текст розміщується одразу після малюнка.

Використано для розміщення на сторінці зображення, файл якого розміщено в тому самому каталозі, що і сам HTML-файл (рис. 9.9).

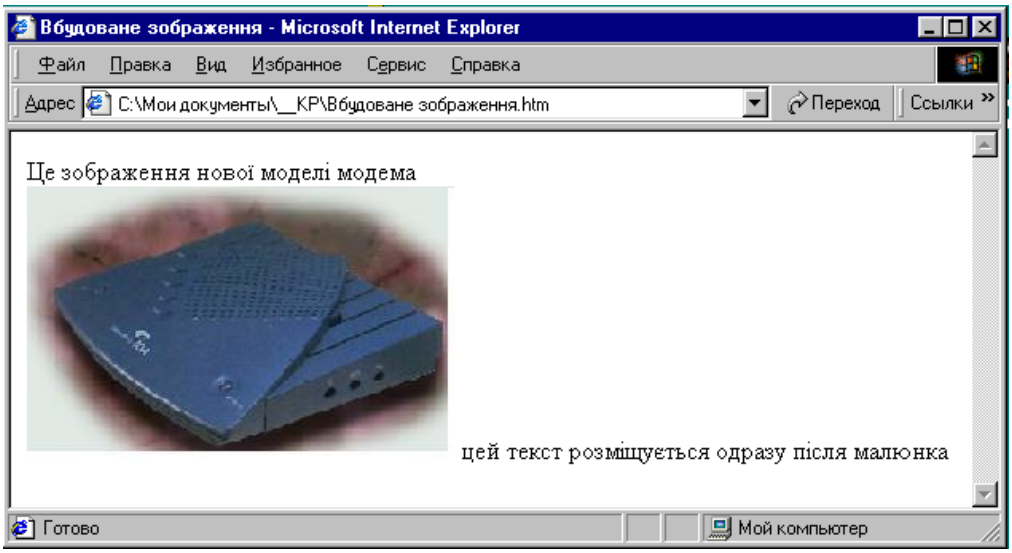


Рис. 9.9. Графічне зображення на вебсторінці

За замовчуванням текст вирівнюється по нижній частині зображення. Для зміни типу вирівнювання використовують дескриптор IMG з атрибутом ALIGN. Нижче наведені значення, які можна присвоїти цьому атрибуту.

Таблиця 9.6

Значення атрибуту	**Призначення атрибуту**
TOP	Вирівнює текст по верхньому краю зображення
MIDDLE	Вирівнює текст по середній частині зображення
BOTTOM	Вирівнює текст по нижньому краю зображення

Значення атрибуту. Призначення атрибуту

TOP вирівнює текст по верхньому краю зображення;

MIDDLE вирівнює текст по середній частині зображення;

BOTTOM вирівнює текст по нижньому краю зображення.

Наведемо код документа, що вміщує три зображення, для кожного з яких використовують один із варіантів вирівнювання. На рис. 9.10 зображено вебсторінку, створену на основі цього коду.

<HTML>

<HEAD>

<TITLE>

Використання атрибута ALIGN в дескрипторі IMG
</TITLE>
</HEAD>
<BODY>
<P> <IMG SRC=”modem.jpg” ALIGN=TOP>
Цей текст вирівнюється по верхньому краю зображення </P>
<P> <IMG SRC=”modem.jpg” ALIGN=MIDDLE>
Цей текст вирівнюється по середній частині зображення </P>
<P> <IMG SRC=”modem.jpg” ALIGN=BOTTOM>
Цей текст вирівнюється по нижньому краю зображення </P>
</BODY>
</HTML>

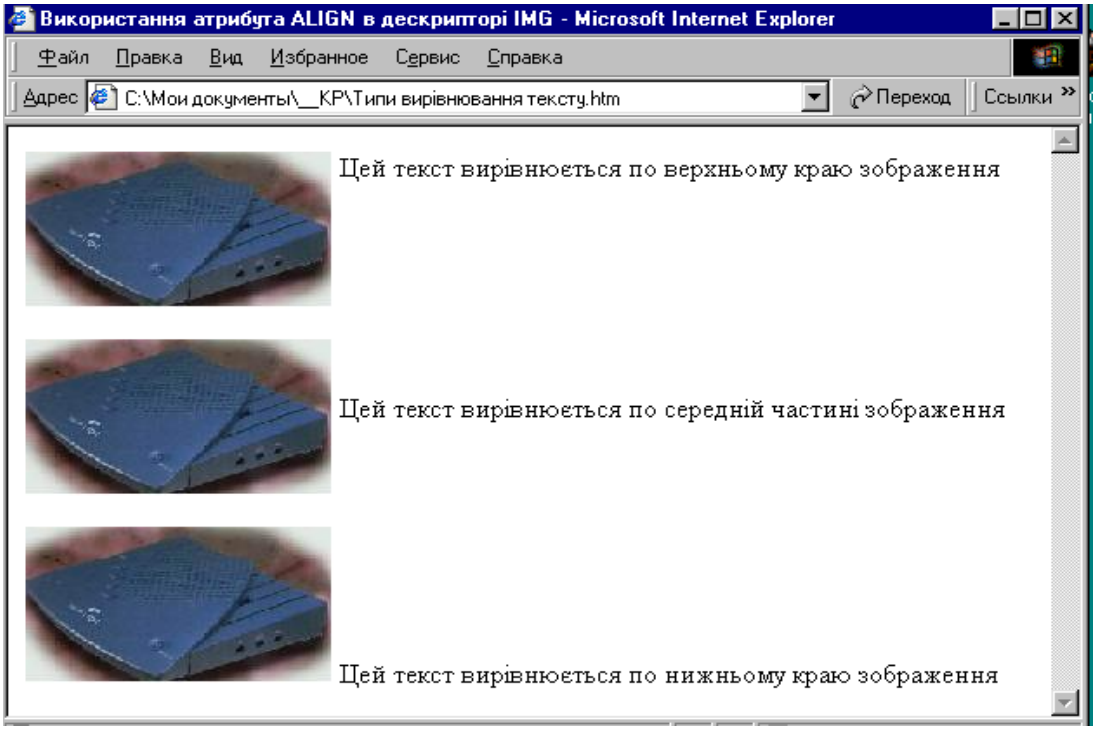


Рис. 9.10. Вирівнювання тексту щодо зображення

Плаваючі зображення – це зображення, що не прив’язуються до одного рядка тексту, а ніби «плавають» вздовж одного з полів і обрамлюються текстом.

Для створення плаваючих зображень використовується дескриптор <IMG> з атрибутом ALIGN. Для плаваючих зображень значення цього атрибута можуть бути LEFT (зображення плаває вздовж лівого краю тексту) або RIGHT (зображення плаває вздовж правого краю тексту). Наприклад:

<IMG SRC=”amd.jpg” ALIGN=LEFT>

Нижче наводиться код HTML-документа, в якому поміщено плаваюче зображення.

```
<HEAD>
<TITLE>
Використання атрибута ALIGN в дескрипторі IMG
</TITLE>
</HEAD>
<BODY>
<P> <IMG SRC=”modem.jpg” ALIGN=LEFT></P>
Цей текст розміщується праворуч від зображення і нижче <BR>
Цей текст розміщується праворуч від зображення і нижче<BR>
Цей текст розміщується праворуч від зображення і нижче<BR>
Цей текст розміщується праворуч від зображення і нижче<BR>
Цей текст розміщується праворуч від зображення і нижче<BR>
Цей текст розміщується праворуч від зображення і нижче<BR>
Цей текст розміщується праворуч від зображення і нижче
</BODY>
</HTML>
```

На рис. 9.11 зображено інтерпретацію цього коду браузером.

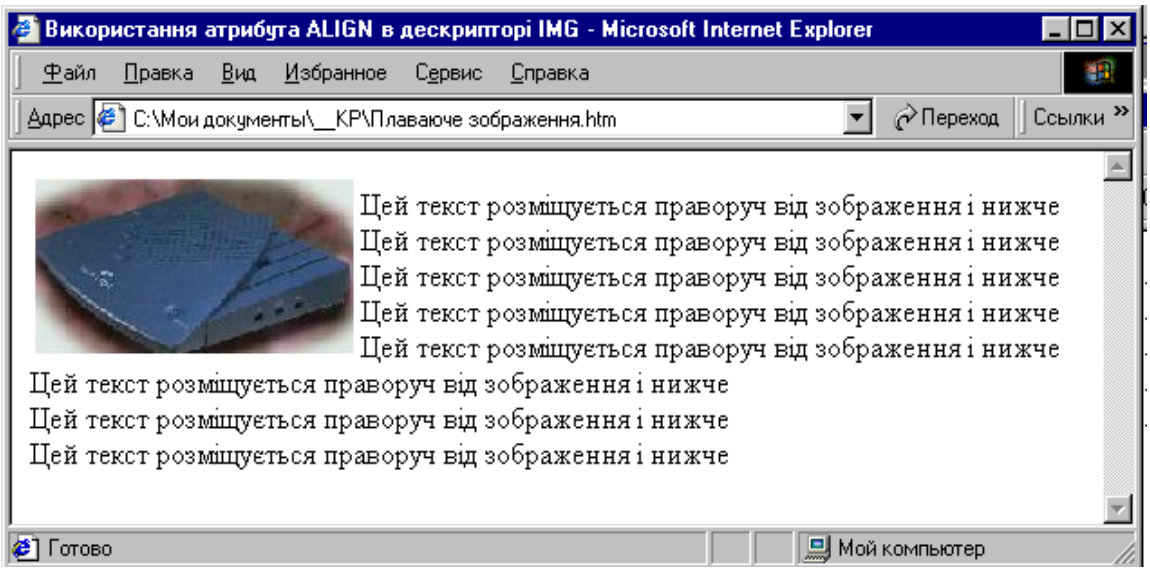


Рис. 9.11. Плаваюче зображення

У дескрипторі <IMG> можна використовувати атрибути HEIGHT і WIDTH для визначення висоти та ширини зображення. За допомогою цих атрибутів «повідомляється» розмір зображення.

Це дозволяє браузеру підготувати макет сторінки та розмістити текст до того, як будуть завантажені всі зображення. За допомогою цих атрибутів також можна змінювати розміри зображення.

Так, наприклад, для збільшення зображення необхідно в атрибутах HEIGHT і WIDTH вказати більші величини, ніж реальні розміри зображення (які можна визначити з допомогою графічного редактора). Зауважимо, що зменшувати розмір зображення таким методом не доцільно, оскільки розмір файлу, а отже, і час завантаження при цьому не змінюються.

За допомогою атрибута BGCOLOR дескриптора <BODY> можна визначити колір фону сторінки, вказавши у ньому шестицифрове шістнадцяткове число. Цей атрибут має такий формат:

<BODY BGCOLOR=#rrggbb>

rr, gg, bb – двоцифрові шістнадцяткові числа, які визначають червону, зелену та голубу складові кольору фону.

Наприклад:

<BODY BGCOLOR=#000000> – задається чорний колір фону;

<BODY BGCOLOR=#ffffff> – задається білий колір фону;

<BODY BGCOLOR=#ffOOOO> – задається червоний колір фону;

<BODY BGCOLOR=#00ff00> – задається зелений колір фону;

<BODY BGCOLOR=#0000ff> – задається синій колір фону.

Для визначення основних кольорів можна використовувати також зарезервовані імена (таблиця 9.7).

Наприклад:

<BODY BGCOLOR=”RED”> – задається червоний колір фону.

Для створення графічного фону використовується атрибут BACKGROUND дескриптора <BODY>. У цьому атрибуті потрібно вказати ім’я файлу графічного зображення. Наприклад:

<BODY BACKGROUND=”my_picture.gif”>

Колір основного тексту можна змінити за допомогою атрибута TEXT дескриптора <BODY>. Колір задається так, як і у випадку зміни кольору фону.

Таблиця 9.7

Зарезервовані імена кольорів

Колір	Шістнадцятковий еквівалент
BLACK (чорний)	#000000
MAROON (темно-бордовий)	#800000
GREEN (зелений)	#008000
OLIVE (оливковий)	#808000
NAVY (темно-синій)	#000080
PURPLE (фіолетовий)	#800080
TEAL (зеленувато-блакитний)	#008080
GRAY (сірий)	#808080
SILVER (сріблястий)	#C0C0C0
RED (червоний)	#FF0000
LIME (лимонний)	#00FF00
YELLOW (жовтий)	#FFFF00
BLUE (синій)	#0000FF
FUCSHIA (фуксія)	#FF00FF
AQUA (морська хвиля)	#00FFFF
WHITE (білий)	#FFFFFF

Зображення-гіперпосилання – це зображення на вебсторінці, клацнувши на якому, можна відкрити нову вебсторінку або службу інтернет. Наприклад, можна помістити на вебсторінці зображення поштової скриньки, клацнувши на якому користувачі зможуть відправляти повідомлення електронної пошти. HTML-код зображення-гіперпосилання має вигляд:

<A HREF=”URL”><IMG SRC=”URL-файлу зображення”></A>

Із ціллю швидшого завантаження вебсторінки має сенс створювати зменшені копії зображень (мініатюри зображень), які є гіперпосиланнями на відповідні їм зображення-оригінали. Мініатюри завантажуються швидко, даючи користувачеві уявлення про зображення-оригінали. Доступ до зображення-оригіналу можна отримати у будь-який момент, клацнувши лівою кнопкою миші на зображенні-гіперпосиланні.

Щоб використовувати мініатюри, необхідно мати дві копії зображення – велике зображення-оригінал і зменшене зображення. Для зменшення розмірів зображень використовують можливості

графічних програм. Щоб зв'язати мініатюру з оригіналом, необхідно помістити дескриптор, що задає мініатюру, всередині дескриптора <A HREF> </A>, який є гіперпосиланням на велике зображення, наприклад:

<A HREF="big.gif"><IMG SRC="small.gif"></A>

Тут задано вказівку браузера розмістити на вебсторінці зображення (файл small.gif), що є гіперпосиланням на зображення, задане файлом big. Gif.

9.6. Гіперпосилання на вебсторінці

Гіперпосилання – це основа HTML. За допомогою гіперпосилань користувачі можуть переходити з однієї сторінки до іншої.

Існує три основних типи гіперпосилань: внутрішні, зовнішні та відносні.

Внутрішні гіперпосилання – це посилання на об'єкти в межах одного документа. З їх допомогою користувач може переміщатись всередині однієї вебсторінки. Такі посилання доцільно використовувати на довгих сторінках, щоб мати можливість швидко переміщуватись між їх розділами.

Зовнішні гіперпосилання – це посилання на сторінки, розміщені на іншому вебсервері.

Відносні або локальні посилання – це посилання на вебсторінки, розміщені на тому самому сервері. Адреси цих посилань задаються відносно адреси вебсторінки, на якій вони розміщені.

У HTML для зовнішніх гіперпосилань використовують синтаксис:

<A HREF="URL">TeKCT посилання </A>,

де URL – повна адреса документа, на який здійснюється посилання. Текст, поміщений у контейнер <A HREF="URL"> </A>, зображається у браузері з підкресленням та іншим кольором (найчастіше синім). Клацнувши на ньому лівою кнопкою миші, даємо вказівку браузеру завантажити сторінку, розміщену за адресою URL.

Наприклад, вказівкою:

<A HREF=”http://www.firma.com/ibm/computers.html”> Комп’ютери компанії IBM </A>

створюється гіперпосилання на вебсторінку, розміщену за адресою http://www.ibm.com/ibm/computers.html.

З’єднуючи гіперпосиланнями сторінки одного сервера, немає необхідності включати в URL доменне ім’я. Наприклад, вказівкою:

<A HREF=”computers.htm”> Комп’ютери виробництва IBM</A>

створюється гіперпосилання на файл computers.htm, який повинен знаходитися на тому самому сервері і в тому самому підкаталозі, що і файл, в якому розміщене це гіперпосилання.

Якщо файл знаходиться в іншому підкаталозі, то необхідно вказати шлях до нього, наприклад:

<A HREF=”ibm/computers.htm”>Комп’ютери виробництва IBM</A>

Є й інші можливості для спрощення записів довгих адрес URL. Так, наприклад, HTML-код:

<HTML>

<HEAD>

<TITLE> Основні розділи вебсайту НПУ імені М. П. Драгоманова</TITLE>

</HEAD>

<BODY>

<A HREF=”http://www.npu.kiev.ua/REKT0RAT/in-rek.htm”> Ректорат</A>

<A HREF=”http://www.npu.kiev.ua/FAKULT/in-fak.Iiit”> Факультети</A>

<A HREF=”http://www.npu.kiev.ua/ASPIRANT/index.htm”> Аспірантура</A>

<A HREF=”http://www.npu.kiev.ua/ABITUR/abit-beg.htm”> Абітурієнту</A>

</BODY>

</HTML>

можна записати компактніше:

<HTML>

<HEAD>

<TITLE> Основні розділи вебсайту НПУ імені М. П. Драгоманова</TITLE>

<BASE HREF=”http://www.npu.kiev.ua/”>

</HEAD>

<BODY>

<A HREF=”REKTORAT/in-rek.htm”>Ректорат</A>

<A HREF=”FAKULT/in-fak.htm”>Факультети</A>

<A HREF=”ASPIRANT/index.htm”>Аспірантура</A>

<A HREF=”ABITUR/abit-beg.htm”>Абітурієнту</A>

</BODY>

</HTML>

Тут використовується дескриптор <BASE>, в якому вказано базову адресу URL, стосовно якої подаються усі інші відносні адреси вебсторінок. Зазначимо, що цей дескриптор використовують всередині контейнера <HEAD> </HEAD>, областю дії його є уся сторінка.

Створюючи великі вебсторінки, необхідно також забезпечити можливість для користувача легко переміщуватись з одного їх розділу до іншого. Це досягається завдяки внутрішнім гіперпосиланням. Для реалізації такого посилання необхідні два елементи: посилання, яке вказує, куди треба перейти, і мітка, яка фіксує місце переходу.

Посилання задається кодом:

<A HREF=”#Назва мітки”>Текст посилання</A>

Мітка також використовує контейнер <A></A> і задається так:

<A NAME =”#назва мітки”>текст</A>

Назва мітки – це довільний текст, який не з’являється на екрані й однозначно визначає місце переходу. Якщо на сторінці є кілька міток, то всі вони повинні мати різні назви. Символ «#» при описі посилання вказує, що за ним записана назва мітки, а не назва файлу.

Приклад використання внутрішніх гіперпосилань:

<HTML>

<HEAD>

<TITLE>Колекція програм мовою VisualBasic</TITLE>

</HEAD>

<BODY>

Нижче наводяться програми мовою VisualBasic:

<! – далі створюємо внутрішні гіперпосилання

<A HREF=”#графік”>Побудова графіків функцій</A>

<A HREF=”#масив”>Сортування масиву</A>

<A HREF=”#рівняння”>Розв’язування квадратних рівнянь</A>

<P>

<! – далі описуємо місця, на які здійснюються посилання

<A NAME=”графік”> Побудова графіків функцій – приклад програми</A>

(текст програми)

<A NAME=”масив”> Впорядкування масиву – приклад програми</A>

(текст програми)

<A NAME=”рівняння”> Розв’язування квадратних рівнянь – приклад програми</A>

(текст програми)

</BODY>

</HTML>

Можна зробити так, щоб після натискування на деякому посиланні звучала музика або відтворювався відеофільм. Посилання на мультимедійні файли.avi,.wav створюються так само, як посилання на HTML-документи. В атрибуті HREF контейнера <A></A> необхідно вказати ім’я файлу, в якому міститься потрібна мелодія або відеофільм. Наприклад:

<P>Натисніть<A HREF=”my_music.wav”>ТуТ</A>, щоб почути мою улюблену композицію.

Файл my_music.wav повинен знаходитись на тому самому сервері, у тому самому каталозі, що і HTML-файл, з якого він викликається. Для створення гіперпосилань на файли мультимедіа, розміщені на інших серверах інтернет, використовують повні URL-адреси.

Часто виникає необхідність включити у вебсторінку посилання на адресу електронної пошти. Для цього в атрибуті HREF необхідно задати адресу електронної пошти у вигляді ”mailto: адреса”. Наприклад, щоб створити посилання на адресу moi@npu. Kiev. Ua, необхідно у код сторінки помістити такий дескриптор:

<A HREF=”mailto: moi@npu.kiev.ua”>moi@npu.kiev.ua</A>,

згідно з яким браузер відображає текст ”moi@npu.kiev.ua” як гіперпосилання. Якщо клацнути на ньому лівою кнопкою миші, браузер відкриє поштову програму в режимі створення листа, і в полі ”Адреса:” помістить адресу moi@npu.kiev.ua. Текст, що поміщається у контейнер <A></A>, може бути будь-яким, проте доцільно використовувати адресу E-mail, оскільки не всі користувачі сторінки можуть користуватися поштовими програмами, налагодженими на комп’ютері, з якого переглядається сторінка. Їм необхідно прочитати реальну адресу електронної пошти.

9.7. Динамічні ефекти та засоби їх створення

Динамічними називаються ефекти, коли графічні зображення на вебсторінці змінюються з часом, елементи сторінки змінюють розміри або навіть свій зміст після клацання над ними мишею, текст «біжить» уздовж екрана тощо.

Розглянемо ефект рухомого тексту у смузі, що має висоту HEIGHT =”висота в пікселах” і фон BGCOLOR = ”колір фону”. Ефект створюється за допомогою парного тега <MARQUEE>...</MARQUEE>, а саме:

<MARQUEE BGCOLOR=”aqua” HEIGHT = 40> Особиста сторінка Ваше прізвище та ім’я</ MARQUEE>. За допомогою цього тегу можна заставити рухатись і графічний об’єкт.

Текст *Особиста сторінка Ваше прізвище та ім'я* буде пробігати у стрічці справа наліво, заходитиме за край екрана і з'являтиметься знову справа. Цей тег рекомендують використовувати до заголовків сторінки.

Ефект відбивання від країв екрана забезпечує параметр BEHAVIOR = "alternate", а зупинити рядок біля лівого краю екрана може значення цього параметра "slide". Значення "right" параметра DIRECTION забезпечить ефект руху у протилежний бік.

Обмеження кількості проходів задається параметром LOOP=число, де цифра – кількість проходів. Смугу можна центрувати за допомогою параметрів HSPASE і VSPACE. Швидкість руху задає параметр SCROLLAMOUNT=число, де число – величина швидкості руху, де конкретне значення вибирають з діапазону від 1 (повільно) до 10 (швидко).

Приклад. У тексті коду вебсторінки у файлі file1.htm вставте рухомий рядок, наприклад такого змісту:

1. <MARQUEE BGCOLOR="aqua" HEIGHT = 40 LOOP=5> Приватна сторінка Ваше прізвище та ім'я</ MARQUEE>.

2. Далі змініть цей рядок на такий:

<MARQUEE BGCOLOR="aqua" HEIGHT = 50 LOOP=4 DIRECTION ="rihgt" SCROLLAMOUNT=5> <FONT FACE = "Courier" SIZE = +4 COLOR = "red"> Особиста сторінка Ваше прізвище та ім'я </FONT></MARQUEE>

перегляньте вебсторінку.

9.8. Поняття про інтерактивні вебсторінки та засоби розробки сценаріїв

Вище було розглянуто основні конструкції мови HTML і можливості її використання для створення вебсторінок. Тепер потрібно з окремих сторінок сконструювати свій вебсайт і розмістити його в інтернеті. Розглянемо спочатку етапи планування і створення сайту на жорсткому диску вашого комп'ютера.

Процес створення вебсайту можна умовно поділити на ряд етапів:

- визначення цілей;
- створення схеми вебсайту;

- створення сторінок;
- публікація сторінок на сервері;
- реклама вебсайту;
- підтримка вебсайту.

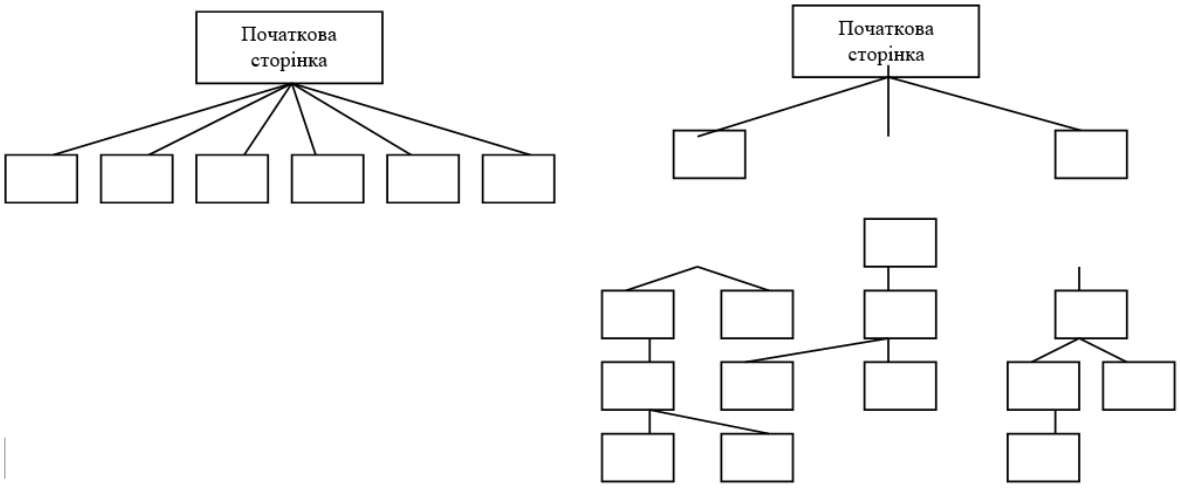
Починаючи розробку свого сайту, потрібно чітко уявити собі його призначення. Сайти створюються для того, щоб заявити про себе чи про свою організацію, повідомити про результати роботи чи про свої досягнення, налагодити ділові зв'язки, дати рекламу про освітні послуги тощо.

Крім призначення сайту, потрібно визначити коло його потенційних відвідувачів, тобто аудиторію, на яку розрахований ваш сайт. Чи то будуть школярі, студенти, чи люди різних віків? У який час вони будуть відвідувати ваш сайт і на яких мовах будуть говорити? Якими браузерами вони будуть користуватися?

Наступним етапом буде підбір матеріалу. Не весь матеріал за тематикою сайту, що ви маєте, варто публікувати в інтернеті. Потрібно оцінити якість матеріалу та його цікавість для відвідувачів. Відібраний матеріал розділіть за темами – ці теми будуть визначати розділи майбутнього сайту. Якщо матеріалу за кожною темою буде досить багато, розсортуйте його за ступенем важливості.

Матеріали, що ви плануєте опублікувати на сайті, потрібно організувати за деякою структурою. Найчастіше для вебсайтів обирається деревоподібна структура організації інформації. На верхньому рівні міститься початкова сторінка сайту, з якої відвідувач за допомогою меню чи посилань і може переходити на сторінки наступного рівня. Деревоподібна структура може складатися з декількох рівнів підпорядкування.

Важливо визначити оптимальне співвідношення між кількістю рівнів (глибиною коренів чи висотою дерева) і кількістю варіантів вибору на кожному рівні (шириною дерева). Якщо дерево виявиться досить широким (рис. 9.12, а), то відвідувачеві важко буде орієнтуватися в поданій інформації й утримувати у пам'яті безліч варіантів вибору. Якщо ж рівнів буде багато, то відвідувачу доведеться довго «докопуватися» до потрібної йому інформації (рис. 9.12, б). Для цього він має зробити безліч клацань мишею і дочекатися завантаження проміжних сторінок.

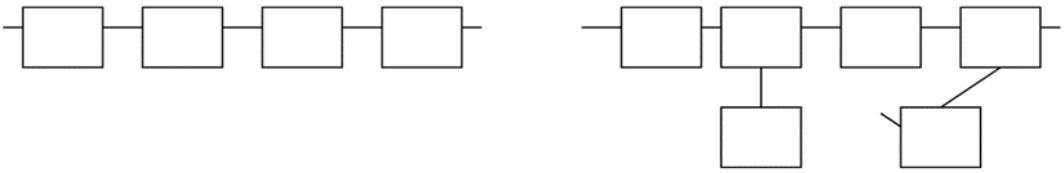


а) б)

Рис. 9.12. Приклади структури сайту:
а) – широке дерево; б) – вузьке дерево

Щоб полегшити подорож усередину сайту, на сторінках верхнього рівня розміщують посилання, що можуть перевести відвідувача відразу ж до сторінок нижнього рівня. На глибоких сторінках також можуть розміщуватися посилання для зворотного переходу вгору ієрархічними рівнями сайту.

Іноді для сайту замість деревоподібної структури доцільно обрати структуру у вигляді простого ланцюга (лінійна структура). При цьому перехід з кожної сторінки може здійснюватися тільки у двох напрямах: на наступну і на попередню сторінки (рис. 9.13, а). Для цього на сторінках розташовуються кнопки Вперед та Назад, за їх допомогою легко переглядати розділи книги чи набір фотографій. На практиці лінійна структура застосовується в поєднанні з різними відгалуженнями й альтернативними переходами (рис. 9.13, б).



а) б)

Рис. 9.13. Приклад лінійної структури сайту:
а) – лінійний ланцюг; б) – лінійна структура з відгалуженням і альтернативним переходом

Відвідувач може скласти уявлення про структуру сайту за допомогою засобів навігації: списків-посилань, меню тощо. Для кращої орієнтації відвідувачів до складу сайту може бути включена окрема сторінка, так звана карта сайту.

Схема розроблюваного сайту може бути такою, як на рис. 9.14.

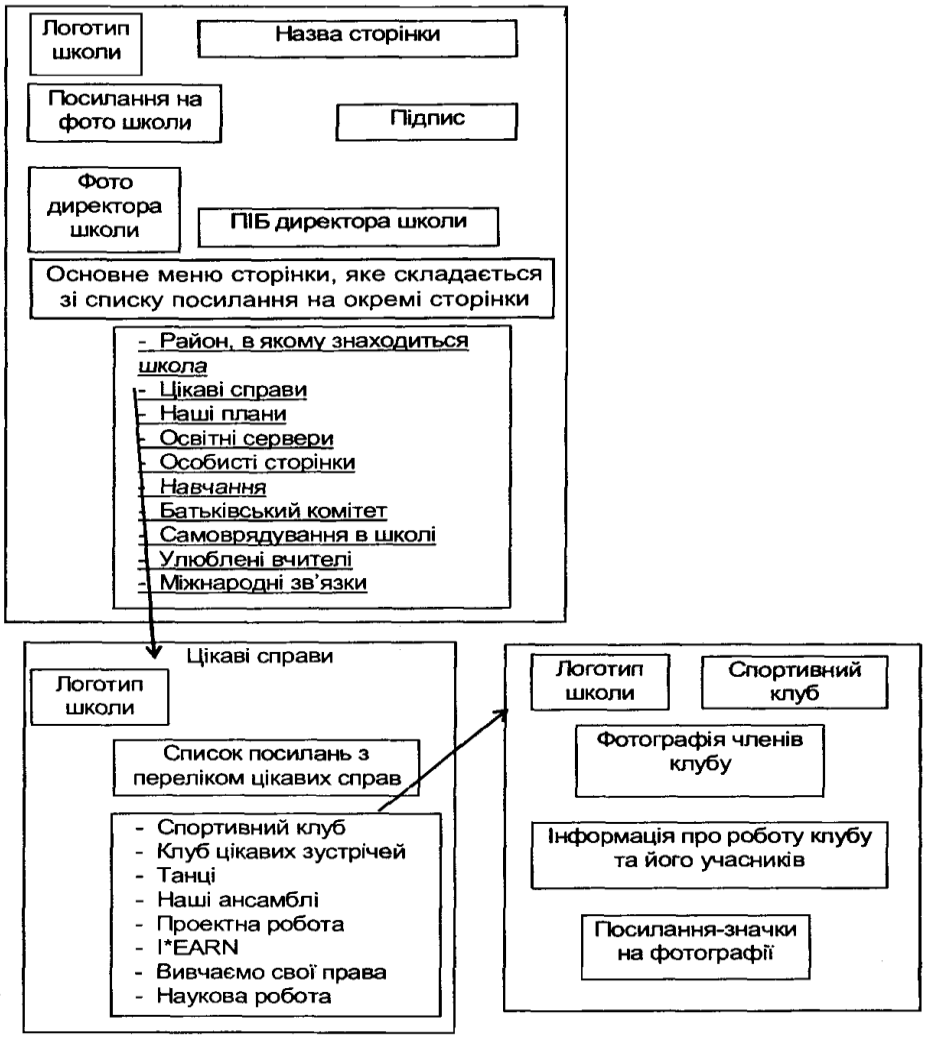


Рис. 9.14

Збирання файлів вебсайту. Коли ви підготували матеріали і подали у вигляді схеми структуру вашого сайту, можете розпочинати безпосередньо розробку, тобто складання HTML-коду. Саме цей етап роботи над сайтом був описаний вище. У результаті на диску вашого комп'ютера будуть створені файли вебсторінок, а також розміщуватися файли впроваджених об'єктів (зображень, аудіо-, відео-).

Скопіюйте всі файли в окрему папку і назвіть її, наприклад, Website. У цій папці має бути файл index.html, що подає початкову сторінку сайту. Інші файли папки Website містять HTML-код інших

сторінок (файли з розширенням.html). У розглянутому прикладі в папці Website розміщений також файл із розширенням.ess, який містить аркуші каскадних стилів (на початковому етапі знайомства з HTML цей файл можна не створювати). Імена усіх файлів повинні бути набрані малими латинськими буквами (рис. 9.15).

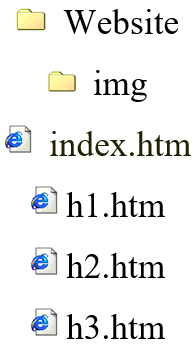


Рис. 9.15. Вміст папки сайту

У папці сайту бажано передбачити папку (у цьому прикладі img), у якій зберігатимуться зображення у форматі GIF чи JPEG.

Якщо під час формування папки сайту ви перейменовували файли, простежте, щоб відповідні зміни були внесені й у код HTML сторінок. Після цього завантажте початкову сторінку сайту в браузер (Internet Explorer або Netscape Navigator) і переконайтеся, що всі переходи на інші сторінки сайту працюють, сторінки відображаються потрібним чином, а малюнки виводяться на сторінки. Якщо будь-яка сторінка чи малюнок не відображаються, то, мабуть, ви помилилися у написанні імен файлів. Виправте ці помилки і ще раз перевірте роботу сайту.

*Вибір вебсерверу***.** Отже, ви створили свій невеликий сайт. Тепер вам потрібно вибрати вебсервер, на якому цей сайт буде розміщений.

Вебсервер виконує збереження, пошук і обмін файлами в WWW. Наприклад, коли клієнт веб запитує файл із WWW, програма браузера відправляє цей запит на вебсервер, на якому розміщується цей файл. Сервер відшукує файл на своїх дисках і відправляє його комп'ютеру-клієнту, від якого був отриманий запит. Обмін інформацією між клієнтом і вебсервером відбувається відповідно до протоколу HTTP – загального протоколу, що відповідає за функціонування World Wide Web.

Вебсервер – це комп'ютер, що працює під керуванням однієї з операційних систем UNIX, Windows NT, Macintosh і на який установлена спеціальна програма вебсерверу.

Найбільш поширеною програмою вебсерверу є Apache Server, яка працює швидко і встановлюється безкоштовно (вузол www.apache.org). Часто використовуються також програми TomCat, Microsoft IIS, NCSA та ін. За допомогою однієї з таких програм свій вебсервер може створити будь-яка організація, школа і навіть приватна особа. Однак це не завжди виправдано, оскільки повноцінний вебсервер повинен мати швидке з'єднання з інтернетом (наприклад, через виділену лінію) і працювати цілодобово. Тому користувачі вирішують проблему вебсерверу за допомогою сторонніх організацій – фірми вашого провайдера чи спеціалізованих фірм, які займаються хостингом.

Хостинг (від англ. слова hosting – спільне розміщення) – це розміщення Web-caumy на обладнанні компанії. Ресурси серверу і лінії зв'язку використовуються спільно безліччю клієнтів.

Під час вибору стороннього вебсерверу керуються двома основними критеріями: ціною і продуктивністю. У професійних розробників вебсайтів вибір визначається можливостями замовника і стосунками з компаніями, що володіють інтернет-серверами.

Нині з'являється усе більше вебсерверів, що здійснюють безкоштовний хостинг (наприклад, сервери www.bigmir.net, www.chat.ru, www.da.ru, www.narod.ru, www.boom.ru). Такі сервери не вимагають оплати за підтримку вебсайтів клієнтів та існують за рахунок рекламодавців. Зареєстрованому клієнту дається обмежений, але достатній простір на диску серверу – від 5 до 50 Мбайт. На ньому клієнт може розмістити як невеликий, так і солідний сайт. Єдине зобов'язання, що бере на себе клієнт безкоштовного вебсерверу, це розміщення на своєму сайті банерів – маленьких помітних зображень (часто анімованих), призначених для реклами. Утім, згоди клієнта на це, як правило, не запитують – банери вставляються й обновляються без участі власника сайту.

Безкоштовні вебсервери мають також обмеження у наданому сервісі: вони, як правило, не підтримують сучасні інтернет-технології PHP, JSP, ASP та ін. Однак свій сайт, що містить

звичайні файли HTML, ви зможете розмістити на безкоштовному сервері без особливих проблем.

(Термін «банер» (від англ. banner – прапор) прийшов у WWW із поліграфії, де він означає газетний заголовок великими літерами на всю ширину смуги – так звану «шапку»).

Розміщення вебсайту в мережі «Інтернет» та його редагування. Створений сайт почне «діяти», якщо розмістити його на вебсервері. Після цього відвідувачі зможуть «заходити» на наш сайт й ознайомлюватися з поданими на ньому матеріалами. Для передачі файлів на вебсервер існує декілька можливостей.

Розглянемо найпростіший спосіб розміщення сайту – на одному з безкоштовних вебсерверів, наприклад www.boom.ru. На цьому сервері користувачеві дається зручний інтерфейс для операцій розміщення і редагування сайту, а також до 50 Мбайт вільного простору на диску.

При передачі файлів на безкоштовний сервер використовується протокол HTTP, тобто протокол, за допомогою якого здійснюється звичайний обмін файлами в WWW. Розглянемо послідовність дій під час розміщення вебсайту.

• Підключіться до інтернету і завантажте початкову сторінку www.boom.ru.

• Клацніть по посиланню *Регистрируйтесь* и начинайте *Создание сайта*. На наступній сторінці натисніть кнопку *Начать регистрацию*, після чого завантажиться сторінка для введення ваших даних.

• Введіть ім’я свого сайту, наприклад «kurrob» (адреса сайту тоді матиме вигляд: www. kurrob.boom.ru). Наберіть також пароль і підтвердіть його в нижньому полі. У поле E-mail введіть свою електронну поштову адресу. На цю адресу ви надалі одержите лист з підтвердженням реєстрації. У розділі *Дополнительная информация о пользователе* наберіть своє прізвище й ім’я, а також псевдонім. У нижній частині сторінки є шестизначний номер. Уведіть наведений номер – він слугуватиме для захисту від автоматичних реєстрацій. Після заповнення полів натисніть кнопку *Отправить*.

• Уведені дані будуть відправлені на сервер. Якщо ім'я вашого сайту не збігається з будь-яким існуючим ім'ям, з'явиться новий діалог, у якому потрібно буде ще раз ввести ім'я сайту, пароль і E-mail. Після цього натисніть кнопку *Далее*. Якщо ж уведене вами ім'я вже існує, то вам запропонують повторно ввести дані на вихідній сторінці. Після успішної реєстрації з'явиться сторінка з вітанням і з адресою вашого сайту.

• Праворуч у верхній частині сторінки виберіть посилання *Управление файлами*, після чого з'явиться екран для вибору файлів, що завантажуються. Натисніть верхню кнопку *Browse* (Огляд) і в наступному стандартному діалозі знайдіть і вкажіть перший файл, що завантажується. Аналогічно за допомогою другої кнопки задайте другий файл для завантаження тощо. Після визначення усіх файлів натисніть кнопку *Загрузить*.

• Почекайте, доки завантажаться файли і з'явиться сторінка вмісту каталогу, на якій буде розміщений список файлів, скопійованих на сервер. Установіть прапорець ліворуч від імені файлу index.html і натисніть кнопку *Сделать главной*.

• На сторінці вмісту є додаткові кнопки вибору файлів – це на випадок, коли на сервер потрібно передати більше шести файлів і ці файли не помістилися на першу сторінку завантаження.

• Створіть окрему папку для файлів зображень, наприклад img. Для цього натисніть кнопку *Создать* і в наступному діалозі установіть перемикач *Что создать*... каталог. Натисніть кнопку *Создать*.

• На наступній аналогічній сторінці ви побачите піктограму папки img. Клацніть двічі по ній мишею, що приведе до відкриття сторінки задання вмісту папки. Задайте потрібні файли і натисніть кнопку *Загрузить*.

Задані вами файли будуть скопійовані на диск вебсерверу. Для редагування вмісту папок вашого сайту на сервері можна використати кнопки на сторінці вмісту. Наприклад, для видалення будь-якого файлу варто поставити поруч з ним прапорець і натиснути кнопку *Удалить*. Далі треба підтвердити видалення в наступному діалозі. Для переміщення файлу треба клацнути на кнопку *Переместить* і у діалоговому вікні визначити папку для переміщення.

Для перевірки роботи створеного сайту в інтернеті варто у вікні браузера набрати адресу створеного сайту й уважно переглянути всі вебсторінки, перевірити можливі переходи між ними.

Після того, як розмістили свій сайт на вебсервері, ви можете до нього звертатися для редагування вмісту (додавати, видаляти, обновляти файли). Якщо ви користуєтеся безкоштовним хостингом, то змінювати файловий склад можна за допомогою передбаченого для цього інтерфейсу.

Після зміни файлового складу сайту не відключайтеся від інтернету, а завантажте програму браузера і зайдіть на сайт, що редагується. Перегляньте уважно, як позначилися внесені вами зміни на зовнішньому вигляді і роботі вебсайту.

Після розміщення сайту на сервері його необхідно «розкрутити», тобто досягти якнайбільшої відвідуваності користувачами. Цей процес, як правило, передбачає використання банерної реклами, розміщення інформації про сайт на пошукових серверах та участь у різноманітних рейтингах.

Реєстрація сайту в пошукових системах є однією з найважливіших частин процесу «розкрути», оскільки майже всі користувачі для пошуку деякої інформації спочатку звертаються саме до пошукових серверів. Процедура реєстрації для кожного серверу різна. Перехід на сторінку реєстрації сайту здійснюється за допомогою гіперпосилання на головній сторінці пошукового сервера, яке в україномовних пошукових системах може бути «Додати сайт», «Додати URL». Потім заповнюється форма, у якій розміщується інформація про зміст сайту, його адреси, дані про організацію, яка підтримує цей сайт.

Після створення сайту, розміщення його на сервері, «розкрутки» настає процес підтримки, що передбачає постійне оновлення й актуалізацію поданої інформації. Причому чим частіше ви оновлюватимете інформацію на сайті, тим більше буде цільових відвідувачів сайту і природно споживачів вашої продукції або послуг.

Контроль відвідуваності сайту здійснюється за лічильниками. Пошукові машини, як правило, мають рейтингові системи, які

ранжирують ресурси за їх відвідуваністю. Для участі в рейтингу варто встановити на головній сторінці свого сайту лічильники рейтингів.

Нині для створення інтерактивних сайтів застосовуються різні сучасні технології: PHP, ASP, Perl, JSP, CSS, бази даних DB2, MSSQL, Oracle, Access тощо. Сучасні сайти, як правило, керовані сайти, тобто сайти, які оснащені CMS (Системою управління контентом – Content Management Systems).

Керовані сайти створюються розробниками для того, щоб інформаційну підтримку і супровід сайту (наприклад, оновлення вмісту або контента сайту) міг здійснювати сам власник сайту.

Залежно від технології створення можна виділити такі типи сайтів:

1. Статичні сайти, що містять статичні HTML або XHTML сторінки. *Статичні вебсторінки* – це статичні файли (набір тексту, таблиць, малюнків тощо), які створюються за допомогою мови розмітки HTML (мають розширення .Html або .Htm) і зберігаються в готовому вигляді у файловій системі сервера.

2. Динамічні сайти, в яких вебсторінки генеруються або формуються (створюються динамічно) в процесі виконання запиту користувача. Динамічні сайти бувають двох типів. У першому типі сайтів вебсторінки генеруються або формуються з даних, зберігаються на сервері в базі даних. У другому типі сайтів вебсторінки генеруються на стороні клієнтського додатка (в браузері).

3. *Флешсайти* – це інтерактивні програми, розроблені в середовищі Macromedia Flash. Основним інструментом розробки флешпрограм є векторна графіка (інтерактивна векторна анімація для Web). Флеш додає сайтам динамічності й інтерактивності.

4. Комбіновані сайти, в яких використовуються вищевикладені технології створення сайтів.

Сайти за взаємодією користувача з ресурсами вебсторінки можна розділити на пасивні та активні або інтерактивні.

Пасивні сайти – це сайти з пасивними вебсторінками. У пасивних сайтах користувач має можливість тільки переглядати інформацію на вебсторінках. Інтерактивні сайти – це сайти з активними вебсторінками. Під час роботи з інтерактивними вебсторінками

користувач має можливість обмінюватися даними з сервером, брати участь в інтерактивному діалозі. Статичні сайти з пасивними вебсторінками.

Технологія створення вебсторінки статичних сайтів: мова HTML (Hyper Text Markup Language), яка є мовою розмітки гіпертексту та каскадні таблиці стилів CSS (Cascading Style Sheets). CSS використовується для оформлення та форматування різних елементів вебсторінок, у результаті чого значно знижують розміри вебсторінок.

Створення вебсторінок статичних сайтів – це трудомісткий процес. Статичні сайти з пасивними вебсторінками створюються вручну, за допомогою будь-якого редактора HTML у файловій системі комп'ютера, потім завантажуються на сайт. Створення нових вебсторінок або редагування існуючих сторінок користувач виконує на ПК у редакторі, а потім знову завантажує на вебсайт.

В основному статичні сайти з пасивними вебсторінками застосовуються для створення невеликих і середніх сайтів з постійною структурою і зовнішнім виглядом сторінок (але кожна сторінка може мати свій шаблон оформлення), які можна розміщувати на будь-яких хостингах, зокрема на безкоштовних, які не підтримують роботу скриптів. Навчання школярів і студентів основам створення сайтів доцільно починати зі створення статичних сайтів з пасивними сторінками, тобто з вивчення мови розмітки HTML і каскадних таблиць стилів CSS.

Для створення сайту використовують різні засоби: редактори тексту типу *Блокнот*, візуальні редактори типу Microsoft FrontPage, Macromedia Dreamweaver і безліч інших редакторів, а також конструктори сайтів (дизайнери). Конструктори вебсайтів розміщуються на деяких сайтах у мережі «Інтернет».

Для навчання доцільно використовувати редактор Microsoft FrontPage – це візуальний HTML редактор, який входить до складу додатків Microsoft Office. Але створення реального сайту краще виконувати в русифікованому редакторі Macromedia Dreamweaver 8 з використанням технології CSS. Необхідно зазначити, що з метою швидкого завантаження вебсторінок їх розміри не повинні перевищувати 20 Кбайт.

Статичні сайти з інтерактивними вебсторінками. Для додання статичним вебсторінкам інтерактивності і динамічності у вебсторінку можна вставляти скрипти на сценаріях javascript і VBScript, виконуваних на стороні клієнта. Скрипти на javascript і VBScript можуть виконуватися або за наявності будь-яких дій користувача, або автоматично під час завантаження вебсторінки.

Крім того, в HTML-документ можна вставляти елементи DHTML (динамічний HTML). DHTML – це спосіб створення інтерактивного вебсайту. Динамічний HTML побудований на мові програмування javascript, каскадних таблицях стилів CSS і DOM (об'єктній моделі документа).

У документ HTML можна вставляти флешфрагменти або флешролики (swf-файли). У документ HTML також можна вставляти флешформи, аналогічні HTML-формам. Флеш забезпечує інтерактивність за рахунок інтерактивної векторної анімації для веб. Для створення флеш використовується мова сценаріїв ActionScript.

Для обміну даними між користувачем і сервером вебсторінку можна вставити у вебдодаток, званий HTML-формою (form). Форма – це частина вебсторінки, в якій користувач може вводити свою інформацію. Запити у формі можуть виконуватися методами GET або POST.

У зв'язку з тим, що скрипти, які виконуються на стороні клієнта, збільшують обсяг вебсторінок, їх кількість і розмір на сторінці повинні бути обмеженими. Створення статичних сайтів з інтерактивними вебсторінками доцільно виконувати в редакторі Macromedia Dreamweaver 8.

Динамічні сайти, вебсторінки яких генеруються або формуються з даних, зберігаються на сервері в базі даних.

Нині для створення динамічних сайтів застосовуються різні вебдодатки. Для розробки вебдодатків застосовуються різні технології, які забезпечують створення динамічних вебсторінок. Динамічні сайти здатні реагувати на запроваджену користувачем інформацію, тобто можуть бути інтерактивними, тому динамічні сайти, як правило, є інтерактивними.

Для розробки вебдодатків використовують два підходи: на основі компіляції модулів та на основі інтерпретації сценаріїв.

Компіляції модулів, компільовані модулі – це модулі типу CGI, які транслюються у виконувані файли і виконуються вебсервером. Першими вебдодатками для створення динамічних сайтів були окремі модулі CGI (сценарії, створені в основному на мові Perl), які виконувалися на сервері. CGI-сценарії є звичайними програмами. Результатом виконання модуля є сторінка у форматі HTML.

В інтернет можна безкоштовно завантажити модулі CGI або CGI-скрипти (гостьові книги, зворотний зв'язок, каталоги посилань, пошук по сайту тощо), які виконуються на вебсервер за запитом користувача. З іншого боку, Common Gateway Interface (CGI) – це стандартний інтерфейс обміну даних, який визначає спосіб взаємодії клієнтської програми та вебсервера. CGI забезпечує запуск скрипта на сервері і взаємодіє з ним. У подальшому для реалізації цього підходу стали застосовувати інтерфейси (серверні розширення) ISAPI і NSAPI.

Підхід на основі інтерпретації сценаріїв. У цьому випадку для створення сайту застосовуються серверні скрипти, так звані мови сценаріїв. Код сценаріїв, як і HTML-код, є інтерпретованим кодом, тому HTML і сценарії можна комбінувати. Найбільш поширені мови серверних скриптів: Perl, ASP, JSP, PHP, Cold Fusion, Python.

Сценарії взаємодіють з об'єктами на сервері і генерують вихідну інформацію у форматі HTML. Тип серверного скрипта визначається з розширення імені файлу (. Php,. Asp,. Aspx,. Jsp,. Cfm). Якщо вебсервер отримує запит на сторінку такого типу, то він інтерпретує всі сценарії, які містяться в ній, у результаті чого генерується вебсторінка у форматі HTML, яка передається назад браузеру.

Найбільш популярними технологіями (середовищем розробки) створення динамічних вебсторінок є: CGI, PHP, ASP, ASP.NET, JSP, Cold Fusion, AJAX, Python, CSS, бази даних DB2, MsSQL, Oracle, Access тощо.

Залежно від розв'язуваних завдань для створення сайту обирають певну мову серверних скриптів. Для створення малих і середніх інтерактивних сайтів доцільно застосувати мову сценаріїв

PHP. Конкурентами PHP є технології ASP, JSP, Cold Fusion, Perl. Гідністю мови PHP є те, що вона є безкоштовною, має відкриті вихідні коди і працює майже на всіх платформах.

Для створення (розробки) і супроводу динамічних сайтів використовують CMS (Content Management System) – систему управління сайтом, яку називають движком сайту. Нині популярними системами управління є Drupal, Joomla та WordPress. На основі цих CMS можна створювати функціональні і легко керовані PHP-сайти. Движки для Drupal, Joomla та WordPress є безкоштовними. Засоби розробки сайтів забезпечують поділ змістовної частини (контенту) від дизайну (шаблону вебсторінки), що дозволяє змінювати зміст вебсторінок, не зачіпаючи їх дизайну, і змінювати шаблон сайту, не зачіпаючи змісту його сторінок.

Для підтримки навчального процесу традиційного навчання школярів і студентів доцільно створювати динамічні інтерактивні сайти, наприклад за допомогою движка Joomla. Цей движок має безліч модулів: форуми, гостьові книги, поштові розсилки, контакти, опитування, форми реєстрації, форми пошуку, систему обміну повідомленнями користувачів сайту та інші товари, які перетворюють сайт із засобу інформації у засіб комунікації.

У цьому випадку сайт буде місцем активного обміну інформацією між користувачами інтернету (учнями або студентами і викладачами). На такий сайт користувачі можуть самостійно додавати електронні навчальні матеріали, статті, фотографії, відео, безкоштовно викачувати освітні ресурси, тобто користувачі мають можливість обмінюватися даними із сервером. Крім того, учні або студенти через опитування можуть оцінювати роботу викладачів, висловлювати свою думку з різних питань, спілкуватися між собою, тобто брати участь в інтерактивному діалозі.

Динамічні сайти, вебсторінки яких генеруються на стороні клієнтського застосування. Для створення таких сайтів використовують мови сценаріїв javascript і VBScript, а також Java-аплети і технологію ActiveX. В інтернеті можна скачати движок для сайту на javascript.

Флешсайти. Технологія флеш призначена для створення векторних графічних додатків. За її допомогою можна створити повноцінну сторінку-ролик для веб, але при цьому інформація розбивається на великі файли, для завантаження яких потрібно багато часу. Нині більш доцільним є застосування флеш як елементів дизайну в HTML-документах (наприклад, для створення логотипів, флешменю, інформерів та інших анімаційних графічних елементів) як анімованих флешбанерів і вхідних флешзаставок.

Флеш доцільно використовувати там, де мало тексту, але потрібні звукові або анімаційні ефекти, тобто там, де флеш забезпечує створення векторних анімаційних файлів з невеликим часом завантаження. Основні недоліки цієї технології створення повноцінних флешсайтів: велика вага вебсторінок і висока вартість розробки сайтів.

Крім того, сайти, створені повністю на основі флеш, погано індексують пошукові системи. Флештехнології в основному застосовуються для створення престижних сайтів. Для створення флешанімацій застосовують технологію Adobe Flash, що забезпечує можливість працювати з мовами: ActionScript і ActionScript 2.0.

Використання сценаріїв. Сценарії використовуються для надання вебсторінкам динамічності. Наприклад, за допомогою сценаріїв можна перевіряти правильність введених у форму даних перед тим, як відправити їх на сервер. Якщо ви будь-що можете помістити на вебсторінку, то можна приєднати сценарій або керувати їм за допомогою сценарію.

Найпоширенішою та найпершою мовою сценаріїв є мова JavaScript, хоча лідерство в неї прехоплює мова VBScript, розроблена фірмою MicroSoft на основі мови Visual Basic. Але, по-перше, мову підтримують JavaScript і Internet Explorer. По-друге, JavaScript – могутня мова сценаріїв, що задовольняє більшість виникаючих вимог. VBScript є альтернативною мовою сценаріїв, і вона являє собою спрощену версію мови Visual Basic.

Для додавання VBScript, JavaScript на вебсторінку використовується дескриптор <SCRIPT>, що має єдиний атрибут LANGUAGE. Привласніть цьому атрибутові значення VBSCRIPT,

щоб вибрати мову VBScript. Оскільки дескриптор <SCRIPT> є контейнером, необхідно використовувати завершальний дескриптор </SCRIPT>. Ось як це виглядає:

```
<SCRIPT LANGUAGE= VBSCRIPT >
Оператори VBScript I </SCRIPT>
```

Можна і не використовувати атрибут LANGUAGE. Якщо ви не вкажете цей атрибут, то більшість браузерів використовуватиме мова JavaScript, хоча це і не визначено ні консорціумом W3C, ні будь-яким стандартом.

Якщо веббраузер не підтримує VBScript чи JavaScript, то він ігнорує дескриптори <SCRIPT> і </SCRIPT>, однак відображає усе, що міститься між ними, припускаючи, що там розміщено якийсь текст HTML.

Для того щоб сховати текст сценарію, використовуйте коментар. Перед блоком коду сценарію необхідно коментар відкрити, а після останнього оператора – закрити. Веббраузери, що підтримують JavaScript, ігнорують коментарі, а інші ігнорують усе, що в них розміщується.

```
<SCRIPT LANGUAGE=JAVASCRIPT>
<! – -
function AddSeries(last)
intTotal = 0;
for(j = 1; j <= last; j++)
{
intTotal+=j;
alert(intTotal);
}
return intTotal;
}
document.write(”Howdy”);
//--> </SCRIPT>
```

Уміння вбудовувати сценарії у вебсторінки і завантажувати їх через інтернет недостатньо, адже іноді вам потрібно буде перехопити якусь подію від об’єктів, розміщених на вебсторінці. Для цього потрібно навчитися підключати сценарії до подій.

Більшість об’єктів на вебсторінці має події.

Таблиця 9.8

Основні події HTML

Назва	Умова виникнення
OnBlur	Поле форми втрачає фокус
OnChange	Вміст поля або списку змінюється
Onclick	Користувач клацає на об’єкті
OnFocus	Поле форми одержує фокус
OnMouseOut	Покажчик миші виходить за межі об’єкта
OnMouseOver	Покажчик миші переміщується над об’єктом
OnSelect	Користувач вибирає поле форми
OnSubmit(submit)	Користувач клацає на кнопці відправлення

Об’єкти, події й оброблювачі подій взаємодіють. Оброблювач подій з’єднується з об’єктом, використовуючи подію. З’єднання здійснюється за допомогою додаткових атрибутів подій, яким привласнюється один оператор, що виконує функцію. Наприклад, для зв’язку функції на JavaScript, що називається MyFunction (), з подією onclick кнопки необхідно, щоб дескриптор <INPUT> виглядав так:

<INPUT TYPE=BUTTON NAME=BUTTON onClick=”MyFunction”>

Так само просто з’єднується оброблювач подій і з іншими подіями. У нашому випадку атрибут події – onClick. Можна додати оброблювач ще однієї події і з’єднати його з будь-якою подією:

<INPUT TYPE=BUTTON NAME=BUTTON onClick= ”MyFunction()” onMouseOut=”ByeMouse()”>

Припустимо, ви завантажили сценарій, що відображає в рядку стану повідомлення, коли користувач наводить покажчик миші на будь-яке посилання на сторінці. Ви розміщуєте цей сценарій на своїй сторінці, а він не працює. Проблема полягає в тому, що він не пов’язаний з подіями. Необхідно з’єднати функцію Message () з подією onMouseOver для кожного дескриптора <A> у файлі HTML, як показано нижче.

```
<SCRIPT LANGUAGE=JAVASCRIPT>
<!--
function Message ()
{
window.status = ”Покажчик миші розташований над посиланням!”;
}
//-->
</SCRIPT>
```

З’єднання нового сценарію.

```
<HTML> <HEAD>
<TITLE>З’єднання нового сценарію</TITLE> </HEAD>
<BODY>
<A HREF=”more.htm” onMouseOver=”Message()”>Уперед</A>
<A HREF=”less.htm” onMouseOver=”Message()”>Назад</A>
</BODY>
</HTML>
```

Створення інтерактивних вебсторінок з допомогою мови сценаріїв VBScript.

VBScript – це мова вебсценаріїв. Вона є однією з діалектів мови VBA і також вбудована в додатки MS Office, як і VBA. Редактор VBScript можна викликати в додатках MS Office, виконавши команду *Сервис|Макрос|Редактор Сценариев* або натиснувши сполучення клавіш Alt+Shift+F11.

VBScript дозволяє:

1. Вставляти в HTML-документи програми обробки зі сторони клієнта.

2. Одержувати доступ до властивостей, методів і подій:

2.1. Елементів HTML-документів;

2.2. Active-X елементів HTML-документів;

2.3. Об’єктів Internet Explorer.

VBScript – інтерпритатор, який вбудований у браузер, і відповідно браузер може запускати програми, написані на VBScript. Тобто ці сценарії виконуються на стороні клієнта.

На відміну від сценаріїв VBScript, ASP-сценарії (Active Service Page) дозволяють створювати сценарії, які виконуватимуться на стороні сервера, тобто ASP-сценарії передають у браузер не саму

програму, а тільки тест, і HTML-дескриптори згенеровані вебсервером. За допомогою моделі ADO (AvtiveX Data Object) ASP також доступні бази даних, які зберігаються на сервері. Але для використання ASP-сценаріїв недостатньо офісних додатків, редакторів Visual Basic for Application і редактора сценаріїв VBScript. Для цього необхідно встановити також вебсервер Internet Information Server, який би міг інтерпретувати ASP-сценарії, а також якщо потрібний доступ до даних, які зберігаються в базі даних, бажано встановити систему управління базами даних Microsoft SQL Server.

На сьогодні досить велика кількість вебсерверів і вебсайтів працюють на платформі Windows з використанням Internet Information Server, і відповідно створено велику кількість програм за допомогою ASP для доступу до інформаційних ресурсів інтернету.

Запитання для самоконтролю

1. Що називається вебсторінкою?
2. Що таке гіпертекст?
3. Що таке тег?
4. За допомогою яких програм створюють вебдокументи?
5. За допомогою яких програм переглядають вебдокументи?
6. Що називається вебсайтом?
7. Яку загальну структуру мають усі HTML-документи?
8. Як здійснюється форматування тексту мовою HTML (заголовки, абзаци, вирівнювання тексту, вибір стилів шрифту, зміна розміру шрифту)?
9. Які типи гіперпосилань використовуються у HTML-документах?
10. Як включити у HTML-документи графічні зображення?
11. Як включити у HTML-документи таблиці?
12. Що таке фрейми і які способи їх використання?
13. Які етапи планування і створення сайту?
14. Як створити вебсайт?
15. Як розмістити вебсайт на вебсервері?
16. У чому полягає процес підтримки власного вебсайту?
17. Що таке світова павутина World Wide Web?

ЛЕКЦІЯ 10
ЗАСТОСУВАННЯ ІНТЕРНЕТ-ТЕХНОЛОГІЙ В ЕКОНОМІЦІ. ЕКСПЕРТНІ І НАВЧАЛЬНІ СИСТЕМИ

10.1. Телеконференції (групи новин) в економічній діяльності

Групи новин (їх також називають телеконференціями) – це засіб, що дозволяє людям, що мають спільні інтереси, спілкуватися між собою за допомогою електронної пошти. Наприклад, учені, що працюють у певній галузі науки, мають свою групу новин, через яку вони обмінюються науковими статтями, різними оголошеннями, ведуть дискусії тощо. У групі новин автолюбителів можна знайти оголошення про продаж і купівлю автомобілів, запчастин, іншу подібну інформацію. Якщо, наприклад, у вас виникли складні технічні проблеми з вашим автомобілем, то ви можете послати в групу новин своє запитання, і серед її читачів знайдеться багато охочих, що надішлють вам відповідь чи дадуть пораду. Отже, групи новин – це своєрідний варіант клубів спілкування за інтересами, в роботі яких можуть брати участь усі бажаючі. Для цього не треба вступати в будь-які організації чи сплачувати членські внески.

Групи новин бувають керовані і некеровані. *Керівник* – це досвідчений фахівець, який переглядає повідомлення, що надійшли у групу новин, і вирішує, чи відповідають вони її тематиці і чи варто їх тут публікувати. Керівник часто сам відповідає на запитання дописувачів і публікує в групі новин свої статті. Переважна більшість груп новин є некерованими. Будь-яка інформація, що надійшла на їх адресу, публікується в них автоматично.

Термін зберігання повідомлень на сервері новин обмежений. Через певний час інформація, що втратила актуальність, вилучається, звільняючи місце для нових надходжень.

Групи новин створюються і припиняють свою роботу відповідно до того, як виникає чи зникає потреба в їх існуванні.

Останнім часом їх популярність взагалі знизилась, бо можливість аналогічного спілкування надається сервісом WWW через так звані форуми. Досить часто наприкінці статті чи іншого повідомлення можна знайти посилання на вебсторінку форуму, де всі бажаючі можуть поставити запитання автору, взяти участь в обговоренні, поділитися власним досвідом тощо.

Можливо, в майбутньому WWW поглине і цей сервіс, проте на сьогодні групи новин ще мають перспективу хоча б тому, що інформація в них поширюється через електронну пошту, безкоштовне користування якою нині загальнодоступне, тоді як можливості безкоштовного доступу до WWW ще досить обмежені.

Підписка на групи новин. Щоб отримати доступ до сервера новин, треба вибрати в меню *Сервис – Учетные записи – Новости – Добавить – Новости…* Потім ввести своє ім'я, електронну адресу та адресу сервера новин, наприклад *news.lucky.net.* Далі натиснути кнопки *Готово, Закрыть* та дати позитивну відповідь на пропозицію завантажити список груп новин цього сервера. Цей список можна також викликати, натиснувши на панелі інструментів кнопку *Группы новостей*, чи обравши однойменну опцію контекстного меню папки цього сервера новин, яка щойно з'явилась на панелі папок.

Якщо список груп новин досить довгий, то в ньому можна провести фільтрацію. Наприклад, якщо вас цікавлять новини, що стосуються комп'ютерів чи музики, то можна відфільтрувати назви груп за ознакою наявності в них слів *comp* чи *music.*

Щоб підписатися чи відмовитись від підписки на отримання повідомлень певних груп новин, треба в цьому списку двічі клацнути мишею по обраній групі або виділити її і скористатися кнопками *Подписаться* чи *Отказаться от подписки*. У результаті на панелі папок вашої програми в папці сервера новин будуть створюватися чи вилучатися папки обраних вами груп.

Завантаження, перегляд, сортування, фільтрація та пошук повідомлень. Звичайні налаштування програми забезпечують завантаження й оновлення переліку повідомлень у папках груп новин тоді, коли ви відкриваєте ці папки для перегляду. Так само й текст повідомлення завантажується тоді, коли ви виділяєте його

заголовок у переліку. Отже, можете переглядати повідомлення в папках груп новин так само, як переглядаєте листи в інших папках програми.

Як правило, повідомлень у папці багато, тому для ефективної роботи з ними треба вміти використовувати різні варіанти представлення, сортування, фільтрації та пошуку повідомлень. Звісно, повідомлення можна сортувати так само, як і листи, клацнувши мишою по відповідному заголовку стовпця в переліку. Однак робота з повідомленнями має й деякі особливості. Наприклад, у групу новин можуть надходити відгуки на повідомлення, розміщені тут раніше. У переліку повідомлень такі повідомлення і відгуки зручно тримати об'єднаними у списки, які можна згортати і розгортати за допомогою миші. Для цього треба встановити позначку в меню *Вид – Текущее представление – Сгруппировать сообщения по теме обсуждения.* Якщо, наприклад, ви хочете швидко переглянути відгуки на своє повідомлення в групі новин, виберіть опцію *Отобразить ответы на мои сообщения.* Багато інших корисних можливостей для роботи з повідомленнями у групах новин знайдете в меню *Сервис – Правила для сообщений – Новости*.

Пошук повідомлень у групі новин виконується за ключовими словами, розміщеними в полі *Тема,* за допомогою опції *Найти* в меню *Правка.* Тому поле *Тема* має бути максимально інформативним. Якщо, наприклад, ви надсилаєте в групу новин оголошення про продаж автомобіля, то слово *продаю*, марку і найголовніші характеристики автомобіля варто вказати у цьому полі, бо саме, переглядаючи інформацію в полі *Тема*, користувач вирішує, чи варто взагалі завантажувати це повідомлення. На відміну від листів повідомлення в групу новин з незаповненим полем *Тема* програма Outlook Express взагалі не надсилає.

Підготовка і відправлення повідомлень у групу новин виконується аналогічно підготовці й відправленню листів, хоча набір кнопок на панелі інструментів під час роботи з групами новин відрізняється від набору кнопок у процесі роботи з листами. Так само й поля редактора, що застосовуються для підготовки тексту повідомлень, дещо відрізняються від полів у редактора листів.

Якщо бажаєте направити в групу новин зовсім нове повідомлення, яке, можливо, започаткує там нову тему для обговорення, то виділіть цю групу на панелі папок і натисніть кнопку *Создать сообщение* 

. Якщо ж ви відповідаєте на інше повідомлення в цій групі новин, тоді треба виділити це повідомлення в переліку і натиснути одну з кнопок 

.

Перша з них, що називається *Ответить в группу,* використовується, щоб надіслати вашу відповідь у групу новин, отже, з нею зможуть ознайомитись усі, хто підписався на цю групу.

Друга кнопка *Ответить отправителю* слугує для того, щоб надіслати звичайного листа автору цього повідомлення. У групу новин нічого не передається.

Третя – *Ответить всем* надсилає вашу відповідь і в групу новин, і автору.

Після натискання кнопки відкривається вікно відповідного редактора, де частина полів може бути вже заповнена деякою попередньою інформацією, яку ви можете скоригувати і доповнити. Закінчивши редагування повідомлення, натисніть кнопку *Отправить*.

Прикладні програми перегляду вебсторінок. Сутність системи World Wide Web (WWW) полягає в застосуванні гіпертекстової моделі до інформаційних ресурсів, розподілених у глобальній мережі. WWW – один з видів сервісу інтернет. WWW надає можливість роботи з документами, в яких об'єднані текст, графічні зображення, звуки, анімація, що значно полегшує сприйняття інформації. Гіпертекстові документи (вебсторінки) створюються за допомогою спеціальної мови розмітки гіпертексту HTML (Hyper Text Markup Language).

Система WWW працює за принципом клієнт-сервер. *Клієнт* – це інтерпретатор HTML, спеціальна програма перегляду, звана WWW-браузер (WWW-browser). WWW-браузер – це прикладна програма, яка взаємодіє із системою WWW, отримує документи, що зажадалися, інтерпретує дані і відображає вміст документів на екрані. Програма клієнт забезпечує доступ практично до всіх інформаційних ресурсів інтернету, які зберігаються на серверах.

Для роботи із системою WWW необхідно встановити на своєму комп'ютері одну з програм перегляду вебсторінок, наприклад Internet Explorer, Mozilla Firefox, MYIE Web Browser, Opera тощо. Більшість браузерів надають доступ до інших серверів інтернету: FTP-серверів, Gopher-серверів і серверів телеконференцій UseNet.

Оглядач Internet Explorer. Програма Internet Explorer встановлюється разом з інсталяцією Windows, і на робочому столі з'являється значок оглядача Internet Explorer. Цей значок може використовуватися для запуску програми Internet Explorer. Після запуску програми перегляду вебсторінок Internet Explorer на екрані з'явиться вікно оглядача (рис. 10.1).

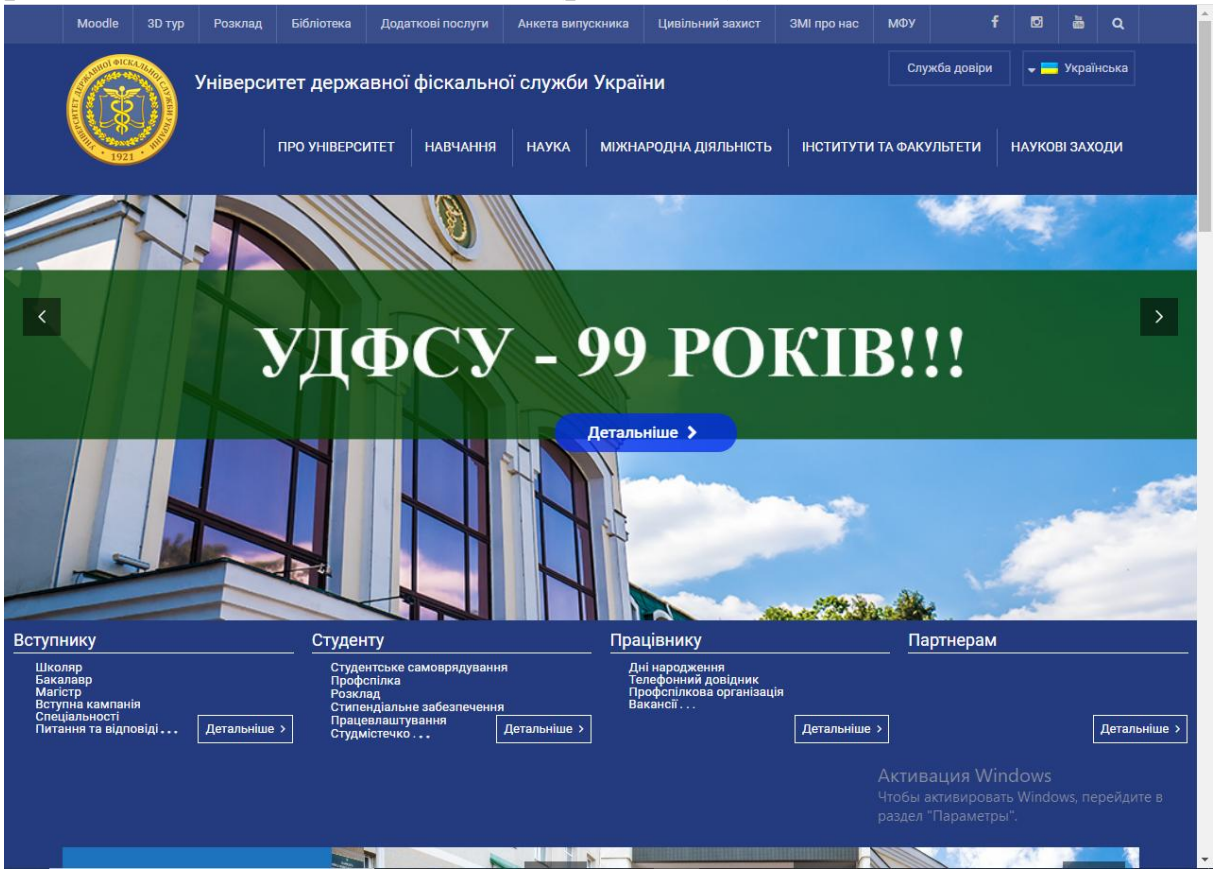


Рис. 10.1. Вікно програми Internet Explorer

Вікно програми перегляду, або оглядача Internet Explorer, містить:

- рядок меню;
- три панелі інструментів: *Обычные кнопки*, *Адрессная строка*, *Ссылки*;
- рядок стану.

Панель *Обычные кнопки* містить усі основні кнопки управління оглядачем. Панелі *Адресс* і *Ссылки* використовуються для введення адреси завантажуваної вебсторінки.

Рядок стану, що перебуває в нижній частині вікна, містить повідомлення про стан вебсторінки. Індикатор процесу в правій частині рядка стану відображає процес завантаження файлу. У рядку стану також є індикатор, що показує встановлений рівень безпеки завантажуваних вебсторінок.

Для переходу на вебсторінки можна використовувати:

- гіперпосилання, що містяться у відкритій сторінці;
- панель *Адресс*, на якій вводиться адреса необхідної вебсторінки;
- панелі *Ссылки* і *Избранное*.

Кожна вебсторінка містить посилання на інші сторінки. Найбільш простим способом переходу від однієї вебсторінки до іншої є перехід за допомогою маніпулятора миша. Для переходу на інші сторінки можна використовувати меню навігація, текстові та графічні гіперпосилання.

Зазвичай текстові гіперпосилання підкреслені, тому їх легко можна знайти. Текстові заслання до тих сторінок, які ви вже відвідали, змінюють свій колір. Більш складно визначити графічні гіперпосилання. Якщо під час наведення курсора на зображення він набирає вигляду долоні з вказуючим перстом, то це гіперпосилання.

У процесі натискання на *Ссылки* вказана в ній вебсторінка за умовчанням відкривається в поточному вікні оглядача. Але її можна відкрити і в новому вікні. Для цього встановіть курсор на *Ссылки*, натискайте праву кнопку миші і з контекстного меню, що з'явилося, виберіть команду *Відкрити* в новому вікні.

Для здійснення переходу на вебсторінку можна використовувати її адресу, яка починається з назви протоколу. Потім слідує ім'я домена і розташування файлу відносно домена. За замовчуванням панель *Адресс* розміщується під основною панеллю інструментів оглядача.

Коли вибирається адреса зі списку, що міститься на цій панелі, то оглядач відкриває вебсторінку за вказаною адресою. Усі адреси, які ви раніше вводили на панелі *Адресс*, зберігаються в списку, що розкривається.

У панель *Адресс* можна вводити URL сторінки безпосередньо з клавіатури. Під час введення адреси можна не вводити початкову частину адреси http://www, оскільки оглядач додасть її сам.

Для використання папок *Ссылки* і *Избранное* для переходу на вебсторінки спочатку в папках *Ссылки* і *Избранное* треба зберегти адреси відібраних вебсторінок. Для збереження адреси відкритої вебсторінки необхідно клацнути мишею на меню Избранное і вибрати команду *Добавить в Избранное*.

Відкриється вікно діалогу *Добавить в Избранное*, в якому необхідно вибрати папку для збереження адреси (зокрема, і папку *Избранное*). Після цього для переходу на необхідну сторінку клацніть на меню *Избранное* і виберіть у папках потрібну адресу вебсторінки, оглядач відкриває вебсторінку, адресу якої ви вказали.

Якщо вас не задовольняє швидкість завантаження вебсторінок, можна відмінити виведення графічної інформації.

Для цього треба виконати такі дії:

1. Виконати команду *Свойства* оглядача меню *Сервис*.
2. У діалоговому вікні *Свойства* оглядача перейти на вкладку *Додатково*.
3. У розділі *Мультимедіа* зняти прапорець *Отображать* рисунки.

Якщо вигляд кодування сторінки не збігається зі встановленим в оглядачі і виводиться безглуздий набір символів, то треба змінити вид кодування за допомогою команди *Вид / Кодировка*. Найбільш поширеними видами кодування україномовного тексту є Кирилиця (КОЇ8) і Кирилиця (Windows).

Вебсторінка складається з HTML-документа й інших файлів, що використовують для оформлення сторінки. За допомогою оглядача Internet Explorer можна зберегти як усю сторінку, так і окремі її фрагменти.

Для збереження вебсторінки треба виконати такі дії:

1. Вибрати команду *Файл / Зберегти*, коли відкриється діалогове вікно *Збереження* вебсторінки.

2. Вибрати папку і задати ім'я створюваного файлу.

3. У полі *Тип файла* вкажіть тип даних, що зберігаються:

- вебсторінка, повністю – застосовується для збереження всієї сторінки повністю;
- вебсервер-архів, один файл – уся вебсторінка збережеться у вигляді одного файлу;
- вебсторінка, лише HTML – використовується для збереження лише HTML-документа без малюнків та інших файлів;
- текстовий файл – для збереження вебсторінки у звичайному текстовому форматі.

4. Натиснути кнопку *Зберегти*.

Для збереження фрагмента тексту виділити його і натиснути праву кнопку миші. Відкриється контекстне меню, з якого вибрати команду Копіювати, фрагмент збережеться в тимчасовому буфері обміну Windows. Тепер його можна вставити, наприклад, у документ Word.

Малюнок можна зберегти у вигляді окремого файлу. Для цього з контекстного меню малюнка виконати команду *Зберегти зображення як*. Під час збереження малюнка можна вибрати тип файлу або зберегти малюнок у форматі bmp.

Електронна пошта. Система обміну повідомленнями є одним з найдоступніших засобів спілкування в інтернеті і локальних мережах.

Сутність роботи поштової системи полягає в такому:

- відкриваєте вікно поштового застосування;
- обираєте режим *Написати лист*;
- набираєте текст листа;
- вказуєте адресу одержувача;
- вказуєте тему листа;
- обираєте команду *Відправити негайно*, тобто поміщаєте в поштову скриньку, що міститься на поштовому сервері;
- з поштової скриньки лист забирається поштовою службою і доставляється в поштову скриньку адресата.

Обмін повідомленнями здійснюється за рахунок взаємодії двох програм – поштового сервера і поштового клієнта. Програмне

забезпечення *Поштовий сервер* встановлюється, наприклад, у провайдера, для здобуття пошти мережею «Інтернет» або в локальній мережі компанії для обміну повідомленнями між співробітниками.

Поштові клієнти – це прикладні програми (наприклад, Outlook Express, Microsoft Outlook, The Bat), які встановлюються на комп'ютерах користувачів, які відправляють або приймають пошту. Усі поштові клієнти мають приблизно однаковий інтерфейс, тому достатньо навчитися працювати з одним із них.

Для роботи з електронною поштою можна використовувати як поштові клієнти, так і поштові вебсервери – інтерфейси, які розташовуються на поштових вебсерверах – серверах. За допомогою вебсервера – інтерфейсу можна працювати з поштою безпосередньо на поштових вебсерверах – серверах. За функціональністю вебсервер – інтерфейс максимально наближений до поштових програм.

Поштові системи на основі WWW дозволяють обробляти поштові повідомлення в інтернеті за допомогою звичайного браузера, а не поштової програми. Вони працюють за принципом «2 в 1», поєднуючи в собі функції поштового сервера і поштового клієнта.

Використання вебсервера – інтерфейсу є доцільним у тому випадку, коли потрібно відправляти або приймати кореспонденцію з різних комп'ютерів, маючи доступ до мережі «Інтернет».

Для того щоб дістати доступ до роботи з електронною поштою, необхідно створити поштову скриньку на одному з поштових серверів. Для створення поштової скриньки необхідно зареєструватися на одному з поштових серверів, наприклад http://www.mail.ru, http://www.hotmail.ru/ тощо.

Для роботи з поштовими програмами (поштовими клієнтами) їх необхідно встановити на ПК. Електронна пошта заснована на двох прикладних програмах: на протоколі для передачі повідомлень SMTP (Simple Mail Transport Protokol) і протоколі для одержання повідомлень.POP3 (Post Office Protokol 3, поштовий протокол 3) або протоколі доступу до повідомлень в інтернеті Internet Message Access PROTOCOI (lMAP4 – остання версія).

Щоб мати можливість відправляти або отримувати пошту за допомогою поштових клієнтів, необхідно створити хоча б один обліковий запис. *Обліковий запис* – це набір даних про користувача поштового сервера (ім'я, електронна адреса, пароль для підключення до сервера, імена серверів вхідної і вихідної пошти). Створення облікових записів здійснюється в поштовій програмі.

Адреса електронної пошти складається з двох частин, розділених символом @, і виглядає приблизно так: moi@ukr.net. Символи, які вказані в лівій частині адреси (до знака @), є ім'ям адресата або ім'ям поштової скриньки, воно часто збігається з логіном користувача. Права сторона електронної адреси, яка слідує після знака @, є доменним ім'ям поштового сервера, на якому розташована поштова скринька.

10.2. Мережеві технології в економіці

Відомо, що інтернет – це засіб телекомунікацій, зберігання і надання різної інформації, а також середовище для економічної діяльності і ведення електронного бізнесу.

Розвиток інформаційних і телекомунікаційних технологій сформував середовище для економічної діяльності в інтернеті. Утворився новий інтерактивний канал взаємодії компаній з бізнес-партнерами і клієнтами, що забезпечило ведення інтерактивного маркетингу і здійснення прямого онлайнового продажу.

Відомо, що мережева економіка – це господарська (економічна) діяльність, що базується на горизонтальних (прямих) тривалих зв'язках між усіма учасниками спільної діяльності в інформаційно-комунікаційному середовищі мережі «Інтернет». Це якісно нова форма управління, яка відрізняється від командно-ієрархічної (централізованої) і ринкової форм управління економічною діяльністю.

Мережева економіка може жити лише в середовищі мережі «Інтернет». Вона стає базисом для електронного бізнесу, основною складовою якого є електронна комерція, яка не може існувати поза мережею.

Наприклад, прямий продаж товарів безпосередньо споживачам, а не через посередників, є новою моделлю ведення бізнесу. Отже, традиційні бізнес-процеси (продаж, маркетинг, постачання тощо) в мережевій економіці набувають нових форм. Сьогодні комерційна діяльність в інтернеті стала доступною всім.

Електронний бізнес – це підприємницька діяльність, заснована на використанні інформаційних і телекомунікаційних технологій, що забезпечують взаємодію суб'єктів економічної діяльності в комп'ютерних мережах, з метою здобуття прибутку.

Розвиток інформаційної інфраструктури загальнодоступної інтерактивної мережі «Інтернет», а також простота використання мережі об'єднали безліч покупців і продавців інформацією, товарами, послугами, а взаємодія суб'єктів економічної діяльності сформувала нові моделі ведення бізнесу.

Інтерес мають основні моделі ведення бізнесу або моделі взаємодії суб'єктів ринку в комп'ютерних мережах:

B2B – бізнес для бізнесу (Business-to-Business), електронний бізнес, орієнтований на бізнес-партнера;

B2C – бізнес для споживача (Business-to-Consumer), електронний бізнес, орієнтований на кінцевого користувача.

B2A – бізнес-адміністрація (Business-to-Administration) – визначає взаємодію компаній з адміністративними органами;

C2A – потребитель-адміністрація (Consumer-to-Administration) – визначає взаємодію споживачів з адміністрацією;

C2C – споживач для споживача (Consumer-to-Consumer), електронний бізнес, орієнтований на кінцевого користувача.

B2B – це бізнес-модель взаємодії компаній між собою за допомогою комп'ютерних мереж;

Основу бізнес-моделі B2C становить роздрібна торгівля, тобто взаємодія компаній із споживачами в мережі «Інтернет»;

C2C – модель продажу товарів і послуг одного споживача іншим споживачам, тобто взаємодія споживачів із споживачами в мережі «Інтернет».

Відповідно до статистичних досліджень у мережі напрям B2B займає перше місці і вважається найбільш перспективною моделлю ведення бізнесу в інтернеті.

Крім того, існує ще ряд моделей ведення бізнесу. У матеріалах Європейської комісії в ESPRIT розглядаються такі моделі електронної комерції:

- електронний магазин;
- електронний довідник-каталог;
- електронний онлайновий аукціон;
- електронний торговельний центр;
- віртуальне співтовариство;
- віртуальний центр розробки;
- інформаційний брокер тощо.

До основних етапів ведення електронної комерції належать: онлайновий маркетинг, оформлення заявок, укладення договорів і контрактів, електронне управління закупівлями сировини і витратних матеріалів, здійснення платежів та інформаційна підтримка доставки товарів, сервісне і післяпродажне обслуговування клієнтів тощо.

Основні можливості бізнесу в інтернеті. Нині існує два основні напрями використання інтернету в бізнесі: інтернет як засіб комунікації, джерело довідкової інформації, засіб реклами і маркетингу для ведення бізнесу (господарської діяльності) поза електронними мережами й інтернет як інструмент ведення електронного бізнесу, заснованого на принципах мережевої економіки.

Інтернет в бізнесі. Інтернет відіграє величезну роль у бізнесі як засіб комунікацій (E-mail, телеконференції, IP-телефонія тощо). Інтернет не просто засіб комунікації, а інтерактивний канал взаємодії підприємців, що дозволяє їм спілкуватися в режимі онлайн.

Велику роль відіграє інтернет як джерело довідкової інформації для компаній. Уся інформація про товари і послуги, що надаються компаніями в інтернеті, представлена у вигляді сайтів. Інтернет є засобом масової інформації, на серверах мережі зберігається значна кількість інформаційних ресурсів (сотні мільйонів вебсторінок), пошук яких забезпечують інформаційно-пошукові системи.

Найважливішим завданням формування іміджу фірми (public relations) є формування привабливого образу фірми в очах громадськості. Це досягається різними засобами: рекламою, добродійними

акціями, спонсорством, випуском пресрелізів та інформаційних матеріалів про діяльність фірми тощо.

Використання інтернету дозволяє істотно понизити витрати на public relations. Зменшення вартості досягається за рахунок зниження накладів і скорочення кількості рекламних матеріалів, так рекламні матеріали можна представити на вебсайті компанії.

Реклама в інтернеті, на відміну від традиційної реклами, має свої особливості. В інтернеті необхідно розрізняти два типи реклами: рекламу вебсайту і рекламу товарів або послуг. Для реклами, або «розкручування» вебсайту зазвичай використовують реєстрацію сайту в пошукових системах, обмін банерами або заславнями, вставку META тегів: title, keywords, description, внесення до тексту ключових слів, розміщення платних заслань на популярних вебсерверах, вказівку адрес електронної пошти й вебсайту у всіх рекламних оголошеннях тощо.

Розсилка e-mail, реклама в новинних розсилках, розміщення реклами на своїх рекламних сайтах і сайтах популярних вебсерверів, а також дошках оголошень є ефективним методом реклами товарів і послуг в інтернеті.

Бізнес в інтернеті. Розвиток інформаційної інфраструктури глобальної мережі «Інтернет» привів до змін способів ведення бізнесу і появи електронного ринку. Відомо, що електронний бізнес – це підприємницька діяльність, в якій використовуються можливості інформаційних технологій з метою створення прибутку.

Створення інформаційних продуктів (книг, інструкцій, програмного забезпечення тощо) і прямий продаж їх через віртуальні магазини є поширеним видом електронного бізнесу або інформаційного бізнесу в мережах «Інтернет». Інформація – це найбільш прибутковий продукт XXI століття.

Основними секторами ринку або моделями взаємодії суб'єктів ринку в комп'ютерних мережах є:

B2B – бізнес для бізнесу (Business-to-Business);

B2C – бізнес для споживача (Business-to-Consumer).

B2B – це бізнес-модель взаємодії компаній між собою за допомогою комп'ютерних мереж.

Основу бізнес-моделі B2C становить роздрібна торгівля, тобто взаємодії компаній зі споживачами в мережі «Інтернет».

10.3. Хмарні технології як основа бізнес-середовища

Хмарні технології (cloud technologies) – це парадигма, що передбачає віддалену обробку та зберігання даних. Ця технологія надає користувачам мережі «Інтернет» доступ до комп'ютерних ресурсів сервера і використання програмного забезпечення як онлайн-сервісу. Тобто якщо користувач має підключення до інтернету, то він може виконувати складні обчислення, опрацьовувати дані, використовуючи потужності віддаленого сервера. Дослівний переклад слова *cloud* означає *«хмара»* (звідси і термін хмарні технології), однак в іншому значенні це слово перекладається як «розсіяний, розподілений». Отже, хмарні технології, по суті, є *«розподіленими технологіями»*, тобто опрацювання даних відбувається з використанням не одного стаціонарного комп'ютера, а розподіляється по комп'ютерах, підключених до інтернету.

Серед переваг хмарних технологій можна виділити такі: економія коштів; відсутність потреби у кваліфікованих фахівцях для установки і підтримки програмного забезпечення; доступ до інформації з будь-якого місця, де є покриття інтернет; немає необхідності в потужному обладнанні; зобов'язання і гарантії захищені договором аутсорсингу; відсутність можливості фізичного доступу до даних.

Зокрема, в роботі юриста «хмарний» сервіс дозволяє працювати з документами та іншими видами інформації з будь-якого комп'ютера, що має доступ до мережі «Інтернет». Ці сервіси можна використовувати з різною метою. Наприклад:

– організації у «хмарі» централізованих тимчасових чи постійних архівів електронних оригінальних документів чи їх сканованих копій, це дозволить мати доступ до архіву всім користувачам, кому буде надано власником «хмари» відповідні права;

– організації у «хмарі» спільної роботи над документами декількох користувачів одночасно (територіальна віддаленість при цьому не має значення). Зміни, які вносяться у документ одним користувачем, видимі одночасно для всіх інших користувачів. У такому режимі роботи можливе додаткове «спілкування» між усіма онлайн-користувачами документа за допомогою групового чату;

– створення табличних документів для постійної оперативної консолідації певної облікової інформації. Кожен, хто має права доступу до документа, вносить свої числові значення у визначені поля, які автоматично відображаються також у відповідних підсумкових даних. З допомогою такої консолідації можна оперативно підбивати щоденні підсумки кількості розглянутих справ у мережі судової системи, робити аналітичну довідку про облік вхідної та вихідної документації з різних відділів підприємства.

Хмарний сервіс OneDrive – це файл-хостинг, створений у серпні 2007 року під управлінням компанії Microsoft. Сервіс OneDrive дозволяє безкоштовно зберігати до 15 ГБ інформації у впорядкованому за допомогою стандартних папок вигляді. Для зображень передбачено попередній перегляд у вигляді ескізів, а також можливість їх перегляду у вигляді слайдів. Для всіх папок та файлів можна визначати рівень доступу від виключно персонального до загального (публічного).

Для користування сервісом потрібно зареєструвати власний акаунт на сайті http://onedrive.live.com. Після реєстрації і входу буде отримано доступ до сервісу OneDrive (рис. 10.2), у корені якого є папки Documents, Pictures, Public. За допомогою кнопки Створити можна створити папку, документ, книгу, презентацію, блокнот, опитування, документ із звичайним текстом. За допомогою кнопки Передати можна перемістити файл (або папку) з твердого диска комп'ютера у відкриту папку на OneDrive.

Якщо виділити об'єкт (файл або папку) чи групу об'єктів, то у вікні браузера з'явиться панель команд (рис. 10.3).

За допомогою кнопки Спільний доступ можна надати спільний доступ до файлу або папки, вказавши електронну адресу іншого користувача.

У Facebook, Gmail, Twitter, LinkedIn та інших службах можна ділитися файлами з OneDrive з будь-ким і в будь-якій мережі, підключеній до вашого облікового запису OneDrive.

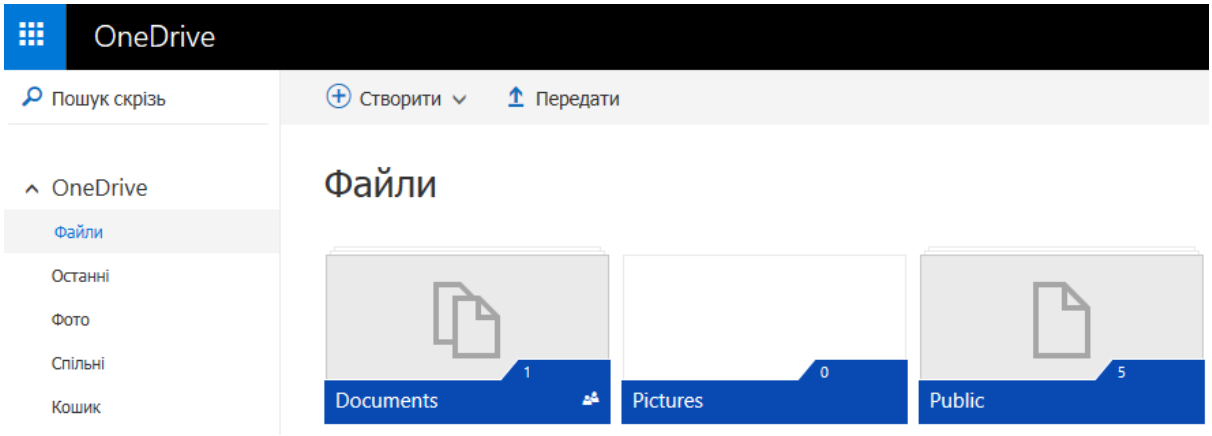


Рис. 10.2. Хмарний сервіс OneDrive

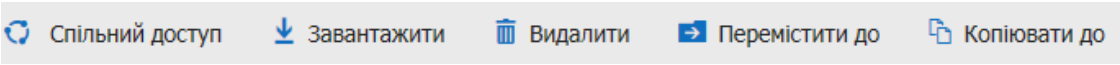


Рис. 10.3. Команди хмарного сервісу OneDrive

До сервісу OneDrive інтегровано деякі сервіси: *Hotmail* (включно з останньою бета-версією Outlook.com) – поштовий сервер; *Microsoft Office Online* – пакет офісних програм; *Bing* – пошукова система, що дозволяє користувачам зберігати історію пошуку у папці *OneDrive*. Перехід до цих сервісів (рис. 10.4) можна здійснити через кнопку .

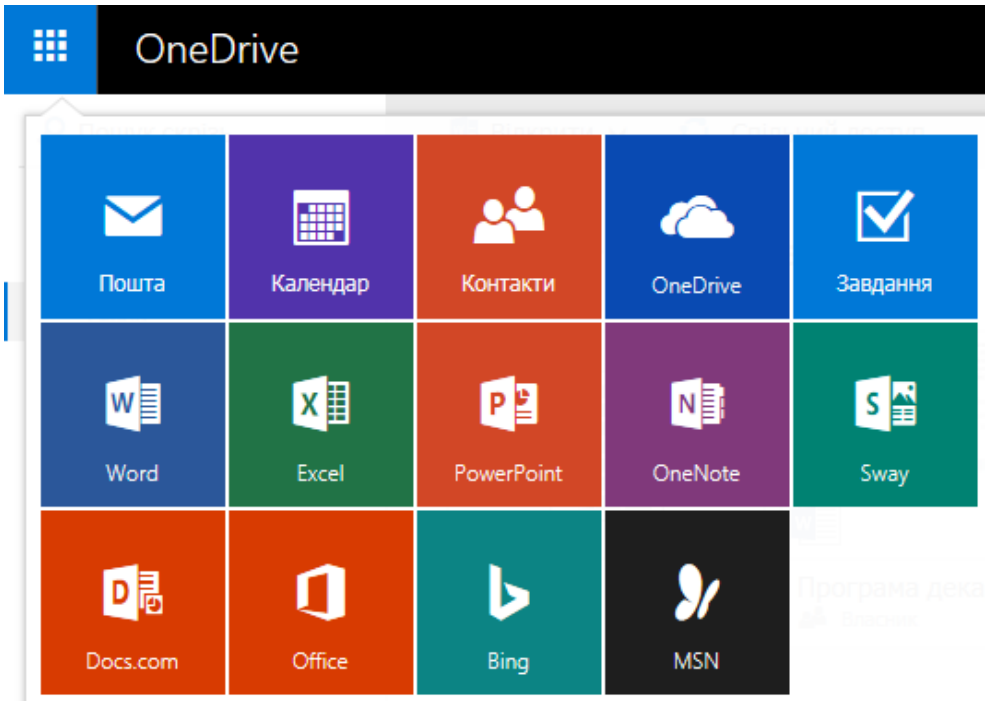


Рис. 10.4. Хмарні сервіси, інтегровані з OneDrive

Microsoft Office Online дозволяє користувачам завантажувати, створювати, редагувати та обмінюватися документами Microsoft Office безпосередньо у веббраузері. Користувачі можуть створювати, переглядати та редагувати документи Word, Excel, PowerPoint та OneNote онлайн. Під час редагування документа OneDrive зберігає всі зміни в одному онлайн-сеансі. Крім того, OneDrive автоматично зберігає до 25 попередніх версій усіх документів, тому вам не потрібно окремо зберігати їх копії. Якщо внесено непотрібні зміни, можна легко відновити документ або завантажити його попередню версію, обравши команду *Попередні версії*.

OneDrive підтримує перегляд формату PDF, а також стандарту ODF. Функція пошуку OneDrive не підтримує пошук документів у форматі PDF, але підтримує пошук за форматами з пакета Microsoft Office: DOC, DOCX, PPT, PPTX, XLS і XLSX.

*Хмарні сервіси компанії Google***.** Метою компанії Google визначено систематизацію всесвітньої інформації та перетворення її на загальнодоступну й корисну. Саме тому серед сервісів Google налічується більше 75 вебдодатків, які вимагають від користувача тільки наявності браузера й інтернет-підключення:

Google Drive – хмарне сховище даних, яке дозволяє користувачам безкоштовно зберігати свої дані обсягом до 15 ГБ і ділитися ними з іншими користувачами в інтернеті;

Google Docs – онлайн-додатки, у яких можна створювати файли, зберігати їх на Google Drive і спільно працювати над ними в режимі реального часу;

Google Sites – сервіс для створення сайтів;

Blogger – це сервіс для ведення блогів;

YouTube – відеохостинг, що надає послуги розміщення відеоматеріалів;

Google ArtProject – сервіс інтерактивно представлених музеїв світу;

Google AdSense – сервіс контекстної реклами, який дозволяє заробити власникам сторінок з великою статистикою відвідування;

та інші: Google Analytics, Google Статистика пошуку, Google AdWords, Google Alerts, Google Bookmarks, Google Calendar, Google Checkout, Google Cloud Print, Google Correlate, Google

Dictionary, Google Finance, Gmail, Google Groups, Google Health, Google Knol, Google Лабораторія, Google Maps, Google Maps API, Google Mars, Google Moon, Google Mobile, Google News, Google Ngram Viewer, Google Orkut, Google Picasa Web, Google Public DNS, Google Reader, Google Talk, Google Search History, Google Translate, Google Trends, Google Voice, Google Webmasters, Picnik, Google One Pass, Google +, Google Building Maker, Google Blog Search, Google Book Search, Google Code Search,Google Custom Search, Froogle, Hackser Style Google, Google Images, Google Government Search Google, Special Searches, Movie Showtimes, Google Patents Search, Google Scholar, Google SSL, Google Suggest, Google Video, Google Weather, Телефонна книга, Мовні інструменти, Калькулятор, Google Earth, про призначення яких можна прочитати на сайті google.com.

Для доступу до хмарних сервісів Google (рис. 10.5) потрібно відкрити сторінку за адресою http://www.google.com.ua та натиснути кнопку . Для реєстрації облікового запису (акаунту) на Google або входу в акаунт потрібно натиснути кнопку Увійти . Для входу у поштову скриньку потрібно натиснути кнопку Gmail .

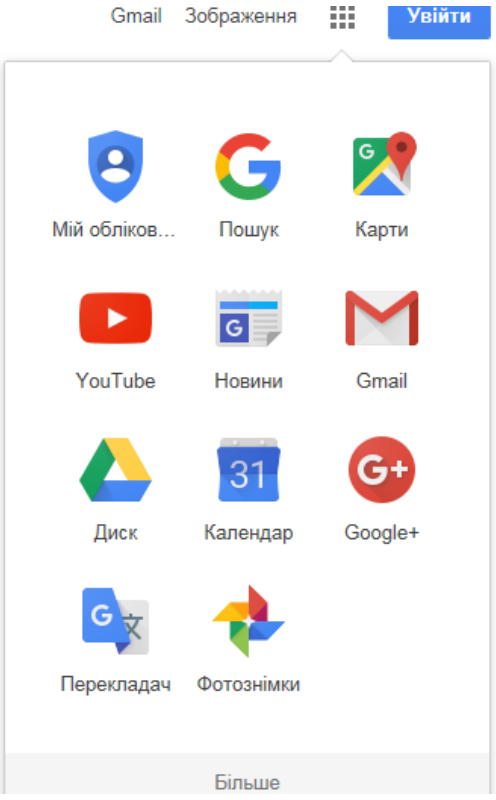


Рис. 10.5. Хмарні сервіси Google

Хмарна технологія Microsoft Azure http://azure.microsoft.com. Платформа *Windows Azure* – це інтернет-платформа служб хмарних обчислень, що розміщується у центрах обробки даних Microsoft. Вона включає в себе операційну систему для хмарних обчислень і набір служб для розробників. Windows Azure – це унікальний інструмент для побудови та розвитку бізнесу, який успішно використовують компанії у всьому світі.

До основних переваг хмарного сервісу Windows Azure належать: створення вебдодатків на основі звичних мов програмування, таких як NET, PHP, Java чи Ruby; забезпечення необхідного рівня захисту персональних даних чи фінансової інформації; автоматизоване управління службами та глобальної мережі центрів обробки даних для швидкого реагування на потреби клієнтів; бухгалтерський сервіс *iFin*; створення первинної документації, ведення бухгалтерського обліку, складської діяльності й автоматичного визначення термінів підготовки звітності та пересилка її до податкової інспекції, пенсійного фонду і статистичного управління тощо (рис. 10.6).

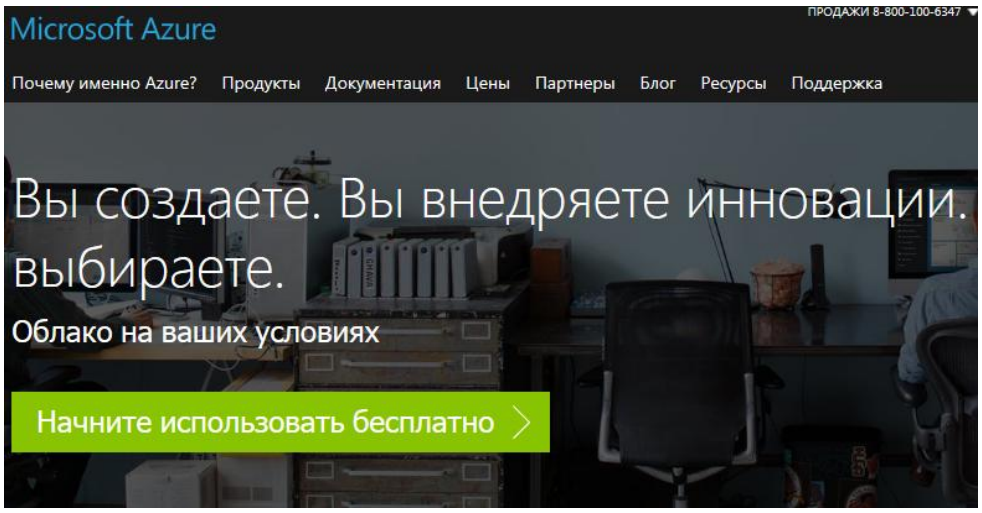


Рис. 10.6. Інтерфейс хмарного сервісу Microsoft Azure

Сьогодні хмарні технології мають широкий поштовх для розвитку і в більшості вони реалізовані у сфері бізнесу, для того щоб допомогти йому, по-перше, користувачеві не потрібно піклуватися про продуктивність свого ПК, переживати про вільне місце на дисковому просторі. З хмарними технологіями це питання автоматично знімається відразу ж на перших етапах; по-друге, користувачеві

не потрібно витрачатися повністю на весь потрібний йому продукт. Тим паче що в подальшому більша його частина може і не знадобитися взагалі. Він платить тільки за послугу, надані можливості і тільки за конкретні функції; по-третє, хмарні технології допомагають бізнесу в сезонному сенсі. Наприклад, якщо компанія продає подарунки до Міжнародного жіночого дня, який, як усі знають, тільки один раз на рік, решту часу (взимку, влітку, восени) її послуги нікому будуть не потрібні. І саме в цей час компанія не буде нести витрати на обслуговування того значного функціоналу, який потрібен їй тільки в піковий період.

Хмарні технології в бізнесі в наших умовах впроваджуються в основному тими компаніями, фінансове становище яких змушує звертати увагу на будь-який засіб, що дозволяє досягти економії.

10.4. Електронна комерція та біржові операції через інтернет

Електронна комерція охоплює такі напрями бізнес-діяльності в мережах інтернет, як: торгівля (роздрібна або опт) через віртуальні магазини, вебвітрини або торговельні центри; надання послуг; електронна реклама, маркетинг тощо.

Для ведення електронної комерції необхідно в інтернеті відкрити свій вебсайт, здійснити його «розкручування». Для автоматизації процесу торгівлі необхідно встановити на сайт спеціалізовану програму (скрипти), так званий *Віртуальний магазин*. Викласти на сайт відповідні товари і застосувати різні методи реклами і MLM-маркетингу з метою просування цих товарів.

Деякі компанії і приватні підприємці здійснюють прямий онлайновий продаж товарів (як матеріальних, так і інформаційних) з віртуальних складів або магазинів. Для просування своїх товарів вони використовують Direct marketing. Електронна комерція (e-commerce) й онлайновий продаж товарів – це майбутнє споживчого ринку.

До електронного бізнесу належить бізнес на валютному ринку, ринку коштовних паперів, товарних ринках і ринках дорогоцінних металів.

Робота трейдером (торгівля) на світовому валютному (фінансовому) ринку FOREX (Foreign Exchange Market). Міжнародний валютний ринок є системою регіональних валютних ринків, що взаємодіють між собою за допомогою інформаційних технологій. Робота на міжнародному валютному ринку є сукупністю різних операцій з купівлі-продажу іноземних валют. Секрет торговельних операцій полягає в тому, щоб купити дешевше, а продати дорожче.

OnLine аукціони (інтернет-аукціон). Міжнародний аукціон EBay – це місце для вигідного придбання необхідних товарів і для ведення власного бізнесу. Аукціон «Ау-ау» – це місце для продажу і покупки товарів.

Бізнес на інвестиціях. Інвестиції – це вклади під відсотки. В інтернеті існує безліч інвестиційних фондів, які під певний відсоток пропонують узяти у позику у вас гроші на певний проміжок часу.

Електронний бізнес включає: укладення договорів і контрактів, електронне управління закупівлями, виробництво ПО і цифрових товарів, формування й обробку замовлень, продажі, доставку товарів, маркетинг, фінансовий аналіз, платежі, пошук кадрів, підтримку клієнтів і підтримку партнерських стосунків.

Основні види (моделі) електронного або мережевого бізнесу викладені на сторінці http://lessons-tva.info/business/bus.html.

10.5. Віртуальна корпорація та віртуальний офіс

Нові форми обміну інформацією зумовили виникнення нових форм організації бізнесу з використанням сучасних засобів передачі даних.

Віртуальний офіс: група людей, що працюють разом для досягнення загальної мети, при цьому не зустрічаючись один з одним щодня.

Сучасний офіс – це набір мережевих служб, таких як електронна пошта, файловий сервер, сервер додатків (може бути), факс-сервер тощо, розташованих на сервері у вашій локальній мережі. Важко уявити такий офіс без персональних комп'ютерів,

копіювальних і факсимільних пристроїв, засобів мобільного зв'язку й інших пристроїв, що дозволяють ефективно вирішувати всі завдання.

Неодмінним атрибутом сучасних офісів є персональні комп'ютери, продуктивність яких за обсягом пам'яті, різноманітністю можливостей і швидкості перевершує продуктивність великих обчислювальних машин минулого.

Крім стаціонарних настільних персональних комп'ютерів, в офісній роботі набули широкого використання портативні комп'ютери, які, в свою чергу, розподіляються на два типи: ноутбуки (або лептопи) і кишенькові комп'ютери (або персональні електронні помічники).

Лептопи (дослівний переклад «наколінний комп'ютер») мають ті самі можливості, що й настільний комп'ютер, і відрізняються від звичайного персонального комп'ютера тільки розмірами (вага їх може не перевищувати 2 кг). З ними можна працювати, розмістивши на колінах під час зустрічі або перельоту.

Кишенькові комп'ютери (персональні електронні помічники (personal digital assistants, PDAs) – це мініатюрні комп'ютери, які важать менше 0,5 кг і легко містяться в кишеню піджака. Продуктивність і пам'ять такого комп'ютера обмежені. Кишенькові комп'ютери використовуються в основному для запису й зберігання телефонних номерів, адрес, записок і заміток, невеликих документів для передачі електронною поштою або факсом, а також для ведення щоденника. З появою стільникового зв'язку PDAs використовували пейджери або для прийому й передачі факсів та електронної пошти.

Крім того, в роботі сучасного офісу застосовуються й інші електронні апаратні засоби: модем, мережева карта, різні копіювальні пристрої, сканери, факси тощо. Коротко розглянемо основні з них.

Мережева карта дозволяє поєднувати комп'ютери в мережу. Комп'ютерна мережа – це об'єднання двох або більше комп'ютерів та інших пристроїв, що уможливлює обмін інформацією між цими пристроями.

Копіювальні апарати призначені для копіювання текстів, креслень, малюнків. Сучасні моделі копіювальних апаратів мають багато додаткових можливостей: двостороннє копіювання, звірення копій, автоматичне скріплення скобами, повнокольорове копіювання тощо.

Факсимільний апарат або факс використовується для електронного пересилання документів з одного місця в інше. Факсом можна передати будь-яку надруковану інформацію або зображення: картинку, графік, рукописні замітки, діловий лист, підписаний контракт тощо.

Сканери призначені для одержання зображення (текстів, таблиць, креслень, схем, малюнків, фотографій) з оригіналів і зберігання отриманого електронного зображення в комп'ютері для наступної роботи з ними.

Багатофункціональні пристрої – пристрої, які можуть виконувати функції копіювального апарата, факсу й сканера.

Сучасні засоби дозволяють керувати досить великою кількістю працівників, які географічно перебувають у різних місцях, і за необхідності консультувати їх.

На Заході послугами віртуальних офісів (колцентрами) користується більшість великих і невеликих компаній. Тільки в США існує більше 70 тисяч колцентрів.

Перший класичний віртуальний офіс в Україні організувала відома пейджингова компанія Beeper. Найбільш «продвинуті» компанії вже встигли скористатися послугами першого українського колцентру і гідно оцінити переваги такої плідної співпраці.

Послугами віртуального офісу користуються не тільки підприємці, що тільки розпочинають свій бізнес і не мають власного офісу. Як це не дивно, але на сьогодні в Україні найбільш активними користувачами й постійними клієнтами колцентрів є великі компанії, що працюють на ринку електроніки й побутової техніки, банки, страхові компанії, фірми, що займаються прямими продажами, тощо.

До послуг віртуального офісу звертаються також фірми, що відкривають філії та представництва в іншому місті. При цьому витрати на телефонні дзвінки зменшуються. За свідченнями

експертів, переваги віртуального офісу полягають також у тому, що власні працівники не відволікаються на консультування із приводу характеристик продукції, умов акції й подібних стандартних питань. А клієнт може одержати консультацію у віртуальному офісі в будь-який час доби.

Особливо цінні послуги віртуального офісу для маркетингової стратегії і тактики будь-якої фірми незалежно від розмірів і спеціалізації.

У разі розширення функцій організації розширюються і можливості віртуального офісу.

У зв'язку з цим є сенс розглянути такі форми організації підприємств, як мережеві і віртуальні корпорації. Мережевий характер підприємств виявляється у тому, що глобальна мережа використовується для обміну необхідною інформацією між географічно видаленими підрозділами компанії.

Нині фірмами широко використовується часткова або повна передача виконання окремих бізнес-функцій і навіть частин бізнес-процесу стороннім особам і організаціям.

Це явище одержало назву аутсорсинг (аутсорсинг – outsourcing (англ.) буквально – процес отримання будь-чого із зовнішніх джерел).

Широкий розвиток на Заході аутсорсинг одержав через ряд причин.

По-перше, це зростання інтенсивності конкурентної боротьби у всіх секторах ринку і пов'язана з нею необхідність досягнення найвищої ефективності всіх операцій компанії, яка прагне до завоювання стабільної і довготривалої переваги над конкурентами. Досягти самостійно максимального підвищення результативності всіх операцій компанії практично неможливо, а іноді і недоцільно. Можна довести до досконалості виконання ключових функцій, а решту роботи довірити тим, у кого вона виходить краще за інших. Отже, для багатьох компаній залучення сторонніх організацій для виконання окремих робіт стало несподіваним і ефективним виходом. Важко не погодитися з тим, що завжди існують компанії, здатні виконувати відносно незалежні функції бізнесу з найбільшою, практично недосяжною ефективністю.

По-друге, це прагнення компаній бути «глобальними», тобто щоб їх продукція і послуги були представлені у всьому світі. Для цього насамперед необхідна відсутність жорсткої «прив'язки» до певної території. Наприклад, власні продуктивні потужності, служба доставки або мережа магазинів є не серйозною перешкодою, а швидше зайвою розкішшю для фірми, що переходить від ринку однієї країни до іншої, принаймні на початковому етапі.

Третій момент перш за все пов'язаний з роллю малих підприємств, що збільшуються, в світовому бізнесі. Аутсорсинг дає можливість глобальної присутності будь-якої компанії на ринках багатьох країн без необхідності практично пропорційного зростання персоналу для обслуговування нових ринків збуту або виробничих потужностей. Тобто відносно невелика компанія може із залученням малих підприємств працювати по всьому світу з центрального офісу, або «штабу», зберігаючи контроль над виконанням поставлених завдань у межах обраної стратегії.

Зв'язок центрального офісу з підрозділами здійснюється, як правило, з використанням електронних засобів і глобальної мережі передачі даних. Революційність такого підходу до створення бізнес-організації полягає хоча б у тому, що, покладаючись на звичні визначення і поняття, важко уявити, що це за організація і де вона розташована. Як приклад, можемо навести фірму, що займається розробкою програмного продукту (ядро, оболонка), зазвичай відбувається із залученням команд з усього світу. Великі промислові фірми, які спеціалізуються на випуску електронної апаратури та комп'ютерів, таких як Compaq та IBM, які раніше повністю самостійно виконували весь цикл розробки і виробництва техніки, тепер замовляють за своїми кресленнями і технологічними картами виробництво апаратури азіатським компаніям. Отже, їм вдається знизити собівартість своєї продукції і легше витримати конкуренцію з виробниками більш дешевих аналогів.

Для зв'язку з партнерами і підрозділами широко використовуються можливості глобальної мережі, такі як електронна пошта і відеоконференції.

Незважаючи на те, що у сфері застосування мережевих технологій комп'ютерщикам «усі карти в руки», найпершою компанією, яка під час розширення і глобалізації бізнесу застосувала

модульну структуру і домоглася приголомшливого успіху, була Nike – лідер американського ринку з виробництва і продажу спортивного одягу та інвентарю.

Мережева або модульна структура дає безліч переваг. У першу чергу, це можливість сконцентрувати зусилля персоналу на вирішенні кількох основних завдань, замовляючи виконання інших функцій, таких як доставка, бухгалтерський облік, а також виробництво фахівцями поза компанією. Модульна корпорація є стрижнем, оточеним гнучкою мережею найкращих постачальників необхідних послуг, які, як модулі в конструкторі, можуть бути задіяні або виключені у зв'язку з необхідністю.

Найважливішою перевагою є присутність організації в багатьох країнах, а також можливість завойовувати ринкові позиції скрізь, де є така можливість. Мережева організація консолідує ресурси по всьому світу з метою домогтися найкращої якості продукції за максимально низької вартості, що є одним з вирішальних чинників для досягнення стійкої переваги над конкурентами. Модульна організація може постійно змінювати свою структуру для того, щоб виходити на ринок з новою продукцією. Інший, не менш важливий аспект – це вища продуктивність і задоволеність виконуваною роботою тими, хто працює в головному офісі, завдяки суттєво гнучкій структурі підприємства. Причому структура мережевої (віртуальної) корпорації – найбільш гнучка з усіх можливих форм існування ділових організацій.

Зрештою, для подолання деяких аспектів недостатнього контролю і підвищення залученості співробітників у загальну справу необхідно використовувати засоби сприяння груповій роботі.

Особливість глобальної мережевої корпорації полягає також у тому, що виникає необхідність у керівниках, які підготовлені до роботи в різних країнах. Незважаючи на те, що основні функції управління: планування, організація, лідерство, контроль, – не змінюються від того, чи проходять операції компанії в одній країні або декількох одночасно, існують чинники відмінності в навколишньому середовищі, а саме:

- соціально-культурні відмінності між країнами;
- відмінності в економічному розвитку;
- відмінності в законодавстві.

Отже, це підвищує частку ризику для бізнесу. Внутрішня структура міжнародної компанії повинна відповідати декільком наборам параметрів зовнішнього середовища, що формує різні культури, усередині яких перебувають клієнти компанії, а також постачальники продукції і різного сервісу, що найбільш важливо для аутсорсинга.

Ідеологія офісу постійно змінюється й розвивається. У значно більшому ступені буде використана технологія віртуальної реальності, тобто створення штучного навколишнього середовища за допомогою комп'ютерних технологій. Широко розповсюдженими стануть такі поняття, як «віртуальний офіс», «віртуальні архіви й бази даних», «віртуальні тури», «віртуальні тренажери», «віртуальні конференції» тощо.

Офіс майбутнього – це не тільки передові інформаційні технології, але й нові підходи до планування будівель і організації внутрішнього простору. Можливо, в майбутньому офісом називатимуться ошатні будинки в оточенні саду або ландшафтного парку, з комфортними меблями й теплим інтер'єром, і тим самим подібні швидше на приватну оселю, а функції бек-офісу передаватимуться спеціалізованим фірмам.

10.6. Експертні системи, їх будова та застосування в економіці

Визначальною рисою комп'ютерних програм, що називаються експертними, є їх здатність накопичувати знання і досвід найбільш кваліфікованих спеціалістів (експертів) у певній предметній галузі. Потім за допомогою цих знань користувачі ЕС, що мають звичайну кваліфікацію, можуть розв'язувати свої задачі так само успішно, як це зробили б самі експерти.

Експертні системи – це програми для комп'ютерів, що акумулюють знання фахівців-експертів у конкретних предметних галузях, які призначені для одержання прийнятних рішень у процесі обробки інформації. Експертні системи трансформують досвід експертів у будь-якій конкретній галузі знань у форму евристичних правил і призначені для консультацій менш кваліфікованих фахівців.

Відомо, що знання існують у двох видах: колективний досвід, особистий досвід. Якщо предметна галузь представлена колективним досвідом (наприклад, вища математика), то вона не потребує експертних систем. Якщо в предметній галузі значна частина знань є особистим досвідом фахівців високого рівня і ці знання є слабоструктурованими, то така галузь потребує експертних систем. Сучасні експертні системи знайшли широке застосування у всіх сферах економіки.

Функціонально експертну систему можна визначити як обчислювальну систему, яка здатна використовувати знання про деяку предметну галузь і приймати рішення в рамках цієї галузі знань на рівні експерта-професіонала.

Експертна система складається з:

– бази знань (складається з робочої пам'яті і бази правил), призначеної для зберігання вихідних і проміжних фактів у робочій пам'яті (її ще називають базою даних), і зберігання моделей та правил маніпулювання моделями в базі правил;

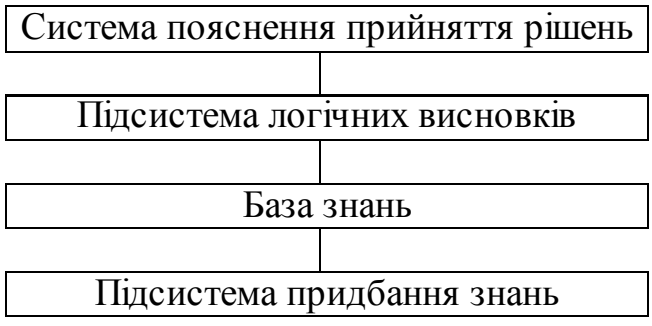
– вирішувача завдань (інтерпретатора), який забезпечує реалізацію послідовності правил для вирішення конкретного завдання на основі фактів і правил, що зберігаються в базах даних і базах знань;

– підсистеми пояснення, яка дозволяє користувачеві отримати відповіді на запитання: «Чому система прийняла таке рішення?»;

– підсистеми набуття знань, призначеної як для додавання в базу знань нових правил, так і модифікації наявних правил;

– інтерфейсу користувача, комплексу програм, що реалізовують діалог користувача із системою на стадії введення інформації й одержання результатів.

Структура експертної системи



Підсистема набуття знань призначена для додавання в базу знань нових правил і модифікації наявних. У її завдання входить приведення правила до виду, що дозволяє підсистемі виводу застосовувати це правило в процесі роботи. У більш складних системах передбачені ще й засоби для перевірки даних, що вводять або модифікують, на несуперечність з наявними правилами.

База знань – найбільш важливий компонент експертної системи, на якому базуються її «інтелектуальні здібності».

База знань містить факти і правила, що оцінюють результативність кожного варіанта з точки зору найшвидшого досягнення мети. Базу знань можна замінити, і це не вплине на процес міркування системи.

База знань є ядром експертної системи. Перехід від даних до знань є наслідком розвитку інформаційних систем. Для зберігання даних застосовуються бази даних, а для зберігання знань – бази знань. У базі даних, як правило, зберігаються значні масиви даних з відносно невеликою вартістю, а в базах знань зберігаються невеликі за об'ємом, але дорогі інформаційні масиви.

База знань – це сукупність знань, описаних з використанням обраної форми їх подання. Наповнення бази знань є одним з найскладніших завдань, яке пов'язане з вибором знань, їх формалізацією та інтерпретацією.

Підсистема логічних висновків – програмний компонент експертних систем, що реалізує процес її міркувань на основі бази знань і робочої множини. Вона виконує дві функції: по-перше, перегляд існуючих фактів з робочої множини й правил з бази знань і додавання (за можливості) у робочу множину нових фактів й, по-друге, визначення порядку перегляду й застосування правил. Ця підсистема управляє процесом консультації, зберігає для користувача інформацію про отримані висновки і запитує в нього інформацію, коли для спрацьовування чергового правила в робочій множині виявляється недостатньо даних.

Ціль ЕС – вивести деякий заданий факт, що називається цільовим твердженням (тобто в результаті застосування правил домогтися того, щоб цей факт був включений у робочу множину), або спростувати цей факт (тобто переконатися, що його вивести

неможливо, отже, за цим рівнем знань системи він є помилковим). Цільове твердження може бути або «закладене» заздалегідь у базу знань системи, або витягається системою з діалогу з користувачем.

Робота системи являє собою послідовність кроків, на кожному з яких з бази обирається деяке правило, що застосовується до поточного вмісту робочої множини. Цикл закінчується, коли виведене або спростоване цільове твердження. Цикл роботи експертної системи інакше називається логічним виводом. Логічний вивід може відбуватися багатьма способами, з яких найпоширеніші – прямий порядок виводу й зворотний порядок виводу.

Прямий порядок виводу – від фактів, які перебувають у робочій множині, до висновку. Якщо такий висновок вдається знайти, то воно заноситься в робочу множину. Прямий вивід часто називають виводом, керованим даними.

Зворотний порядок виводу: висновки переглядаються доти, доки не будуть виявлені в робочій пам'яті або отримані від користувача факти, що підтверджують одне з них. У системах зі зворотним висновком спочатку висувається деяка гіпотеза, а потім механізм виводу в процесі роботи, ніби повертається назад, переходячи від її до фактів, і намагається знайти серед них ті, які підтверджують цю гіпотезу. Якщо вона виявилася правильною, то обирається наступна гіпотеза, що деталізує першу, яка є щодо неї підціллю. Далі відшукуються факти, що підтверджують істинність підлеглої гіпотези. Висновок такого типу називається керованим цілями. Зворотний пошук застосовується в тих випадках, коли цілі відомі і їх порівняно небагато.

Система пояснення прийняття рішень показує, за допомогою яких міркувань системою був отриманий результат у зрозумілому людині вигляді.

Оскільки системи, засновані на знаннях, реалізуються на комп'ютерах, то й вхідна інформація сприймається у вигляді, зрозумілому комп'ютеру, тобто в бітах і байтах. Однак для того, щоб міг взаємодіяти непідготовлений користувач, у неї потрібно включити засоби спілкування природною мовою. Переважна більшість систем, заснованих на знаннях, має досить примітивний інтерфейс природною мовою – припустимі вхідні повідомлення

користувача, обмежені набором понять, що втримуються в базі знань. Через це були ще включені блоки для пояснення кожного кроку міркувань, для користувача-непрограміста такий спосіб пояснення може здатися зайво деталізованим і занадто формальним, а потрібно було б пояснення на «більш високому рівні» з погляду здорового глузду й без поглиблення в деталі. Подібні моделі пояснення тільки починають з'являтися в ЕС і вимагають більш складної організації знань.

До характеристик, притаманних ЕС як системам штучного інтелекту, можна віднести:

- компетентність, тобто здатність приймати рішення, адекватні рішенням професіонала-експерта високого рівня;
- здатність будувати «міркування» на основі символьних перетворень;
- здатність використовувати як загальні, так і окремі схеми породження рішення;
- здатність розв'язувати завдання в реальних предметних галузях;
- здатність до інтерпретації формулювання запитів і завдань;
- здатність до аналізу своєї роботи.

Експертні системи відрізняються від традиційних систем обробки даних тим, що в них, як правило, використовується символьний спосіб подання, символьний вивід і евристичний пошук рішень. Для вирішення завдань, що слабо формалізуються або не формалізуються, більш перспективними є нейронні мережі або нейрокомп'ютери.

Основу нейрокомп'ютерів становлять нейронні мережі – ієрархічні організовані паралельні з'єднання адаптивних елементів – нейронів, які забезпечують взаємодію з об'єктами реального світу так само, як і біологічна нервова система.

Великі успіхи використання нейромереж досягнуті під час створення самонавчальних експертних систем. Мережу набудовують, тобто навчають, пропускаючи через неї всі відомі рішення і досягаючи одержання необхідних відповідей на виході. Налаштування полягає в підборі параметрів нейронів. Часто використовують спеціалізовану програму навчання, яка займається навчанням мережі. Після навчання система готова до роботи.

Якщо в експертну систему її творці заздалегідь закладають знання в певній формі, то в нейронних мережах невідомо навіть розробникам, як формуються знання в її структурі в процесі навчання і самонавчання, тобто мережа є «чорним ящиком».

Нейрокомп'ютери як системи штучного інтелекту є перспективними і можуть нескінченно удосконалюватися в своєму розвитку. Нині системи штучного інтелекту у формі експертних систем і нейронних мереж знаходять широке застосування під час вирішення фінансово-економічних проблем.

Запитання для самоконтролю

1. Яку структуру має ЕС?
2. Які переваги та недоліки штучної експертизи?
3. Як можна представити правила економічного аналізу в експертній системі?
4. Які напрями застосування ЕС в управлінській діяльності?
5. Які особливості технології побудови ЕС у середовищі електронних таблиць?
6. Які сучасні тенденції подальшого розвитку економічної інформатики?
7. У чому сутність нового наукового напряму, що отримав назву концептуальної інформатики?
8. У чому сутність нового наукового напряму нейроінформатики?
9. У чому сутність інформаційних технологій електронного навчання?
10. У чому сутність наукового напряму, що отримав назву квантова інформатика?
11. Які перспективи розвитку інформаційних технологій?
12. У чому сутність управлінських інформаційних систем?
13. Які етапи створення управлінських інформаційних систем?
14. У чому сутність діаграмного підходу до опису управлінських процесів, відомих під абревіатурою IDEF?

ТЕСТОВІ ЗАВДАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Яку клавішу треба натиснути, щоб перемістити текстовий курсор на кінець рядка:

1) Alt;
2) End;
3) Esc;
4) Home;
5) Num Lock.

2. Яку клавішу треба натиснути, щоб стерти символ, який міститься праворуч від курсора:

1) Ctrl;
2) BS (Backspace);
3) Scroll Lock;
4) Tab;
5) Del(ete).

3. Яку клавішу треба натиснути, щоб стерти символ, який міститься ліворуч від курсора:

1) Ctrl;
2) BS (Backspace);
3) Scroll Lock;
4) Tab;
5) Del(ete).

4. З чого складається повне ім'я файлу:

1) шлях до файлу, ім'я файлу;
2) адреса, ім'я файлу;
3) папка, ім'я файлу;
4) шлях до файлу, ім'я файлу, тип файлу;
5) шлях до файлу, розширення.

5. Яке розширення мають документи, створені у середовищі графічного редактора Paint:

1) bat;
2) doc;
3) jpg;
4) bmp;
5) rtf.

6. Яке розширення за замовчуванням мають документи, створені у середовищі редактора WordPad:

1) bat;
2) doc;
3) jpg;
4) bmp;
5) rtf.

7. Яке розширення має файл з умовами пошуку:

1) txt;
2) fnd;
3) tip;
4) rtf;
5) eml.

8. Яке розширення має файл архіву:

1) txt;
2) fnd;
3) zip;
4) rar;
5) eml.

9. Які автофігури можна створити у тексті:

1) Лінії, Основні фігури, Фігурні стрілки, Блок-схема, Зірки і стрічки;
2) Фігурні стрілки, Блок-схема, Нові фігури;
3) Основні таблиці, Фігурні стрілки, Блок-схема;
4) Лінії, Основні фігури, Фігурні формули, Блок-схема;
5) Виноски.

10. Які є способи збереження документів MS Word 2013:

1) Вставка – Зберегти;

2) CTRL+S;

3) Вставка – Зберегти як;

4) Файл – Зберегти;

5) Файл – Зберегти як.

11. Режими перегляду документа у редакторі MS Word 2013:

1) Розмітка сторінки, Режим читання;

2) Вебдизайн, Структура, Чистовик;

3) Вебдокумент, Структура, Черновик;

4) Лінійка, Одна сторінка;

5) Сфера навігації.

12. Як перетворити набраний текст у таблицю в програмі MS Word 2013:

1) Файл – Таблиця – Перетворити в таблицю;

2) Вставка – Таблиця – Перетворити в таблицю;

3) Таблиця – Перетворити в таблицю;

4) Формат – Таблиця – Перетворити в таблицю;

5) Головна – Таблиця – Перетворити в таблицю.

13. Включення і відключення зображення лінійок на екрані програми MS Word 2013:

1) Файл – Лінійка;

2) Вставка – Лінійка;

3) Вид – Лінійка;

4) Формат – Лінійка;

5) Меню – Лінійка.

14. Якими способами можна змінити масштаб перегляду документа в програмі MS Word 2013:

1) Файл – Масштаб;

2) Вид – Масштаб;

3) Формат – Масштаб;

4) перетягування повзунка масштабування у відповідному місці рядка стану;

5) перетягування повзунка масштабування у відповідному місці рядка заголовка.

15. Які стандартні розширення файлів, збережених у програмі PowerPoint 2013:

1) doc;

2) ppsx;

3) xls;

4) pptx;

5) txt.

16. Які найбільш поширені операції для форматування комірок можна виконати:

1) за допомогою пункту Формат листа;

2) за допомогою пункту Формат ячеек;

3) за допомогою пункту Формат області листа;

4) за допомогою пункту Формат тексту;

5) за допомогою пункту Формат книги.

17. Яке повідомлення виводиться, коли число не поміщається в комірці:

1) #ДЕЛ/0!

2) ####

3) #ИМЯ?

4) #ЗНАЧ!

5) #ПУСТО!

18. Яке повідомлення виводиться у разі помилки у назві функції:

1) #ДЕЛ/0!
2) #Н/Д
3) #ИМЯ?
4) ####
5) #ЗНАЧ!

19. Яке повідомлення виводиться у разі спроби поділити на нуль:

1) #ДЕЛ/0!
2) #Н/Д
3) #ИМЯ?
4) #ПУСТО
5) #ЗНАЧ!

20. Який формат має логічна команда ЕСЛИ:

1) =ЕСЛИ(логічна умова; Дія 1; Дія 2; Дія 3);
2) =ЕСЛИ(логічна умова; Дія 1; Дія 2);
3) =ЕСЛИ(логічна умова; Дія 1+ Дія 2);
4) =ЕСЛИ(Дія 1; Дія 2; логічна умова);
5) =ЕСЛИ(Дія 1; логічна умова; Дія 2).

21. Який загальний вигляд має формула електронної таблиці:

1) =ФУНКЦІЯ(АРГУМЕНТ);
2) =ФУНКЦІЯ/АРГУМЕНТ;
3) =ФУНКЦІЯ*АРГУМЕНТ;
4) =ФУНКЦІЯ-АРГУМЕНТ;
5) ФУНКЦІЯ(АРГУМЕНТ).

22. Який формат числа встановлюється для загального способу виведення чисел з обраною кількістю десяткових знаків:

1) Загальний;
2) Фінансовий;
3) Відсотковий;
4) Текстовий;
5) Числовий.

23. Вкажіть правильні записи формули:

1) =(R18+L21)/(G8-87,16E12);

2) =(R18+L21)(G8-87,16*E12);

3) =(R18+L21)/(G8-87,16*E12);

4) =(R18+L-21)/(G8-87,16*E12);

5) =(R18+L21)/G8-87,16*E12.

24. Конструктор – це:

1) програмний модуль для виведення даних;

2) програмний модуль для виконання допустимих операцій над об'єктами;

3) режим роботи БД, у якому здійснюється побудова таблиць або форм;

4) режим роботи БД, у якому здійснюється виведення таблиці або форми на друк.

25. Сортування даних в електронних таблицях – це:

1) впорядкування записів за значеннями одного з полів;

2) впорядкування назв полів за наборами значень властивостей.

26. Під час сортування за зростанням числа впорядковуються у такому порядку:

1) від найменшого від'ємного до найбільшого додатного;

2) у алфавітному порядку: числа, знаки, латинський алфавіт, український алфавіт;

3) у хронологічному порядку.

27. Під час сортування за зростанням дані текстового типу впорядковуються у такому порядку:

1) від найменшого від'ємного до найбільшого додатного;

2) в алфавітному порядку: числа, знаки, латинський алфавіт, український алфавіт;

3) у хронологічному порядку.

28. Під час сортування за зростанням дані типу «дата та час» впорядковуються у такому порядку:

1) від найменшого від'ємного до найбільшого додатного;

2) в алфавітному порядку: числа, знаки, латинський алфавіт, український алфавіт;

3) у хронологічному порядку.

29. У стовпчику А здійснено впорядкування:

	A	B	C	D
1	-10	бит	суббота, Январь 01, 2000	20:30
2	-5	bit	понедельник, Март 03, 2003	16:30
3	0	$	понедельник, Январь 03, 2004	12:30
4	1	5	среда, Март 03, 2004	8:30
5	5	1	среда, Январь 12, 2005	4:30

1) чисел за зростанням;

2) тексту за спаданням;

3) дат за зростанням;

4) часу за спаданням.

30. У стовпчику В здійснено впорядкування:

	A	B	C	D
1	-10	бит	суббота, Январь 01, 2000	20:30
2	-5	bit	понедельник, Март 03, 2003	16:30
3	0	$	понедельник, Январь 03, 2004	12:30
4	1	5	среда, Март 03, 2004	8:30
5	5	1	среда, Январь 12, 2005	4:30

1) чисел за зростанням;

2) тексту за спаданням;

3) дат за зростанням;

4) часу за спаданням.

31. Як зробити посилання на поштову скриньку most@ukr.net:

1) Пошта

2) Пошта

3) Пошта

4) Пошта

32. Що означає параметр bgcolor:

1) колір фону сторінки;

2) зображення-обої;

3) колір тексту.

33. Який параметр визначає колір переглянутого посилання:

1) vlink;

2) alink;

3) bulink;

4) link.

34. Який параметр тега body дозволяє вставити зображення як фон документа:

1) bgcolor;

2) background;

3) bgground;

4) backimage.

35. Що містить URL-адреса:

1) правильне кодування вебсторінки;

2) назву протокола, адресу вебсервера, шлях до документа на цьому сервері;

3) доменне ім’я вебсервера, на якому розміщено сайт;

4) ім’я вхідної сторінки сайту у Всесвітній павутині.

36. Що таке протокол:

1) правила об’єднання тексту, графіки, звуки, відео- та гіперпосилань в єдиний гіпертекстовий документ;

2) програма для доступу до інформаційних ресурсів інтернету;

3) пристрій для узгодження роботи комп’ютера та телефонної лінії зв’язку у разі виходу в інтернет;

4) набір правил, що визначають порядок подання і пересилання інформації в інтернеті, реалізований у програмах або комунікаційних пристроях.

37. Яке призначення пошукових систем:

1) пошукова система накопичує інформацію про документи, що зберігаються в інтернеті, і за запитом користувачів підбирає адреси документів, в яких користувач може знайти потрібну інформацію;

2) пошукова система дає відповіді на запитання користувачів;

3) пошукова система постійно шукає нову інформацію в інтернеті.

38. Для чого призначений протокол ftp:

1) для передачі гіпертексту у Всесвітній павутині;

2) для роботи з електронною поштою;

3) для узгодження роботи комп'ютера і телефонної лінії зв'язку під час виходу в інтернет;

4) для передачі файлів у глобальній мережі інтернет.

39. Що таке HTML (HyperText Markup Language):

1) універсальна програма для доступу до інформаційних ресурсів інтернету, а саме для перегляду гіпертекстових документів Всесвітньої павутини;

2) мова каскадних стилів, призначена для визначення способу відображення елементів гіпертекстового документа;

3) мова розмітки гіпертексту, якою описують логічну структуру гіипертекстового документа;

4) протокол передачі гіпертексту, основа Всесвітньої павутини.

40. Що таке візуальний редактор вебсторінок:

1) програма для доступу до інформаційних ресурсів інтернету;

2) пристрій для узгодження роботи комп'ютера і телефонної лінії зв'язку під час виходу в інтернет;

3) спеціалізована програма, призначена для автоматичної генерації HTML-коду документа за його візуальним образом.

41. Які блоки обов'язково входять у структуру HTML-документа:

1) мета-теги з описом документа і ключовими словами;

2) адреса документа в інтернеті;

3) заголовок і тіло документа.

42. Як називається фрагмент HTML-коду, розміщений у кутових дужках:

1) елемент;
2) контейнер;
3) атрибут;
4) тег;
5) параметр.

43. Що таке параметр циклу:

1) величина, що обчислюється у циклі;
2) величина, що управляє роботою циклу;
3) лічильник кількості повторень;
4) умова виконання циклу;
5) немає правильної відповіді.

44. В якому типі циклу відома кількість його повторень:

1) Do While... Loop;
2) For … Next;
3) Do Until... Loop;
4) Do... Loop Until;
5) Do... Loop While.

45. Яке значення матиме параметр циклу після виходу з такого циклу:

```
Dim i As Integer
For i=3 To 23 Step 1
a = a + 1
Next i
```

Виберіть один із 5 варіантів відповіді:

1) 10;
2) 3;
3) 24;
4) 23;
5) 13.

46. Масив – це:

1) послідовність випадкових чисел;

2) скінченний набір елементів одного типу, які зберігаються в послідовних комірках оперативної пам’яті і мають спільну назву;

3) послідовність символьних величин, які зберігаються в послідовно розташованих комірках оперативної пам’яті.

47. Масив, елементами якого є числа, називається:

1) числовим;

2) літерним;

3) символьним.

48. Оголошення одновимірного масиву M дійсних чисел:

1) Dim M(50) As Integer;

2) Dim M(10,20) As Integer;

3) Dim M (10,20) As Single;

4) Dim M(50) As Single.

49. Оголошення двовимірного масиву цілих чисел з іменем M:

1) Dim M(50) As Integer;

2) Dim M(10,20) As Integer;

3) Dim M () As Single;

4) Dim M(50) As Single.

50. Вкажіть кількість елементів масиву a(m TO n):

1) n-m;

2) m-n+1;

3) n-m+1;

4) m-n.

ВІДПОВІДІ ДО ТЕСТОВИХ ЗАВДАНЬ

1) правильні відповіді: 2;
2) правильні відповіді: 5;
3) правильні відповіді: 2;
4) правильні відповіді: 4;
5) правильні відповіді: 3; 4;
6) правильні відповіді: 5;
7) правильні відповіді: 2;
8) правильні відповіді: 3; 4;
9) правильні відповіді: 1; 5;
10) правильні відповіді: 2;4;5;
11) правильні відповіді: 1; 3;
12) правильні відповіді: 2;
13) правильні відповіді: 3;
14) правильні відповіді: 2; 4;
15) правильні відповіді: 2; 4;
16) правильні відповіді: 1;
17) правильні відповіді: 2; 5;
18) правильні відповіді: 5;
19) правильні відповіді: 2;
20) правильні відповіді: 2;
21) правильні відповіді: 3;
22) правильні відповіді: 1;
23) правильні відповіді: 2;
24) правильні відповіді: 1;
25) правильні відповіді: 5;
26) правильні відповіді: 3; 5;
27) правильні відповіді: 3;
28) правильні відповіді: 1;
29) правильні відповіді: 1;
30) правильні відповіді: 2;
31) правильні відповіді: 3;
32) правильні відповіді: 1;
33) правильні відповіді: 2;
34) правильні відповіді: 4;
35) правильні відповіді: 2;
36) правильні відповіді: 1;
37) правильні відповіді: 2;
38) правильні відповіді: 2;
39) правильні відповіді: 4;
40) правильні відповіді: 1;
41) правильні відповіді: 4;
42) правильні відповіді: 3;
43) правильні відповіді: 3;
44) правильні відповіді: 3;
45) правильні відповіді: 2; 4;
46) правильні відповіді: 2; 3;
47) правильні відповіді: 2;
48) правильні відповіді: 3;
49) правильні відповіді: 2;
50) правильні відповіді: 1.

ГЛОСАРІЙ ОСНОВНИХ ТЕРМІНІВ

А

Абсолютна адреса (absolute address) – посилання у формулі на адресу клітини електронної таблиці, яка не змінюється під час копіювання або переміщення формули в інше місце робочого аркуша або на інший робочий аркуш. Посилання на абсолютну адресу клітини включає букву(и) імені стовпця і номер рядка, перед якими стоїть знак долара ($). Наприклад, $F$8. Абсолютні адреси застосовуються до імен комірок, що містять ключові значення, такі як курс долара, значення рівня прибуткового податку тощо.

Автоматизація (automation) – розробка та використання методів і засобів (зокрема, ІТ), що дозволяють здійснювати роботи й технологічні процеси з мінімальною участю людини (оператора). Наприклад, автоматизація виробничих процесів, використання систем автоматизації проєктування (САПР), автоматизація програмування та ін.

Автоматизована інформаційна система [АІС] (automatic information system) – організаційно-технічна система, що використовує автоматизовані інформаційні технології в цілях навчання, інформаційно-аналітичного забезпечення науково-інженерних робіт і процесів керування. Відповідно до цього визначення ГІС потрапляє в клас автоматизованих інформаційних систем.

Автоматизована обробка даних (computer-aided data processing) – обробка даних, що виконується автоматичними засобами за можливою участю людини.

Адаптер локальної мережі (local area network adapter, LAN adapter) – адаптер, призначений для підключення комп'ютера до локальної мережі комп'ютерів. Наприклад, для підключення ПК до мережі Ethernet використовується адаптер NE-2000.

Адреса (address) – число, код або ідентифікатор, що специфікують регістр, елемент пам'яті, сферу запам'ятовуючого пристрою, зовнішній пристрій або вузол мережі. Унікальний ідентифікатор або номер, що привласнюється пристрою або об'єкту для

операцій з ним. Одне з основоположних понять обчислювальної техніки. Адреси бувають фізичними, логічними й мережними. Наприклад, фізична адреса може являти собою:

а) унікальний номер ділянки (напівпровідникової) пам'яті, який слугує для її ідентифікації під час операцій читання/запису;

б) номер регістра або порту вводу-виводу, закодований в команді; в) номер доріжки і номер сектора для визначення адреси для дискових операцій тощо. Прикладом логічної адреси є номер кластера на диску або адреса клітини електронної таблиці. Мережні адреси, такі як адреси електронної пошти, вебсерверів у різних мережах, влаштовані по-різному. Зазвичай це одна або декілька груп символів, що містяться в тексті повідомлення, за якими визначається одержувач або відправник даних, що пересилаються. Стандарти IEEE 802.3 і 802.5 рекомендують наявність унікальної адреси в світі для кожного пристрою.

Адреса електронної пошти (e-mail address) (*IP-address*) – закодоване позначення пункту відправлення або призначення даних у мережі «Інтернет»; ідентифікація об'єкта (наприклад, об'єкта мережі). Будується за так званою доменною системою адресації. Це означає, що адреса користувача мережі складається з двох частин: ідентифікатора користувача і назви домена з роздільовим символом @ (офіційна назва символу – «комерційне at (ет)», жарг. – «собака»): (user)@(domain). Як назва домена, так і ідентифікатор користувача можуть ділитися на сегменти, що розділяються крапкою. В адресі допустимі латинські букви, цифри та деякі інші символи.

Аналоговий сигнал (analog signal) – сигнал, сфера визначення якого є безперервним простором, тобто простором, що не є дискретним. Іншими словами, безперервний сигнал, наприклад звук людської мови, що приймає безліч значень з деякого діапазону величин. Аналогові сигнали описуються безперервними функціями часу, тому аналоговий сигнал іноді називають безперервним сигналом. Аналоговим сигналам протиставляються дискретні (квантовані, цифрові).

Анімація (animation) (*графіка анімаційна*) – створення рухомих зображень на екрані дисплея шляхом відображення декількох нерухомих картинок для створення ефекту руху. Тобто вивід на екран послідовності зображень, що злегка розрізняються, для створення ілюзії руху. Для цього моделюється розвиток у часі певних процесів, який відображається у вигляді послідовності кадрів. Анімація широко використовується перш за все в комп'ютерних іграх. Є також штучним представленням руху в кіно і на телебаченні шляхом послідовної зміни мальованих кадрів з певною швидкістю. Варто розуміти, що якщо під час зйомки кінофільму або відеофільму безперервний рух розбивається на окремі кадри, то анімація, навпаки, із спеціально підготовлених окремих кадрів створює ефект безперервного руху.

Архітектура комп'ютера (computer architecture) (архітектура) – організаційна структура комп'ютера (обчислювальної системи), що включає потоки та представлення даних, інтерфейси, набір команд, способи адресації, регістри, апаратне та програмне забезпечення. Термін введений корпорацією IBM під час створення сімейства сумісних ЕОМ System/360.

Аутентифікація (authentication) (авторизація, ідентифікація) – перевірка (підтвердження) достовірності. Процес ідентифікації користувача, перевірка його посвідчення особи в певній системі авторизації. Засіб захисту, що визначає достовірність даних користувача і законність його роботи. Процес ідентифікації користувача або комп'ютера, що намагаються отримати (інтерактивний) доступ до певної категорії інформації, комп'ютерної системи, обчислювальної мережі або електронної пошти. Виконується в цілях забезпечення безпеки, ґрунтується на використанні додаткових ідентифікаторів: біометричної інформації, паролів, спеціальних карток, цифрового підпису та ін. У комп'ютерній системі користувачеві зазвичай привласнюється символічне ім'я або ідентифікаційний код (SID), які використовуються надалі процедурами керування доступом. Аутентифікація не відповідає на запитання.

Б

Багатозадачність [багатозадачна робота, багатозадачний режим, мультизадачність] (multitasking) – здатність ОС підтримувати одночасне виконання двох або більше завдань (сегментів програми). Для користувача це виглядає як паралельне (одночасне) виконання декількох застосувань. Отже, він може редагувати документ за допомогою MS Word і паралельно слухати музику у застосуванні WinAmp. Такий режим роботи виконується шляхом почергового виділення коротких квантів часу процесам кожного із застосувань процесором ПК. Відповідно до цього ОС поділяють на однозадачні (наприклад, MS DOS) і багатозадачні – MS Windows і Linux.

База даних (database, DB) (*ММDB, бази даних створення, СКБД*) – набір таблиць, що є організованим набором записів і пов'язаних з ними допоміжних файлів. Сукупність великих структурованих наборів постійних даних, що зазвичай асоціюються з ПЗ, призначеним для виконання оновлення і вибірки необхідних елементів сукупності. Проста БД може бути одиночним файлом, що містить багато записів, які мають однакову структуру полів, кожне з яких має фіксовану ширину. Є сукупністю даних, істотною для деякої діяльності у відповідній предметній галузі.

База знань (knowledge database, knowledge base, KB) – семантична модель, що описує предметну галузь і дозволяє відповідати на такі запитання з цієї предметної галузі, відповіді на які не містяться в цій базі. База знань є основним компонентом інтелектуальних і експертних систем. Зазвичай знання, що стосуються певної предметної галузі, записуються на будь-якою мовою представлення знань.

Буфер обміну [проф. кліпбоард] (clipboard) (*ОС Windows*) – спеціальна сфера ОЗП, що обслуговується ОС. Вона легкодоступна всім застосуванням і використовується для передачі даних між ними. Для забезпечення сумісності між застосуваннями буфер обміну зберігає дані, які передаються, одночасно в декількох різних форматах. Отже, під час роботи з Windows можна легко переміщати фрагменти тексту, графічні зображення й таблиці між такими застосуваннями, як Word, Excel, Access, Powerpoint тощо.

В

Вебзастосування (web application) – набір клієнтських і серверних програм, які працюють разом, щоб забезпечити вирішення проблеми обробки контента, що розташовується на вебсерверах. ПЗ, що міститься на вебсервері, забезпечує логіку роботи сайту, генерацію сторінок HTML і обробку запитів користувачів. Працюючи з вебзастосуванням, користувач має значні можливості з обробки інформації, що зберігається на вебсервері. Зокрема, йому стають доступні такі операції, як додавання, видалення, редагування, переупорядковування, пошук та інші дії з різнорідними даними, організованими різними способами і, можливо, будь-якими користувачами вебзастосування в різних БД (наприклад, MySQL, PostgreSQL та ін.).

Вебінтерфейс (web interface) – сукупність засобів, за допомогою яких користувач взаємодіє з вебсайтом через вебзастосування. Інтерфейс заснований на використанні браузера для відображення інформації, елементів керування й мінімальної логіки на клієнтській стороні під час розміщення основних функціональних елементів на вебсервері.

Вебсайт (web site) (*вебвузол*, *сайт*) – являє собою простір, що адресується загальним доменним ім'ям (IP-address), яке містить різні види контента (з'єднані вебсторінки, графічні, відео- і бінарні файли та ін.), розташовуваного на вебсерверах. Існують різні типи вебсайтів: а) домашні сторінки користувачів (не треба плутати з домашньою сторінкою вебсайту); б) інформаційні (контент) сайти й портали; в) сайти електронної комерції; г) представницькі сайти веброзробників та ін. Деякі фахівці вважають, що терміни «вебсайт» і «вебвузол» є синонімами. Крім вебсайтів, у мережі «Інтернет» так само доступні WAP-сайти для мобільних телефонів.

Вебтехнології (web technologies) – засоби розробки, створення, розміщення й пересилання інформації в мережі в різних форматах. Припускають використання скриптових мов програмування й технологій роботи на стороні клієнта та стороні сервера. Вебтехнології: HTML, HTTP, WAIS (Wide Area Information Server) та ін.

Відеоінформація (video data) (*формат відеофайлів*) – інформація, що подається у формі відеокліпів (відеороликів), тобто наборів послідовно виведених один за одним взаємозалежних зображень-кадрів (відеокадрів). Якщо швидкість появи відеокадрів перевищує частоту злиття мелькань (порядку 25 кадрів), то в користувача створюється враження безперервного руху об'єктів (full-motion video – повнокадрове відео). Цей принцип був реалізований у кіно й у цей час залишається основним у процесі оцифровування відеозображень. Відеозображення можуть відтворюватися як в окремому вікні програми перегляду на частині екрана, так і в розмірах усього екрана (full-screen video – повноекранне відео). Обсяг однієї секунди відеоролика з частотою 30 кадр/с під час розрізнення 640 × 480 пікселів, представлених 8-розрядним кодом (256 кольорів), становить 9 Мбайт. У разі використання 24-розрядної колірної палітри (16 млн кольорів) і розрізнення 1 280 × 1 024 ця цифра збільшиться до 114 Мбайт, у зв'язку з чим особливу актуальність здобувають питання зберігання та передачі відеоінформації.

Відеоконференція (video conferencing) – методологія проведення нарад і дискусій між групами віддалених користувачів з використанням трансляції зображення та звуку в мережі «Інтернет».

Відеопам'ять [буфер відеоінформації] (video buffer, video memory) (*відеокарта, графічний процесор, відеоконтролер, цифроаналоговий перетворювач, відео-ПЗП, система охолодження*) – *п*ристрій, що виконує роль кадрового буфера, у якому зберігається зображення, постійно змінюване графічним процесором і виведене на екран монітора (або декількох моніторів). У відеопам'яті зберігаються також проміжні невидимі на екрані елементи зображення та інших даних. Відеопам'ять буває декількох типів, що розрізняються за швидкістю доступу й робочої частоти. Сучасні відеокарти комплектуються пам'яттю типу DDR, DDR2, GDDR3, GDDR4 і GDDR5. Варто також мати на увазі, що, крім відеопам'яті, що перебуває на відеокарті, сучасні графічні процесори зазвичай використовують у своїй роботі частину загальної системної пам'яті комп'ютера, прямий доступ до якої організується драйвером відеоадаптера через шину AGP або PCIE. У випадку використання архітектури UMA, як відеопам'ять, використовується частина системної пам'яті комп'ютера.

Віртуальний офіс (virtual office) (*віртуальне підприємство*) – інтернет-ресурс або його частина, що дозволяють географічно роз'єднаним співробітникам компанії взаємодіяти за допомогою єдиної системи комунікації для обміну, зберігання, обробки й передачі інформації та керуючих дій. Є способом електронної (мережної) організації робочого процесу, який не потребує оренди чи побудови відповідних приміщень для виконавців. У такому офісі можуть працювати співробітники трьох типів: надомні працівники (telecommuter), працівники віддалених офісів (teleworker) і мобільні працівники (mobile worker).

Вірус (virus) – тип програм, що мають здатність прихованого від користувача саморозмноження для поразки інших програм, комп'ютерів або мереж. Програма здатна підключатися до інших програм (тобто заражати їх). Звичайна дія вірусу призводить до небажаних і руйнівних наслідків для програм і даних. Існує безліч видів таких програм. Термін запропонований Фредом Коеном (Кохеном) 1983 р., коли він був ще студентом університету шт. Південна Каліфорнія.

Г

Геолокація (geolocation) (*геотаргетинг*) – виявлення реального географічного положення (країна, місто) комп'ютера, з'єднаного з мережею «Інтернет» або мобільного пристрою, одержуваного на основі IP-адреси, даних, вбудованих у програмне або апаратне забезпечення, або іншої інформації. Метою геолокації є збір статистичних даних про відвідувачів сайтів, а також зміна змісту сайту чи блокування доступу до сайту в цілому або до деякої інформації залежно від його географічного положення.

Гіпертекст (hypertext) – текст, відображений у вікні браузера, що містить гіперпосилання. Крім тексту, гіпертекст може містити таблиці, зображення та інші засоби презентації. Часто під гіпертекстовими сторінками мають на увазі HTML-документи. Розширеним прикладом гіпертексту сьогодні є весь WWW.

Д

Двійковий код (binary code) – спосіб представлення будь-яких даних (чисел, слів та ін.) у вигляді комбінації двох знаків, які можна позначити як 0 і 1. Знаки або одиниці двійкового коду називають бітами. Одним з обґрунтувань застосування двійкового коду є простота і надійність накопичення інформації в будь-якому носієві у вигляді комбінації всього двох його фізичних станів, наприклад у вигляді зміни або постійності магнітного потоку в деякій ділянці носія магнітного запису. Найбільше число, яке може бути виражене двійковим кодом, залежить від кількості використовуваних розрядів, тобто від кількості бітів у комбінації, що виражає число. Наприклад, для вираження числових значень від 0 до 7 достатньо мати 3-розрядний або 3-бітовий код, оскільки 710 = 1 112.

Дефрагментація [файлу] (defragmentation) (*фрагментація файлу*) – усунення фрагментації файлів або, наприклад, дискової пам'яті. Операція, що припускає перепис розміщених на диску файлів наново так, щоб вони займали безперервну ділянку секторів диска.

Динамічна вебсторінка (dynamic web page) – вебсторінка, що може виглядати по-різному, залежно від деяких змінюваних ззовні параметрів (конкретного відвідувача, часу відвідування сайту та ін.). Існують два способи динамічного формування інформаційного наповнення вебсторінки (текст, зображення, поля форми та ін.): а) використовуючи створення сценарію на стороні клієнта, щоб змінити поводження інтерфейсу в межах певної вебсторінки, у відповідь на дії миші, клавіатури або за зазначеними подіями синхронізації; б) використовуючи створення сценарію на стороні сервера, щоб забезпечити послідовність завантаження або перезавантаження вебсторінок, причому відповіді сервера можуть бути визначені такими умовами, як дані у відправленій HTML-формі, параметр URL, тип використовуваного браузера, стан сервера або БД. Однак обидва способи можуть використовуватися й одночасно.

Дистанційне навчання (distance education, [teaching, training]) – спосіб реалізації процесу навчання, побудований на використанні сучасних інформаційних і телекомунікаційних технологій, що дозволяють здійснювати навчання на відстані безпосереднього (особистого, face-to-face) контакту між викладачем і учнем.

Діалогове меню (pop-up menu, pull down menu) (*меню, контекстне меню*) – елемент графічного інтерфейсу користувача. Зазвичай є спливаючим меню. Невидиме на екрані меню, яке з'являється тільки після того, як користувач виконає дію, що змушує програму або ОС вивести її на екран. Один з видів діалогового меню – контекстне меню, яке викликається клацанням правої кнопки миші в будь-якому місці екрану.

Документ текстовий (text document) – документ, виконаний у вигляді твердої копії, тобто на паперовому носії. Текстове представлення цифрового документа на екрані комп'ютера застосуванням, в якому він розроблений. Оскільки концепцією корпорації Microsoft, закладеній в офісний пакет Microsoft Office, є представлення будь-яких даних у вигляді документа або складеного документа, то будь-який файл, відкритий створеним ним застосуванням (Word, Excel, Power Point та ін.), зображується на екрані у вигляді текстового документа. Вивід на принтер такого документа, як правило, дозволяє отримати його тверду копію.

Домен (domain) – сфера в інтернеті (пойменована група вузлів мережі). Визначається частиною адреси, що йде за символом @. Увесь простір імен в інтернеті поділено на домени, організовані за ієрархічним принципом (менші домени входять до складу більших). Домени визначаються IP-адресами. Вважається, що всі пристрої, «що ділять між собою» загальну частину IP-адреси, містяться в одному й тому ж домені (*DNS, IP-address).*

Драйвер (driver) – системна програма, що реалізовує на фізичному рівні всі процеси вводу-виводу для роботи з конкретним пристроєм (класом пристроїв) або будь-яким ресурсом. Драйвери основних пристроїв зазвичай включаються розробниками до складу ОС. Крім того, створюється спеціальний механізм для включення в систему будь-яких додаткових драйверів, що поставляються

виробниками додаткового й нестандартного устаткування. У цілому драйвер є програмою, що керує роботою зовнішнього пристрою (миші, клавіатури, принтера, флешпам'яті тощо), а також елементом програмного інтерфейсу між програмами вводу-виводу ОС і конкретним пристроєм (принтером, дисководом, дисплеєм та ін.). Кожен зовнішній пристрій характеризується своїм унікальним інтерфейсом, узгодження якого з ОС здійснює відповідний цьому пристрою драйвер. У цілому драйвер приймає на вхід стандартні команди ОС ПК і цифрові дані, які він перетворює на специфічні команди та електромеханічні процеси й дії відповідного фізичного пристрою.

Е

Експертна система (expert system) (*штучний інтелект*) (*ШІ*). Комп'ютерна програма, що акумулює знання експертів і фундаментальні знання певній наочній галузі, має здібність до логічних виводів і використовується як електронний консультант для осіб, що приймають рішення, а також слугує для забезпечення високоефективного вирішення завдань у деякій вузькій наочній галузі. Такі програми, як правило, представляють знання символічно, досліджують і пояснюють свої процеси міркування і призначені для тих прикладних галузей, в яких людям для досягнення майстерності необхідні роки спеціального навчання та практики.

Електронна комерція [еКомерція] (*eCommerce*), **електронна обчислювальна машина [ЕОМ]** (machine, computer) (*комп'ютер*) – старе найменування комп'ютерів у ті часи, коли вони вироблялися на електронних лампах, а потім на діодтранзисторній елементній базі. Однією з найбільш поширених ЕОМ в Україні періоду 70–80-х років XX ст. була БЭСМ-6 із швидкодією 1 млн оп./с. У тих галузях, де ЕОМ використовуються до нині, вони називаються успадкованими системами або мейнфреймами, на відміну від сучасних суперкомп'ютерів і кластерів.

Електронна торгівля (eTrade) – торгівля, що здійснюється за допомогою електронного документообігу в інтернеті.

Елементарні функції (elementary function) – клас функцій, що складається з багаточленів, показових функцій, логарифмічних функцій, тригонометричних функцій і зворотних тригонометричних функцій, а також функцій, що виходять з перерахованих вище за допомогою чотирьох арифметичних дій і суперпозиції (утворення складної функції), застосованих кінцеве число разів. Клас елементарних функцій найбільш вивчений і найчастіше зустрічається в застосуваннях математики, а також у мовах програмування при математичних обчисленнях.

Ж

Жорсткий диск (hard disc, hard disc drive) – магнітний диск, вінчестер, накопичувач на жорстких магнітних дисках.

З

З'єднання [скріплення (імені і сутності)] (binding) (*ім'я, bind*) – відображення імені на відповідну сутність. У рамках поточного контексту ім'я може мати не більш ніж одне з'єднання. Допускається посилання за допомогою іншого імені на контекст. Структуроване або складене ім'я формується у вигляді комбінації імені контексту та імені сутності в цьому контексті, яка розділена крапкою. Наприклад, у мові *Visual Basic for Application* з'єднання імені об'єкта (*Object*) з його властивостями (*Property*) проводиться через розділення крапкою так: *Object.Property*. Тобто звернення до властивості *StatusBar* об'єкта Application запишеться у вигляді: *Application.StatusBar*.

Змінна (variable, argument) (*ідентифікатор*) – програмний об'єкт, що має ім'я та значення, яке може бути отримане і змінене програмою. Тобто іменоване місце в пам'яті, що може містити елемент даних, який, у свою чергу, може модифікуватися (змінюватися) під час виконання програми. Кожна змінна має унікальне ім'я (ідентифікатор), що ідентифікує її в загальній безлічі змінних. У мові VBA тип даних може специфікуватися або ні. Ім'я змінної

повинне починатися з буквеного символу, бути унікальним (єдиним) у використовуваному наборі, не перевищувати довжину в 255 символів і не може містити заборонені символи. У мовах з чіткою типізацією (strong typing, наприклад, Turbo Pascal) визначення типу змінної є обов'язковим і у цьому випадку ім'я такої змінної визначає: а) спосіб представлення у ній значення (текстове, цифрове, логічне); б) довжину ділянки пам'яті у байтах, що виділяється для її значення; в) множину її допустимих значень; г) множину операцій, що є допустимими для її обробки.

І

Ідентифікатор (identifier) (*змінна*) – символьне ім'я клітини або сфери пам'яті. Імена, що привласнюються змінним, константам, структурам даних, класам, процедурам, функціям, методам та іншим програмним об'єктам. Деякі мови програмування вимагають оголошення ідентифікаторів до їх використання в програмі (*explicit declaration*).

Ідентифікація користувача (user identification) – упізнання користувачів (за іменем і паролем) для визначення його повноважень, тобто права на доступ до даних і ресурсів, а також вибору режиму їх використання.

ІКТ [інформаційно-комп'ютерні технології] (information-computing technologies) – термін, що визначає комплекс взаємозалежних і взаємовпливових програмних і апаратних компонентів, що перебувають у постійному розвитку.

Ім'я файлу (file name) – спеціальний набір символів, який використовується, щоб однозначно визначити файл, що зберігається у файловій системі комп'ютера. Ряд ОС ідентифікує каталоги (директорії, папки) у такий самий спосіб. Різні ОС вводять будь-які обмеження для довжини й припустимого набору символів в іменах файлів. З урахуванням можливих засобів доступу ім'я файлу може включати ряд компонентів: а) протокол (або схема) – метод доступу (наприклад, *http*, *ftp* та ін.); б) хост – ім'я хост-комп'ютера, IP-адреса, доменне ім'я або ім'я в локальній мережі (наприклад,

www.cnn.com, 207.142.131.206, \\MYCOMPUTER, SYS: та ін.); в) пристрій (або вузол) – порт, роз'єм, диск або том (наприклад, *C:, /, SYSLI та ін.*); г) каталог (або шлях) – дерево каталогів (наприклад, */usr/bin, \TEMP, [USR.LI.SRC]* та ін.); д) файл – ім'я файлу; е) тип файлу (формат або розширення імені) – вказує тип інформаційного наповнення файлу (наприклад, розширення *.txt,.exe,.COM* та ін.).

Індекс (index) (див. *база даних, масив*) (*БД*) – структура даних, що забезпечує доступ до даних у БД за ключем. Компактний файл у СКБД, що зберігає список ключів, кожен з яких визначає унікальний запис у БД і містить інформацію про її фізичне розташування. Слугує для прискорення пошуку і сортування даних. Отже, програма для доступу до даних використовує цей індекс, а не простежує всю БД від початку до кінця.

Інтеграція (integration) – відновлення та/або підвищення якісного рівня взаємозв'язків між елементами системи, а також процес створення з декількох різнорідних систем єдиної системи з метою виключення (до технічно необхідного мінімуму) функціональної та структурної надмірності й підвищення загальної ефективності функціонування.

Інтернет [мережа] (Internet) – всесвітня інформаційна система загального користування, яка логічно пов'язана з глобальним адресним простором та базується на інтернет-протоколі, визначеному міжнародними стандартами. Високошвидкісна оптоволоконна мережа, що об'єднує решту всіх мереж нижчих рівнів (національні, регіональні, *WAN, LAN, WLAN, WMAN, Mobile WMAN, Wi- Fi* та ін.) по всьому світу і використовує для передачі даних транспортний протокол *TCP/IP*. Слугує засобом комунікації користувачів за допомогою використання *e-mail*, засобів передачі даних і файлів програм із застосуванням протоколів *HTTP* та *FTP*, а також пошуку інформації в *World Wide Web*. Крім того, інтернет забезпечує віддалений доступ до комп'ютерних систем з метою використання програмних компонентів у розподілених обчисленнях, роботи з онлайновими електронними каталогами і БД-засобами технології комутації пакетів (*packet switching*) та

вирішення інформаційно-пошукових завдань. Інтернет був заснований 1969 р. під егідою проєкта міністерства оборони США *Arpanet* і являє собою (на відміну від *World Wide Web*) тільки засоби комунікації, тобто лінії зв'язку й супутні їм апаратні засоби: маршрутизатори, хаби, перемикачі та ін. Для цієї платформи створено і створюється безліч архітектур, моделей обчислень і застосувань (наприклад, *Java*,.*NET*, вебсервіси тощо), що дозволяють ефективно використовувати наявні можливості глобальної комп'ютерної мережі – колосальні обсяги інформаційних даних за різними галузями знань і надшвидкісні лінії їх передачі.

Інтерфейс (interface) – точка з'єднання двох елементів або загальна межа між двома взаємопов'язаними системами (чи рівнями однієї системи), яка забезпечує сумісність пристроїв незалежно від їх типу та виробника. Засіб сполучення двох або більше систем (сутностей). Інтерфейс між комп'ютером і людиною називається користувацьким інтерфейсом (*User interface, UI*). Він визначає спосіб взаємодії людини з комп'ютером і включає набір команд і правил їх використання. Також представляє служби для організації повнофункціональної роботи із застосуванням. Це означає, що він надає засоби для виконання таких операцій: а) переміщення компонентів; б) введення даних; в) представлення даних; г) виведення даних; д) маніпулювання з даними; е) допомога користувачу (*help*), і разом з тим він є зовнішньою візуальною частиною програми, безпосередньо зверненою до користувача. Залежно від конкретної реалізації інтерфейс користувача може бути частиною застосування або ОС, а може бути окремим клієнтським застосуванням. Зазвичай складається з набору кнопок, списків команд (меню), команд ОС, форматів графічних дисплеїв та інших пристроїв, що надаються комп'ютером або програмою, які забезпечують користувачу можливість «спілкуватися» з комп'ютером або програмою. Графічний інтерфейс користувача (*GUI*) забезпечує особам, що використовують його, «образно орієнтований» (*picture-oriented*) спосіб взаємодіяти з новими технічними засобами й технологіями. *GUI* більш зручний під час роботи з новими комп'ютерними і цифровими системами.

Інтерфейс командного рядка (command-line interface, CLI) – засіб взаємодії користувача з ОС або ПЗ шляхом набору на клавіатурі текстових команд для виконання певних дій, наприклад в ОС *MS DOS* і консольних застосуваннях *MS Windows*.

Інформаційна система (information system, IS) – термін, що взагалі використовується для посилань на системи, які забезпечують збирання, збереження й доступ користувачів до накопичених даних. Такі системи зазвичай включають до свого складу апаратні й технічні засоби підтримки вищевказаних процесів. Завжди вирішують завдання пошуку або логічної обробки інформації. Найбільш загальним прикладом інформаційної системи є банківська система (наприклад, Національного банку України), що включає програмні та апаратні засоби обробки запитів клієнтів і проведення трансакцій, розподілені БД державного масштабу, банкомати, відповідні мережні технології (*LAN* або *WAN*) та ін. У широкому значенні це – організаційно впорядкована сукупність документів (масивів документів) та інформаційно-комп'ютерних технологій, зокрема з використанням засобів комп'ютерної техніки й зв'язку, що реалізовують інформаційні процеси. За приналежністю до конкретної предметної галузі можна розділити інформаційні системи на три класи: технічні, економічні й інформаційно-аналітичні. До технічних відносять автоматизовані системи наукових досліджень, системи автоматизованого проєктування, гнучкі виробничі системи, робототехнічні комплекси та ін. Прикладом економічних систем можуть служити автоматизовані системи керування, бухгалтерські інформаційні системи, банківські інформаційні системи, біржові інформаційні системи, маркетингові інформаційні системи та ін. Інформаційно-аналітичні автоматизовані системи включають: автоматизовані довідкові інформаційні системи, БД, експертні системи, статистичні інформаційні системи та ін.

Інформаційний ресурс (information resource) – дані в будь-якому вигляді, які можна багаторазово використовувати для вирішення проблем користувачів. Наприклад, це може бути файл, документ, вебсайт, портал, фотографія, відеофрагмент.

Інформаційні технології [комп'ютинг, обробка даних] (computing) – галузь науки, техніки й виробництва, що охоплює дослідження теоретичних і методичних основ, розробку й створення технологій інформаційної індустрії, пов'язаних із збором, виробництвом, обробкою, передачею, розповсюдженням, зберіганням, експлуатацією, представленням, використанням і захистом різних видів інформації. Нині під інформаційними технологіями найчастіше розуміють комп'ютерні технології. Зокрема, інформаційні технології мають справу з використанням комп'ютерів і ПЗ для зберігання, перетворення, захисту, обробки, передачі й отримання інформації. Фахівців з комп'ютерної техніки й програмування часто називають IT-спеціалістами.

Інформаційно-пошукова мова (information retrieval language, IRL) – формалізована штучна мова, призначена для індексування документів, формування інформаційних запитів і опису фактів з метою наступного зберігання й пошуку. Як правило, такі засоби є внутрішніми мовами розвинених інформаційно-пошукових систем, таких як *Lycos, AltaVista, Yahoo, OpenText, Infoseek* і *WAIS*.

К

Кеш-пам'ять 1-го рівня [первинна кеш-пам'ять] (L1 cache, primary cache) – пам'ять, вбудована в ядро процесора *Intel 486* і вище. Кеш-пам'ять, що розміщується на кристалі процесора, на відміну від вторинної кеш-пам'яті, яка зазвичай розміщується в процесорному модулі або на системній платі.

Ключ бази даних (data base key) – елемент даних, значення якого використовується для пошуку окремих сукупностей даних у БД, найчастіше записів або сегментів.

Кодування (coding) – трансформація призначеного для передачі значення в послання або сигнал, який може бути переданий. Кодування в комунікаційному процесі починається з вибору системи кодових знаків – носіїв інформації, які потім передаються за допомогою каналів. Носіями (каналами передавання) можуть бути звук, світло, радіохвилі, температура, запах, смак, атмосферний тиск, фізичні тіла та ін.

Комунікація (communication) – у перекладі українською мовою це слово може означати: зв'язок, повідомлення, засіб зв'язку, інформацію, засіб інформації, а також контакт, спілкування або з'єднання.

Конфігурування (configuration) – процес адаптації (настроювання) апаратних засобів або програми для правильної роботи конкретного комп'ютера. Включає проведення робіт з ОС комп'ютера, пов'язаною з підключенням до неї драйверів, які використовуються в цій системі зовнішніх пристроїв (блоків пам'яті, принтерів, моніторів та ін.), також можливо виконання необхідних установок у *BIOS*.

Користувач (user) – фізична або юридична особа, що застосовує обчислювальну систему чи програмний засіб (застосування).

Л

Лінія зв'язку (communication line) – сукупність технічних пристроїв і фізичного середовища, що забезпечує розповсюдження сигналів від передавача до приймача. Лінія може бути провідна, повітряна, кабельна, радіо, радіорелейна та ін.

Логічна структура (logical structure) – сукупність визначень, яка встановлює порядок і принципи взаємодії окремих частин системи.

Локальна обчислювальна мережа (local area network, LAN) – україномовний синонім локальних (*LAN*) і регіональних (*WAN*) обчислювальних мереж. Група станцій користувачів, кожна з яких може здійснювати зв'язок принаймні з однією з них, використовуючи спільне середовище передавання та спільне керування. Зазвичай ЛОМ є мережею передачі даних, що охоплює невелику територію (наприклад, підприємство, будівля, поверх будівлі). Ця мережа сполучає певну кількість комп'ютерів та інших пристроїв поміж собою або ж до них підключається центральний сервер, який і забезпечує між ними високошвидкісну передачу даних.

М

Маршрутизатор [*жарг.* роутер] (router) – вузол мережі, що спрямовує пакети на основі спеціального критерію й забезпечує оптимальне використання мереж між вихідним пунктом та пунктом призначення пакетів. Є або апаратним пристроєм, або програмним забезпеченням. Зазвичай спеціалізований комп'ютер або пакет програмного забезпечення відповідає за з'єднання й підтримку комунікаційних процесів між двома або декількома мережами. Маршрутизатори відшукують у пакетах дані, що передаються через них, адреси одержувачів і визначають, за яким маршрутом потрібно передавати ці пакети.

Меню (menu) – елемент інтерфейсу користувача, що дозволяє вибрати одну (у найпростішому випадку) з декількох перерахованих опцій. У сучасних ОС меню є основним елементом графічного інтерфейсу користувача, на відміну від інтерфейсу командного рядка (*command, command mode*), де команди комп'ютера вводяться у формі текстових команд. Вибір даних з меню може бути здійснений різними методами (інтерфейсами): а) натисканням однієї або більше клавіш на клавіатурі або миші; б) установленням курсору або смуги негативного зображення, використовуючи клавіатуру або мишу; в) використовуючи електромеханічний покажчик (світлове перо); г) торканням екрана дисплея пальцем; д) розмовляючи із системою розпізнавання мови.

Мікропроцесор (microprocessor, MP, MPU) – центральний процесор, пристрій або функціональна частина цифрової обчислювальної системи, виготовлена в монолітному кристалі і призначена для обробки даних та інтерпретації програм. Велика інтегральна схема (ВІС) або мікросхема, здатна виконувати функції центрального процесора. Зазвичай інтегральна схема складається з арифметико-логічного пристрою, пристрою керування й іноді числового співпроцесора. Перший комерційний мікропроцесор був розроблений корпорацією *Intel* 1972 р. (*4004).*

Мова програмування (programming language) – формальна мова опису даних (інформації) з метою їх обробки на комп'ютері. Алфавіт, граматика і синтаксис, що використовуються для побудови набору інструкцій, які змушують комп'ютер виконувати

певні дії. Інструкції, написані мовою програмування, називають початковим (вихідним) кодом. Перед тим, як початковий (вихідний) код буде реально виконаний комп'ютером, його потрібно або інтерпретувати, або компілювати в машинний код. Мови програмування є штучними мовами, в яких синтаксис і семантика чітко визначені. Тому під час їх застосування за призначенням вони не допускають вільного тлумачення виразів, що характерно для природних мов.

Модель даних ієрархічна (hierarchical data model) (*модель даних даталогічна, модель даних інфологічна, модель даних фізична)* – модель, що є однією з найбільш простих і структурно визначених. У цій моделі даних зв'язки між її частинами тверді, а структурна діаграма – упорядковане дерево. Найважливішим поняттям для цієї моделі є рівень. Граф ієрархічної моделі являє собою дерево й включає два типи елементів: дуги (ребра) і вузли (вершини). Дуги, що відповідають функціональним зв'язкам, завжди повинні бути спрямовані від кореня в листи, тобто граф має орієнтацію (орграф). Розглядаючи послідовність зв'язків, можна визначити батьківську й дочірню вершини (запису) і в такий спосіб визначити ієрархічний шлях записів. Прикладом є структура файлової системи.

Модель даних реляційна (relational database) – модель, побудована на математичній теорії відносин, яка спирається на систему понять: таблиця, відношення, рядок, стовпець, ключ і домен.

Н

Накопичувач на жорсткому магнітному диску [привід жорсткого диска, жорсткий диск, дисковод] (hard disk, HD, hard disk drive, HDD) – енергонезалежний пристрій зовнішньої пам'яті з прямим доступом, призначений для постійного зберігання великих обсягів програм і даних практично у всіх комп'ютерах. Тобто вимикання та включення комп'ютера не відбивається на збереженні й цілісності даних, записаних на диск. Зазвичай використовує декілька (від двох до одинадцяти) негнучких пластин (алюмінієвих або керамічних дисків), покритих магніточутливим матеріалом і поміщених разом у герметично закритий корпус. Має досить високі

показники ємності (від 60 Гб і вище), швидкості обертання (від 3 600 об/с і вище), швидкості обміну даними з оперативною пам'яттю (від 3 Мбайт/с). Зчитуючі головки у робочому режимі не торкаються поверхні пластин завдяки прошарку потоку набігаючого повітря, утвореного на поверхні під час швидкого обертання. Відстань між головкою й диском становить кілька нанометрів (у сучасних дисках близько 10 нм), а відсутність механічного контакту забезпечує довгий термін служби пристрою. За відсутності обертання дисків головки перебувають у шпинделя або за межами диска в безпечній зоні, де виключений їх позаштатний контакт з поверхнею дисків. Засобами намагнічування (запису) на сектори, доріжки, поверхні й інші елементи структури дисків, тобто записуючими головками, наносяться тільки нулі й одиниці, що є компонентами цифрового подання збереженої інформації.

О

Об'єктно-орієнтована мова програмування [оомп, мова ооп] (object-oriented programming language, OOPL) – мова, призначена для підтримки об'єктно-орієнтованого програмування (ООП), на якій будується програма шляхом опису поведінки сукупності взаємопов'язаних об'єктів. Базовою концепцією в цьому підході є те, що об'єкт, який є структурою даних (абстрактним типом даних), інкапсульований у набір підпрограм, що іменуються методами, які призначені проводити операції з цими даними. Операції над даними можуть бути виконані тільки за допомогою цих методів, загальними для всіх об'єктів, які є екземплярами конкретних класів (з елементами спадкоємства). Отже, інтерфейс об'єктів добре визначений і дозволяє змінювати реалізацію коду методів без зміни вказаних інтерфейсів. Кожен клас є окремим модулем і має своє місце в ієрархії класів. Методи або програмні коди одного класу можуть застосовуватися вниз за ієрархією до підкласів нижніх рівнів або успадковуватися із суперкласів. Виклики процедур описуються в термінах передачі повідомлень. Повідомлення дають імена методам і можуть опціонально включати інші аргументи. Коли повідомлення надсилається об'єкту, обирається метод, відповідний цій операції.

П

Периферійні пристрої – пристрої в комп'ютерній техніці, які є доповненням до комп'ютера (наприклад, дискові накопичувачі, монітори, принтери, термінали) і сполучені з ним інтерфейсним кабелем. Часто називаються зовнішніми пристроями ПК.

Персональний комп'ютер [ПК] – мейнфрейм, робоча станція, сервер, персональний комп'ютер, персональний суперкомп'ютер, комп'ютер-моноблок, мультимедийний комп'ютер, десктоп, лептоп, портативний комп'ютер, портативні робочі станції, ноутбук, мініноутбук, субноутбук, нетбук, ультра мобільний персональний комп'ютер, смартбук, мобільний інтернет-пристрій, планшетний ноутбук, планшетний нетбук, палмтоп, комп'ютер у вигляді записника, кишеньковий ПК, хендхелд, надолонний комп'ютер, персональний цифровий секретар, підприємницький цифровий секретар, планшетний комп'ютер, інтернет-планшет, мобільний пристрій, комунікатор, смартфон, мобільний телефон, стільниковий телефон, камерофон.

Протокол мережний [протоколи: HTTP, http, FTP, TCP/IP та ін.**]** (network protocol) – стандарт, що визначає спосіб обміну даними в мережі (зокрема, і в інтернеті). Формальний набір правил і форматів (семантика і синтаксис), розроблений для форматування, тактування й керування мережними комунікаціями, що дозволяє здійснювати з'єднання й обмін даними між двома включеними в мережу комп'ютерами. Зазвичай протокол реалізується як група послуг на одному або двох рівнях моделі *OSI*. Протокол може описувати деталі взаємодії устаткування на нижніх рівнях або обміну інформацією між програмами на верхніх рівнях. Різні протоколи найчастіше описують лише різні сторони одного типу зв'язку. Узяті разом, вони утворюють стек протоколів. Назви «протокол» і «стек протоколів» також указують на ПЗ, яким реалізується відповідний протокол. Нові протоколи для інтернету визначаються *IETF*, а інші протоколи – *IEEE* або *ISO*. Організація *ITU-T* займається телекомунікаційними протоколами й форматами. Найпоширенішою системою класифікації мережних протоколів є так звана модель *OSI*, відповідно до якої протоколи поділяються на 7 рівнів за своїм призначенням – від фізичного (формування й розпізнавання електричних або інших сигналів) до прикладного (*API* для передачі інформації застосуваннями).

Протокол передачі даних (protocol) – набір правил, яким слідують комп'ютери й програми під час обміну інформацією. Існує велика кількість різних протоколів, які керують усіма аспектами зв'язку й передачі даних, – від апаратного до прикладного рівня, але всі вони подібні в тому, що задають правила, які роблять зв'язок можливим.

Р

Реєстр (registry) – спеціальна база даних ОС *Windows*, в якій міститься інформація про поточну конфігурацію комп'ютера (встановлені апаратні компоненти), встановлені програми, користувачів комп'ютера та інші параметри конкретної комп'ютерної системи.

С

Сайт [вебсайт] – сукупність вебсторінок, об'єднаних спільним змістом. Розміщується на будь-якому вебсервері під певним доменним ім'ям і реалізує віртуальне представництво організації або окремої людини в інтернеті.

Системне програмне забезпечення – програмне забезпечення, яке використовується для розробки й виконання прикладних програм. Поняття «системний» і «прикладний» досить відносні: компілятор є прикладною програмою щодо операційної системи і системною щодо компільованої програми.

У

Утиліта [допоміжна програма, програма-утиліта] – програма, що допомагає керувати файлами, отримувати інформацію про комп'ютер, діагностувати й усувати проблеми, що часто зустрічаються, і забезпечувати ефективну роботу системи. Найбільш популярні комплекти утиліт *PC Tools i Norton Utilities* пропонують такі можливості: а) керування файлами й каталогами; б) відновлення видалених файлів; в) захист від вірусів; г) отримання інформації про систему; д) ремонт диска; е) дефрагментація диска; ж) захист файлів паролем.

Ф

Файл – пойменована сфера на магнітному (жорсткому) диску або на будь-якому фізичному носієві (гнучкому диску, *CD-ROM*, флешпам'яті та ін.). У файлі може міститися будь-яка інформація, доступна для кодування в подвійному виді. Файли бувають: захищені, приховані, системні, архівні, каталоги, звичайні та ін. Використання файлів ґрунтується на т.з. файловій системі, яка підтримує систему каталогів. Каталог – це файл, який має ім'я, містить у своїй структурі інші каталоги та файли, а також може перебувати в іншому каталозі, будучи при цьому підкаталогом або підлеглим каталогом. Так утворюється ієрархічна деревовидна файлова система в операційних системах *DOS*, *Windows* і деяких інших. На кожному дисковому носієві існує кореневий каталог, той, в якому реєструються звичайні файли й підкаталоги 1-го рівня. В останніх, у свою чергу, реєструються файли й підкаталоги 2-го рівня тощо. Власне ім'я файлу утворюється з двох слів: імені й типу (розширення), розділених крапкою. Повне ім'я файлу включає: а) логічне ім'я дисковода; б) шлях до каталогу, що містить файл (послідовність імен надкаталогів, розділених знаками бекслеш "\"); в) ім'я файлу з розширенням. Наприклад, якщо файл *WORK.PAS* міститься в каталозі *PROGRAMS*, що є підкаталогом надкаталогів *SYS1* і *MODULES*, що перебувають на логічному диску *D:*, то його *повне ім'я (або шлях до нього)* запишеться у такому вигляді: *D:\SYS1\MODULES\PROGRAMS\WORK.PAS*.

Файл архіву (archive file) – файл, що складений з одного або більше файлів разом з метаданими, може включати структуру підкаталогів, інформацію для відновлення, коментарі файлів, а також використовує деяку форму стиску без втрат. Файли архіву можуть також бути зашифровані частково або цілком. Файли архіву використовуються, щоб зібрати файли різних даних разом у єдиний файл для полегшення мобільності та зберігання.

Файлова система (file system) – організаційна модель взаємодії комп'ютера з дисковими пристроями накопичення, зберігання й маніпулювання даними. Основним завданням файлової системи є приховування особливості процесу вводу-виводу на

конкретний дисковий пристрій і надання програмісту простої абстрактної моделі файлів, незалежної від використовуваних пристроїв. Для читання, створення, видалення запису, відкриття й закриття файлів в операційній системі є значна категорія системних викликів (*create, delete, open, close, read, write*). З організацією файлової системи зазвичай пов'язані такі поняття, як *каталог, поточний каталог, кореневий каталог, шлях*, для маніпулювання якими в ОС також є відповідні системні виклики.

X

«Хмарні обчислення» (cloud computing) – нова концепція, що сформувалася 2008 р. з метою надання користувачам обчислювальних ресурсів за допомогою інтернету. «Хмарою» зазвичай зветься пул віртуальних ресурсів (апаратне забезпечення, платформи розробки або сервіси), для яких забезпечені легкий доступ і простота використання. Є високотехнологічною комплексною стратегією, що одночасно базується на можливостях мережі «Інтернет» (асоціюється з «хмарою, яка накриває й об'єднує все навкруги») і використанні найсучасніших комп'ютерних технології зі сфери «комп'ютингу». Концептуально – це парадигма, відповідно до якої користувачі абстрагуються від деталей, які могли б вимагати від них відповідних знань, досвіду або будь-якого керування в інфраструктурі цієї технології (тобто «у хмарі», що їх підтримує). Звичайно, це пов'язано з наданням кінцевим користувачам засобами інтернету динамічно масштабованих і, як правило, потужних віртуальних ресурсів у вигляді замовлених ними сервісів. Термін «хмара» використовується як метафора інтернету у зв'язку з широким розповсюдженням відображення його на схемах і діаграмах у вигляді символу хмари й одночасно є абстракцією, що лежить в основі побудови його інфраструктури. Типові постачальники (провайдери) «хмарних обчислень» доставляють користувачам потрібні бізнес-застосування онлайн через звичайні веббраузери, тоді як ПЗ і дані зберігаються на серверах інфраструктури. Концепція «хмарних обчислень» базується на

ряді понять, які можна умовно поділити на такі категорії: а) «все є сервісом» (*Everything as a service, EaaS*); б) «дані як сервіс» (*Data as a Service, DaaS*); в) «інфраструктура як сервіс» (*Infrastructure as a Service, IaaS*); г) «платформа як сервіс» (*Platform as a Service, PaaS*); д) «ПЗ як сервіс» (*Software as a service, SaaS*); е) «робоче місце як сервіс» (*Workplace as a service, WaaS*); ж) «зв'язок як сервіс» (*Communications as a Service, CaaS*). Уся *Chrome OS* фактично складається тільки з одного браузера *Chrome*. Звичайних (інстальованих у ПК) програм у системі немає, є зовнішні щодо неї вебзастосування, які виконують відповідні функції «віддалено». На спеціальній панелі розташовані великі піктограми таких застосувань. Наприклад, сервіс *Gmail* реалізує функції «поштовика», а вебзастосування *Google Docs* підтримує функції офісного пакета тощо. Іншим учасником процесу створення технологічних новинок у цій галузі є корпорація *Microsoft*, яка розробила власну «хмарну» платформу *MS Windows Azure* (хмарні [лазурні] вікна). Одночасно випущена доступна для розроблювачів у *Windows Azure* версія сервера застосувань *Windows Server AppFabric*, що спрощує розгортання й керування застосуваннями, які запускаються на локальних серверах і в «хмарі». *AppFabric* поєднує хостингові й кешуючі технології, відомі раніше під кодовими іменами *Dublin* і *Velocity*, які об'єднуються зі службами *Windows Azure AppFabric Service Bus* і *AppFabric Access Control*, що раніше були відомі як *.NET Services*.

Хостинг [послуги з розміщення інформації на вузлах веб] (hosting, hosting service) – принцип та спектр послуг з розміщення й підтримки вебсторінок (сайтів і ресурсів клієнта) на сервері вебпровайдера із забезпеченням цілодобового доступу до них. Дозволяє розмістити на устаткуванні провайдера віртуальний вебсервер виду *www.name.ua* (**.com* або **.zel.ua*), де «*name*» – це ідентифікатор, пропонований користувачем і не зайнятий в обраному домені. При цьому матеріали стають доступними для інших користувачів Всесвітньої мережі. Іншими словами, хостинг забезпечує фізичне розміщення сайту на дисковому просторі спеціалізованого комп'ютера. До цього комп'ютера надходитимуть запити браузерів, в адресному рядку яких вказана *IP*-адреса або

доменне ім’я цього сайту. Провайдер бере на себе роботу з реєстрації доменного імені в організаціях, що адмініструють відповідний домен. Плата стягується за обсяг орендованої дискової пам’яті, інтенсивність трафіка та додаткові послуги.

Я

Ярлик (shortcut) – файл, що містить вказівник (посилання) на деякий об’єкт у дереві ресурсів *MS Windows*: папку, застосування, файл або принтер. Забезпечує безпосередній доступ до об’єкта з іншої папки, зокрема з робочого столу ОС *Windows*. Має розширення *.lnk* і розпізнається за заломленою стрілкою в лівому нижньому куті його значка. Роль ярликів виконують також *PIF*- і *URL*-файли. Ярлик зазвичай має піктограму, яка відповідає типу файлу застосування, що його створило (*MS Word, MS Excel* та ін.), або ж самого застосування, на яке ярлик посилається.

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Анісімов А. В. Інформаційні системи та бази даних : навчальний посібник для студентів факультету комп'ютерних наук та кібернетики / А. В. Анісімов, П. П. Кулябко. – Київ, 2017. – 110 с.

2. Антоненко В. М. Сучасні інформаційні системи і технології: управління знаннями : навч. посібник / В. М. Антоненко, С. Д. Мамченко, Ю. В. Рогушина. – Ірпінь : Нац. університет ДПС України, 2016. – 212 с.

3. Билл Джелен. Сводные таблицы в Microsoft Excel 2016 / Билл Джелен, Майкл Александер. – Издательство «Вильямс», 2017. – 480 с.

4. Леонтьев В. Office 2016. Новейший самоучитель / В. Леонтьев. – Москва : Эксмо-Пресс, 2015. – 368 с.

5. Наукові дослідження: економіко-математичний напрям : підручник / В. Д. Дербенцев, Ю. В. Пасічник, І. М. Пріхно, Г. О. Волощук, Л. А. Гордієнко. – К. : Центр учбової літератури, 2016. – 725 с.

6. Распознавание информационных операций / А. Г. Додонов, Д. В. Ландэ, В. В. Цыганок, О. В. Андрейчук, С. В. Каденко, А. Н. Грайворонская. – К. : ООО «Инжиниринг», 2017. – 282 с.

7. Інформатика та обчислювальна техніка : практикум / В. А. Одинець, О. В. Гладченко, Т. В. Ратушняк та ін. – Ірпінь : Видавництво Національного університету ДПС України, 2015. – 394 с.

8. Інформаційні системи і технології в юридичній практиці : навч. посіб. / Т. В. Ратушняк, О. В. Гладченко, В. О. Ніжегородцев, В. А. Одинець ; Державна фіскальна служба України, Ун-т ДФС України. – Ірпінь, 2017. – 302 с.

9. Інформаційні системи та технології на підприємствах та в міжнародному бізнесі : навчальний посібник для студентів напрямів підготовки 6.030504 «Економіка підприємства», 6.030503 «Міжнародна економіка» / уклад. В. В. Чаговець ; Харківський держ. ун-т харчування та торгівлі. – Харків : ХДУХТ, 2016. – 168 с.

10. Костріков С. В. Географічні інформаційні системи : навчально-методичний посібник / С. В. Костріков, К. Ю. Сегіда. – Харків, 2016. – 82 с.

11. Красюк Ю. М. Прикладна інформатика. Опорний конспект (для студентів факультету економіки та управління) / Ю. М. Красюк, М. В. Сільченко. – К. : Центр навчальної літератури, 2019. – 181 с.

12. Красюк Ю. М. Бізнес-інформатика у структурно-логічних схемах та прикладах. Опорний конспект / Ю. М. Красюк, Т. О. Кучерява, М. В. Сільченко. – К. : Центр навчальної літератури, 2019. – 134 с.

13. Козлов Д. А. EXCEL 2016. Полное руководство / Д. А. Козлов, В. В. Серогодский. – Издательство «Наука и техника», 2017. – 416 с.

14. Конспект лекцій з дисципліни «Геоінформатика» (для студентів напряму підготовки «Геодезія, картографія та землеустрій»). – Частина І / О. М. Шикула, І. М. Шквир, А. А. Москаленко, Т. А. Гезь. – Київ, 2015. – 241 с.

15. Конспект лекцій з дисципліни «Геоінформатика» (для студентів напряму підготовки «Геодезія, картографія та землеустрій»). – Частина ІІ / О. М. Шикула, І. М. Шквир, А. А. Москаленко, Т. А. Гезь. – Київ, 2015. – 305 с.

16. Рудикова Л. Microsoft Office Excel 2016 / Л. Рудикова. – Издательство «БХВ-Петербург», 2017. – 640 с.

17. Комп'ютери та комп'ютерні технології : навчальний посібник / В. П. Лисенко, М. О. Кіктєв, Д. С. Комарчук, Р. М. Вдовин. – Частина 2. – К. : ЦП «Компринт», 2015. – 703 с.

18. Лучко М. Р. Інформаційні системи і технології в обліку й аудиті : навчальний посібник / М. Р. Лучко, О. В. Адамик. – Тернопіль : ТНЕУ, 2016. – 252 с.

19. Морзе Н. В. Інформаційні системи : навч. посібн. /за наук. ред. Н. В. Морзе ; Н. В. Морзе, О. З. Піх. – Івано-Франківськ : ЛілеяНВ, 2015. – 384 с.

20. Майкл Александер. Формулы в Microsoft Excel 2016 / Майкл Александер. – Издательство «Вильямс», 2017. – 784 с. : ил.

21. Нелюбов В. О. Основи інформатики. Microsoft Word 2016 : електронний навчальний посібник / В. О. Нелюбов, О. С. Куруца. – Ужгород : ДВНЗ «УжНУ», 2018. – 96 с.

22. Нелюбов В. О. Основи інформатики. Microsoft Excel 2016 : навчальний посібник / В. О. Нелюбов, О. С. Куруца. – Ужгород : ДВНЗ «УжНУ», 2018. – 58 с.

23. Комолова Н. Программирование на VBA в Excel 2016 / Н. Комолова, Е. Яковлева. – Издательство «БХВ-Петербург», 2017. – 432 с.

24. Іванов В. Г. Основи інформатики та обчислювальної техніки : підручник / В. Г. Іванов, В. В. Карасюк, М. В. Гвозденко ; за заг. ред. В. Г. Іванова. – Х. : Право, 2015. – 312 с.

25. Правове регулювання інформаційної діяльності : навч. посіб. / В. Ю. Жарких, О. С. Білоусов, І. Б. Кривдіна, Д. Д. Татакі, І. М. Чістякова, Б. Г. Шевченко, С. Х. Яворський. – К. : Каравела, 2015. – 216 с.

26. Ратушняк Т. В. Розробка навчального модуля «Особові рахунки» для АРМ працівника банку засобами MS ACCESS [Електронний ресурс] / Т. В. Ратушняк, А. С. Шостак // Проблеми впровадження інформаційних технологій в економіці : матеріали IX Міжнародної науково-практичної інтернет-конференції, 17–18 травня 2018 року. – Ірпінь : УДФСУ, 2018. – С. 241–242. – Режим доступу : http://ir.nusta.edu.ua/jspui/handle/doc/3005

27. Садко М. Г. Бази даних та системи управління базами даних : навч.-метод. посіб. / М. Г. Садко, П. М. Сорока, С. П. Саяпін. – К. : ЦП «Компринт», 2016. – 119 с.

28. Сільченко М. В. Інформатика. Комп’ютерний аналіз економічних даних: моніторинг знань : зб. практ. завдань [Електронний ресурс] / М. В. Сільченко, Т. О. Кучерява, Ю. М. Красюк ; за заг. ред. О. Д. Шарапова ; М-во освіти і науки, молоді та спорту України, ДВНЗ «Київський нац. екон. ун-т ім. В. Гетьмана». – К. : КНЕУ, 2013. – 354, [6] с.

29. Сільченко М. В. Прикладна інформатика. Опорний конспект (для студентів факультету фінансів та факультету міжнародної економіки та менеджменту) / М. В. Сільченко, Т. О. Кучерява. – К. : Центр навчальної літератури, 2019. – 181 с.

30. Сільченко М. В. Business Intellegence (usingMS Office). Опорний конспект / М. В. Сільченко. – К. : Центр навчальної літератури, 2019. – 116 с.

31. Серогодский В. В. Microsoft Office 2016 / Office 365. Полное руководство / В. В. Серогодский. – Издательство «Наука и техника», 2017. – 448 с.

32. Сорока П. М. Інформаційні системи і технології в управлінні організацією : навч. посіб. / П. М. Сорока, В. В. Харченко, Г. А. Харченко. – К. : ЦП «Компринт», 2019. – 518 с.

33. Сорока П. М. Засоби обробки таблиць : навч.-метод. посіб. з дисципліни «Сучасні інформаційні технології» / П. М. Сорока, Н. П. Тмєнова. – К. : ТОВ «Гліф Медіа», 2017. – 185 с.

34. Сорока П. М. Основи операційних систем : навч.-метод. посіб. з дисципліни «Сучасні інформаційні технології» / П. М. Сорока. – К. : ТОВ «Гліф Медіа», 2017. – 65 с.

35. Сорока П. М. Засоби обробки текстів : навч.-метод. посіб. з дисципліни «Сучасні інформаційні технології» / П. М. Сорока, О. Є. Іларіонов. – К. : ТОВ «Гліф Медіа», 2017. – 112 с.

36. Сорока П. М. Інформаційні системи і технології у фінансах : навч. посіб. / П. М. Сорока. – К. : ТОВ «Аграр Медіа Груп», 2016. – 260 с.

37. Супрунюк Г. М. Прикладна інформатика. Опорний конспект (для студентів факультету управління персоналом соціології та психології) / Г. М. Супрунюк, М. В. Сільченко, Т. О. Кучерява. – К. : Центр навчальної літератури, 2019. – 143 с.

38. Тарасюк Г. М. Планування комерційної діяльності : навч. посібник / Г. М. Тарасюк. – К. : Каравела, 2015. – 400 с.

39. Тмєнова Н. П. Комп'ютерна графіка : навчально-методичний посібник / Н. П. Тмєнова. – К. : ВПЦ «Київський університет», 2017. – 111 с.

40. Трофименко О. Г. Організація баз даних : навчальний посібник / О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, І. М. Копитчук. – 2-ге вид. виправ. і доповн. – Одеса : Фенікс, 2019. – 246 с.

41. Трофименко О. Г. Сучасні інформаційні та комунікаційні системи і технології : метод. вказівки для практ. занять / О. Г. Трофименко, Н. І. Логінова. – Одеса : ВЦ НУ «ОЮА», 2016. – 120 с.

42. Трофименко О. Г. Бази даних: створення та опрацювання : навч. посібн. / О. Г. Трофименко, Л. М. Буката, Ю. В. Прокоп. – Одеса, 2016. – 226 с.

43. Трофименко О. Г. Електронні засоби автоматизації документообігу. Робот а в M SWord / О. Г. Трофименко, Л. М. Буката, М. В. Малигіна. – ОНАЗ, 2015. – 104 с.

44. Федусенко О. В., Доманецька І. М., Красовська Г. В. Проєктування систем електронного документообігу : навчальний посібник / О. В. Федусенко, І. М. Доманецька, Г. В. Красовська. – К. : КНУБА, 2016. – 88 с.

45. Windows 10. Все, що потрібно знати про нову ОС [Електронний ресурс]. – Режим доступу : http://tsn.ua/special-projects/windows10/ – Назва з екрану.

46. Windows 10: переваги та недоліки [Електронний ресурс]. – Режим доступу : https://www.ua5.org/windows/408-windows-10-perevagi-ta-nedoliki.html – Назва з екрану.

47. E-Learning Glossary. Kaplan-Leiserson Eva. WEB-сайт [Електронний ресурс]. – Режим доступу : http://www.learningcircuits.org/glossary.html

48. Encyclopedia Britannica. WEB-сайт [Електронний ресурс]. – Режим доступу : http://www.Britannica.com

49. Glossary. WEB-сайт [Електронний ресурс]. – Режим доступу : http://www.ifsworld.com/about_ifs/glossary.asp

50. Glossary of MSDN. WEB-сайт [Електронний ресурс]. – Режим доступу : http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netstart/html/cpconGlossary.asp

51. Glossary of e-Commerce. WEB-сайт [Електронний ресурс]. – Режим доступу : http://www.ifsworld.com/about_ifs/glossary.asp

52. Glossary of internet & intranet. WEB-сайт [Електронний ресурс]. – Режим доступу : http://www.nur.yamal.ru/operating_systems/internet_intranet/nbg2iig.shtml

53. Glossary of terms and abbreviations for the Anvil WEB site. WEB-сайт [Електронний ресурс]. – Режим доступу : http://www.anvil.eu.com/find/Glossary-english.htm#EU

54. Glossary of terms for internet resources. WEB-сайт [Електронний ресурс]. – Режим доступу : http://www.ucm.es/INET/hytelnet_html/glossary.html

55. Glossary of terms found in the WEB services architecture. WEB-сайт [Електронний ресурс]. – Режим доступу : http://www.w3.org/TR/ws-gloss/

56. IT glossary. WEB-сайт [Електронний ресурс]. – Режим доступу : http://www.sdnp.org.gy/it/glossary

57. Soroka P. M. Information system sandtechnologies. Laboratory Practical: Educationalbook / P. M. Soroka, V. V. Kharchenko. – K. : ЦП «Компринт», 2016. – 153 p.

Навчальне видання

СЕРІЯ «НА ДОПОМОГУ СТУДЕНТУ УДФСУ»

Гладченко Оксана Вікторівна,
Ніжегородцев Владислав Олександрович,
Одинець Володимир Андрійович,
Поденежко Олександр Васильович,
Ратушняк Тетяна Володимирівна

ЕКОНОМІЧНА ІНФОРМАТИКА: КУРС ЛЕКЦІЙ

Навчальний посібник

Відповідальний за випуск — *Д. Ф. Салахова*

Відповідальний редактор — *І. Є. Врублевська*
Редактори — *Н. І. Грицюк, Л. Б. Дьомена, О. В. Данильчик*

Форматування та комп'ютерна верстка — *Д. П. Завальніцька*

Здано до друку 12.03.2021. Формат 60×84/14
Папір офсетний № 1. Гарнітура «Times New Roman»
Ум. друк. арк. 24.8
Наклад 300 прим. Замовлення № 943

Підготовлено до друку Видавничо-поліграфічним центром
Університету ДФС України
08205, вул. Університетська, 31, м. Ірпінь, Київська область,
Україна

Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції Серія ДК № 5104 від 20.05.2016

СЕРІЯ «НА ДОПОМОГУ СТУДЕНТУ УДФСУ»

Серію «На допомогу студенту УДФСУ» засновано 2016 року.

Редакційна колегія:
Шевчук О. А., д.е.н.
Топчій В. В., д.ю.н.
Краєвський В. М., д.е.н.
Кужелєв М. О., д.е.н.
Шевчук С. В., д.е.н.
Суліма Є. М., д.філос.н., член-кореспондент НАПНУ
Горбовий А. Ю., д.т.н.
Чмелюк В. В., к.ю.н.
Малинський І. Й., к.н.фіз.вих.
Шевчук В. А., к.ю.н.

У СЕРІЇ «НА ДОПОМОГУ СТУДЕНТУ УДФСУ» ВИЙШЛИ ДРУКОМ:

2016

«Методичні основи спеціальної фізичної та технічної підготовки студентів за розділом «Легка атлетика»

«Самостійна робота студента як одна з форм впливу на функціональну, фізичну та психологічну підготовленість»

«Організація роботи командира механізованого взводу»

«5.45-мм автомати Калашникова (АК-74, АКС-74, АК-74Н, АКС-74Н) та 5.45-мм ручні кулемети Калашникова (РПК-74, РПКС-74, РПК-74Н, РПКС-74Н)»

«Гранатомет підствольний ГП-25»

«Ручні гранати»

«Кулемети Калашникова – 7.62, ПК, УЖМ, ПКТ»

«Ручний протитанковий гранатомет РПГ-7»

«9-мм пістолет Макарова (ПМ)»

2017

«Вища та прикладна математика»

«Цивільний захист»

«Програмування мовою JAVA : практикум»

«Інформаційні системи і технології в юридичній практиці»

«Дослідження операцій : практикум»

«Чисельні методи»

«English for Students of Finance»

«Основи військової розвідки»

2018

«CASE-технології. Міждисциплінарне інформаційне моделювання»

«Економічна інформатика: практикум»

«Економічна теорія (політекономія, мікроекономіка, макроекономіка). Політекономія»

«Економічна теорія (політекономія, мікроекономіка, макроекономіка). Мікроекономіка»

«Економічна теорія (політекономія, мікроекономіка, макроекономіка). Макроекономіка»

«Охорона праці»

«Економіка і організація діяльності об'єднань підприємств»

«Основи християнської культури»

«Економіка підприємства»

«Фізика»

«Трудове право України»

2019

«Основи тактичної медицини»
«Аудит»
«Збірник задач. Вища та прикладна математика»
«Міжнародні розрахунки та валютні операції»
«Підготовка озброєння механізованого взводу до бойового застосування»
«Контролінг в управлінні підприємством»
«Актуальні питання судової експертизи (у питаннях і відповідях)»
«Інноваційний менеджмент»
«Організація навчальних занять з фізичного виховання за методом колового тренування»
«Теорія судових доказів (у таблицях і схемах)»
«Статистика»
«Фінансовий менеджмент проектів і програм»
«Ручний протитанковий гранатомет РПГ-7 (РПГ-7Д)»
«Організація роботи командира механізованого відділення»
«Психологія управління»
«Deutsch und Wirtschaft»
«Основи тактичної підготовки працівників правоохоронних органів»
«Основи кінології»
«Моделювання систем»
«UML. Уніфікована мова моделювання інформаційних систем»
«Історія держави і права України»
«Економічна теорія»

2020

«Менеджмент»
«Бізнес-аналітика та моделювання»
«Безпека життєдіяльності»
«Статистичний аналіз даних»
«Хімія»
«Мікробіологія»
«Медичне право»
«Бухгалтерський облік в управлінні підприємством»
«Загальна психологія»
«Хортинг у закладах вищої освіти»
«Теорія і методика викладання оздоровчого фітнесу»
«Організація охорони державної таємниці в Україні»
«Цивільне право України (особлива частина) »
«Цивільний процес України»
«Оздоровчий туризм»
«English Grammar for Philology Students»
«Фітнес-технології фізичного виховання у закладах вищої освіти»
«Конкурентна розвідка»
«Теорія та методика бойового хортингу»
«Логіка»
«Інституціональна економіка»
«Економетрика: прикладний аспект»
«Основні поняття сучасної алгебри»

2021

«Теоретичні основи товарознавства»
«Економічна інформатика: курс лекцій»