Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра ІСМ



**Звіт**

до лабораторної роботи № 2

з дисципліни

​*Спеціалізовані мови програмування*

на тему:

“**Основи побудови об’єктно-орієнтованих додатків на Python** ”

Виконав студент РІ-31

**Гвоздь Северин**

Прийняв: Щербак С.С.

Львів – 2024

**Мета роботи:** Розробка консольного калькулятора в об’єктно орієнтованому стилі з використанням класів

**Завдання лабораторної роботи**

Завдання 1: Створення класу Calculator Створіть клас Calculator, який буде служити основою для додатка калькулятора. Завдання 2: Ініціалізація калькулятора Реалізуйте метод \_\_init\_\_ у класі Calculator для ініціалізації необхідних атрибутів або змінних. Завдання 3: Введення користувача Перемістіть функціональність введення користувача в метод у межах класу Calculator. Метод повинен приймати введення для двох чисел і оператора. Завдання 4: Перевірка оператора Реалізуйте метод у класі Calculator, щоб перевірити, чи введений оператор є дійсним (тобто одним із +, -, \*, /). Відобразіть повідомлення про помилку, якщо він не є дійсним. Завдання 5: Обчислення Створіть метод у класі Calculator, який виконує обчислення на основі введення користувача (наприклад, додавання, віднімання, множення, ділення). Завдання 6: Обробка помилок Реалізуйте обробку помилок у межах класу Calculator для обробки ділення на нуль або інших потенційних помилок. Відобразіть відповідні повідомлення про помилку. Завдання 7: Повторення обчислень Додайте метод до класу Calculator, щоб запитати користувача, чи він хоче виконати ще одне обчислення. Якщо так, дозвольте йому ввести нові числа і оператор. Якщо ні, вийдіть з програми. Завдання 8: Десяткові числа Модифікуйте клас Calculator для обробки десяткових чисел (плаваюча кома) для більш точних обчислень. Завдання 9: Додаткові операції Розширте клас Calculator, щоб підтримувати додаткові операції, такі як піднесення до степеня (^), квадратний корінь (√) та залишок від ділення (%). Завдання 10: Інтерфейс, зрозумілий для користувача Покращте інтерфейс користувача у межах класу Calculator, надавши чіткі запити, повідомлення та форматування виводу для зручності читання.

Реалізація програми:

from colorama import Fore, Style

from calculator import BasicCalculator, ScientificCalculator

def main():

    print("Виберіть тип калькулятора:")

    print("1. Базовий калькулятор")

    print("2. Науковий калькулятор")

    while True:

        choice = input("Введіть номер типу калькулятора (1-2): ")

        if choice == '1':

            calculator = BasicCalculator()

            break

        elif choice == '2':

            calculator = ScientificCalculator()

            break

        else:

            print("Невірний вибір. Спробуйте ще раз.")

    color\_map = {

        '1': Fore.WHITE,

        '2': Fore.RED,

        '3': Fore.GREEN,

        '4': Fore.YELLOW,

        '5': Fore.BLUE,

        '6': Fore.MAGENTA,

        '7': Fore.CYAN,

        '8': Fore.BLACK

    }

    while True:

        choice = input("Введіть номер кольору (1-8): ")

        try:

            if choice in color\_map:

                print(color\_map[choice] + "Вибрано колір тексту" + Style.RESET\_ALL)

                break

            else:

                raise ValueError("Невірний вибір. Введіть число від 1 до 8.")

        except ValueError as e:

            print(e)

    calculator.set\_decimal\_places()

    while True:

        num1 = float(input("Введіть перше число: "))

        operator = input("Введіть оператор (+, -, \*, /, ^, √, %): ")

        if operator != '√':

            num2 = float(input("Введіть друге число: "))

        else:

            num2 = None

        result = calculator.calculate(num1, num2, operator)

        print(f"Результат: {result:.{calculator.decimal\_places}f}")

        again = input("Виконати ще одне обчислення? (y/n): ")

        if again.lower() != 'y':

            break

    calculator.show\_history()

    print("Програма завершена.")

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

    main()

**Висновок**

Виконавши ці завдання, я перетворив консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проект допоміг мені вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.