МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА
на тему:

РОЗРОБКА ЧАТ-БОТУ TELEGRAM

Здобувача вищої освіти першого (бакалаврського) рівня,
4 курсу, групи КБ-19-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»

_____ Жебровський Леонід Сергійович
(підпис) (прізвище, ім'я, по батькові)

Науковий керівник _____ Гладченко Оксана Вікторівна
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-професійної програми _____ Федотова – Півень Ірина Миколаївна, к.т.н, доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри
протокол від «26» травня 2023 № 13

Завідувач кафедри _____ Філоненко Михайло Миколайович
(підпис) к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2023

Реєстрація № 23 06.06.2023 _____
(дата) (підпис відповідальної особи)

Результати перевірки До захисту 29.05.2023 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 4 (добре) 15.06.2023 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ В.М. Франчук
(підпис)

Члени комісії:

_____ М.М. Філоненко
(підпис)

_____ О.В. Редич
(підпис)

_____ Є.В. Котух
(підпис)

БЛАНК ОЦІНЮВАННЯ ВИПУСКНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	28		15.06.2023 Франчук В.М. Філоненко М.М Редич О.В. Котух Є.В. Суховецька Л.П.
2. Дотримання вимог щодо оформлення	від 0 до 10	7		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		
4. Рецензент	від 0 до 5	4		
РАЗОМ БАЛІВ	від 0 до 50	43		

Оцінювання кваліфікаційної роботи науковим керівником – 42 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалоюЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
85	B	4 (добре)	15.06.2023	Франчук В.М. Філоненко М.М Редич О.В. Котух Є.В. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Інформаційні управляючі системи і технології (за галузями)»

Затверджую
Завідувач кафедри
_____ Філоненко М.М.
«16» 11 2022

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Жебровський Леонід Сергійович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Розробка чат боту Telegram
Затверджена наказом ректора від «15» листопада 2022 № 1205 (зі змінами наказ від «04» квітня 2023 № 361)
Затверджена наказом ректора від «15» листопада 2022 № 1205
2. Строк здачі закінченої роботи «26» травня 2023
3. Цільова установка
Комплексний аналіз розробки Telegram боту
Мета роботи
Створення боту у месенджері Telegram
Об'єкт дослідження
Месенджер Telegram та його складові
Предмет дослідження
Мова програмування Python, її головні складові та бібліотеки.
База дослідження
PyCharm - гібридна платформа, розроблена JetBrains як IDE для Python.
4. Вихідні дані до роботи
Бібліотеки Telegram api та сам Telegram
5. Перелік графічного матеріалу
Рисунки, знімки екрану та блок-схеми
6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)
РОЗДІЛ 1 1.1. Теоретичні відомості про Telegram, 1.2. Огляд і порівняльний аналіз інтернет-ботів, 1.3. Формулювання вимог до проєкту
РОЗДІЛ 2 2.1. Python, 2.2. IDE, 2.3. PyCharm
РОЗДІЛ 3 3.1. Створення бота через BotFather та завантаження PyCharm, 3.2. Написання бота у PyCharm, 3.3. Результат роботи бота в Telegram.

4

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційноїроботи	Строк виконанн	
		За планом	
1	2	3	
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 15.11.2022	
2	**І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	01.04.2023	
3	**ІІ розділ** (аналіз експериментальної частини про методики досліджень)	20.04.2023	
4	**ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проєктні рішення)	14.04.2023	
5	**Висновки**	21.05.2022	
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	26.05.2023	
7	Отримання відгуку керівника	06.06.2023	0
8	Отримання зовнішньої рецензії	08.06.2023	0

8. Дата видачі завдання «16» листопада 2022

9. Керівник випускної кваліфікаційної роботи _____ (підпис) Гладченко О.В (прізвище, ініціали)

10. Керівник освітньо-професійної програми _____ (підпис) Федотова-Півен (прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти _____ (підпис) Жебровський Л (прізвище, ініціали)

АНОТОЦІЯ

РОЗРОБКА ЧАТ-БОТУ TELEGRAM

Мета роботи полягає в створення інтерактивного та автоматизованого інструменту комунікації з користувачами.

Завданням роботи є створення чат-боту, який здатний відповідати на запитання, надавати інформацію та виконувати завдання, що перед ним ставляться.

Предметом дослідження є мова програмування Python, її головні складові та бібліотеки.

Об'єктом дослідження є месенджер Telegram та його складові.

Базою дослідження виступає гібридна платформа, розроблена JetBrains як IDE для Python. Він зазвичай використовується для розробки додатків Python, а це саме PyCharm.

За результатами дослідження сформульовані висновки, що мова програмування Python та гібридна платформа IDE PyCharm э ідеальним середовищем для розробки і написання Telegram боту.

Одержанні результати можуть бути використані в подальшому формуванні і написанні інших, більш складних і повноцінних ботів в Telegram.

Кваліфікаційна робота містить 48 сторінок, 24 рисунка, список використаних джерел з 15 найменувань.

Ключові слова : Python, IDE, PyCharm, чат-бот, Telegram.

Рік виконання кваліфікаційної роботи 2023

Рік захисту роботи 2023

The goal of the work is to create an interactive and automated tool for communication with users.

The task of the work is to create a chatbot that is able to answer questions, provide information and perform the tasks set before it.

The subject of research is the Python programming language, its main components and libraries.

The object of the study is the Telegram messenger and its components.

The basis of the research is a hybrid platform developed by JetBrains as an IDE for Python. It is commonly used to develop Python applications, namely PyCharm.

Based on the results of the research, the conclusions were drawn that the Python programming language and the PyCharm IDE hybrid platform are an ideal environment for developing and writing a Telegram bot.

The obtained results can be used in the further formation and writing of other, more complex and full-fledged bots in Telegram.

The qualification work contains 48 pages, 24 figures, a list of used sources from 15 titles.

Keywords: Python, IDE, PyCharm, chatbot, Telegram.

The year of performance of qualification work is 2023

The year of job protection is 2023

ЗМІСТ

ВСТУП 7
РОЗДІЛ 1. ОСНОВНІ ПОНЯТТЯ ПРО TELEGRAM, ІНТЕРНЕТ-БОТИ ТА ТЕХНОЛОГІЇ ЇХ РОЗРОБКИ 8
1.1. Теоретичні відомості про Telegram 8
1.2. Огляд і порівняльний аналіз інтернет-ботів 12
1.3. Формулювання вимог до проєкту 14
РОЗДІЛ 2 ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБКИ 18
2.1. Python 19
2.2. IDE 29
2.3. PyCharm 29
РОЗДІЛ 3 РОЗРОБКА TELEGRAM БОТА 33
3.1. Створення бота через BotFather та завантаження PyCharm 33
3.2. Написання бота у PyCharm 38
3.3. Результат роботи бота в Telegram 44
ВИСНОВКИ 47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 48

ВСТУП

У сучасному світі інформаційні технології постійно розвиваються, а способи комунікації з користувачами стають все розмаїтішими та ефективнішими. Одним з інноваційних засобів комунікації є чат-боти, які здатні автоматизувати та спростити взаємодію з користувачами.

Особливо популярними серед месенджерів, які надають платформу для реалізації чат-ботів, є Telegram. Його зручний інтерфейс, широкий функціонал та велика аудиторія користувачів роблять його привабливим вибором для розробки чат-ботів.

Ця дипломна робота присвячена розробці чат-боту для платформи Telegram з метою створення інтерактивного та автоматизованого інструменту комунікації з користувачами. Головна мета роботи - створення чат-боту, який здатний відповідати на запитання, надавати інформацію та виконувати завдання, що перед ним ставляться.

Для досягнення цієї мети, у роботі будуть розглянуті ключові вимоги до чат-боту, проведений аналіз існуючих рішень та технологій для розробки чат-ботів. В результаті буде обрано найбільш відповідні технології та інструменти для реалізації проекту.

Після реалізації чат-боту буде проведено тестування його функціональності та ефективності. Виявлені помилки та недоліки будуть виправлені, а результати тестування допоможуть оцінити працездатність та ефективність розробленого чат-боту.

Отриманий чат-бот може знайти застосування в різних галузях, починаючи від підтримки клієнтів і закінчуючи автоматизацією бізнес-процесів.

Актуальність та практична значимість дипломної роботи обумовлена високою популярністю месенджерів та інтернет-ботів серед користувачів мережі Інтернет.

РОЗДІЛ 1. ОСНОВНІ ПОНЯТТЯ ПРО TELEGRAM, ІНТЕРНЕТ-БОТИ ТА ТЕХНОЛОГІЇ ЇХ РОЗРОБКИ

Telegram – це безкоштовний месенджер для смартфонів та персональних комп'ютерів під управлінням всіх найбільш поширених на сьогоднішній день операційних систем, що дозволяє обмінюватися не тільки текстовими повідомленнями, а й різними медіафайлами (малюнки, музика, архіви, текстові документи).

1.1. Теоретичні відомості про Telegram

Функціонально Telegram схожий на інші месенджери, його головні переваги перед конкурентами – швидкість, захищеність, зберігання даних у хмарі (віддалений сервер) та безкоштовність.

Телеграм є досить популярним месенджером.

Telegram має більше 550 мільйонів активних користувачів щомісяця (і це вже застарілі данні, зараз набагато більше).

Тільки за останні 3 з половиною роки додаток для обміну повідомленнями додав свою базу користувачів на 350 мільйонів (трохи більше 60% усіх існуючих активних користувачів Telegram).

За словами засновника Telegram Павла Дурова, 25 мільйонів нових користувачів приєдналися до Telegram під час 72-годинного сплеску в січні 2021 року. Це дорівнює приблизно 8,33 мільйонам нових реєстрацій на день. І тільки в січні Telegram почали використовувати понад 100 мільйонів нових користувачів.

Насправді база користувачів Telegram зростає більш ніж на 40% щороку з моменту його запуску в 2013 році.

Згідно з даними Apptopia, у Telegram щоденно 55,2 мільйона активних користувачів на iOS і Android. Зростання більш ніж на 30% з листопада 2020 року.

Telegram має 55,2 мільйона щоденних активних користувачів [8].

Станом на березень 2014 року Telegram мав 15 мільйонів щоденних активних користувачів. Збільшення порівняно зі 100 000 щоденних активних користувачів у жовтні 2013 року.

4 жовтня 2021 року під час припинення роботи Facebook і Instagram кількість користувачів Telegram за один день збільшилася на 70 мільйонів користувачів.

Згідно з даними Sensor Tower, у січні 2021 року Telegram був найбільш завантажуваним додатком (в App Store і Google Play) у світі з понад 63 мільйонами завантажень.

Станом на 31 серпня 2021 року Telegram отримав приголомшливий 1 мільярд завантажень і став 15-м додатком, який зробив це в усьому світі.

Telegram є сьомим за популярністю неігровим додатком у Google Play за кількістю завантажень у всьому світі.

Станом на серпень 2021 року за загальною кількістю завантажень Telegram був 7-м найпопулярнішим мобільним додатком для iOS і Android.

Середній користувач Telegram на Android витрачає 2,9 години на місяць на використання програми. Це трохи більше, ніж 2,7 години на місяць, які люди витрачають на Facebook Messenger.

Зараз команда розробників Telegram знаходиться в Дубаї, Об’єднані Арабські Емірати. Емітентом мобільного додатку Telegram є Telegram FZ-LLC, організація, зареєстрована в Об’єднаних Арабських Еміратах.

Юридична адреса материнської компанії Telegram (Telegram Group Inc.) розташована на Британських Віргінських островах.

Законним представником законодавства про захист даних є компанія Telegram UK Holdings Ltd, розташована у Великобританії.

За даними Statista, Telegram є 10-ю найбільш використовуваною платформою соціальних медіа в усьому світі з 550 мільйонами активних користувачів.

База користувачів Telegram складається з 41,4% жінок і 58,6% чоловіків.

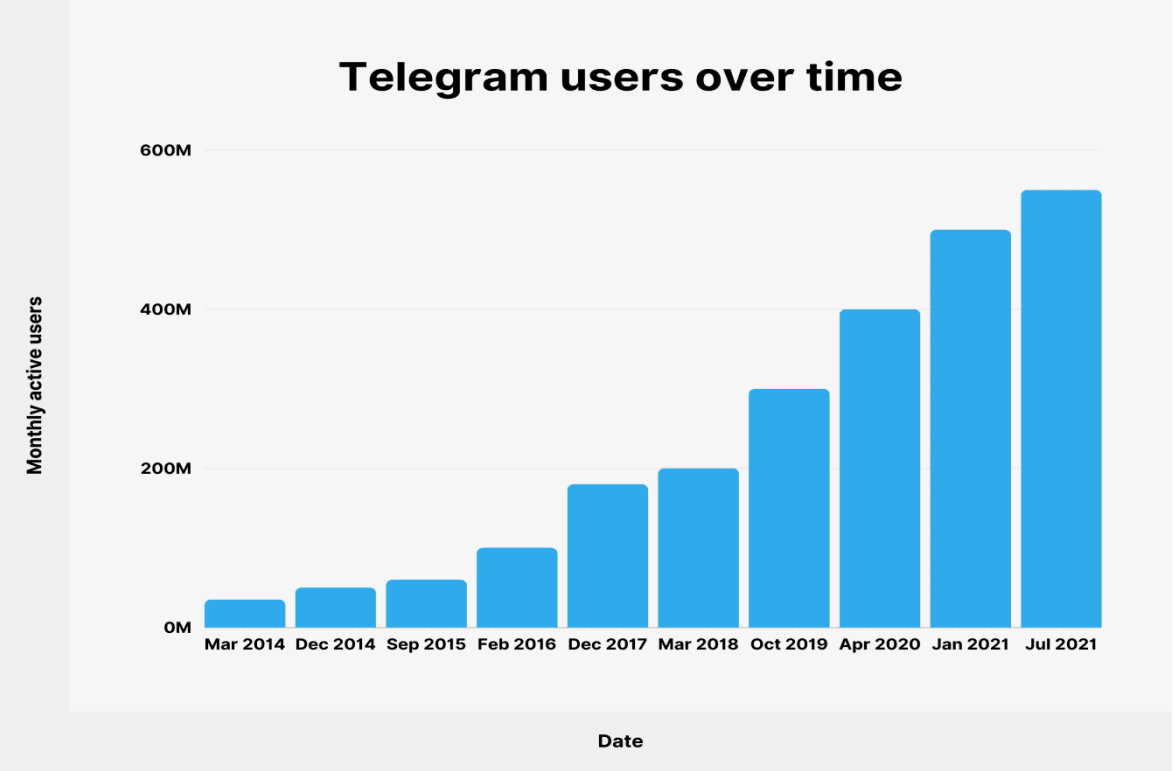


Рис. 1. Кількість користувачів Telegram рік від року

Свій трафік Telegram надійно зашифровує, всі обчислення виробляються на віддаленому сервері, а клієнтська частина лише шифрує дані та відправляє їх на сервер. Для стійкої роботи було створено унікальний протокол MTProto, що суттєво підвищило безпеку та захист від несанкціонованої втрати інформації.

При одночасному обміні повідомленнями та фотографіями учасників чата, конкуренти значно поступаються Telegram за швидкістю доставки інформації.

Боти – це спеціальні програми, що виконують різні функції та полегшують життя їх користувачів. Написані для платформи Telegram, вони призначені для виконання різних функцій: від отримання новин до пошуку інформації та навіть торгівлі акціями. Головна мета робота полягає у автоматичній відповіді після введеної користувачем команди. При цьому, працюючи безпосередньо через інтерфейсу Telegram, програма імітує події живого користувача, за рахунок чого користування таким ботом набагато зручніше і зрозуміло.

Самі боти діляться на кілька напрямків:

- Чат-боти є найпростішим чатом, що імітує спілкування на задану тематику.

- Боти-інформатори – окремий вид ботів, головна мета яких заключається в інформуванні користувача про ті чи інші події (новини, заходи, публікації тощо).
- Ігрові боти – це боти, з якими можна пограти в різні ігри.
- Боти-ассистенти – це боти, розроблені різними онлайн-сервісами як доповнення до основної веб-версії.

Алгоритм роботи бот-утиліт полягає в наступному: повідомлення, команди та запити, надіслані користувачами, передаються на програмне забезпечення, запущене на серверах розробників. Посередницький анонімний сервер Telegram обробляє шифрування та здійснює зворотний зв'язок між утилітою та користувачем.

Взаємодія між користувачем та ботом виглядає наступним чином: користувач віддає команду боту, бот передає команду на сервер, після цього програма на сервері обробляє отриманий від робота запит, потім сервер дає відповідь роботу, в результаті бот виводить відповідь на екран додатки користувачеві. І цей цикл повторюється щоразу, коли користувач взаємодіє з будь-яким телеграм-ботом.

Взаємодія з серверами відбувається за допомогою простого HTTPS інтерфейсу, який є спрощеною версією API Telegram. Інакше цей інтерфейс можна назвати програмним каталогом чи боталгоритмом. Нові bot-утиліти створюються за допомогою спеціальної утиліти @BotFather.

Бот в Telegram створюється за допомогою іншого бота під назвою BotFather. Для цього потрібно надіслати йому команду /newbot, вибрати ім'я, яке відображатиметься у списку контактів, та адресу.

Якщо адреса не зайнята, а ім'я введено правильно, BotFather надішле в відповідь повідомлення з токеном – "ключом" для доступу до створеного бота [9].

Коли бот зареєстрований, його потрібно наділити функціоналом. І тому можна використовувати різні методи. Існують різні послуги, з допомогою яких можна наділяти робота певними функціями, але для більш гнучкого налаштування потрібна робота з мовою програмування.

У мовах програмування існують різні фреймворки, спрямовані працювати безпосередньо з Telegram. Вибір мови буде розглянуто нижче, у відповідному розділі [4].

1.2. Огляд і порівняльний аналіз інтернет-ботів

Вимоги до сервісу повинні містити найбільш ефективні для даної задачі розв'язання. З цього випливає, що при вивченні та порівнянні сервісу з аналогами основними завданнями можна виділити чітке формулювання вимог до свого проєкту та знаходження найбільш успішних рішень у вже існуючих розробках.

Вивчення вже розроблених конкретних аналогів, розгляд їх переваг та недоліків, маркетингових моделей, технічних рішень допоможе краще зрозуміти, яким має бути створюваний сервіс, які елементи необхідно включити, а яких немає потреби, сформулювати вимоги до проєкту.

Для цього було проведено аналіз різних сервісів на підставі їх власного позиціонування на власних платформах, статей та відгуків про них простих користувачів, особистий досвід використання.

Основними критеріями були готова версія продукту, затребуваність у цільової аудиторії споживачів, різні технічні чи маркетингові рішення сервісів вивчення кількох підходів до реалізації товару.

Під ці критерії підійшли такі проєкти: ControllerBot, Combot, Channely.

ControllerBot. ControllerBot є найпопулярнішим ботом для публікації контенту. У його функціонал входять: відкладена публікація, додавання коментарів під публікацією, відкладене видалення публікації.

До його переваг можна віднести:

- Простий та зрозумілий інтерфейс.
- Безстрокову безкоштовну версію.

Також є низка серйозних недоліків:

- Невеликий функціонал.
- Відсутність web-версії.
- Не завжди працює стабільно.
- Немає можливості опублікувати фотографії у форматі "Альбом".

Combot. Серед його можливостей – детальна аналітика, налаштування модерації, система репутації, тригери, попередження, а також система антиспам та різні звіти.

Переваги:

- Є безкоштовний тариф до 200 передплатників.
- Різноманітність функцій та налаштувань.
- Глибока статистика каналом.
- Наявність мобільної версії.

Недоліки:

- Мінімальний платний тариф 9.99$ на місяць.
- Складний інтерфейс.

Channely. У його можливості входить: відкладений постінг, автовидалення, додавання кнопок та реакцій, закріплення повідомлень, редагування надісланих, надсилання кількох повідомлень. З відмінностей: мультипостинг, пересилання постів.

Переваги:

- Широкий функціонал.
- Унікальні функції мультипостингу та пересилання публікацій у інші канали.
- Наявність мобільної версії.

Недоліки:

- Мінімальний платний тариф 5$ на місяць за канал.
- Довга відсутність оновлень.

Більшість ботів мають маркетингову стратегію, що полягає у наявності постійної або тимчасової безкоштовної версії телеграм-бота та розширеної платної версії. Такий підхід дозволяє залучити ширше коло користувачів, оскільки дає можливість вивчити основний функціонал без фінансових витрат.

Через обмеження в інтерфейсі роботів самого Телеграма, наявність webінтерфейсу значно полегшує взаємодію користувача з продуктом, даючи можливість бачити та взаємодіяти з великим кількістю параметрів та опцій, ніж у месенджері. Можливість налаштувати власний інтерфейс у web-версії дозволяє

зробити його більш зручним та зрозумілим для широкої аудиторії користувачів. Багато телеграм-ботів мають функціонал, необхідний для управління медійними каналами, але не потрібно простим користувачам для організацій конференцій, локальних спільнот.

Наявність великої кількості функцій може ускладнювати інтерфейс для сприйняття звичайних користувачів.

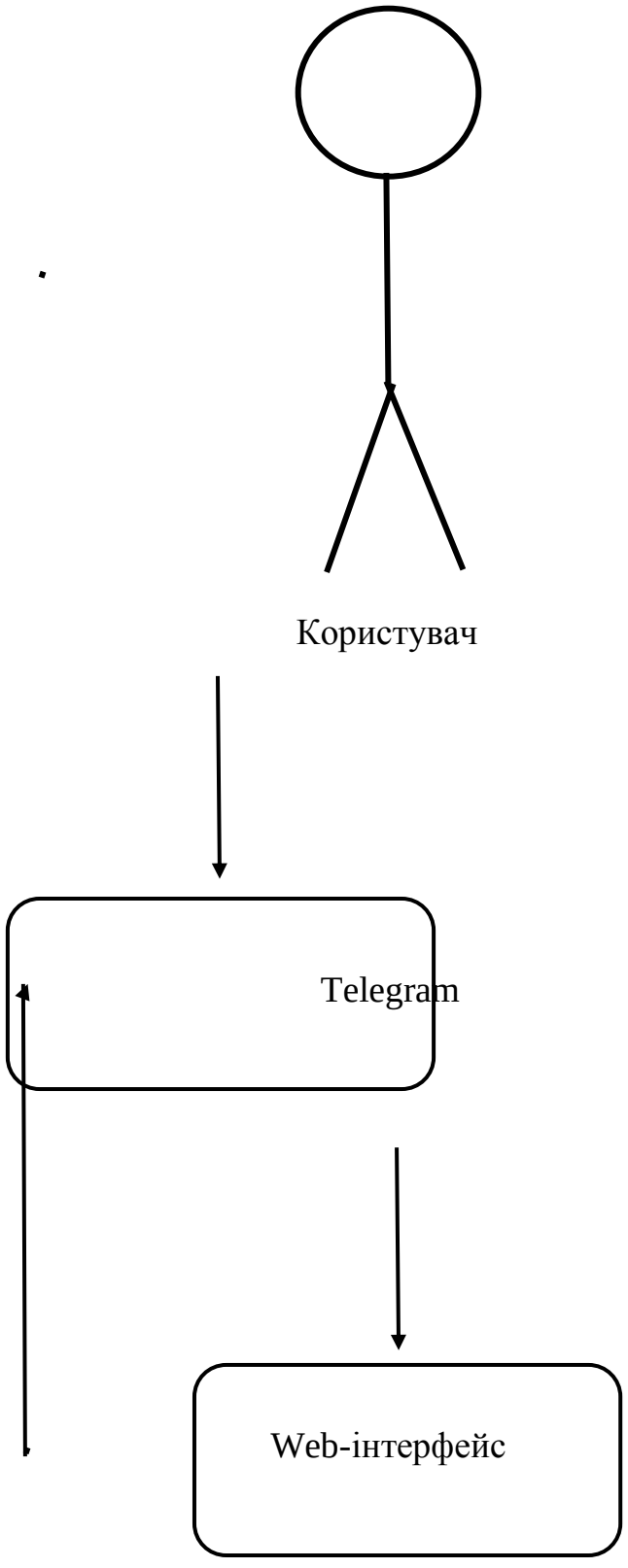
1.3. Формулювання вимог до проєкту

Правильне висунення вимог – один із ключових етапів формування проєкту. Поганий чи неопрацьований план може зробити процес розробки неефективним, змінити кінцевий результат, віддаливши його від початкової мети сервісу, зробити проєкт неможливим для виконання.

У формуванні вимог слід орієнтуватися на переваги користувачів. Виходячи з їх переваг і потреб, буде складено список вимог, що поєднуються із поставленими цілями.

Найбільш поширеними потребами користувачів можуть бути:

- Швидкий доступ до конфігурацій каналу;
- Зручний інтерфейс;
- Гнучкі налаштування;
- Можливість повного доступу до функції сервісу.



Схама 1. Принцип фунуціонування бота

Ця схема наочно показує принцип функціонування програми, що розроблювалася, та взаємозв’язок між об’єктами. Під зв’язком мається на увазі передача даних від одного блоку до іншого.

Формулювання вимог для програми важливе не тільки розробнику проєкту, а й безпосередньому керівнику розробки.

Керівник розробки розбиває проєкт на підзавдання та визначає терміни їх виконання тестує продукт на різних стадіях розробки. Формулювання вимог дозволять скласти список підзавдань, планувати розробку.

Схема 1 загалом відображає основну модель взаємодії та роботи різних елементів програми, але не є досить докладною та наочною для розробника. Для вираження конкретних підзадач на основі специфікації вимог буде потрібно більше докладно розписати кожну взаємодію блоків один з одним:

- Користувач-Telegram.
- Користувач запускає бота для налаштування каналу.
- Telegram – web-інтерфейс.

Для створення опитувань, налаштувань каналу, перегляду не анонімного статистики опитувань, користувач вибирає в роботі Telegram потрібну опцію, та його перенаправляє на web-інтерфейс для подальших дій.

- Web-інтерфейс – Telegram.

Налаштування, зроблені в web-інтерфейсі, відтворюються в каналі Telegram.

Схематичне відображення принципів роботи програми та більше докладний опис зв’язків дає чітке уявлення про обсяг роботи та вимоги до проєкту.

РОЗДІЛ 2 ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБКИ

Почнемо з Python. Python – це інтерпретована мова програмування високого рівня, відома своєю простотою, читабельністю та універсальністю. Філософія дизайну Python наголошує на зручності читання коду та чистому синтаксисі, що полегшує його розуміння та написання.

Синтаксис Python розроблено таким чином, щоб бути зрозумілим і легким для читання, з акцентом на зрозумілий для людини код. Він використовує відступи та пробіли для визначення блоків коду, а не покладається на явні дужки чи ключові слова. Це дозволяє розробникам писати код, який є більш читабельним і зручним для обслуговування.

Однією з ключових переваг Python є його універсальність. Це мова загального призначення, що означає, що її можна використовувати для широкого спектру програм. Python підтримує різні парадигми програмування, включаючи процедурний, об’єктно-орієнтований та функціональний стилі програмування. Ця гнучкість дозволяє розробникам вибрати найбільш прийнятний підхід для своїх конкретних потреб.

Python є інтерпретованою мовою, що означає, що код, написаний на Python, не потребує компіляції перед запуском. Замість цього інтерпретатор виконує код рядок за рядком. Ця функція дозволяє швидко розробляти та тестувати, що робить Python популярним вибором для сценаріїв, створення прототипів і створення програм.

Python також типізується динамічно, тобто типи змінних визначаються під час виконання. Розробникам не потрібно явно оголошувати типи змінних, що дозволяє створити більш гнучкий і стислий код. Однак ця динамічна типізація іноді може призвести до компромісу з продуктивністю.

На додаток до основних функцій мови, Python має величезну стандартну бібліотеку, яка надає широкий спектр модулів і інструментів для різних завдань, таких як робота з файлами, мережа, доступ до бази даних тощо. Ця обширна стандартна бібліотека допомагає розробникам заощаджувати час і зусилля, надаючи готові функції для стандартних завдань.

Крім того, Python має динамічну та активну спільноту, яка сприяє його зростанню та розвитку. Існує широкий спектр сторонніх бібліотек і фреймворків, які розширюють можливості Python і дозволяють розробникам ефективно створювати складні програми.

Популярність Python значно зросла з роками, що зробило його однією з найпоширеніших мов програмування сьогодні. Його зазвичай використовують у веб-розробці, аналізі даних, машинному навчанні, наукових обчисленнях, автоматизації тощо. Його простота, зрозумілість і обширна екосистема бібліотек і інструментів роблять Python чудовим вибором як для початківців, так і для досвідчених розробників.

2.1. Python

Python був створений Гвідо ван Россумом, голландським програмістом, наприкінці 1980-х років. Гвідо почав розробляти Python у грудні 1989 року, працюючи в Національному науково-дослідному інституті математики та комп'ютерних наук у Нідерландах. Його мета полягала в тому, щоб створити мову, яка була б простою, читабельною та легкою у використанні, з акцентом на продуктивність розробника.

Назва «Python» була надихнута любов'ю Гвідо до британського комедійного серіалу «Літаючий цирк Монті Пайтона». Це також відображає акцент мови на чистому та цікавому синтаксисі. Guido випустив першу версію Python, Python 0.9.0, у лютому 1991 року. Цей початковий випуск використовувався в основному внутрішньо в дослідницькому інституті та привернув увагу та популярність через онлайн-спільноту.

Філософія дизайну Python, яку часто називають «дзен Python», наголошує на читабельності, простоті та зрозумілості. Він сприяє зрозумілості коду та має на меті полегшити програмістам написання чистого коду, який зручно підтримувати. Дзен Python інкапсульовано в набір керівних принципів, доступ до яких можна отримати, ввівши «імпортувати це» в інтерпретаторі Python.

Протягом 1990-х років Python продовжував розвиватися. Python 1.0 був випущений у січні 1994 року, представивши кілька нових функцій, таких як

лямбда, карта, фільтр і функції зменшення. Мова набрала обертів і почала залучати зростаючу спільноту розробників.

На початку 2000-х було випущено серію Python 2.x, а Python 2.0 вийшов у жовтні 2000 року. Ця серія побачила широке впровадження та розширення екосистеми Python із розробкою численних бібліотек і фреймворків. Python 2.3, випущений у 2003 році, представив такі важливі функції, як генератори та уніфікований збирач сміття. Python 2.5, випущений у 2006 році, містив такі вдосконалення, як декоратори та умовні вирази.

Основний випуск Python 3.0 був представлений у грудні 2008 року. Він був розроблений для усунення кількох фундаментальних недоліків і невідповідностей у мові та її стандартній бібліотеці. Python 3.0 представив значні зміни, які не були зворотно сумісними з попередніми версіями Python, що спричинило деякий початковий опір прийняттю. Проте спільнота Python визнала важливість цих удосконалень, і з часом впровадження Python 3 неухильно зростало.

Незважаючи на труднощі, пов’язані з переходом з Python 2 на Python 3, Python Software Foundation (PSF) оголосила про офіційне завершення терміну служби Python 2 у січні 2020 року. Цей крок спонукав розробників переходити на Python 3 і проклав шлях для безперервного росту і розвитоку мови.

Починаючи з Python 3.0, спільнота Python випустила кілька наступних версій, включаючи Python 3.1, 3.2, 3.3 і так далі. Кожен випуск привносив у мову нові функції, вдосконалення та оптимізацію. Відомі версії включають Python 3.4, який представив модуль asyncio для асинхронного програмування, і Python 3.5, який представив підказки типу та подальшу підтримку асинхронного програмування.

Python 3.6, 3.7, 3.8 і 3.9 продовжували вдосконалювати мову за допомогою нових функцій і вдосконалень, таких як f-рядки, анотації змінних, оптимізація продуктивності та оператори об’єднання словників.

Очікується, що Python 3.10 буде випущено в 2021 році, і вона представить цікаві функції, такі як зіставлення шаблонів і додаткові покращення синтаксису, що ще більше розширить можливості мови.

Python пережив величезний ріст і популярність, ставши однією з найпоширеніших мов програмування в різних областях. Він має активну та підтримуючу спільноту, розгалужену бібліотечну екосистему (включаючи такі популярні бібліотеки, як NumPy, Pandas і TensorFlow), і широко поширений.

Як працює інтерпретатор Python?

Власне, будь-який перекладач починається з аналізу вихідного коду. Тут інтерпретатор Python отримує вихідний код і ініціалізує деякі інструкції для виконання наступних речей: Він дотримується правила відступів і перевіряє синтаксис Python. Можливо, є деякі неправильні рядки, тому програма зупинить виконання, щоб показати повідомлення про помилку.

Цей етап називається лексичним аналізом, що означає поділ файлів вихідного коду на список токенів. На наступному кроці інтерпретатор згенерує байт-коди. Давайте подивимося, як це працює.

Генерація байт-коду.

Як тільки аналізатор інтерпретатора Python отримує токени, він починає маніпулювати лексичними токенами. Він створює велику структуру під назвою AST (абстрактне синтаксичне дерево). Інтерпретатор перетворює цей AST у байт-код, який означає машинну мову. У Python байт-код можна зберегти у файлі з розширенням «.рус». У наступному розділі ви побачите, як інтерпретатор python виконує ці байтові коди.

Віртуальна машина Python (PVM).

Інтерпретатор Python ініціалізує свій механізм виконання під назвою PVM, який є віртуальною машиною Python. Інтерпретатор завантажує машинну мову з модулями бібліотеки та вводить її в PVM. Це перетворює байт-код у виконуваний код, наприклад, 0 і 1 (двійковий). А потім друкує результати.

Зауважте, що якщо під час процесу PVM станеться помилка, виконавець негайно припинить операцію, щоб відобразити помилку.Тепер ви дізнаєтесь, як інсталювати Python у вашій операційній системі. Якщо немає програмного забезпечення Python або використовується мобільний пристрій, то можна скористатися будь-яким онлайн-компілятором Python [2].

Особливості Python

1. Python легко вивчити.

Однією з найбільших перешкод для тих, хто зацікавлений у програмуванні, є те, що мови програмування насправді є їх власними мовами; вони мають власні правила, синтаксис, граматичні структури тощо, і вони часто вимагають вивчення абсолютно нового словника.

Але Python інший. Більше ніж будь-яка інша мова програмування, Python читає та пише дуже подібно до стандартної англійської мови. Він використовує спрощений синтаксис з акцентом на природній мові, що значно полегшує процес навчання для початківців. І, оскільки Python є безкоштовним для використання та підтримується надзвичайно великою екосистемою бібліотек і пакетів, це часто є мовою першого вибору для нових розробників. Ці та інші фактори допомагають продемонструвати, чому Python є найкращим вибором для тих, хто не має належного досвіду програмування.

2. Python має активну, підтримуючу спільноту.

Жоден програміст не є островом; вони залежать від необхідної документації та підтримки, тому, коли вони стикаються з неочікуваними проблемами або новими проблемами, які потрібно вирішити, вони мають куди звернутися, щоб знайти відповіді. Python існує вже понад три десятиліття, що є більш ніж достатнім часом, щоб навколо нього виросла спільнота відданих користувачів. Спільнота Python включає розробників усіх рівнів кваліфікації та забезпечує легкий доступ до документації, посібників, підручників тощо.

Водночас спільнота Python надзвичайно активна. Коли розробники стикаються з дедлайнами та відчайдушно потребують допомоги, вони можуть співпрацювати з спільнотою, щоб знайти швидкі та ефективні рішення.

3. Python гнучкий.

Python часто описують як мову програмування загального призначення. Це означає, що на відміну від доменно-орієнтованих мов, які розроблені лише для певних типів програм, Python можна використовувати для розробки практично будь-яких програм у будь-якій галузі чи сфері.

Для чого використовується Python? Python дуже ефективно використовувався у веб-розробці, аналітиці даних, машинному навчанні, науці

про дані, розробці даних і навіть у машинному навчанні та штучному інтелекті. Багато провідних компаній і програмних компаній залежать від Python, включаючи Facebook, Google, Netflix, Instagram та інші. Завдяки підтримці низки фреймворків і бібліотек, Python не може впоратися з кодуванням.

4. Python пропонує різноманітні рішення для веб-розробки.

Хоча Python є ефективним вибором для багатьох типів проєктів розробки, його корисність у веб-розробці варта особливого визнання. Використовуючи доступні бібліотеки з відкритим вихідним кодом, розробники Python можуть швидко та легко запустити свої веб-програми.

І хоча інші мови, такі як Java або .NET, можуть запропонувати підвищену продуктивність, швидкість і досвід розробника, які надає Python, роблять його очевидним вибором для тих, кому потрібне швидке рішення, на яке вони можуть покластися. У той же час різноманітність доступних ресурсів Python пропонує унікальну можливість інтегрувати інші типи програм у веб-сайти.

5. Python добре підходить для науки про дані та аналітики.

Багато факторів, які роблять python привабливим вибором для початківців, також виділяють його як надійний варіант для обробки даних та аналізу даних. Простота використання, підтримка та гнучкість Python зробили його важливим інструментом для тих, хто працює з машинним навчанням, хмарними обчисленнями та великими даними.

Python особливо ефективний для аналізу та організації наборів даних. Фактично, для проєктів з науки про дані та аналітики Python поступається за популярністю лише мові R. Його готові можливості аналізу даних у поєднанні зі зростаючою екосистемою фреймворків, орієнтованих на дані, допомагають гарантувати, що Python залишається популярним рішенням для програмування даних.

6. Python ефективний, швидкий і надійний.

Інколи розробник, який спеціалізується на іншій мові програмування, може запитати: «Чому Python повільний?» І так, порівняно з деякими іншими мовами, такими як Java, C#, Go, JavaScript або C++, Python часто має дещо нижчу швидкість виконання. Однак у сучасному світі час розробки набагато

важливіший за час роботи комп'ютера. А з точки зору часу виходу на ринок, Python просто неможливо перемогти.

Подібним чином Python є ефективним і надійним, що дозволяє розробникам створювати потужні програми з мінімумом зусиль. Завершити проєкти програмування легко, а не забрати багато часу, і результати можуть порівнюватися з програмами, створеними з використанням більш вимогливих мов.

telegram.ext – це бібліотека або пакет Python, який розширює функціональні можливості бібліотеки python-telegram-bot, яка є популярним фреймворком для розробки ботів Telegram. Він надає інтерфейс вищого рівня та додаткові функції для спрощення процесу створення ботів Telegram за допомогою API Telegram Bot.

Telegram Bot API дозволяє розробникам взаємодіяти з Telegram і створювати ботів, які можуть надсилати й отримувати повідомлення, відповідати на команди, обробляти вбудовані запити тощо. Однак робота безпосередньо з API Telegram Bot може бути складною та потребувати написання великої кількості шаблонного коду.

telegram.ext вирішує цю проблему, надаючи набір абстракцій і попередньо визначених обробників, які спрощують розробку ботів Telegram. Він побудований на основі бібліотеки python-telegram-bot і розширює її можливості.

Одним із ключових компонентів telegram.ext є клас Updater. Програма оновлення діє як міст між вашим ботом і Telegram Bot API. Програма оновлення обробляє зв'язок із серверами Telegram і отримує оновлення, наприклад, нові повідомлення або запити зворотного виклику, від Telegram.

Ключові особливості класу Updater:

Підтримка Webhook і довгого опитування: клас Updater підтримує як методи webhook, так і довгого опитування для отримання оновлень від Telegram. Веб-хуки дозволяють Telegram надсилати оновлення вашому боту через HTTP-запити, тоді як тривале опитування передбачає постійне опитування серверів Telegram щодо нових оновлень.

Автоматичне отримання оновлень: клас Updater автоматично отримує оновлення з Telegram і передає їх диспетчеру для обробки. Він забезпечує простий і зручний спосіб отримання та обробки вхідних оновлень без необхідності ручного опитування.

Механізм обробки помилок і повторних спроб: клас Updater містить вбудовані механізми обробки помилок і повторних спроб. Він обробляє поширені помилки, які можуть виникнути під час зв'язку з серверами Telegram, і повторює невдалі запити, щоб забезпечити надійну доставку повідомлень.

Обробка потоків і подій: клас Updater може працювати в окремому потоці, дозволяючи вашому боту обробляти оновлення одночасно. Він також надає обробники подій, які можна використовувати для виконання коду, коли відбуваються певні події, наприклад, коли бот запускається або зупиняється.

Клас Dispatcher у telegram.ext є ключовим компонентом, який обробляє маршрутизацію вхідних оновлень до відповідних обробників. Він визначає, який обробник має обробити конкретне оновлення на основі вмісту та типу оновлення. Клас Dispatcher надає інтерфейс для реєстрації обробників для різних типів оновлень і подій.

Ключові особливості класу Dispatcher включають:

Маршрутизація оновлень: основним обов'язком класу Dispatcher є маршрутизація вхідних оновлень до відповідних обробників. Він аналізує вміст і тип кожного оновлення та зіставляє їх із зареєстрованими обробниками, щоб визначити, який із них має обробляти оновлення.

Програмування, кероване подіями: клас Dispatcher дає змогу програмувати, кероване подіями, у контексті обробки оновлень Telegram. Ви можете зареєструвати обробники для різних подій, таких як нові повідомлення, вбудовані запити, запити зворотного виклику, публікації на каналі тощо. Коли відбувається подія, запускається відповідна функція обробки, що дозволяє вам відповідати або вживати відповідних дій.

Попередньо визначені обробники: telegram.ext надає ряд попередньо визначених обробників, які можна використовувати для обробки різних типів оновлень. Наприклад, існують обробники для обробки команд, повідомлень,

вбудованих запитів, запитів зворотного виклику, відредагованих повідомлень тощо. Ці попередньо визначені обробники спрощують процес обробки певних типів оновлень, надаючи готовий до використання інтерфейс.

Фільтри: клас Dispatcher працює разом із фільтрами, щоб визначити, які обробники слід активувати для певних оновлень. Фільтри дозволяють вказати умови, яким має відповідати оновлення для виклику обробника. telegram.ext надає різноманітні вбудовані фільтри на основі тексту повідомлення, ідентифікатора користувача, типу чату тощо. Фільтри дозволяють вибірково обробляти оновлення на основі певних критеріїв.

Проміжне програмне забезпечення: клас Dispatcher підтримує використання функцій проміжного програмного забезпечення. Функції проміжного програмного забезпечення — це функції, які можуть перехоплювати та змінювати оновлення до того, як вони будуть передані обробникам. Ця функція дозволяє виконувати завдання попередньої обробки оновлень, наприклад журналювання, автентифікацію або маніпулювання даними.

Обробка помилок: клас Dispatcher містить механізми обробки помилок для обробки винятків, які можуть виникнути під час обробки оновлень. Він надає спосіб визначення обробників помилок, які виконуються, коли виникає виняток у функції обробника. Обробники помилок дозволяють витончено обробляти помилки та надавати відповідні відповіді або резервні дії.

Клас Dispatcher у telegram.ext підвищує гнучкість і модульність вашого бота Telegram, надаючи централізований механізм для маршрутизації оновлень і запуску відповідних обробників. Це спрощує процес обробки різних типів оновлень і дозволяє кероване подіями програмування для створення інтерактивних та чуйних ботів.

Використовуючи можливості класу Dispatcher, ви можете створити ботів Telegram, які відповідають на повідомлення користувачів, обробляють команди, обробляють вбудовані запити та взаємодіють з користувачами різноманітними способами, і все це в рамках чіткої та організованої структури.

telegram.ext надає ряд попередньо визначених обробників, які спрощують процес обробки різних типів оновлень. Наприклад, існують обробники для

обробки команд, повідомлень, вбудованих запитів, запитів зворотного виклику тощо. Ці обробники дозволяють визначати функції або методи, які будуть викликані, коли відбудеться певна подія.

Пакет також містить клас Filters, який дозволяє вказати умови, яким має відповідати оновлення, щоб запустити обробник. Ви можете використовувати вбудовані фільтри для фільтрації оновлень на основі тексту повідомлення, ідентифікатора користувача, типу чату тощо. Це дає змогу вибірково обробляти оновлення на основі певних критеріїв.

Ще одна потужна функція telegram.ext – клас ConversationHandler. ConversationHandler призначений для спрощення розробки розмовних ботів, які передбачають багатоетапну взаємодію з користувачами. Він керує потоком розмов між ботом і користувачем, визначаючи серію станів і відповідні обробники для кожного стану.

Ключові особливості класу ConversationHandler включають:

Керування станом: ConversationHandler керує станом розмови, відстежуючи поточний стан і переходячи між станами на основі введення користувача. Це дозволяє визначити серію станів і пов’язати обробники з кожним станом.

Покрокова взаємодія: за допомогою ConversationHandler ви можете визначити послідовність кроків або запитань, які бот задає користувачеві. Кожен крок може мати пов’язану функцію обробки, яка обробляє відповідь користувача та визначає наступний крок у розмові.

Постійність: ConversationHandler підтримує постійність, яка дозволяє зберігати та відновлювати стан розмови, навіть якщо бот перезавантажується або виявляється помилка. Це гарантує безперебійне відновлення розмови.

Розширена обробка помилок: ConversationHandler надає механізми обробки помилок для обробки несподіваних введених користувачем даних і ефективного відновлення після помилок. Це дозволяє визначати обробники помилок, які виконуються, коли виникає помилка, дозволяючи надавати відповідні відповіді або підказки користувачеві.

Час очікування розмови: ConversationHandler дозволяє встановити час очікування розмови. Якщо користувач не відповідає протягом зазначеного часу, розмова може бути автоматично припинено або переведено в інший стан.

ConversationHandler спрощує процес створення інтерактивних і динамічних ботів, які можуть брати участь у складних розмовах з користувачами. Це абстрагує складність керування потоком розмов і станом, полегшуючи створення складних чат-ботів.

За допомогою telegram.ext можемо створювати ботів Telegram ефективніше та з меншим кодом. Він абстрагує багато низькорівневих деталей роботи з API Telegram Bot, дозволяючи вам зосередитися на реалізації логіки та функціональності вашого бота.

Щоб використовувати telegram.ext, мені потрібно встановити бібліотеку python-telegram-bot, яка включає пакет telegram.ext. Можемо знайти більше інформації, прикладів і документації в офіційному репозиторії python-telegram-bot GitHub.

2.2. IDE

Щоб зрозуміти «Що таке PyCharm?» і «Для чого використовується PyCharm?», спочатку дамо відповідь на запитання «Що таке IDE?»

IDE складається з редактора та компілятора, які використовуємо для написання та компіляції програм. Він має комбінацію функцій, необхідних для розробки програмного забезпечення.

Наявність IDE значно полегшує процес розробки та програмування. Він інтерпретує те, що ми вводимо, і пропонує відповідне ключове слово для вставки. Можемо розрізнити клас і метод, оскільки IDE надає їм різні кольори. IDE також надає різні кольори для правильних і неправильних ключових слів. Якщо пишемо неправильне ключове слово, він намагається передбачити ключове слово, яке ми маємо намір написати, і автоматично завершує його.

Чому ми повинні використовувати IDE?

Нижче наведено основні причини використання IDE для розробки:

- IDE складається з вікна текстового редактора, де можемо писати наші програми.
- Він складається з вікна редактора проєкту, де зберігаємо всі необхідні файли проєкту програмного забезпечення.
- Ми можемо надавати різні вхідні дані та перевіряти ефективність нашої програми, перевіряючи вихідні дані, які отримуємо у вікні виводу.
- У разі виникнення будь-якої помилки IDE показує попередження та пропозиції у вікні виводу, щоб могли її вирішити.
- IDE містить стелаж із модулями та пакетами в одному місці, що допомагає додавати функції в наші програми.

Це допомагає підвищити ефективність створення програмного забезпечення. Тепер ви хочете знати, яку IDE використовуємо для програмування на Python і розробки додатків? Найпопулярнішим і широко використовуваним середовищем розробки програм Python є PyCharm.

2.3. PyCharm

PyCharm — це гібридна платформа, розроблена JetBrains як IDE для Python. Він зазвичай використовується для розробки додатків Python. Деякі з організацій єдинорога, такі як Twitter, Facebook, Amazon і Pinterest, використовують PyCharm як своє середовище розробки Python! Він підтримує дві версії: v2.x і v3.x.

Ми можемо запускати PyCharm на Windows, Linux або Mac OS. Крім того, він містить модулі та пакети, які допомагають програмістам розробляти програмне забезпечення за допомогою Python за менший час і з мінімальними зусиллями. Крім того, його також можна налаштувати відповідно до вимог розробників. Особливості PyCharm: чому ми повинні використовувати його для нашого наступного проєкту Python? Нижче зібрано деякі основні функції, які надає PyCharm.

1. Інтелектуальний редактор коду:

Це допомагає нам писати високоякісні коди! Він складається з кольорових схем для ключових слів, класів і функцій. Це допомагає підвищити читабельність і розуміння коду. Це допомагає легко виявляти помилки.

Він забезпечує функцію автозаповнення та інструкції для завершення коду.

2. Навігація по коду

Це допомагає розробникам редагувати та вдосконалювати код із меншими зусиллями та часом. За допомогою навігації по коду розробник може легко перейти до функції, класу або файлу. Програміст може швидко знайти елемент, символ або змінну у вихідному коді. Крім того, використовуючи режим лінзи, розробник може ретельно перевірити та налагодити весь вихідний код.

3. Рефакторинг

Перевагою є внесення ефективних і швидких змін як до локальних, так і до глобальних змінних. Рефакторинг у PyCharm дозволяє розробникам покращувати внутрішню структуру без зміни зовнішньої продуктивності коду. Це також допомагає розділяти більш розширені класи та функції за допомогою методу екстракту.

4. Допомога для багатьох інших веб-технологій:

Він допомагає розробникам створювати веб-програми на Python. Він підтримує такі популярні веб-технології, як HTML, CSS і JavaScript. Розробники мають можливість редагувати в реальному часі за допомогою цієї IDE. Водночас вони можуть попередньо переглянути створену/оновлену веб-сторінку. Розробники можуть стежити за змінами безпосередньо у веб-браузері. PyCharm також підтримує AnglularJS і NodeJS для розробки веб-додатків.

5. Підтримка популярних веб-фреймворків Python

PyCharm підтримує такі веб-фреймворки, як Django. Він забезпечує функцію автозаповнення та пропозиції щодо параметрів Django. Це допомагає в налагодженні кодів Django. Він також допомагає web2py та Pyramid, іншим популярним веб-фреймворкам.

6. Допомога для наукових бібліотек Python

PyCharm підтримує наукові бібліотеки Python, такі як Matplotlib, NumPy і Anaconda. Ці наукові бібліотеки допомагають створювати проєкти Data Science і Machine Learning. Він складається з інтерактивних графіків, які допомагають розробникам зрозуміти дані. Він здатний інтегруватися з різними інструментами,

такими як IPython, Django та Pytest. Ця інтеграція допомагає створювати інноваційні унікальні рішення.

РОЗДІЛ 3 РОЗРОБКА TELEGRAM БОТА

Спочатку за допомогою блок схеми буде наведено як працює бот.



Схама 2. Послідовність дій при роботі з ботом

3.1. Створення бота через BotFather та завантаження PyCharm

BotFather.. Спочатку необхідно зареєструвати саму оболонку бота самому месенджері, отримати присвоєний ей ID, який є також токеном. В Телеграмм допомога при такому реєстрації надає віртуальний помічник @BotFather.

1. Ввод команди /start для запиту інструкції.

2. Відправка установки /newbot.

3. @BotFather пропонує придумати ім’я для створюваного бота, яке повинно закінчуватися словом «бот».

4. Далі помічник присилає токен нового віртуального користувача і url для додавання його в список контактів.

Токен бота в телеграмі – це програмний засіб ідентифікації, який представлений у вигляді зашифрованої послідовності символів. За допомогою токенів сервер ідентифікує віртуального помічника та визначає його рівень привілеїв. Він генерується системою авторизації автоматично та має прив’язку до конкретного об’єкта (бота).

Таким чином, токен Телеграм-бота дозволяє встановити справжність віртуального помічника, а також пов’язати його із серверами месенджера та сервісом програмного управління Bot API.

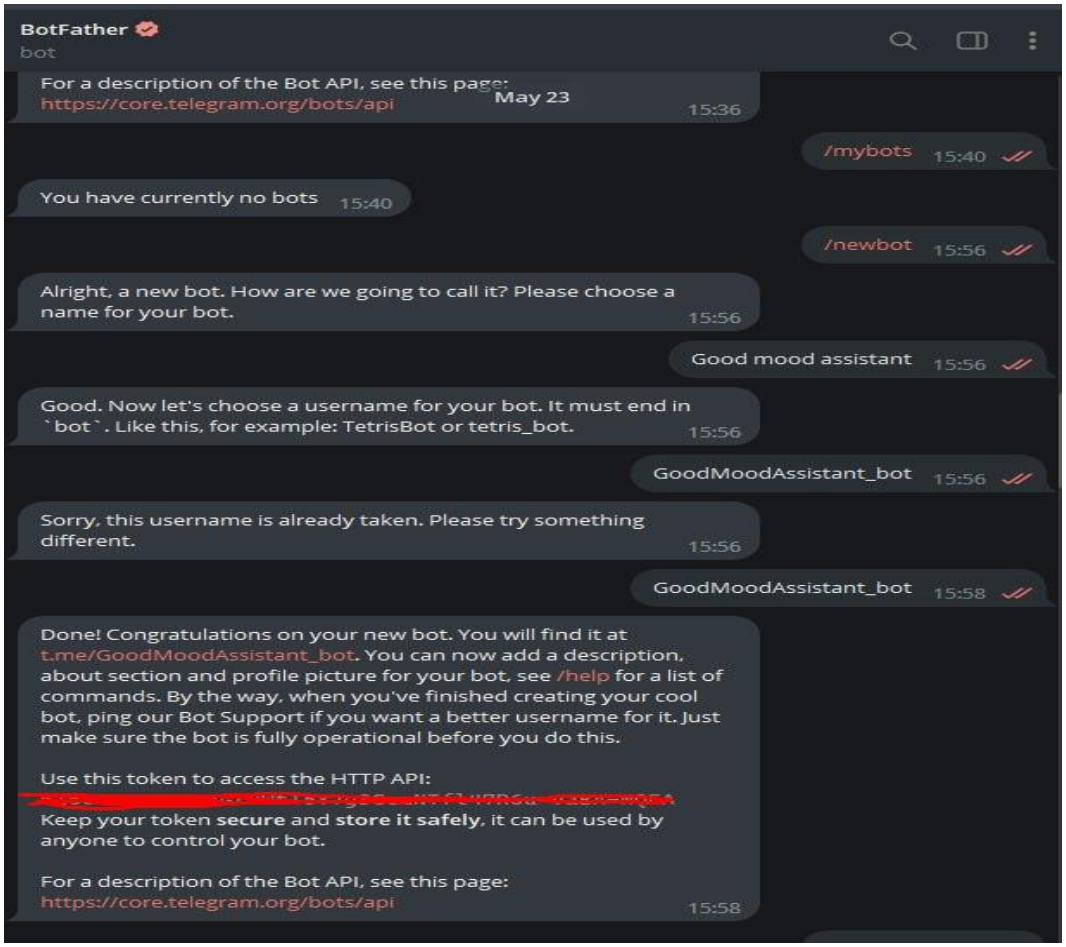


Рис. 2. Реєстрація бота

В BotFather можемо зробити набагато більше ніж просто створити бота, а саме доступні такі команди:

/setuserpic – встановлює аватарку бота

/setcommands – встановлює стандартні команди бота, які з’являтимуться в інлайн режимі при введенні "/" в чаті

/deletebot – видаляє бота

/token – генерує новий токен

/revoke – відкликає старий токен

/Setinline – включає інлайн-режим. Він дозволяє використовувати функції бота прямо з рядка введення

/setinlinegeo — якщо увімкнути цю функцію, боти можуть запитувати дані про місцезнаходження користувачів в режимі онлайн.

/setjoingroups – налаштування можливості додавання вашого бота в канали та чати

/setprivacy — обмеження доступу бота до конфіденційної інформації та листування в чатах та каналах

/setname – Встановлює ім’я бота. Воно буде відображатися у вікні діалогу

/setdescription – встановлює опис бота

/setabouttext – встановлює вітальне повідомлення бота

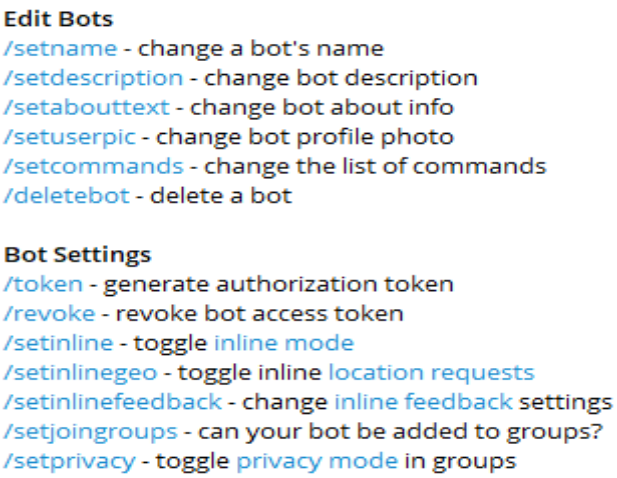


Рис. 3. Частина з доступних команд

Наступним кроком нам треба завантажити PyCharm. Отже, переходимо на сторінку завантаження PyCharm. Для завантаження доступно дві версії: професійна та версія для спільноти. Версія для спільноти безкоштовна.

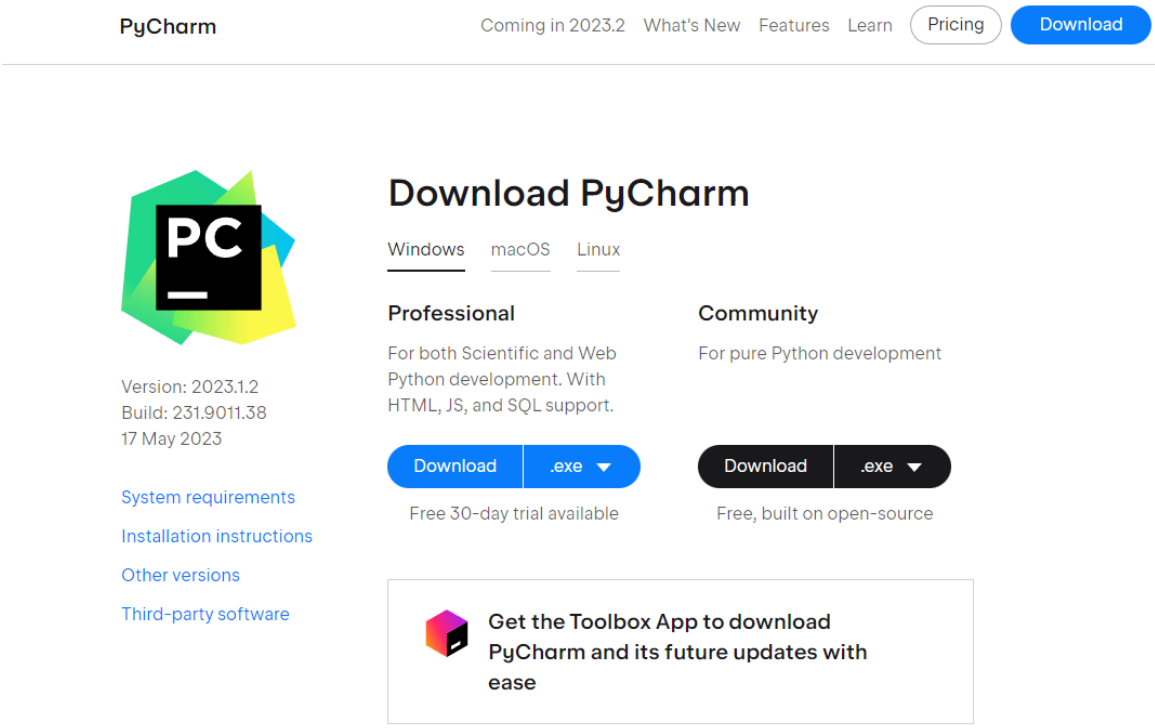


Рис. 4. Завантаження PyCharm

Запускаємо завантажений .exe файл. У першому вікні нас вітає сам установник. Сміливо клацаємо «Next». Далі треба зазначити місце встановлення середовища.

На наступному вікні виділяємо галочку "Download and install JRE x86 by JetBrains", якщо на комп’ютері не встановлена Java. Також виділяємо галочку ".py" для того, щоб усі файли з програмним кодом за промовчанням відкривалися в PyCharm. Це зручно.

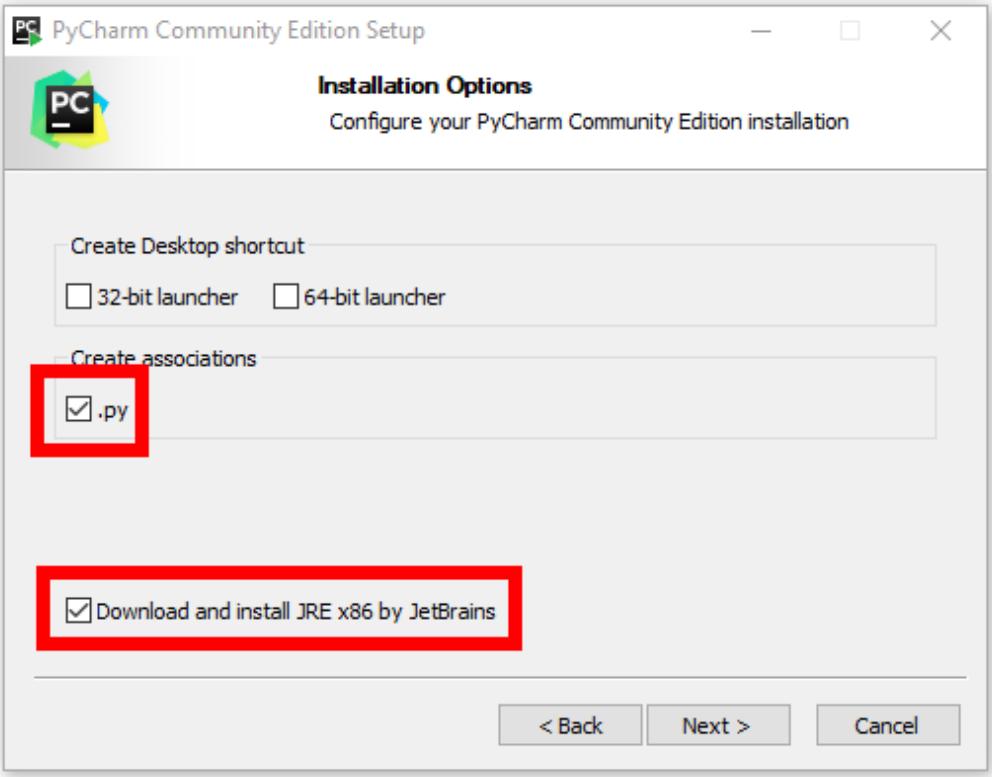


Рис. 5. Завантаження PyCharm на комп’ютер

Після закінчення установки запускаємо PyCharm.

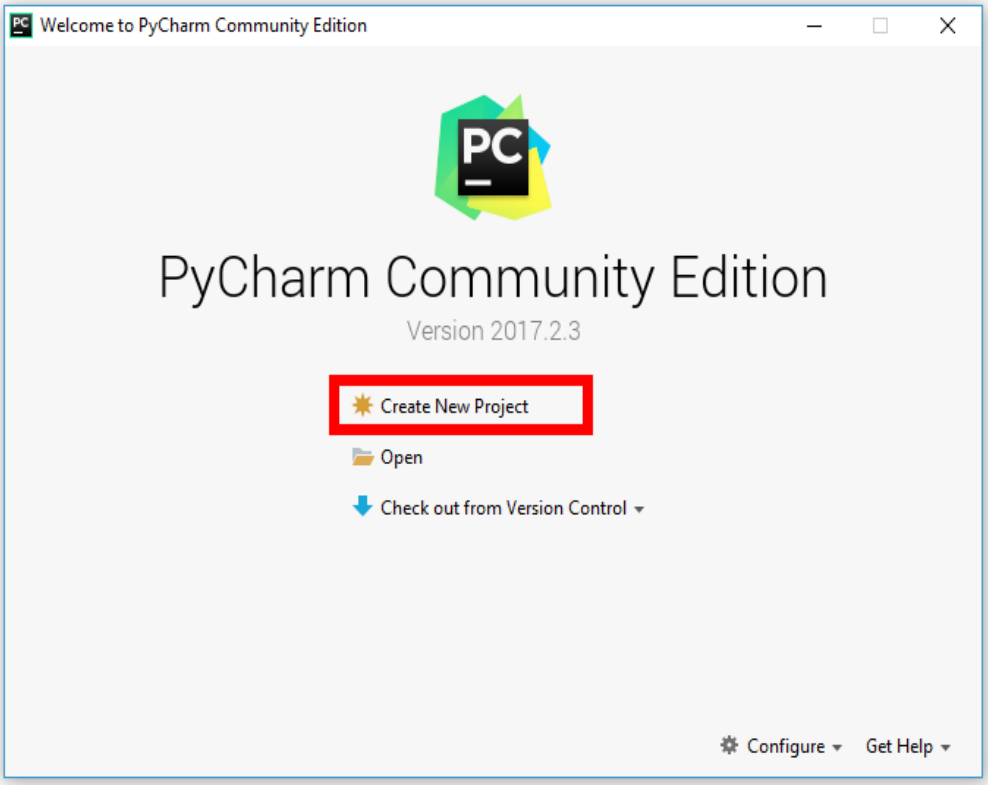


Рис. 6. Створення нового пректу

Створюємо новий проєкт.

3.2. Написання бота у PyCharm

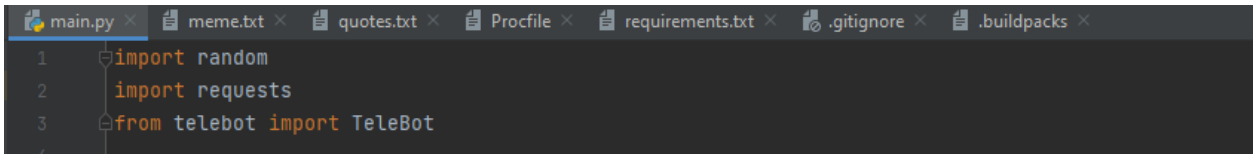


Рис. 7. Бібліотеки

Першим ділом після створення проєкту відвантажуємо ці модулі та бібліотеки. Telebot (pyTelegramBotAPI) це гарна і легка бібліотека для створення бота на python для телеграм. Далі нам потрібен токен який нам видав BotFather і створюємо змінну під назвою token, в ній зберігатимемо наш токен.

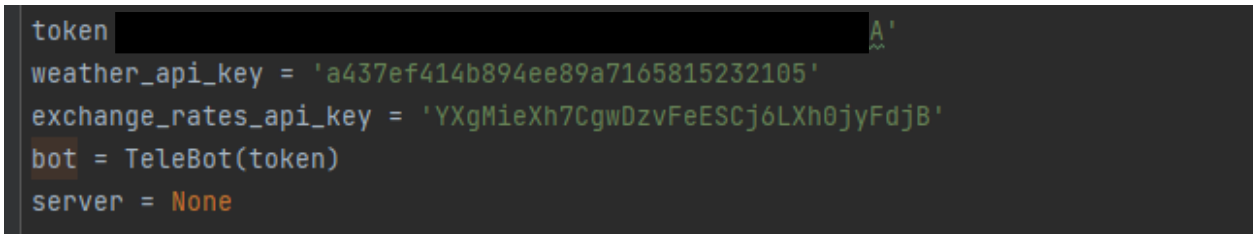


Рис. 8. Токени и апі ключі для погоди і курса валют

Тепер прописуємо декоратор @bot.message_handler для відслідковування команди /start, зазвичай саме з цієї команди запускається весь бот. Далі передаємо ще один параметр, а саме commands, він в якості значення приймає деякий список і в цьому списку можемо вказати які команди будемо обробляти. Далі створюємо функцію start і ця функція приймає лише один параметр. Зараз це буде параметр message. Це той параметр котрий буде зберігати всю інформацію про користувача котрий працює з ботом и плюсом зберігати всю інформацію про сам по собі чат. Далі прописуємо той текс і список команд які будуть функціоналом мого бота. Коли юзер буде вписувати команду /start то бот буде йому відправляти такий текст:

Привіт! Я бот, який може надіслати вам рандомну цитату, анекдот або інформацію про погоду.

Доступні команди:

/quote – надіслати рандомну цитату.

/joke – надіслати рандомний анекдот.

/weather <місто>, <країна> – отримати погоду у вказаному місті.

/currency <опція> <базова_валюта> <цільова_валюта>, щоб отримати курс обміну валют.

/manual – надіслати інструкцію використання бота.

Іструкція використання бота:

1. Використайте команду /quote, щоб отримати рандомну цитату.

2. Використайте команду /joke, щоб отримати рандомний анекдот.

3. Використайте команду /weather <city>, <country>, щоб отримати погоду у вказаному місті.

4. Використайте команду /currency <опція> <базова_валюта> <цільова_валюта>, щоб отримати курс обміну валют.

```python
@bot.message_handler(commands=['start'])
def start(message):
    response = "Привіт! Я бот, який може надіслати вам рандомну цитату, анекдот або інформацію про погоду.\n\n" \
               "Доступні команди:\n" \
               "/quote - надіслати рандомну цитату\n" \
               "/joke - надіслати рандомний анекдот\n" \
               "/weather <місто>, <країна> - отримати погоду у вказаному місті\n" \
               "/currency <опція> <базова_валюта> <цільова_валюта>, щоб отримати курс обміну валют.\n\n" \
               "/manual - надіслати інструкцію використання бота"
    print("processed command start from: " + str(message.chat.id))
    bot.send_message(message.chat.id, response)
```

Рис.9. Команда start

Далі потрібно зробити так, щоб ці команди працювали, а не були просто текстом. І почнемо з команди /manual котра відповідає за відправку такого тексту:

Іструкція використання бота:

Використайте команду /quote, щоб отримати рандомну цитату.

Використайте команду /joke, щоб отримати рандомний анекдот.

Використайте команду /weather <city>, <country>, щоб отримати погоду у вказаному місті.

Використайте команду /currency <опція> <базова_валюта> <цільова_валюта>, щоб отримати курс обміну валют.

Наприклад:

/currency USD EUR

Тут можемо переглянути список доступних команд і інструкцію по їх використанню. Для створення команди /manual прописуємо декоратор @bot.message_handler для відслідковування команди і далі все так само, як і при своренні команди /start.

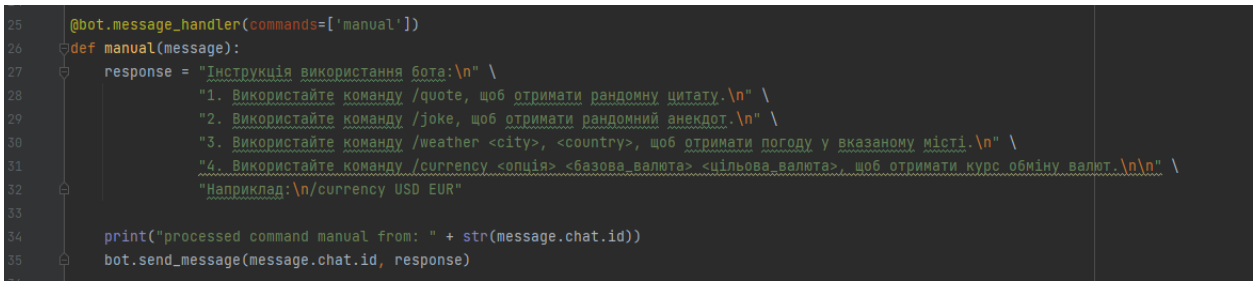


Рис. 10. Команда manual

Далі перейдемо до команди /quote, вона відповідає за отримання рандомоної цитати на тему програмування. Для цього нам треба створити текстовий документ і завантажити в нього всі цікаві цитати. Потім цей текстовий документ додаємо до проєкту, щоб дот міг звертатися до нього кожен раз коли користувач вписуватиме команду /quote.



Рис. 11. Команда quote

Наступною буде команда /joke, початок точно той самий як і при минулих командах. Далі була можливість завантажити текстовий документ із жартами як це було і з цитатами, або просто вписати в сам код всі ці жарти і при зверненні до

бота по цій команді він би обирав одну випадкову і видавав її.

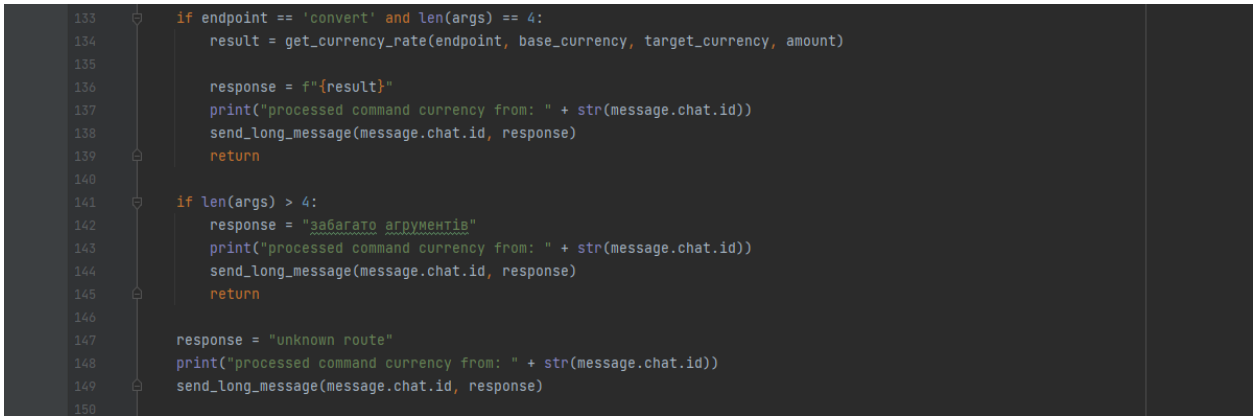


Рис. 12. Команда joke

Користуючись інтернет ресурсом і документацією в ньому, ресурс буде наведений в списку літератури, також підключаємо команду / currency, вона відповідатиме за курс валют через бота.



Рис. 13. Команда currency



```
    if endpoint == 'convert' and len(args) == 4:
        result = get_currency_rate(endpoint, base_currency, target_currency, amount)

        response = f"{result}"
        print("processed command currency from: " + str(message.chat.id))
        send_long_message(message.chat.id, response)
        return

    if len(args) > 4:
        response = "забагато аргументів"
        print("processed command currency from: " + str(message.chat.id))
        send_long_message(message.chat.id, response)
        return

    response = "unknown route"
    print("processed command currency from: " + str(message.chat.id))
    send_long_message(message.chat.id, response)
```

Рис. 14. Команда currency

І останньою командою буде /weather. Ця команда буде показувати погоду в місті вписаному самим користувачем. Прописую декоратор @bot.message_handler для відслідковування команди, потім створюю функцію def get_weathe і приймаємо один обов’язковий параметр “message”.

Отримувати інформацію про погоду будемо з окремого стороннього сервіса, а саме з weatherapi, не цьому сайті можемо отримати апі ключ, а за рахунок цього ключа зможемо вже отримувати інформацію про погоду. Отриманий апі ключ просто копіюємо і на початку коду створюємо нову змінну котра буде називатись weather_api_key і підставляємо туди сам ключ.

Для того, щоб користувач після того як впише місто і країну погода в котрій його цікавить, отримував інформацію поро погоду, боту потрібно буде відправляти данні про місто і країну по відповідній url адресі. Також по стандарту в документації до апі буде йти те, що бот буде видавати поготу в фаренгейтах, у коді це змінено на цельсії.

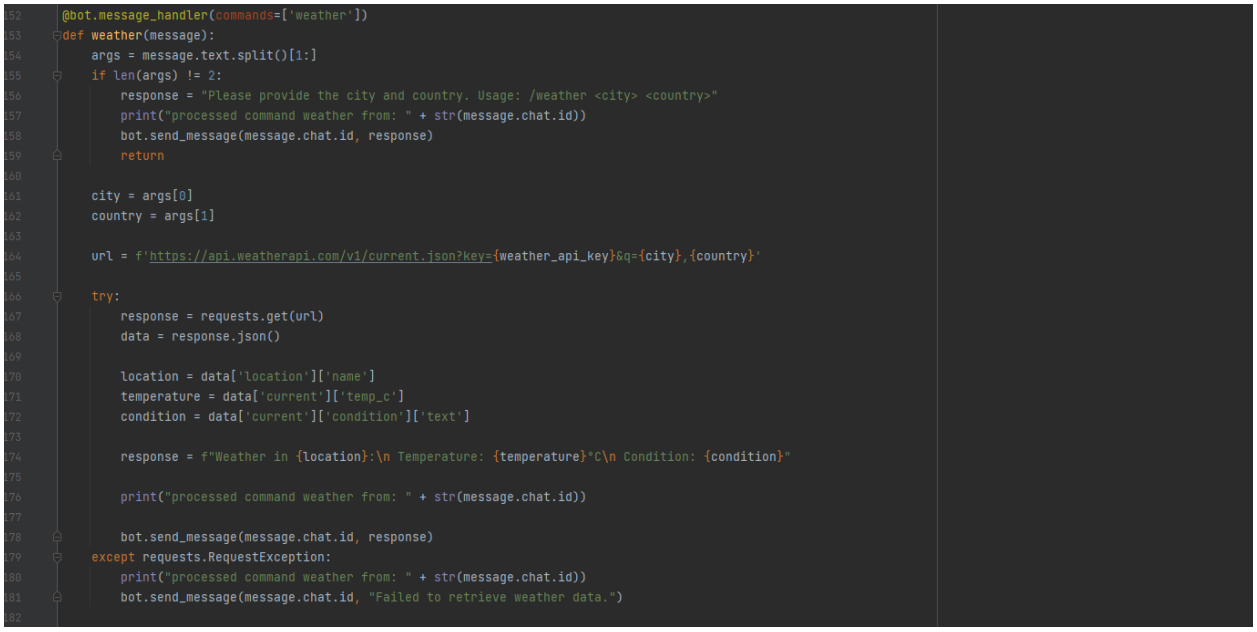


Рис. 15. Команда weather

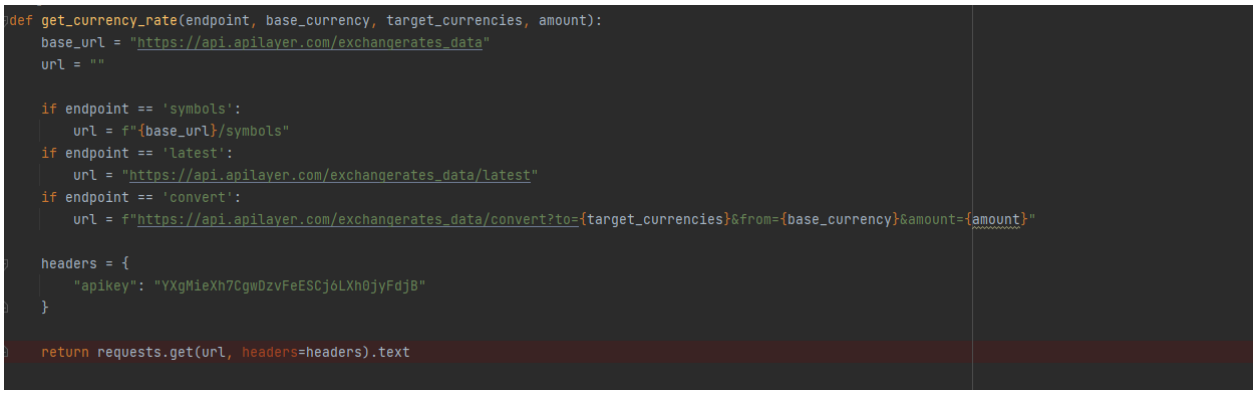


Рис. 16. Метод отримати курс валюти

Це метод, який звертається до апі для конвертації валют та перегляду курсу валют.

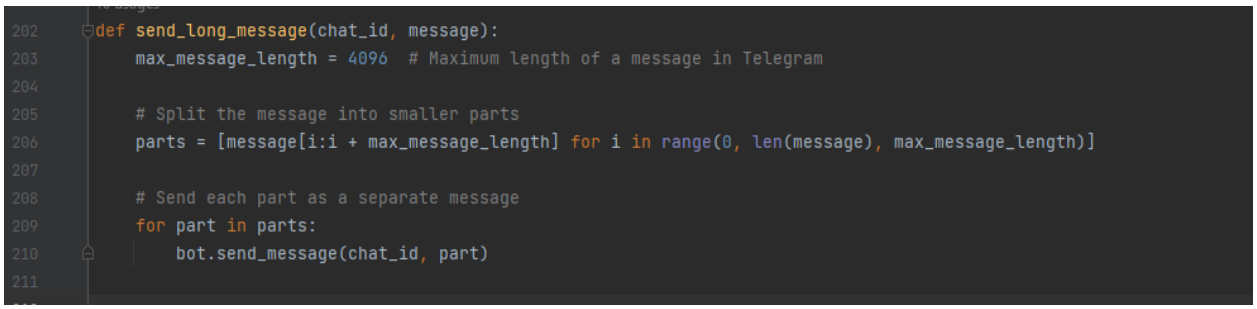


Рис. 17. Максимальний розмір

Метод встановлює максимальна довжину повідомлення в Telegram.

```
1 usage
def run_bot():
    print("Bot is running!")
    bot.polling()


if __name__ == "__main__":
    run_bot()

```

Рис. 18. Заключна частина коду

На цьому етапі завершене написання коду, далі буде продемонстровано те як бот реагує на команди.

3.3. Результат роботи бота в Telegram

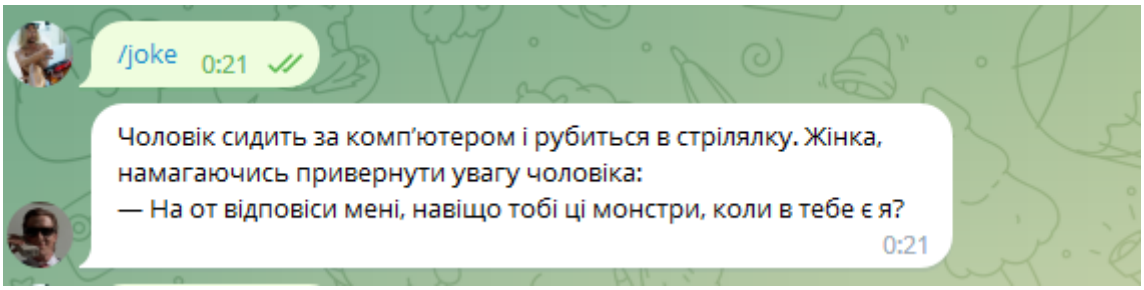


Рис. 19. реагування на команду /joke

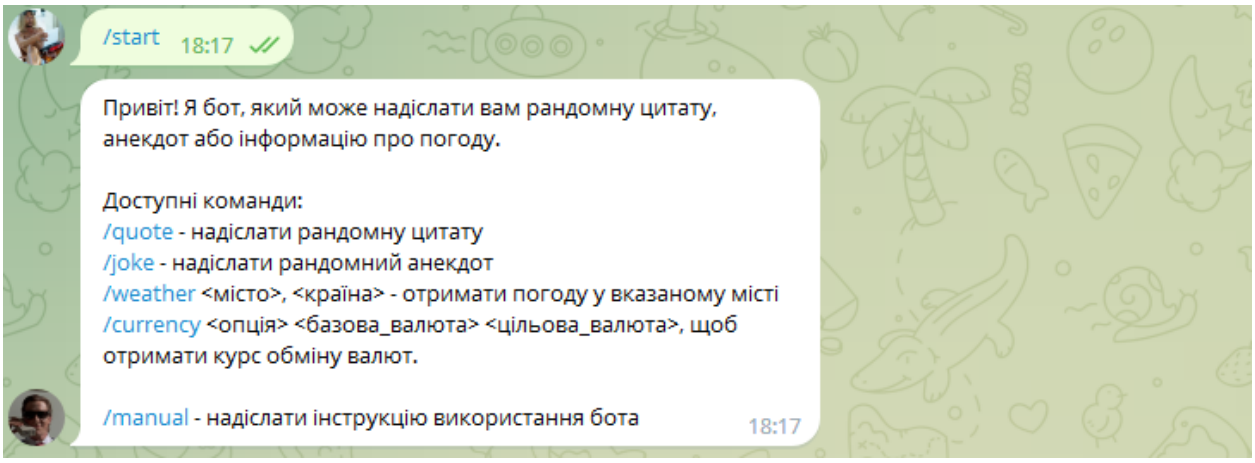


Рис. 20. Реагування на команду /start

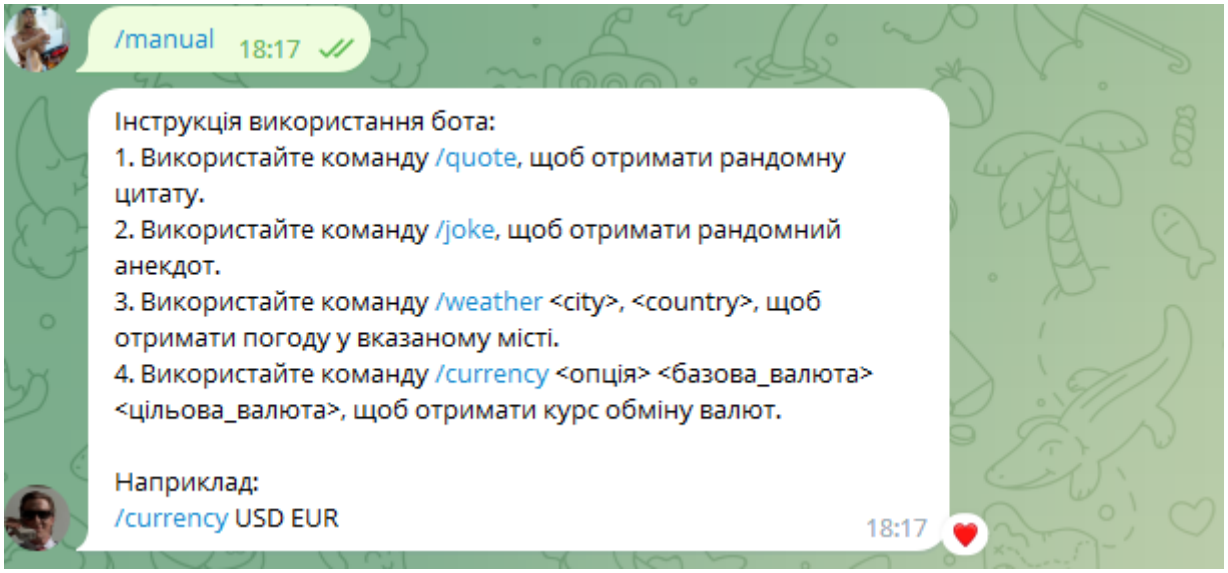


Рис. 21. Реагування на команду /manual

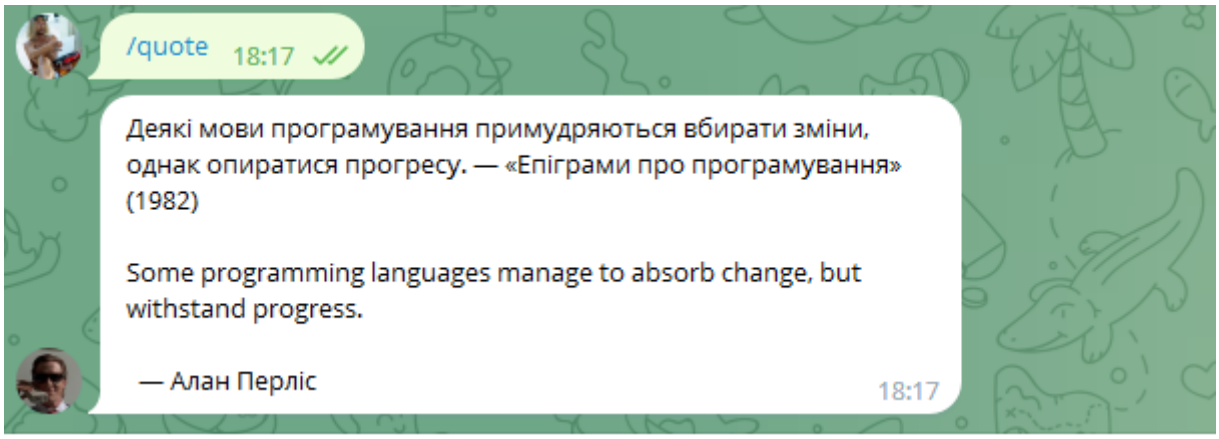


Рис. 22. Реагування на команду /quote

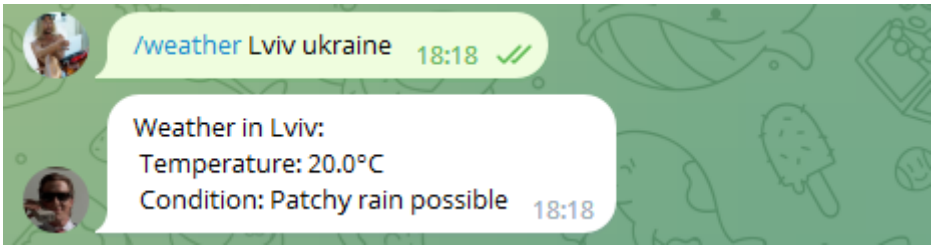


Рис. 23. Реагування на команду /weather



Рис. 24. Реагування на команду /currency

На цьому було завершено створення Телеграм боту, він працює.

ВИСНОВКИ

У ході дипломної роботи була проведена розробка чат-боту для платформи Telegram. Ця робота спрямована на створення інтерактивного і автоматизованого інструменту комунікації з користувачами за допомогою месенджера Telegram.

У процесі дослідження були визначені ключові вимоги до чат-боту, а також проведений аналіз існуючих рішень та технологій для розробки чат-ботів. Було вибрано відповідні технології та інструменти для реалізації проекту.

На основі отриманого аналізу та розробки було реалізовано чат-бот для Telegram, який здатний взаємодіяти з користувачами, надавати їм інформацію та виконувати деякі завдання.

В ході тестування було перевірено працездатність та функціональність розробленого чат-боту. Були виявлені деякі недоліки та помилки, які були виправлені. Загалом, результати тестування свідчать про задовільну роботу чат-боту та його здатність виконувати поставлені завдання.

Отриманий чат-бот може бути використаний в різних галузях, таких як підтримка клієнтів, автоматизація деяких процесів або надання інформації користувачам.

На основі проведеної роботи можна зробити висновок, що розробка чат-боту для Telegram є актуальним і перспективним напрямом в сфері інформаційних технологій. Розроблений чат-бот має потенціал для подальшого розширення функціональності та вдосконалення його роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. JSON and XML Weather API and Geolocation Developer API. – URL: https://www.weatherapi.com/

2. What is Python? How the Interpreter Works and How to Write "Hello World" in Python. – URL: https://www.freecodecamp.org/news/what-is-python-beginners-guide/

3. requests 2.31.0. – URL: https://pypi.org/project/requests/

4. Telegram APIs. – URL: https://core.telegram.org/api

5. CurrencyConverter 0.17.7. – URL: https://pypi.org/project/CurrencyConverter/

6. Telegram Bot на Python. – URL: https://qsy.by/telegram-bot-na-python-chast-2-pycharm-python3-pishem-kod/

7. Что такое токен бота в Телеграмме: как получить, узнать и обновить. – URL: https://telega-blog.ru/chto-takoe-token-bota-v-telegramme-kak-poluchit-uznat-i-obnovit/

8. How Many People Use Telegram in 2023? 55 Telegram Stats – URL: https://backlinko.com/telegram-users

9. Чат-бот. Что это такое и как он работает? – URL: https://ikraine.net/chat-bot-chto-eto-takoe-i-kak-rabotaet/

10. pyTelegramBotAPI 4.12.0. – URL: https://pypi.org/project/pyTelegramBotAPI/

11. Build data-driven apps with fully managed data services. – URL: https://www.heroku.com/

12. Телеграм бот на Python. – URL: https://www.youtube.com/watch?v=v8Nqa78v2tE&list=PL0lO_mIqDDFUev1gp9yEwmwcy8SicqKbt&index=8

13. Як створити телеграм-бота за допомогою Python. – URL: https://www.freecodecamp.org/ukrainian/news/yak-stvoryty-telehram-bota-za-dopomohoyu-python/

14. Створюємо Telegram бота на Python. Частина 1. – URL: https://codeguida.com/post/410

15. Як створити телеграм-бота на Python. – URL: https://spacelab.ua/articles/yak-stvoriti-telegram-bota-na-python/

16. How to Create a Bot for Telegram – Short and Simple Guide for Beginners. – URL: https://flowxo.com/how-to-create-a-bot-for-telegram-short-and-simple-guide-for-beginners/

17. "Теле бот" Розробка телеграм-бота для OPC-технології програмування вбудованої системи управління на базі ПЛ. – URL: https://nure.ua/wp-content/uploads/2020/Konkurs/tele-bot.pdf

18. Как сделать чат-бот для Telegram-канала - инструкция для администраторов. – URL: https://netpeak.net/ru/blog/kak-sozdat-chat-bot-dlya-telegram-instruktsiya-dlya-administratorov-kanalov/

19. Топ-10 корисних Telegram-ботів для українських користувачів. – URL: https://www.epravda.com.ua/publications/2017/08/7/627822/

20. Building Your First Telegram Bot: A Step by Step Guide. – URL: https://www.toptal.com/python/telegram-bot-tutorial-python

21. A step-by-step guide to building your first Telegram bot. – URL: https://thenextweb.com/news/a-step-by-step-guide-to-building-your-first-telegram-bot