

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення  
Дисципліна: Скриптові мови програмування (Python)

**Лабораторна робота №8**  
**Тема: «СТВОРЕННЯ ЧАТ-БОТУ ДЛЯ TELEGRAM**  
**ВИКОРИСТОВУЮЧИ PYTHON»**

Виконав: ст. гр. КН-24  
Куріщенко П. В.  
Перевірив: асистент  
Ткаченко О.С.

Кропивницький 2025

*Мета роботи* - навчитися створювати найпростіші реалізації чат-ботів на Python.

## ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

### Завдання:

1. Реалізувати приклад вказаний у даній лабораторній роботі.
2. Додати боту можливість реагувати на інші команди або отримані повідомлення (не забувайте додавати свої команди в менеджері ботів `@BotFather`).
3. Додати до відповідей бота використання емої або стікерів.

### Принцип роботи бота "Телефончик"

Бот працює на основі бібліотеки `telebot`, яка дозволяє легко реагувати на повідомлення та команди користувача. Усі команди обробляються за допомогою декораторів `@bot.message_handler`.



### Список команд і реакцій:

**/start** - Надсилає привітальне повідомлення та підказку використати `/help`.

**/help** - Показує список доступних команд з коротким описом кожної.

**/time** - Показує поточний системний час у форматі год:хв:сек.

**/echo [текст]** - Повторює текст, який користувач додав після команди.

**/sticker** - Надсилає випадковий стікер з попередньо визначеного списку.

**/github** - Надсилає посилання на GitHub-акаунт автора бота.

**/clear** - Намагається видалити всі попередні повідомлення в чаті (працює лише з правами адміністратора).

**/stop** - Зупиняє роботу бота (лише в межах скрипту — не вимикає Telegram-сесію).

### Реакція на звичайний текст

Бот також обробляє деякі звичайні повідомлення без команд:

- "Привіт" / "Доброго дня" / "Доброго ранку" — бот вітається.
- "Дякую" — відповідає вдячністю.
- "Як справи?" — відповідає, що все добре.
- "Котра година?" — показує поточний час.
- "Стікер" — надсилає випадковий стікер.
- "Оцінка" — надсилає оцінку 5/5.
- "Погода" — відповідає мотиваційно, що погода гарна.
- Інше — якщо фраза не розпізнана, бот радить ввести /help.

### Лістинг бота:

```
import telebot
from safe import token

bot = telebot.TeleBot(token)

@bot.message_handler(commands=['start']) #/start
def start_message(message):
    bot.send_message(message.chat.id,
"""
📞 Привіт! Я *Телефончик* – твій бот-помічник.
Введи /help, щоб дізнатись, що я вмію.
🎧 Я слухаю тебе уважно...""", parse_mode='Markdown')

@bot.message_handler(commands=['help']) #/help
def help_message(message):
    bot.send_message(message.chat.id,
"""
/time – Показати поточний час
/echo _текст_ – Повторити те, що ти написав
/sticker – Відправити стікер
/github – Показати посилання на GitHub
/clear – Очистити чат
/stop – Зупинити бота""", parse_mode='Markdown')
```

```

@bot.message_handler(commands=['time'])#/time
def time_message(message):
    import datetime as dt
    now = dt.datetime.now()
    current_time = now.strftime("%H:%M:%S")
    bot.send_message(message.chat.id, f"⌚ Поточний час: {current_time}")

@bot.message_handler(commands=['echo'])#/echo text
def echo_message(message):
    text = message.text.split(' ', 1)
    if len(text) > 1: bot.send_message(message.chat.id, f"👉 {text[1]}")
    else: bot.send_message(message.chat.id, "👉 ")

@bot.message_handler(commands=['sticker'])#/sticker
def sticker_message(message):
    import random as r
    stickers = [
        'CAACAgIAAxkBAAE0g8BoKd3zPnkLFGj5f-tIpQ7avEXFeAACDicAAh5v0EmbtU7CYB0N-zYE',
        'CAACAgIAAxkBAAE0g8RoKd4lgJQ7IF5yo715JpdaoIzdAACjxIAAiXVsEi_4Q-PyoPsCjYE',
        'CAACAgIAAxkBAAE0g8toKd5bXOeEqWiWrMTeBEDnWm-5wAACSBQAAmoz4U1QCUG0FKMZiTYE',
        'CAACAgIAAxkBAAE0g81oKd6Y6zFIjj6mMfbhaRIWAAEWBlkAAioAA9cyTx5rOenqNdJhsDYE',
        'CAACAgIAAxkBAAE0g89oKd7ZuseK0dYcsDCOW5gKb2Y1vwACfEYAAAsw0yUrTDUa1JsYxFjYE',
        'CAACAgIAAxkBAAE0g9NoKd-QsJoGDHhUg5hdAAFq2vY-ndUAAoYVAALIfNBLwdWIqWyiYM42BA',
        'CAACAgIAAxkBAAE0g9VoKeBIKsQMa0QMNI9agn0HreFugAAC8iAAAmUl2EonKSfn8aLdXDYE',
        'CAACAgIAAxkBAAE0g9loKeC8IfpIJ0WvL-HQtzIodiaEYgACpBcAAmSD0Evr0stI8mPpVDYE'
    ]
    sticker_id = r.choice(stickers)
    bot.send_sticker(message.chat.id, sticker_id)

@bot.message_handler(commands=['github'])#/github
def github_message(message): bot.send_message(message.chat.id, "🔗 *Мій GitHub:* [movavok](https://github.com/movavok)", parse_mode='Markdown')

@bot.message_handler(commands=['stop'])#/stop
def stop_message(message):
    bot.send_message(message.chat.id, "👋 Бувай! Якщо захочеш повернутись – ввімкни мене знову.")
    bot.stop_polling()

```

```

@bot.message_handler(commands=['clear'])#/clear
def clear_message(message):
    try:
        for i in range(message.message_id, 0, -1):
            try: bot.delete_message(message.chat.id, i)
            except: continue
        bot.send_message(message.chat.id, "⚡ Чат очищено! Тепер тут чисто.")
    except: bot.send_message(message.chat.id, "❌ На жаль, я не можу видалити всі повідомлення. Можливо, у мене немає прав адміністратора.")

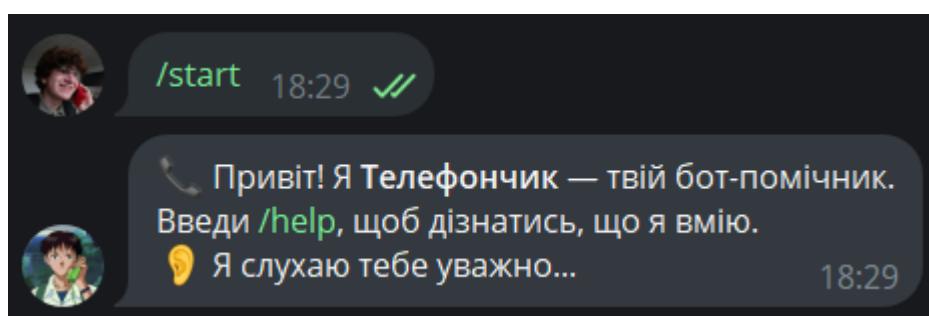
def unknown_message(message): bot.send_message(message.chat.id, "❓ Я не знаю, що з цим робити. Введи /help, щоб дізнатись, що я вмію.")

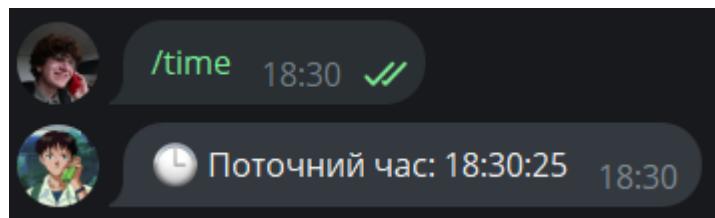
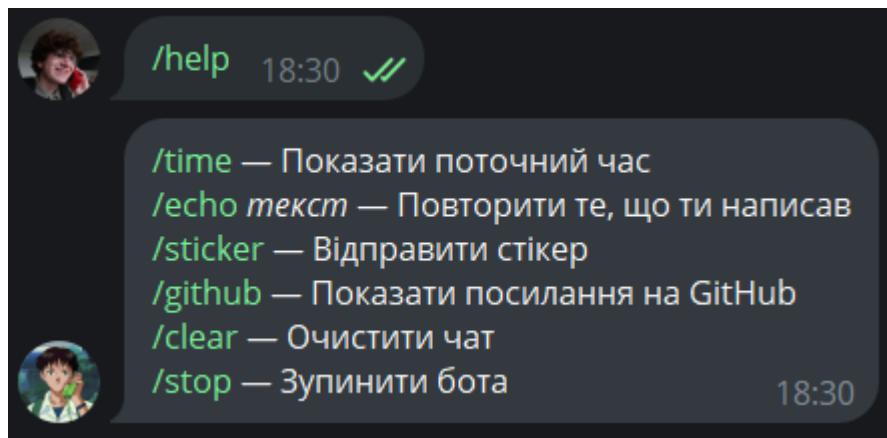
@bot.message_handler(content_types=['text'])#розпізнавання тексту
def handle_specific_text(message):
    text = message.text.lower()
    if text in ["привіт", "доброго ранку", "доброго дня", "доброго вечора",
               "привіт!"]:
        bot.send_message(message.chat.id, "👋 Вітаю! /help, щоб дізнатись, що я вмію.")
    elif text in ["дякую", "дякую!"]:
        bot.send_message(message.chat.id, "👉 Завжди радий допомогти!")
    elif text == "як справи?":
        bot.send_message(message.chat.id, "⌚ У мене все добре, дякую!")
    elif text == "котра година?":
        time_message(message)
    elif text == "стікер":
        sticker_message(message)
    elif text == "оцінка":
        bot.send_message(message.chat.id, "⭐ 5/5")
    elif text == "погода":
        bot.send_message(message.chat.id, "☁️ Погода завжди гарна, коли ти поруч!")
    else:unknown_message(message)

bot.polling(none_stop=True)

```

## Результати роботи:







/github 18:34 ✓

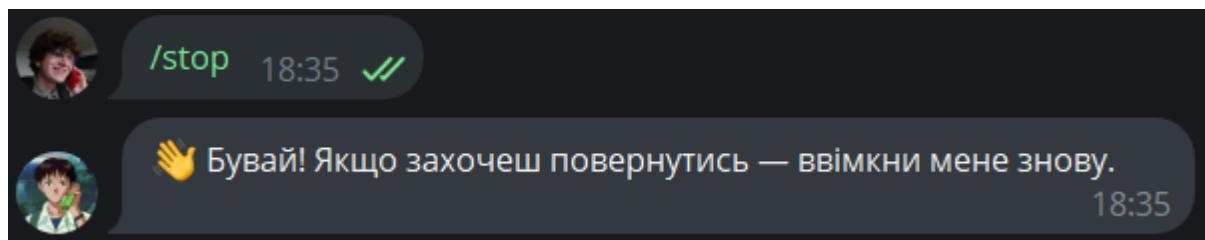
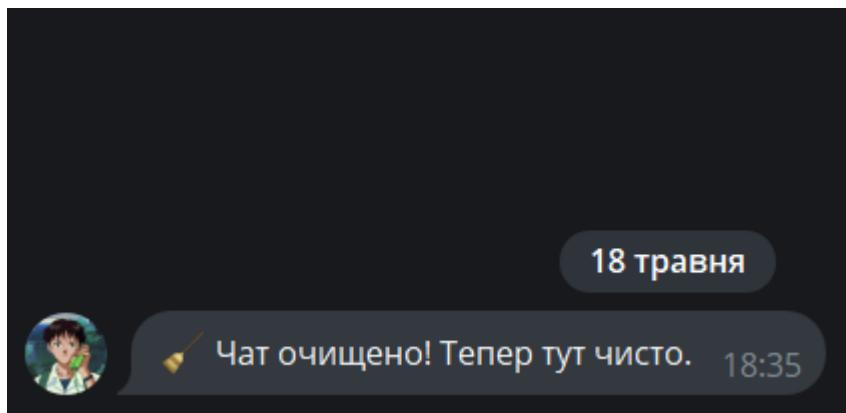
∅ Мій GitHub: [movavok](#)

[GitHub](#)

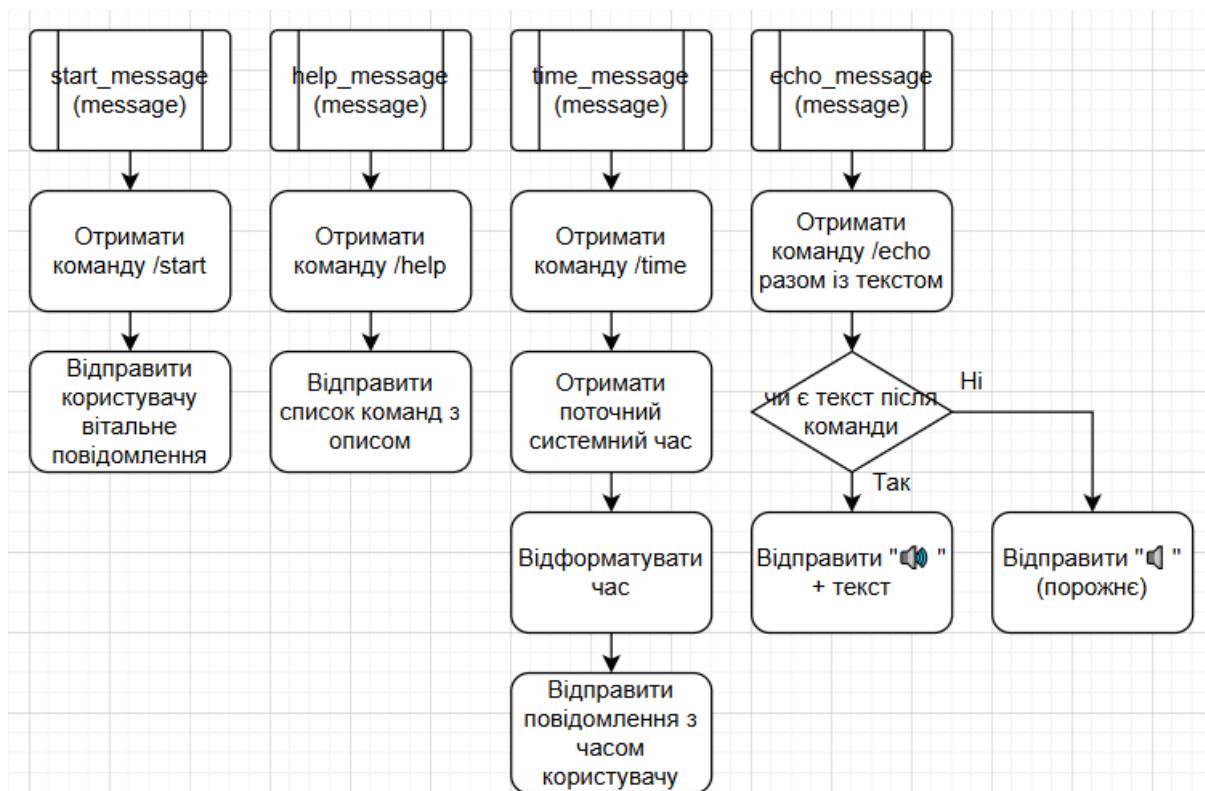
**movavok - Overview**

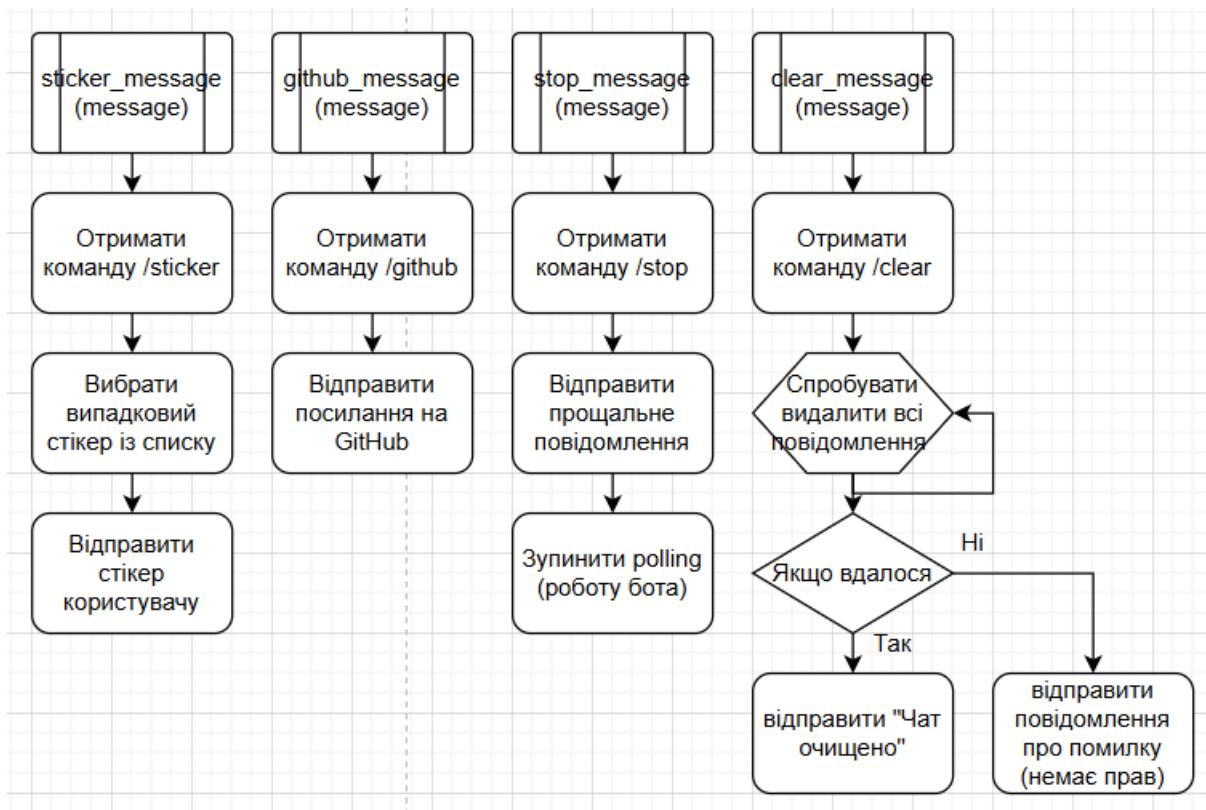
movavok has 9 repositories available. Follow their code on GitHub.

18:34



### Блок схеми:







## КОНТРОЛЬНІ ЗАПИТАННЯ

**1. Які бібліотеки мови Python можна використати для розробки чат-боту для телеграму?**

- **pyTelegramBotAPI (telebot)** — проста та популярна бібліотека для створення ботів Telegram.
- **python-telegram-bot** — багатофункціональна бібліотека з підтримкою багатьох функцій Telegram API.

- **aiogram** — асинхронна бібліотека, підходить для масштабних проектів.
- **telethon** — бібліотека для роботи з Telegram API на низькому рівні (боти і клієнти).

## **2. Що таке API токен?**

- **API токен** — унікальний секретний ключ, який видає Telegram BotFather при створенні нового бота.
- Він використовується для ідентифікації і авторизації вашого бота в Telegram API, дозволяє програмі взаємодіяти з Telegram від імені бота.

## **3. Як запустити розроблений чат-бот для телеграму на своєму комп'ютері?**

- Встановити Python та потрібні бібліотеки (наприклад, pip install pyTelegramBotAPI).
- Створити файл із кодом бота (наприклад, bot.py), вставити туди токен.
- Запустити скрипт через консоль командою python bot.py.
- Бот підключиться до Telegram і почне приймати повідомлення.

## **4. Як запустити розроблений чат-бот на комп'ютерах користувачів?**

- Найпростіший варіант — запустити бот на власному сервері чи хмарному хостингу, щоб він працював постійно.
- Якщо потрібно запускати локально на комп'ютерах користувачів, кожен користувач має встановити Python, залежності, і запустити бот.
- Для зручності можна зробити інсталятор або пакет, але це складніше і рідко використовується.

## **5. Як розмістити розроблений чат-бот на сервері?**

- Обрати сервер (VPS, хмарний сервер, наприклад, DigitalOcean, AWS, Heroku).
  - Завантажити туди файли бота.
  - Встановити Python і залежності.
  - Запустити бота в режимі постійної роботи (наприклад, через screen, tmux або systemd-сервіс).
    - Для кращої надійності можна налаштувати автозапуск і моніторинг роботи бота.