

## МЕТОДИЧНІ АСПЕКТИ ЗАСТОСУВАННЯ СУЧАСНИХ ПАРАДИГМ ПРОГРАМУВАННЯ

Сирий В.М., ст. викл.  
Кірюхіна А.С., здобувач РВО бакалавр  
Державний біотехнологічний університет  
м. Харків, Україна, 0506995128@btu.kharkov.ua

**Анотація:** Розглянуто методичні аспекти застосування сучасних парадигм програмування у навчанні на прикладі задачі розробки алгоритму автентифікації.

**Ключові слова:** інформаційні технології, програмування, методичне забезпечення

Опанування сучасними інформаційними технологіями багато в чому залежить від набуття здобувачами компетентності застосовувати сучасні парадигми програмування. Так, розробники мови Python є прихильниками відомої філософії програмування «The Zen of Python» [1, 2].

У даній роботі ми дозволили собі власне тлумачення частини принципів «Дзену Пайтона» тезою про доцільність комплексного творчого підходу до вивчення мовних конструкцій Python вже з перших кроків навчання.

Наприклад, у дослідженні методів об'єктів варто одразу розглядати можливості їх ефективного прикладного використання, поєднувати технології уведення-виведення даних на консоль із залученням файлової системи, оптимізувати початковий код застосуванням функцій користувача, опанування операторами керування супроводжувати демонстраційними прикладами «одягання» початкового коду в графічний інтерфейс користувача (GUI) тощо.

Зазначений методичний підхід продемонструємо на прикладі розв'язання відомої задачі програмування алгоритму автентифікації доступу до уявного інформаційного ресурсу. Така задача вдало демонструє вибудовування логіки алгоритму поєднанням уведення параметрів доступу з консолі з їх циклічною перевіркою операторами розгалуження.

На першому етапі здобувачі розробляють початковий код із розташуванням бази даних параметрів автентифікації в тілі програми та консольним інтерфейсом користувача. Потім виносять бази даних параметрів доступу в окремі CSV-файли проєкту. Далі знайомляться із синтаксисом функції та програмують блоки розгалуження «так» і «ні» за допомогою функцій, які згодом будуть використані в подієво-орієнтованому програмуванні віджетів GUI.

Практика розробки функцій спонукає здобувачів до розв'язання колізії внутрішніх і зовнішніх змінних та потреби в їх декларуванні як глобальних.

З іншого боку на даному прикладі здобувачі отримують досвід застосування функцій з метою оптимізації початкового коду. Тут функції обслуговують типові реакції застосунку на уведення коректних чи хибних параметрів автентифікації, надання чи блокування доступу до уявного ресурсу.

Нарешті для застосунку розглядається демонстраційний приклад створення графічного інтерфейсу користувача. Попервах здобувачі знайомляться з аспектами організації діалогових вікон, застосування віджетів

елементів управління та їх бібліотеками у першому наближенні. На демонстраційних прикладах для інших задач ознайомлення з технологією GUI стає більш докладним і поступово здобувачі отримують сталі навички розробки графічного інтерфейсу та звикають орієнтувати на нього майбутні проекти.

У даному прикладі застосовується уніфіковане динамічне діалогове вікно (рис. 1) з табличним впорядкуванням віджетів і виведенням поточного стану автентифікації у вигляді міток.

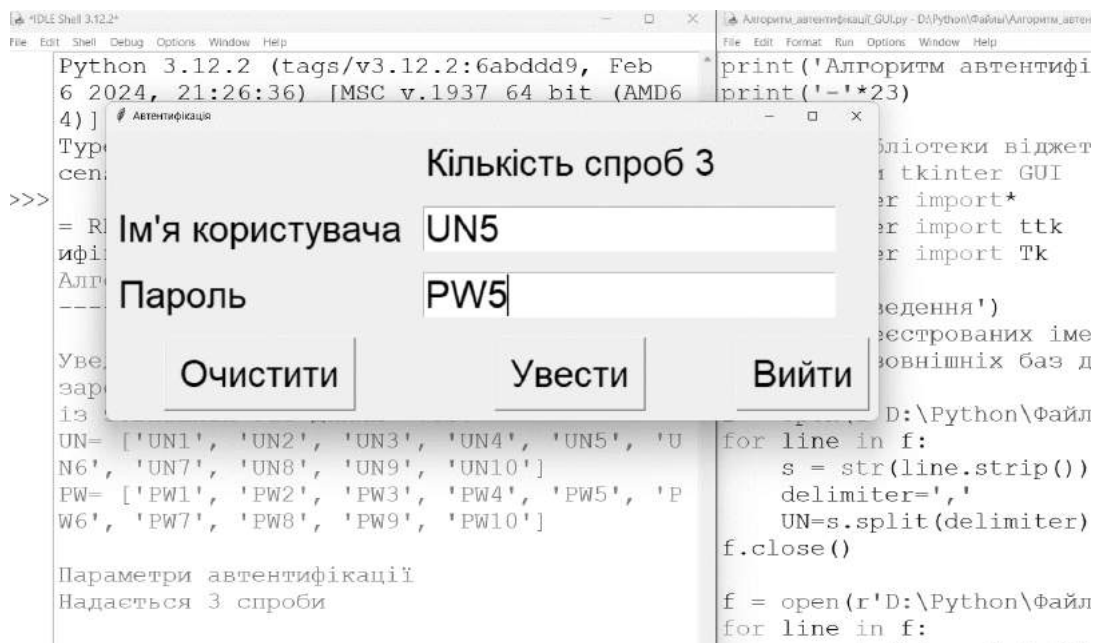


Рис. 1. Тестування проєкту алгоритму автентифікації

Динаміка діалогового вікна забезпечується створенням та видаленням (чи приховуванням) елементів управління. При цьому розміри вікна автоматично підлаштовуються під їх поточний склад.

Окремим вагомим дослідженням, яке може бути проведеним на даному прикладі є робота із зовнішніми джерелами інформації, що прокладає шлях майбутнім розробникам до застосування Python в алгоритмах для структурованих даних [3].

У методичному аспекті вважаємо доцільним, також, супроводжувати вивчення фундаментальних операторів: присвоєння, виразів та управління, які складають остов програмного коду, оглядом їх синтаксису іншими актуальними мовами програмування, як то: діалекти C, Java Script тощо, з метою легкого старту в сучасні ІТ.

#### Список літератури

1. PEP 20 – The Zen of Python. URL: <https://peps.python.org/pep-0020/> (дата звернення: 09.04.2024).
2. Python 3.12.2 documentation. URL: <https://docs.python.org/uk/3/> (дата звернення: 09.04.2024).
3. Крєневич А.П. Алгоритми і структури даних. Підручник. К.: ВПЦ «Київський Університет», 2021. 200 с.