

УДК 621.762:678-19

Пастернак В. В., канд. техн. наук, доцент  
Сватко А. В., студент

Волинський національний університет імені Лесі Українки, м. Луцьк, [Shyberko@ur.net](mailto:Shyberko@ur.net)

## РОЗРОБКА СКРИПТІВ ДЛЯ АВТОМАТИЗАЦІЇ З ВИКОРИСТАННЯМ МОВИ ПРОГРАМУВАННЯ PYTHON

У добу глобалізації, інтеграції та стрімкого розвитку інновацій і технологій важко уявити сьогодення і сучасний світ без WEB-простору, без пошукових систем, різноманітних сайтів, додатків, а також без скриптів для автоматизації. Автоматизацією користується майже кожна людина і тому одну із провідних ланок в IT-індустрії займає саме вибір правильної мови програмування для розробки скриптів. При створенні коду, який надсилає скарги на телеграм канали можна використовувати наступні мови програмування, як: JavaScript, Java, PowerShell-PowerShell, Ruby і т.д. Однак найбільш доцільно використовувати високорівневу мову програмування – Python. За допомогою цієї мови будуються нейронні мережі, створюються штучні інтелекти, розробляються WEB-сайти, а також створюються скрипти для автоматизації рутинних задач. До найпопулярніших бібліотек та фреймворків Python, які можна використовувати під час написання коду можна віднести наступні: Selenium, PyAutoGUI, Requests, Pandas, Django.

Нами був розроблений код з використанням мови програмування Python, який автоматично надсилає скарги на телеграм канали і безпосередньо фіксує скріншоти виконаної роботи. Реалізація на мові програмування Python із використанням основних операцій, циклів та прописаних коментарів представлено нижче:

```
from time import sleep # Підключаємо метод sleep() з бібліотеки time
from keyboard import is_pressed # Підключаємо метод is_pressed() з бібліотеки keyboard
import pyautogui # Підключаємо бібліотеку pyautogui
m = 1 # Створюємо змінну для нумерації наших скріншотів, яку в подальшому будемо збільшувати
f = open('BlocTg/test.txt', 'r') # Відкриваємо завчасно заготовлений текстовий документ test.txt з посиланнями на канали на які потрібно кинути скаргу
for i in f: # За допомогою циклу for вибираємо потрібні посилання із заготовленого документу
    if is_pressed('Ctrl'): # Відслідковуємо взаємодію користувача із утриманням клавіши Ctrl і якщо умова виконується, то виконуємо наступний код:
        break # Вихід із циклу
        pyautogui.click(212, 1053) # Відкриваємо браузер
        pyautogui.click(371, 67) # Активуємо поле вводу
        pyautogui.write(i) # Вводимо вибране посилання
        pyautogui.press('enter') # Застосовуємо клавішу enter для пошуку
        sleep(3) # Чекаємо поки відкриється потрібний телеграм канал
        pyautogui.click(1893, 64) # Відкриваємо діалогове вікно для взаємодії з каналом
        sleep(0.5) # Чекаємо 0.5 секунд
        image_location = pyautogui.locateOnScreen('BlocTg/test.png') # Шукаємо завчасно піктограму «поскаржитися», яка розташована на екрані монітора
        if image_location: # У випадку, якщо піктограма знайдена, то виконуються наступні дії:
```

```
x, y = pyautogui.center(image_location) # Отримуємо координати центру знайденої піктограми «поскаржитися»
pyautogui.click(x,y) # Застосовуємо знайдені координати центру піктограми «поскаржитися» за допомогою методу click() з бібліотеки pyautogui
pyautogui.click(875,664) # Обираємо розділ «інше»
pyautogui.write('/') # Вписуємо будь який символ
pyautogui.click(1070,623) # Застосовуємо піктограму "поскаржитися"
sleep(0.5) # Чекаємо
pyautogui.screenshot(f'BlocTg/{m}test.png', region = (430,30,1580,990)) # Робимо скріншот виконаної роботи і зберігаємо його у вказаній дерикторії, де зазначаємо потрібне ім'я
else: # У випадку, якщо піктограма «поскаржитися» не знайдена, то виконуємо наступні дії:
pyautogui.screenshot(f'BlocTg/{m} Заблокований.png', region = (430,30,1580,990)) # Робимо скріншот уже заблокованого каналу і зберігаємо його у вказаній дерикторії з потрібним ім'ям
pyautogui.click(212, 1053) # Повертаємось назад у браузер для повторного циклу
m += 1 # Збільшуємо нашу змінну m на одиницю
```

Слід відмітити, що реалізація даного коду дозволяє здійснювати цілий ряд переваг, а саме: 1) використовувати безкоштовні прив'язки бібліотек: PyAutoGUI, Time, Keyboard; 2) має наявність великої кількості модулів, які забезпечують різноманітні додаткові можливості; 3) розроблений скрипт працює набагато швидше завдяки своїй компактності та використанню меншої кількості рядків (операцій, циклів). Важливо теж відмітити, що у даній розробці скрипту ми використовуємо метод sleep(), який дозволив нам зупинити виконання коду на зазначену кількість секунд. Для коректної роботи коду на мові програмування Python потрібно виконати певні умови: 1) завчасно відкрити браузер, у якому буде проходити уся робота відкриття телеграм каналів, а також звернута піктограма даного браузера повинна стояти першою (зліва) у переліку від усіх інших відкритих програм; 2) програма телеграм повинна бути розгорнута на весь екран монітору комп'ютера, а також важливим є те, що панель функцій самої програми телеграм має бути згорнута під час роботи скрипту; 3) користувач не повинен бути учасником телеграм каналів з яким безпосередньо взаємодіє скрипт. Важливим є і те, що панель функцій розташована завжди з правого боку відкритого телеграм каналу. У результаті розроблений код на мові програмування Python зберігає скріншоти у вказану дерикторію, тобто скріншот кожного каналу, на якому видно, що на канал була кинута скарга. Тобто, якщо каналів 200, то і скріншотів буде 200. У випадку заблокованого каналу розроблений скрипт зберігає скріншот, але підписує його як заблокований.

**Висновок.** У даній роботі ми розробили скрипт для автоматизації з використанням мови програмування Python, а також обґрунтували цілий ряд переваг використання даної мови програмування. Варто зазначити, що розроблена теорія скриптів для автоматизації може вирішувати сучасні виклики сьогодення, які потребують уважного управління, планування, постійного вдосконалення та розуміння її впливу на постійний розвиток суспільства.

#### Список посилань

1. Яковенко А.В. Основи програмування. Python. Частина 1 / за ред. А.В. Яковенко. – Київ: КПІ ім. Ігоря Сікорського, 2018. – 195 с. Режим доступу: <https://ela.kpi.ua/bitstream/123456789/25111/1/Python.pdf>.

2. Пастернак В.В. Особливості моделювання методами StarUML. / В.В.Пастернак // Математичні методи та моделі технічних і економічних систем: тези доп. Міжнар. наук.-техн. конф. (22-23 лист. 2022 р.). – Тернопіль, 2022. – с. 146-147.

3. Пастернак В.В. Еволюція розвитку інформаційних систем та технологій. В.В.Пастернак // Комплексне забезпечення якості технологічних процесів та систем: тези доп. XII Міжнар. наук.-практ. конф. (26-27 травн. 2022 р.). – Чернігів, 2022. – с. 180-181.

УДК: 004.056.6

**Рзаєва С. Л., канд. техн. наук, доцент**

Державний торговельно-економічний університет, м. Київ, rzaevasl@knu.edu.ua

**Рзаєв Д. О., старший викладач**

Київський національний економічний університет імені Вадима Гетьмана, ditomas@ukr.net

**Ковальова Л. І., спеціаліст**

Державний торговельно-економічний університет, м. Київ, l.kovalova@knu.edu.ua

## **ОСНОВНІ СКЛАДОВІ БЕЗПЕКИ СХОВИЩА ДАНИХ ТА МЕТОДИ ЗАХИСТУ**

Проблеми безпеки сховища даних, пов'язані з процесом забезпечення безпеки всіх даних, незважаючи на те, що зберігається маса/велика інформація. Безпека даних пов'язана зі стійкістю та надійністю інформації.

На етапі сховища, доступ до даних зазвичай матимуть більше груп користувачів, ніж у базах даних. Необхідно застосовувати керування даними, щоб захистити певні фрагменти конфіденційної інформації від доступу не тих користувачів організації. Незалежно від того, чи йдеться про особисту інформацію (PII), чи фінансову інформацію, забезпечення конфіденційності даних має бути протягом усього «шляху» від БД до сховища даних. Запобігання оприлюдненню такої інформації є ключовим, і до нього можна підходити різними способами.

Кожна компанія зберігає інформацію, яка не може бути відкритою для всіх, хто працює в компанії. При переході від БД до Data Warehouse більше людей отримують доступ до даних. Адміністратору потрібно переконатися в узгодженості конфіденційної інформації, її збереження та обмеження у сховищі даних, і як до неї можна отримати доступ за допомогою інструментів бізнес-аналітики (BI).

Є кілька способів вирішення цього питання та кілька запитань, на які потрібно відповісти.

- 1) Де зараз обробляються конфіденційні дані (ідентифікаційна та фінансова інформація)?
- 2) Чи будуть ці конфіденційні дані все ще присутні в сховищі даних, а потім очищені?
- 3) Як цю інформацію буде видалено або обмежено з відкритих наборів даних - сценаріїв на шляху до сховища, вітрин даних, створених зі сховища?

На ці запитання потрібно відповісти, перш ніж підключати ці джерела до інструментів BI. У великих компаніях часто всі внутрішні дані вважаються конфіденційними. Навіть у внутрішньому плані департаментам необхідно знати дані інших департаментів. Проблеми виникають, коли компанія підключає своє сховище даних до своєї платформи BI або загалом надає доступ для запитів різним відділам. Це призводить до потенційного доступу до конфіденційних даних неавторизованими користувачами.

*Як захистити конфіденційні дані в сховищі даних.* Найпряміший спосіб обмежити доступ належним людям - це застосувати правила на рівні бази даних (сховище даних використовуються як реляційна база). Це можна зробити шляхом створення дозволів нарівні «лише для читання», створення спеціальних груп користувачів і шифрування конфіденційних даних.

*Лише для читання.* Налаштування сховища «лише для читання» за замовчуванням – запобігає виконанню будь-яких небезпечних інструкцій SQL-запису для даних.