

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА**  
**ШЕВЧЕНКА**

**кафедра інформаційних систем та технологій**

**ЗВІТ**

із лабораторної роботи № 6

з дисципліни “**Технології програмування об’єктів лінгвістичної**  
**предметної галузі”**

на тему: “Взаємодія з месенджером Telegram”

Варіант № **2-05**

Виконав:

Студент групи №2

Кличлієв Кирило

Перевірив:

доц. Костіков М. П.

**Київ — 2023**

## **Мета лабораторної роботи**

Ознайомитися з бібліотекою telethon мови Python, що дозволяє взаємодіяти з месенджером Telegram і вилучати з нього потрібну інформацію.

## **Дані з індивідуального варіанту №2-05**

Контрольне питання:

“Способи (засоби) під’єднання до Telegram для розробників.”

## **Середовище розробки**

Мова програмування: Python

Назва та версія IDE: Visual Studio Code 1.78.2

Назва та версія ОС: Linux Mint 21 Vanessa

Мова ОС: англійська

## Хід роботи

1. Створюємо консольний проект мовою Python, що при запуску виводить ім'я, прізвище та групи виконавця, а також номер лабораторної роботи:

```
print("Кличлієв Кирило\nГрупа №2\nЛабораторна робота №6")
```

✓ 0.0s

Кличлієв Кирило  
Група №2  
Лабораторна робота №6

2. Імпортуємо бібліотеки, необхідні для виконання поставлених задач:

```
import telethon  
from pytz import timezone  
from tokenize_uk import tokenize_words
```

3. Реєструємось на сторінці розробників Telegram: [my.telegram.org/apps](https://my.telegram.org/apps)
4. Отримуємо та зберігаємо у змінних проекту параметри `api_id` та `api_hash`, а також власний `username` в Telegram і номер телефону:

```
api_id = "..."  
api_hash = "..."  
username = "@klychliiev"  
phone_number = "..."
```

5. Створюємо та запускаємо клієнт для з'єднання з Telegram, використовуючи вищезазначені параметри:

```
client = telethon.TelegramClient(username, api_id, api_hash)
```

6. Проходимо авторизацію при першому запуску клієнта для збереження сесії.
7. Створюємо в Telegram власний публічний канал.
8. Наповнюємо канал приблизно 10 постами з довільним текстом.
9. Під'єднуємося до свого каналу, зчитуємо та виводимо в консоль його ID, назву, `username` і дату створення:

```

async def get_channel_info():
    await client.start()
    channel_name = 'chernihiv_now'
    channel_entity = await client.get_entity(channel_name)

    local_tz = timezone('Europe/Kiev')
    date = channel_entity.date.astimezone(local_tz)

    print(f"ID каналу: {channel_entity.id}")
    print(f"Username каналу: {channel_entity.username}")
    print(f"Назва каналу: {channel_entity.title}")
    print(f"Дата й час створення каналу: {date.strftime('%Y-%m-%d %H:%M:%S')}")

with client:
    client.loop.run_until_complete(get_channel_info())

```

### Результат роботи коду:

```

klychiieafx@svarthamar:~/Desktop/lab_tg$ /bin/python3 /home/klychiieafx/Desktop/lab_tg/main.py
ID каналу: 1798996234
Username каналу: chernihiv_now
Назва каналу: ChT (Чернігів туристичний)
Дата й час створення каналу: 2023-05-26 21:23:03

```

10. Зчитуємо з каналу 4 останні пости і для кожного виводимо в консоль:

- Дату й час публікації. Час приводимо до часового пояса Києва (UTC+2)
- Зміст (текст)
- Кількість символів
- Кількість слів

```

async def get_posts():
    await client.start()
    channel_name = "chernihiv_now"
    channel_entity = await client.get_entity(channel_name)
    msgs = await client.get_messages(channel_entity, limit=4)

    local_tz = timezone('Europe/Kiev')

    for msg in msgs:
        entire_date = msg.date.astimezone(local_tz)
        print(f'Дата публікації: {entire_date.strftime("%Y-%m-%d")}')
        print(f'Час публікації: {entire_date.strftime("%H:%M:%S")}')
        print(f"Текст посту:\n{msg.message}")
        print(f"Кількість символів: {len(msg.message)}")
        words = [word for word in tokenize_words(
            msg.message) if word.isalpha()]
        print(f"Кількість слів: {len(words)}\n")

with client:
    client.loop.run_until_complete(get_posts())

```

### Результат роботи коду:

klychiieafx@svarthamar:~/Desktop/lab\_tg\$ /bin/python3 /home/klychiieafx/Desktop/lab\_tg/main.py

Дата публікації: 2023-05-26

Час публікації: 23:03:13

Текст посту:

Цікаве місце Чернігова #10 - Чернігівський пивзавод

Звісно, куди ж без нього? Історія марки «Чернігівське» починається у 1988 році, коли Чернігів готувався святкувати 1300-річчя. Ця марка з'явилася в результаті багатомісячної роботи колективу Чернігівського пивоварного заводу «Десна», з розробки рецептури нового пива, яке назвали на честь міста-іменинника. У 2006 році, разом з Миколаївки пивзаводом «Янтар» та Харківським «Рогань», утворює об'єднання, яке увійшло до складу найбільшого пивоварного концерну AB In Bev. Пивзавод приймає бажаючих дорослих на екскурсію та дегустацію 9 сортів пива свого виробництва.

Кількість символів: 615

Кількість слів: 75

Дата публікації: 2023-05-26

Час публікації: 23:02:19

Текст посту:

Цікаве місце Чернігова #9 - Борисоглібський собор

Свідок князівської доби є усипальницею князів роду Давидовичів, а пізніше головний храм кафедрального монастиря. Відома пам'ятка архітектури міста Чернігова домонгольської доби

названа на честь Св. страстотерпців князів Бориса й Гліба. Окрасою експозиції собору є срібні з позолотою царські врата, масою близько 56кг. Вони були виготовлені на початку XVIII ст. за наказом гетьмана І.С.Мазепи, в майстерні Пилипа Якоба Дрентветта з міста Аусбурга.

Кількість символів: 496

Кількість слів: 67

Дата публікації: 2023-05-26

Час публікації: 23:01:24

Текст посту:

Цікаве місце Чернігова #8 - Болдині гори

В 1069 р. преподобний Антоній прибув до Чернігова, де для майбутнього свого поселення вподобав Болдині гори. Назва місця походить від давньоруського слова «болд» – дуб. Це дерево має підвищену енергетику і його здавна шанували місцеві жителі; свого часу тут було язичницьке капище. Імовірно, з цього антонієвого вибору і почалися різні «дивацтва» цих місць... Прекрасний оглядовий майданчик, з якого Чернігівський дитинець і частина міста проглядається, як на долоні.

Кількість символів: 506

Кількість слів: 70

Дата публікації: 2023-05-26

Час публікації: 23:00:18

Текст посту:

Цікаве місце Чернігова #7 - Антонієві печери

Для одних це – християнська святиня, для інших – примітний зразок підземної архітектури, для третіх – притулок привидів... Сьогодні Антонієві печери є комплексом підземних споруд, що складається з церков святого Антонія, святого Феодосія Тотемського, Ніколи Святенника, каплиці з похованнями, келії Антонія та кімнати, сполучених між собою галереями.

Кількість символів: 395

Кількість слів: 49

11. Зберігаємо всі пости з каналу в текстовий файл (з розширенням .txt), проставляючи перед кожним постом дату та час публікації. Між окремими публікаціями робимо відступ в один рядок:

```

my_file = open('chernihiv_posts.txt', 'a')

async def write_file():
    await client.start()
    channel_name = "chernihiv_now"
    channel_entity = await client.get_entity(channel_name)
    limit = None
    msgs = await client.get_messages(channel_entity, limit=limit)
    local_tz = timezone('Europe/Kiev')

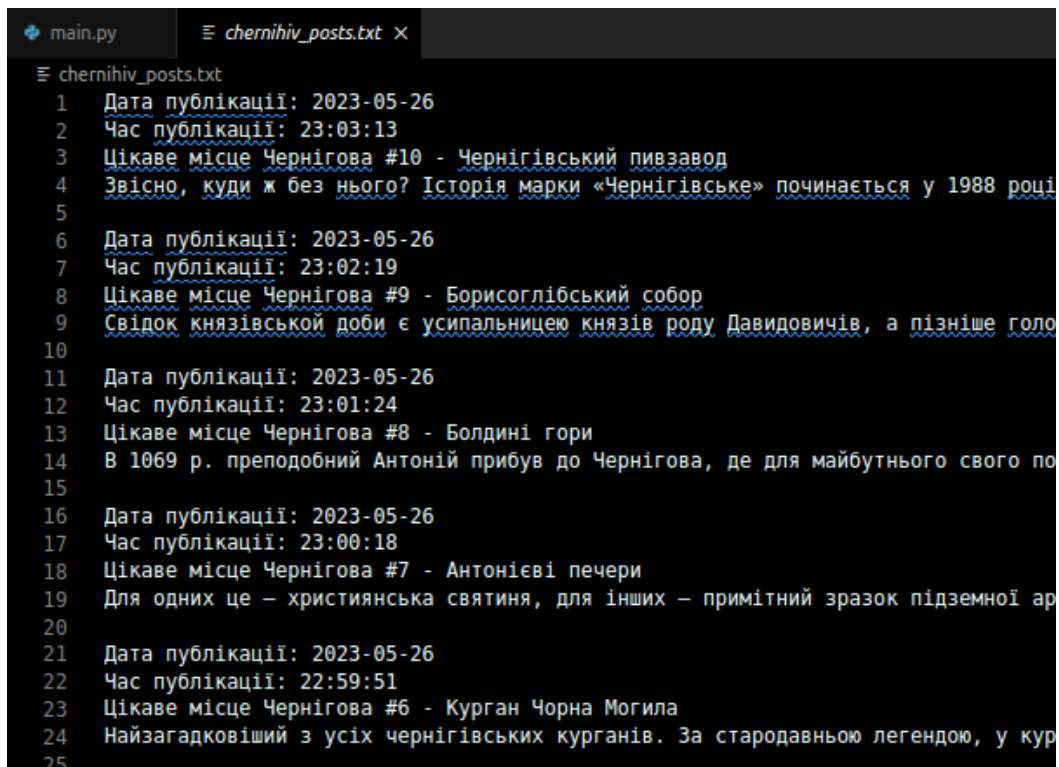
    for msg in msgs:
        if msg.message != None:
            entire_date = msg.date.astimezone(local_tz)
            date = f'Дата публікації: {entire_date.strftime("%Y-%m-%d")}\n'
            time = f'Час публікації: {entire_date.strftime("%H:%M:%S")}\n'
            my_file.write(f"{date}{time}{msg.message}\n\n")
        else:
            entire_date = msg.date.astimezone(local_tz)
            date = f'Дата публікації: {entire_date.strftime("%Y-%m-%d")}\n'
            time = f'Час публікації: {entire_date.strftime("%H:%M:%S")}\n'
            text = 'Текст публікації відсутній'
            my_file.write(f"{date}{time}{text}\n\n")

    await client.disconnect()

with client:
    client.loop.run_until_complete(write_file())

```

Вміст новоствореного файлу виглядає наступним чином:



```

main.py  chernihiv_posts.txt x
chernihiv_posts.txt
1  Дата публікації: 2023-05-26
2  Час публікації: 23:03:13
3  Цікаве місце Чернігова #10 - Чернігівський пивзавод
4  Звісно, куди ж без нього? Історія марки «Чернігівське» починається у 1988 році
5
6  Дата публікації: 2023-05-26
7  Час публікації: 23:02:19
8  Цікаве місце Чернігова #9 - Борисоглібський собор
9  Свідок князівської доби є усипальницею князів роду Давидовичів, а пізніше голо
10
11  Дата публікації: 2023-05-26
12  Час публікації: 23:01:24
13  Цікаве місце Чернігова #8 - Болдині гори
14  В 1069 р. преподобний Антоній прибув до Чернігова, де для майбутнього свого по
15
16  Дата публікації: 2023-05-26
17  Час публікації: 23:00:18
18  Цікаве місце Чернігова #7 - Антонієві печери
19  Для одних це – християнська святиня, для інших – примітний зразок підземної ар
20
21  Дата публікації: 2023-05-26
22  Час публікації: 22:59:51
23  Цікаве місце Чернігова #6 - Курган Чорна Могила
24  Найзагадковіший з усіх чернігівських курганів. За стародавньою легендою, у кур
25

```

12. Під'єднуємося до каналу `tpolph3_2022` та виводимо в консоль загальну кількість постів на каналі

```
async def get_tpolph_info():
    await client.start()
    channel_name = "tpolph3_2022"
    channel_entity = await client.get_entity(channel_name)
    msgs = await client.get_messages(channel_entity)
    print(
        f"Загальна кількість повідомлень на каналі tpolph3_2022: {msgs.total}")

with client:
    client.loop.run_until_complete(get_tpolph_info())
```

### Результат роботи коду:

```
klychiieffx@svarthamar:~/Desktop/lab_tg$ /bin/python3 /home/klychiieffx/Desktop/lab_tg/main.py
Загальна кількість повідомлень на каналі tpolph3_2022: 135
```

13. Відповідь на контроль питання: “Способи (засоби) під'єднання до Telegram для розробників”

Для розробників, що використовують функціонал месенджера Telegram чи інформацію з нього, існують різні способи під'єднання та взаємодії з платформою. Ось декілька засобів, які розробники можуть використовувати для під'єднання до Telegram:

1. Telegram Bot API: Telegram надає API, яке дозволяє створювати та управляти ботами в месенджері. Можна використовувати Bot API для створення ботів з різними функціями, зокрема, прийом повідомлень, відправка повідомлень, керування групами/каналами тощо.
2. Telegram API: Крім Bot API, Telegram надає також загальний Telegram API, який дозволяє взаємодіяти безпосередньо з користувачами та групами. API можна використовувати для створення клієнтів, ботів або інтеграції Telegram з іншими додатками.
3. Telegram MTProto API: Це протокол, який використовується Telegram для забезпечення з'єднання з серверами та передачі повідомлень. Розробники можуть використовувати MTProto API для створення власних клієнтів або інших програм, які взаємодіють з Telegram.



4. Telegram CLI: Це інтерфейс командного рядка для Telegram, який дозволяє взаємодіяти з месенджером через команди. Telegram CLI можна використовувати для автоматизації деяких завдань або інтеграції з іншими програмами.
5. Telegram Bot SDK та бібліотеки. Існують різні сторонні бібліотеки та фреймворки, які допомагають у розробці Telegram-ботів та інших застосунків. Наприклад, є SDK для мов програмування Python, JavaScript, Java, PHP та ін.

### **Висновки**

Під час виконання лабораторної роботи №6 ми ознайомилися з бібліотекою `telethon` мови Python, яка дозволяє взаємодіяти з месенджером Telegram та інформацією, наявною в ньому. Зокрема, за допомогою цього модуля ми підключалися до новоствореного для цього проекту каналу та вилучили з нього наступну інформацію:

1. ID, назва, `username`, дата й час створення каналу;
2. Тексти постів, кількість символів і слів у кожному з них.

Тексти всіх постів, а також дата й час публікації кожного з них були записані за допомогою вбудованих методів мови Python у новий файл "`chernihiv_posts.txt`". За допомогою функціоналу бібліотеки `timezone` час публікації постів і створення каналу відображений за київським часом (UTC+2). Для вже давно існуючого каналу (`tpolp3_2022`) за допомогою методу `total` було визначено загальну кількість постів на каналі (135).