Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Кафедра ФІКТ

Група:ВТ-21-1

Програмування мовою Python

Лабораторна робота №

«unit-тестування в мові Python»

Виконав: Вигнич О. С.

Прийняв: Морозов Д. С.

**Мета роботи:** ознайомитися з алгоритмами послідовної (лінійної) структури, з процедурами запуску програм, які реалізують ці алгоритми на мові Python; знайомство з інтегрованим середовищем розробки – integrated development environment (IDLE).

1. Онлайн-магазин (Завдання 8 з Л.Р. №8).

a. Створіть клас з ім’ям Shop(). Клас Shop() повинен містити два

атрибути: shop\_name і store\_type. Створіть метод describe\_shop(), який

виводить два атрибути, і метод open\_shop(), який виводить

повідомлення про те, що онлайн-магазин відкритий. Створіть на основі

класу екземпляр з ім’ям store. Виведіть два атрибути окремо, потім

викличте обидва методи.

b. Створіть три різних екземпляри класу, викличте для кожного

екземпляру метод describe\_shop().

c. Додайте атрибут number\_of\_units зі значенням за замовчуванням 0; він

представляє кількість видів товару у магазині. Створіть екземпляр з

ім’ям store. Виведіть значення number\_of\_units, а потім змініть

number\_of\_units і виведіть знову.

d. Додайте метод з ім’ям set\_number\_of\_units(), що дозволяє задати

кількість видів товару. Викличте метод з новим числом, знову виведіть

значення. Додайте метод з ім’ям increment\_number\_of\_units(), який

збільшує кількість видів товару на задану величину. Викличте цей

метод.

e. Напишіть клас Discount(), що успадковує від класу Shop(). Додайте

атрибут з ім’ям discount\_products для зберігання списку товарів, на які

встановлена знижка. Напишіть метод get\_discounts\_ptoducts, який

виводить цей список. Створіть екземпляр store\_discount і викличте цей

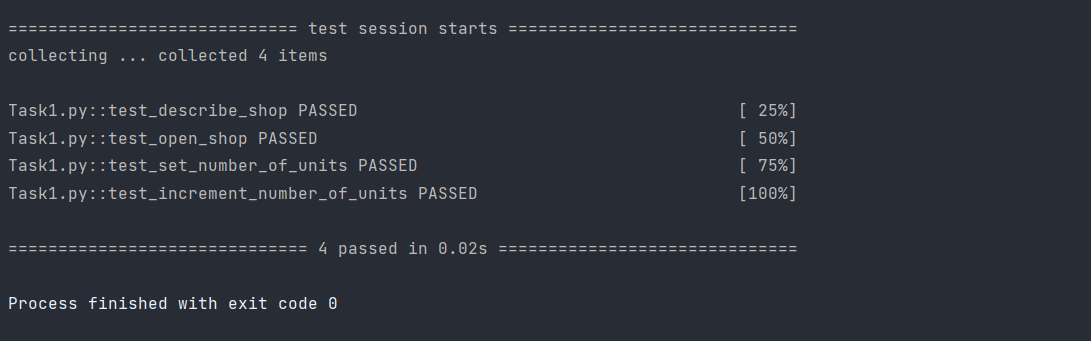
метод.

f. Збережіть код класу Shop() у модулі. Створіть окремий файл, що

імпортує клас Shop(). Створіть екземпляр all\_store і викличте один з

методів Shop(), щоб перевірити, що команда import працює правильно.

*import* pytest  
*from* shop *import* Shop  
  
shop = Shop('АТБ', 'продуктовий')  
*def* test\_describe\_shop():  
 *assert* shop.describe\_shop() == "Name=>АТБ Type=>продуктовий"  
*def* test\_open\_shop():  
 *assert* shop.open\_shop() == "АТБ is open"  
*def* test\_set\_number\_of\_units():  
 shop.set\_number\_of\_units(4)  
 *assert* shop.number\_of\_units == 4  
*def* test\_increment\_number\_of\_units():  
 shop.increment\_number\_of\_units()  
 *assert* shop.number\_of\_units == 5



2. Облік користувачів на сайті (Завдання 9 з Л.Р. №8).

a. Створіть клас з ім’ям User. Створіть два атрибути first\_name і

last\_name, а потім ще кілька атрибутів, які зазвичай зберігаються у

профілі користувача (поштова адреса, нікнейм, що відображається на

сайті, згода на розсилку новин з форуму). Напишіть метод

describe\_user який виводить повне ім’я користувача. Створіть ще один

метод greeting\_user() для виведення персонального вітання для

користувача. Створіть кілька примірників, які представляють різних

користувачів. Викличте обидва методи для кожного користувача.

b. Додайте атрибут login\_attempts у клас User. Напишіть метод

increment\_login\_attempts(), що збільшує значення login\_attempts на 1.

Напишіть інший метод з ім’ям reset\_login\_attempts(), обнуляє

значення login\_attempts. Створіть екземпляр класу User і викличте

increment\_login\_attempts() кілька разів. Виведіть значення

login\_attempts, щоб переконатися у тому, що значення було змінено

правильно, а потім викличте reset\_login\_attempts(). Знову виведіть

login\_attempts і переконайтеся у тому, що значення обнулилося.

c. Адміністратор - користувач з повними адміністративними

привілеями. Напишіть клас з ім’ям Admin, що успадковує від класу

User. Додайте атрибут privileges для зберігання списку рядків виду

«Allowed to add message», «Allowed to delete users», «Allowed to ban

users» і т. д. Напишіть метод show\_privileges() для виведення набору

привілеїв адміністратора. Створіть екземпляр Admin і викличте

метод.

d. Напишіть клас Privileges. Клас повинен містити всього один атрибут

privileges зі списком, який треба забрати із класу Admin. Водночас,

необхідно перемістити метод show\_privileges() у клас Privileges із

класу Admin. Створіть екземпляр priv як атрибут класу Admin.

Створіть новий екземпляр admin і використайте метод для виведення

списку привілеїв.

e. Збережіть клас User в одному модулі, а класи Privileges і Admin у

іншому модулі. В окремому файлі створіть екземпляр admin і

викличте метод show\_privileges(), щоб перевірити, що все працює

правильно.

*from* user *import* User  
*from* PrivAndAdmin *import* Admin  
  
user = User("User1(First)", "User1(last)","User1(nick)")  
*def* test\_describe\_user():  
 *assert* user.describe\_user() == 'Name:User1(First) Surname:User1(last) Nackname:User1(nick)'  
*def* test\_greeting\_user():  
 *assert* user.greeting\_user() == 'Hello User1(nick)'  
  
user1 = User("User(First)", "User(last)","User(nick)", 1)  
*def* test\_increment\_login\_attempts():  
 user1.increment\_login\_attempts()  
 user1.increment\_login\_attempts()  
 user1.increment\_login\_attempts()  
 *assert* user1.login\_attempts == 4  
  
*def* test\_reset\_login\_attempts():  
 user1.reset\_login\_attempts()  
 *assert* user1.login\_attempts == 0  
  
*def* test\_show\_privileges():  
 admin = Admin("Admin(First)", "Admin(last)", "Admin(nick)", 1)  
 *assert* admin.show\_privileges() == ["Allowed to add message", "Allowed to delete users", "Allowed to ban users"]  
  
*def* test\_show\_privileges2():  
 admin = Admin("Admin(First)", "Admin(last)","Admin(nick)", 1)  
 *assert* admin.priv.show\_privileges() == ["Allowed to add message", "Allowed to delete users", "Allowed to ban users"]

