Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



ЗВІТ

до лабораторної роботи №9

З дисципліни: «Кросплатформні засоби програмування»

На тему: «ОСНОВИ ОБ’ЄКТНО-ОРІЄНТОВАНОГО

ПРОГРАМУВАННЯ У PYTHON»

Варіант 20

**Виконав:**

ст. групи КІ-306

Петрук О.С.

**Прийняв:**

доцент кафедри ЕОМ

Олексів М.В.

Львів – 2024

**Мета:** оволодіти навиками реалізації парадигм об’єктно-орієнтованого

програмування використовуючи засоби мови Python.

**Завдання:**

1. Написати та налагодити програму на мові Python згідно варіанту. Програма має задовольняти наступним вимогам:

• класи програми мають розміщуватися в окремих модулях в одному пакеті;

• точка входу в програму (main) має бути в окремому модулі;

• мають бути реалізовані базовий і похідний класи предметної області згідно варіанту;

• програма має містити коментарі.

2. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.

4. Дати відповідь на контрольні запитання.

**Базовий клас згідно варіанту №20: «Взуття»**

**Похідний клас згідно варіанту №20: «Чоботи»**

**Хід роботи**

Код програми:

***Shoes.py***

*class Shoes:*

*"""*

*Абстрактний клас Shoes, що описує основні характеристики та функціональність взуття.*

*"""*

*def \_\_init\_\_(self, brand, size, color):*

*"""*

*Конструктор для створення взуття.*

*:param brand: Бренд взуття.*

*:param size: Розмір взуття.*

*:param color: Колір взуття.*

*"""*

*self.brand = brand*

*self.size = size*

*self.color = color*

*self.laced = False*

*self.worn = False*

*self.distance\_run = 0.0*

*def needs\_repair(self):*

*"""*

*Перевіряє, чи потрібно ремонтувати взуття.*

*:return: True, якщо пробіг >= 100 км.*

*"""*

*return self.distance\_run >= 100*

*def add\_distance(self, distance):*

*"""*

*Додає дистанцію до загального пробігу.*

*:param distance: Пробіг (км).*

*"""*

*self.distance\_run += distance*

*# Геттери та сеттери для атрибутів класу*

*def get\_brand(self):*

*return self.brand*

*def get\_size(self):*

*return self.size*

*def get\_color(self):*

*return self.color*

*def is\_laced(self):*

*return self.laced*

*def set\_laced(self, laced):*

*self.laced = laced*

*def is\_worn(self):*

*return self.worn*

*def set\_worn(self, worn):*

*self.worn = worn*

*def get\_distance\_run(self):*

*return self.distance\_run*

***Snekers.py***

*from shoes import Shoes*

*class Sneakers(Shoes):*

*"""*

*Клас Sneakers, що представляє кросівки. Наслідує клас Shoes та реалізує інтерфейс ShoeOperations.*

*"""*

*def \_\_init\_\_(self, brand, size, color):*

*"""*

*Конструктор для створення кросівок.*

*:param brand: Бренд кросівок.*

*:param size: Розмір кросівок.*

*:param color: Колір кросівок.*

*"""*

*super().\_\_init\_\_(brand, size, color)*

*def wear(self):*

*"""*

*Одягає кросівки, якщо вони не одягнуті.*

*"""*

*if self.is\_worn():*

*print("Кросівки вже одягнені.")*

*else:*

*self.set\_worn(True)*

*print("Кросівки одягнені.")*

*self.log\_operation("Кросівки одягнені.")*

*def remove(self):*

*"""*

*Знімає кросівки, якщо вони надягнуті.*

*"""*

*if not self.is\_worn():*

*print("Кросівки вже зняті.")*

*else:*

*self.set\_worn(False)*

*print("Кросівки зняті.")*

*self.log\_operation("Кросівки зняті.")*

*def clean(self):*

*"""*

*Очищає кросівки.*

*"""*

*print("Кросівки очищені.")*

*self.log\_operation("Кросівки очищені.")*

*def repair(self):*

*"""*

*Відремонтувати кросівки, якщо вони зняті.*

*"""*

*if self.is\_worn():*

*print("Спочатку зніміть кросівки, щоб відремонтувати їх.")*

*else:*

*print("Кросівки відремонтовані.")*

*self.log\_operation("Кросівки відремонтовані.")*

*self.add\_distance(-100)  # Скидаємо пробіг на 100 км після ремонту*

*def polish(self):*

*"""*

*Відполірувати кросівки, якщо вони зняті.*

*"""*

*if self.is\_worn():*

*print("Спочатку зніміть кросівки, щоб відполірувати їх.")*

*else:*

*print("Кросівки відполіровані.")*

*self.log\_operation("Кросівки відполіровані.")*

*def lace\_up(self):*

*"""*

*Зав'язує шнурки, якщо кросівки надягнуті.*

*"""*

*if not self.is\_worn():*

*print("Спочатку одягніть кросівки, щоб зав'язати шнурки.")*

*elif self.is\_laced():*

*print("Шнурки вже зав'язані.")*

*else:*

*self.set\_laced(True)*

*print("Шнурки зав'язані.")*

*self.log\_operation("Шнурки зав'язані.")*

*def unlace(self):*

*"""*

*Розв'язує шнурки, якщо вони зав'язані.*

*"""*

*if not self.is\_laced():*

*print("Шнурки вже розв'язані.")*

*else:*

*self.set\_laced(False)*

*print("Шнурки розв'язані.")*

*self.log\_operation("Шнурки розв'язані.")*

*def check\_status(self):*

*"""*

*Перевіряє та виводить поточний стан кросівок.*

*"""*

*status = (*

*f"Бренд: {self.get\_brand()}, Розмір: {self.get\_size()}, Колір: {self.get\_color()}, "*

*f"Шнурки зав'язані: {'так' if self.is\_laced() else 'ні'}, "*

*f"Одягнені: {'так' if self.is\_worn() else 'ні'}, "*

*f"Пробіг: {self.get\_distance\_run()} км"*

*)*

*if self.needs\_repair():*

*status += " (Потребує ремонту)"*

*print(status)*

*self.log\_operation(f"Перевірено стан: {status}")*

*def run(self, distance):*

*"""*

*Виконує пробіг на задану відстань, якщо кросівки в належному стані.*

*:param distance: Відстань для пробігу в км.*

*"""*

*if self.needs\_repair():*

*print("Кросівки потребують ремонту. Спочатку відремонтуйте їх.")*

*return*

*if not self.is\_laced():*

*print("Потрібно зав'язати шнурки перед пробігом.")*

*return*

*self.add\_distance(distance)*

*print(f"Ви пробігли {distance} км.")*

*self.log\_operation(f"Пробігли {distance} км.")*

*if self.needs\_repair():*

*print("Кросівки потребують ремонту після пробігу 100 км.")*

*def log\_operation(self, operation):*

*"""*

*Записує проведену операцію в log-файл.*

*"""*

*# Тут буде код для запису в log-файл.*

*pass*

*ShoesOpertions.py*

*class ShoeOperations:*

*"""*

*Інтерфейс ShoeOperations, що визначає основні операції з взуттям.*

*"""*

*def wear(self):*

*pass*

*def remove(self):*

*pass*

*def clean(self):*

*pass*

*def repair(self):*

*pass*

*def polish(self):*

*pass*

*def lace\_up(self):*

*pass*

*def unlace(self):*

*pass*

*def check\_status(self):*

*pass*

***Main.py***

*from sneakers import Sneakers*

*def main():*

*"""*

*Точка входу програми.*

*"""*

*sneakers = Sneakers("Nike", 42.5, "Білий")*

*while True:*

*print("\nОберіть дію:")*

*if not sneakers.is\_worn():*

*print("1. Одягнути взуття")*

*else:*

*print("1. Зняти взуття")*

*if not sneakers.is\_laced():*

*print("2. Зав'язати шнурки")*

*else:*

*print("2. Розв'язати шнурки")*

*print("3. Очистити взуття")*

*print("4. Відремонтувати взуття")*

*print("5. Відполірувати взуття")*

*print("6. Перевірити стан взуття")*

*print("7. Пробігти")*

*print("8. Вийти")*

*choice = input("Ваш вибір: ")*

*if choice == "1":*

*if not sneakers.is\_worn():*

*sneakers.wear()*

*else:*

*sneakers.remove()*

*elif choice == "2":*

*if not sneakers.is\_laced():*

*sneakers.lace\_up()*

*else:*

*sneakers.unlace()*

*elif choice == "3":*

*sneakers.clean()*

*elif choice == "4":*

*sneakers.repair()*

*elif choice == "5":*

*sneakers.polish()*

*elif choice == "6":*

*sneakers.check\_status()*

*elif choice == "7":*

*if sneakers.is\_worn():*

*try:*

*distance = float(input("Введіть відстань для пробігу (км): "))*

*sneakers.run(distance)*

*except ValueError:*

*print("Помилка: введіть числове значення.")*

*else:*

*print("Спочатку одягніть взуття.")*

*elif choice == "8":*

*print("Програма завершена.")*

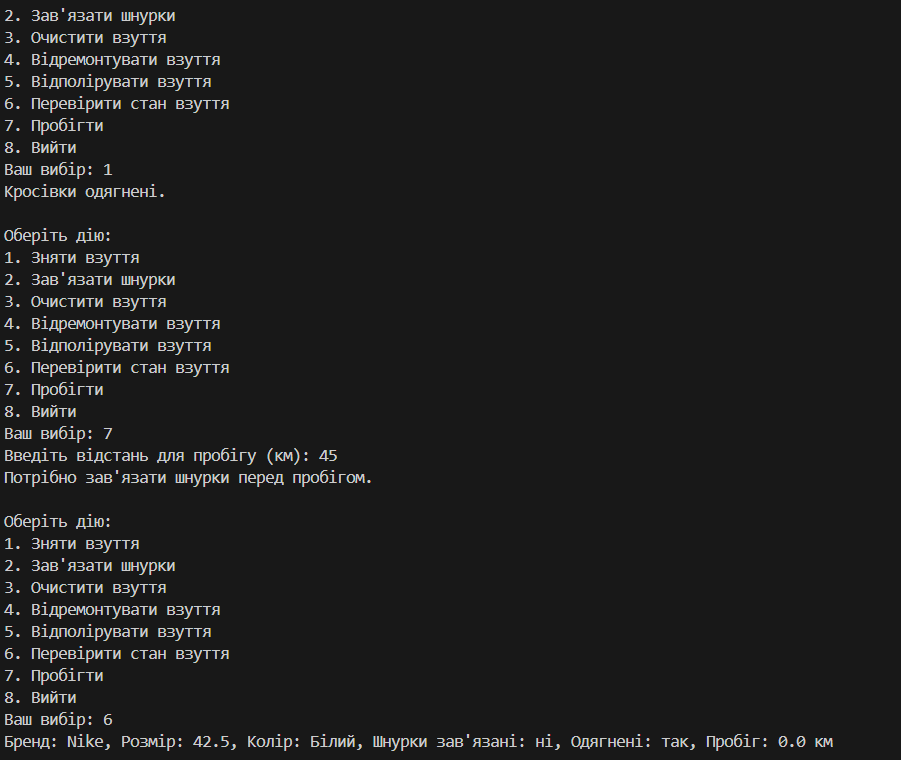
*break*

*else:*

*print("Невірний вибір, спробуйте ще раз.")*

*if \_\_name\_\_ == "\_\_main\_\_":*

*main()*



**Рис.1 Вивід результату у консоль**

**Висновок:** На лабораторній роботі я оволодів навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python.