МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Лабораторна робота №2

з курсу

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконала студентка

групи ІТ-21сп

**Нечайко П.В.**

Прийняв

**Щербак С.С.**

Львів - 2023

**Мета:** розробка консольного калькулятора в об’єктно орієнтованому стилі з використанням класів.

**План роботи**

**Завдання 1:** Створення класу Calculator.

Створити клас Calculator, який буде служити основою для додатка калькулятора.

**Завдання 2:** Ініціалізація калькулятора.

Реалізувати метод \_\_init\_\_ у класі Calculator для ініціалізації необхідних атрибутів або змінних.

**Завдання 3:** Введення користувача.

Перемістити функціональність введення користувача в метод у межах класу Calculator. Метод повинен приймати введення для двох чисел і оператора.

**Завдання 4:** Перевірка оператора.

Реалізувати метод у класі Calculator, щоб перевірити, чи введений оператор є дійсним (тобто одним із +, -, \*, /). Відобразити повідомлення про помилку, якщо він не є дійсним.

**Завдання 5:** Обчислення.

Створити метод у класі Calculator, який виконує обчислення на основі введення користувача (наприклад, додавання, віднімання, множення, ділення).

**Завдання 6:** Обробка помилок.

Реалізувати обробку помилок у межах класу Calculator для обробки ділення на нуль або інших потенційних помилок. Відобразити відповідні повідомлення про помилку.

**Завдання 7:** Повторення обчислень.

Додати метод до класу Calculator, щоб запитати користувача, чи він хоче виконати ще одне обчислення. Якщо так, дозволити йому ввести нові числа і оператор. Якщо ні, вийдіть з програми.

**Завдання 8:** Десяткові числа.

Модифікувати клас Calculator для обробки десяткових чисел (плаваюча кома) для більш точних обчислень.

**Завдання 9:** Додаткові операції.

Розширити клас Calculator, щоб підтримувати додаткові операції, такі як піднесення до степеня (^), квадратний корінь (√) та залишок від ділення (%).

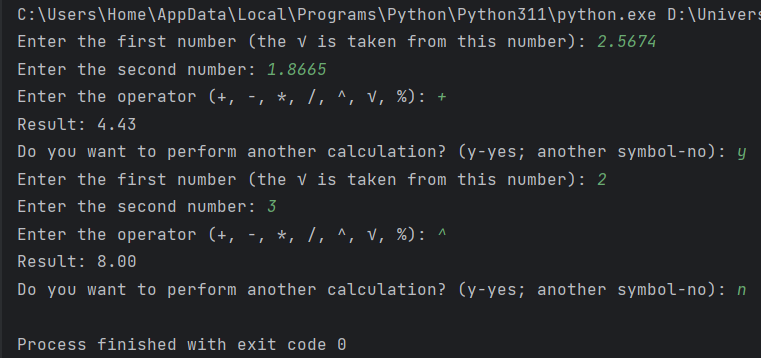
**Завдання 10:** Інтерфейс, зрозумілий для користувача.

Покращити інтерфейс користувача у межах класу Calculator, надавши чіткі запити, повідомлення та форматування виводу для зручності читання.

Код виконаних завдань представлено нижче.

import math  
  
# Task 1: Creating the Calculator class  
class Calculator:  
  
  
#Task 2: Initializing the calculator  
 def \_\_init\_\_(self):  
 self.result = None  
  
#Task3: User input  
#Task 10: User-friendly interface  
 def user\_input(self):  
#Task 8: Decimal numbers  
 number1 = float(input("Enter the first number (the √ is taken from this number): "))  
 number2 = float(input("Enter the second number: "))  
 operator = input("Enter the operator (+, -, \*, /, ^, √, %): ")  
 return number1, operator, number2  
  
#Task4: Operator check  
 def validate\_operator(self, operator):  
 valid\_operators = ['+', '-', '\*', '/', '^', '√', '%']  
 if operator not in valid\_operators:  
 print("Error: Invalid operator entered.")  
 return False  
 return True  
  
#Task5: Calculation  
 def calculate(self, number1, operator, number2):  
 if operator == '+':  
 self.result = number1 + number2  
 elif operator == '-':  
 self.result = number1 - number2  
 elif operator == '\*':  
 self.result = number1 \* number2  
  
#Task6: Error handling  
 elif operator == '/':  
 if number2 == 0:  
 print("Error: Division by zero is not possible.")  
 return  
 self.result = number1 / number2  
  
#Task 9: Additional operations  
 elif operator == '^':  
 self.result = number1 \*\* number2  
 elif operator == '√':  
self.result = math.sqrt(number1)  
 elif operator == '%':  
 self.result = number1 % number2  
  
#Task7: Repetition of calculations  
 def run\_calculator(self):  
 while True:  
 number1, operator, number2 = self.user\_input()  
 if not self.validate\_operator(operator):  
 continue  
 self.calculate(number1, operator, number2)  
 print(f"Result: {self.result:.2f}")  
 choice = input("Do you want to perform another calculation? (y-yes; another symbol-no): ").lower()  
 if choice != 'y':  
 break  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 calc = Calculator()  
 calc.run\_calculator()

На рисунку 1 зображено результат виконання програми.



*Рис.1 Робота калькулятора*

**Висновок:** під час виконання лабораторної роботи я навчилася перетворювати консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проєкт допоміг мені вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.