

РОЗДІЛ 5 ІНФОРМАТИКА ТА КІБЕРНЕТИКА

УДК 37.0:004.3

Антонюк О. В., ст. магістратури факультету кібернетики; науковий керівник – д.ф.-м.н., професор Джунь Й. В. (Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука, м. Рівне)

РОЗРОБКА TELEGRAM-БОТА ЗА ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ PYTHON І СЕРИДОВИЩА РОЗРОБКИ PYCHARM

***Анотація.** В статті розглянуто сучасний метод програмування Telegram-бота на мові Python і здійснено його розробку в спеціальному середовищі PyCharm. Охарактеризовано сучасний і спрощений метод створення ботів, розкрито новизну цього методу, зокрема показано, що в ньому спрощення досягається за рахунок обсягів написаного коду. Виявлено головну проблему, яка викликає складнощі створення і адаптації ботів та наведено підходи до вирішення цієї проблеми. Запропоновано метод швидкої розробки ботів на мові програмування Python.*

***Ключові слова:** Telegram-боти, середовище PyCharm, мова програмування Python, швидка розробка ботів.*

***Abstract.** The article describes the modern method of programming a Telegram bot in Python and developed it in a special PyCharm environment. The modern and simplified method of creation of bots is described, the novelty of this method is revealed, in particular it is shown that in it simplification is achieved due to the volumes of written code. The main problem that causes the complexity of creating and adapting bots is identified and approaches are presented to solve this problem. The method of rapid development of bots in Python programming language is suggested.*

***Keywords:** Telegram-bots, PyCharm environment, Python programming language, rapid development of bots.*

Telegram – це швидкий, зручний і безкоштовний месенджер від Павла Дурова, екс-власника «Вконтакте». Він стрімко розвивається з 2013 року. За перший рік його існування кількість користувачів становила 60 000 000 чоловік. Це в 2 рази перевищило результати його основного конкурента WhatsApp за той же самий період часу. Судячи з того, що вже зараз Telegram за своїм функціоналом і швидкості передачі даних обганяє WhatsApp, Viber, Facebook messenger та ін., очевидно, що він є лідером в своєму сегменті [1; 2].

Занепокоєність рівнем розробки Telegram-ботів є нині актуальною, але вони є одним із нелегких напрямів у програмуванні, оскільки, ціни на розробку таких ботів сягають від 100 до 3000 \$. Тому не дивно що складність і відсутність належної літератури для вивчення ботів і їх розробки, відштовхує молодих фахівців.

Telegram-боти за допомогою мов програмування досліджувала низка науковців та програмістів, зокрема Д. Бригс [3], Е. Ховченко [1], Ч. Фаулер [4] та ін., проте, ця проблема потребує подальших досліджень.

Метою нашої статті є дослідження існуючих методів розробки ботів та запропонування методу, який спростить їх розробку. Завданням нашого дослідження є демонстрація того, що створення ботів насправді є захоплюючим, легким і цікавим.

Існує три основні підготовчі етапи [5], які необхідно здійснити перед створенням ботів, а саме:

1. Встановити мову програмування Python. Зробити це дуже просто, так як вона є у вільному доступі і безкоштовна. При цьому необхідно поставити «Галочку» під час інсталяції програми, оскільки без цього не можна коректно написати код (рис. 1);



Рис. 1. Фініш першого підготовчого етапу

2. Встановити бібліотеку для розробки ботів, а саме PyTelegramBotAPI. Для того, щоб це зробити потрібно перейти в cmd і в консолі написати `pip install pyTelegramBotAPI` та натиснути Enter, після чого встановиться потрібна бібліотека;

3. Встановити PyCharm Community Edition, яке є безкоштовним середовищем розробки від JetBrains і також є у вільному доступі.

Розглянемо порядок створення бота. Для цього потрібно створити його в telegram, отже для здійснення цього необхідно перейти в BotFather (рис. 2).



Рис. 2. Вид вікна BotFather

Дальше потрібно створити бота за допомогою команди `/newbot` та ввести його ім'я і юзернейм (рис. 3).

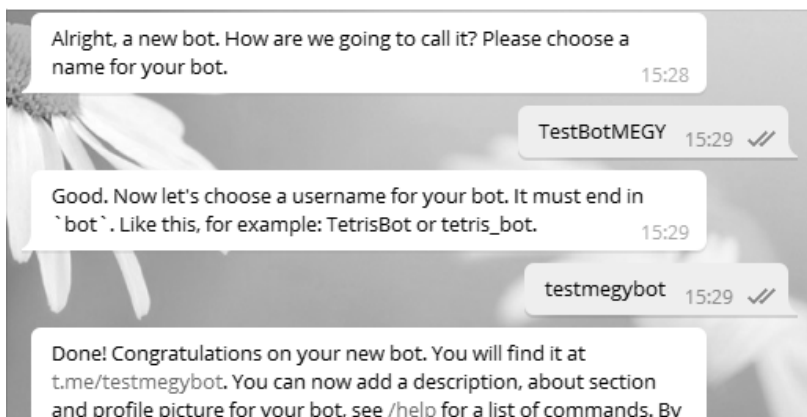


Рис. 3. Фрагмент створеного бота

Таким чином, буде отриманий бот і наступний token, для того щоб ним керувати (рис. 4).


Use this token to access the HTTP API:
893005653:  lca1KitZqtq08ABbWMyiuoRI

Рис. 4. Токен telegram бота

Обов'язково потрібно запам'ятати цей токен і нікому його не передавати. Частина токена автор повинен тримати в секреті, для того, щоб ніхто сторонній не зміг їм користуватись у власних цілях.

Наступний крок найважливіший – написання боту. Створимо бота, який буде вітатись, і видавати певне інформаційне меню з правом вибору.

Для цього, перейдемо у PyCharm і створимо проєкт (рис. 5)

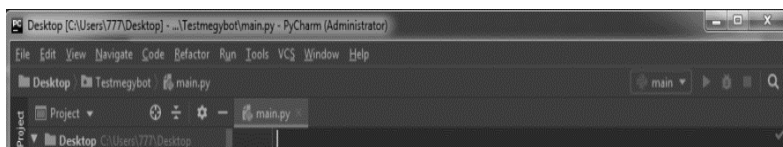
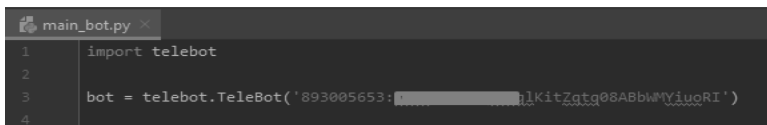


Рис. 5. Вікно проєкту

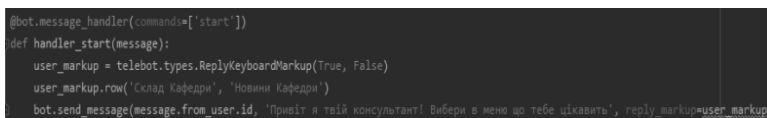
Дальше необхідно написати код, але перед початком його написання потрібно підключити бібліотеку для створення бота (рис. 6)



```
main_bot.py
1 import telebot
2
3 bot = telebot.TeleBot('893005653:AAHj3Lz0uJc0kK1Zqtg08ABbWYiu0RI')
4
```

Рис. 6. Вікно бібліотеки для створення бота

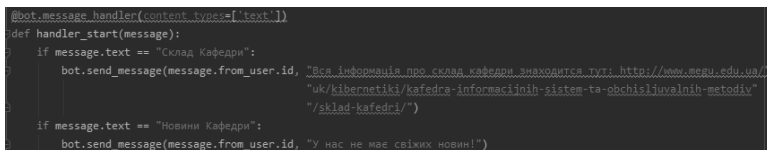
Наступним кроком є створення оброблювача команд, або як кажуть хендлери – для обробки команди start, тобто для запуску бота і хендлера для кастовної клавіатури (рис. 7).



```
@bot.message_handler(commands=['start'])
def handler_start(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    user_markup.row('Склад кафедри', 'Новини кафедри')
    bot.send_message(message.from_user.id, 'Привіт я твій консультант! Вибери в мене що тебе цікавить', reply_markup=user_markup)
```

Рис. 7. Вікно: @bot.message_handler

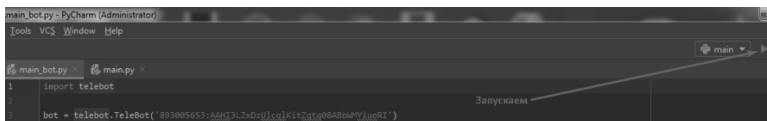
Крім цього необхідно передбачити варіанти для кожного user_markup.row, тобто, задати, що буде коли користувач вибере певний пункт меню з двох доступних (Рис. 8).



```
@bot.message_handler(content_types=['text'])
def handler_start(message):
    if message.text == "Склад Кафедри":
        bot.send_message(message.from_user.id, "Вся інформація про склад кафедри знаходиться тут: http://www.megu.edu.ua/uk/libernetiki/kafedra-informacijih-sisten-ta-obchisluvajnih-metodiv/" /sklad-kafedri/")
    if message.text == "Новини Кафедри":
        bot.send_message(message.from_user.id, "У нас не має свіжих новин!")
```

Рис. 8. Вікно варіантів для кожного user_markup row

Для того, щоб весь код успішно працював необхідно в кінці обов'язково прописати bot.polling(none_stop=True), щоб не дивлячись на різні помилки або збої Telegram bot працював, тобто запустити наш файл у середовищі PyCharm (рис. 9).



```
main_bot.py - PyCharm (Administrator)
Tools VCS Window Help
main_bot.py main_bot.py
1 import telebot
bot = telebot.TeleBot('893005653:AAHj3Lz0uJc0kK1Zqtg08ABbWYiu0RI')
Запускаємо
```

Рис. 9. Вікно main botу PyCharm (Administrator)

Продемонструємо тепер вікна отриманих результатів (рис. 10)

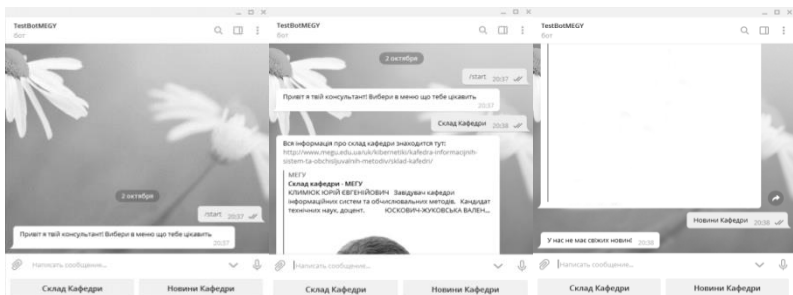


Рис. 10. Вікна отриманих результатів

Код програми має такий вигляд:

```
import telebot
bot = telebot.TeleBot('token ')
@bot.message_handler(commands=['start'])
def handler_start(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    user_markup.row('Склад Кафедри', 'Новини Кафедри')
    bot.send_message(message.from_user.id, 'Привіт я твій консультант!
Вибери в меню що тебе цікавить', reply_markup=user_markup)
@bot.message_handler(content_types=['text'])
def handler_start(message):
    if message.text == «Склад Кафедри»:
        bot.send_message(message.from_user.id, «Вся інформація про склад
кафедри знаходиться тут: http://www.megu.edu.ua/uk/kibernetiki/kafedra-informacijnih-sistem-ta-obchisljuvalnih-metodiv»
        «/sklad-kafedri/»)
    if message.text == «Новини Кафедри»:
        bot.send_message(message.from_user.id, «У нас не має свіжих новин!»)
bot.polling(none_stop=True)
```

Щоб використати цей код і краще зрозуміти запропонований метод, а також найголовніші поняття при написанні ботів на мові програмування Python, нами був складений невеликий голосарій.

Зокрема:

1. *import telebot* без цього поняття не можна запрограмувати сценарій, і використовувати бібліотеку telebot. Тому імпортуємо цю бібліотеку;

2. *bot = telebot.TeleBot('token')*, це потрібно для того щоб керувати ботом в Telegram, і був задіяний обмін даними з сервером телеграмм, у дужки необхідно вписати токен, який видає BotFather;

3. *@bot.message_handler(commands=['start'])*, це так званий хендлер, який задає команду старт під час запуску бота;

4. `def handler_start(message): user_markup = telebot.types.ReplyKeyboardMarkup(True, False)user_markup.row ('Склад Кафедри', 'Новини Кафедри')` тут задано типому, звичайну клавіатуру, де вказано оптимальний розмір True, False, а також два головних меню – 'Склад Кафедри' і 'Новини Кафедри';

5. `bot.send_message(message.from_user.id, 'Привіт я твій консультант! Вибери в меню що тебе цікавить', reply_markup=user_markup)`, це одна з цікавих функцій, яка передбачає, що коли ви зайшли в бот як новий користувач вибрати команду **start**, нам бот видасть повідомлення 'Привіт я твій консультант! Вибери в меню що тебе цікавить';

6. `@bot.message_handler(content_types=['text'])`, цим задається хендлер який відповідає за тип «текст», і через нього видає певну інформацію користувачу;

7. `def handler_start(message): if message.text == «Склад Кафедри»: bot.send_message(message.from_user.id, «Вся інформація про склад кафедри» знаходиться тут: http://www.megu.edu.ua/uk/kibernetiki/kafedra-informacijnih-sistem-ta-obchisljuvalnih-metodiv/sklad-kafedri/) if message.text == "Новини Кафедри": bot.send_message(message.from_user.id, «У нас не має свіжих новин!»)`, це наповнення клавіатури даними, тобто є хендлер який дає два варіанта в меню «Склад Кафедри» і «Новини Кафедри» якщо вибрати певний з них, за кожним закріплена відповідь, яка буде відправлятися ботом, користувачу, відповідно до його вибору;

8. `bot.polling(none_stop=True)`, останній параметр, що задає боту команду на працю без збоїв, і не закінчувати його роботу із-за якихось помилок на сервері, що може бути дуже часто.

Узагальнюючи результати проведеного дослідження, можна зробити висновок, що запропонований код, який має обсяг всього в 20 рядків, може виконувати близько 3 функцій і суттєво пришвидшує програмування. Запропонована нами бібліотека `pyTelegramBotAPI` істотно полегшує роботу програміста, виключає написання величезних конструкцій типу `if-elif-else`. Тому бібліотека створена для вирішення цієї важливої проблеми. Оскільки, на превеликий жаль, про цю бібліотеку мало кому відомо, то ознайомлення з нею читачів цієї статті безсумнівно буде сприяти необхідному прогресу в програмуванні.

1. Ховченко Е. Азбука Телеграм для підприємця. Как «поселить» свой бизнес в месенджер № 1 в мире. Лаборатория Бизнес Кейсов. С. 1–2. 2. Офіційна документація на `pyTelegramBotAPI`. URL: <https://core.telegram.org/bots/api> (дата звернення: 29.09.2019). 3. Бригс Д. Python для детей «Манн, Иванов и Фербер». 2017. 67 с. 4. Фаулер Ч. Чистый код. Питер, 2015. 158 с. 5. Основы создания ботов. URL: <https://groosha.gitbook.io/telegram-bot-lessons/> (дата звернення: 30.09.2019).