МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Лабораторна робота №2

з курсу

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконав студент

групи ІТ-21сп

Кузьо О.Р.

Прийняв

Щербак С.С.

Львів – 2023

Лабораторна робота №2

Мета: Розробка консольноо калькулятора в обʼєктно орнієнтованому стилі з використанням класів

**Хід роботи**

Завдання 1: Створення класу Calculator

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

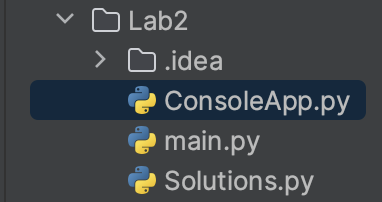


Рис.1. створення класу для основи додатка калькулятора

Завдання 2: Ініціалізація калькулятора

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

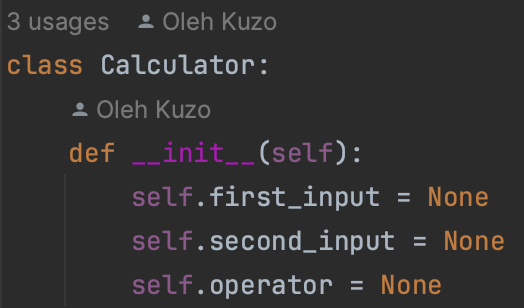


Рис.2. реалізація методу \_\_init\_\_.py

Завдання 3: Введення користувача

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

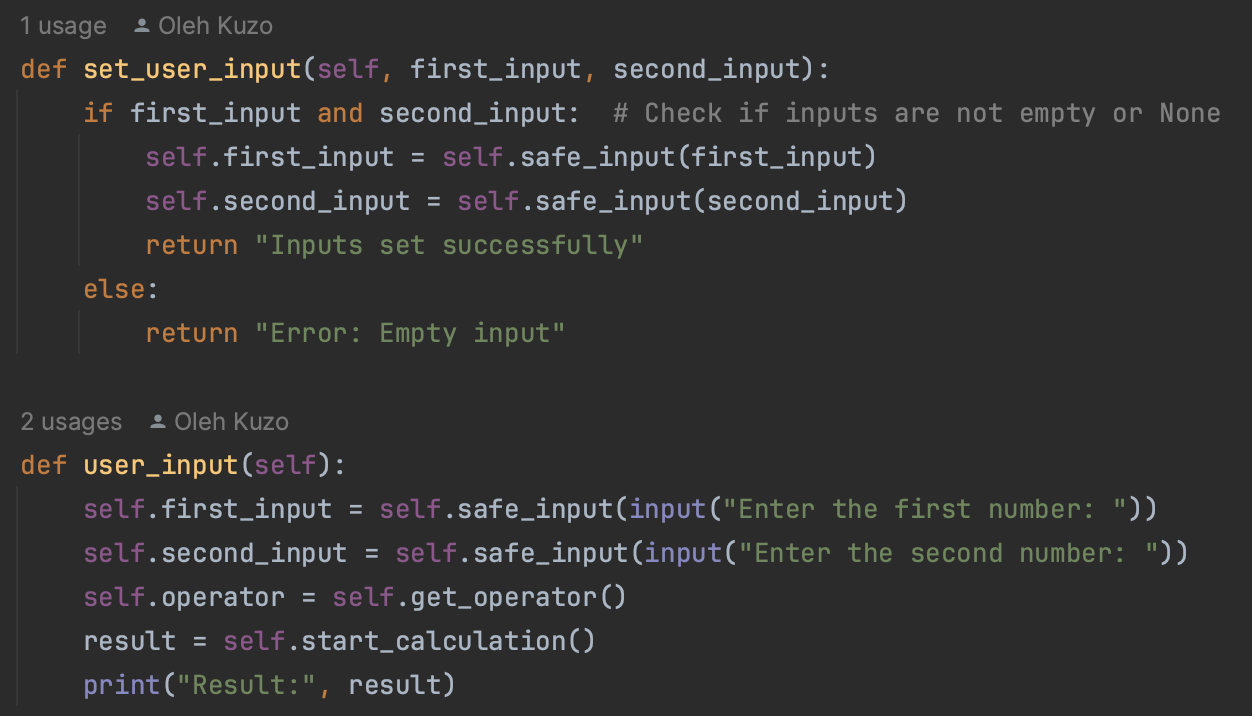


Рис.3. функціональність введення. Користувача.

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.

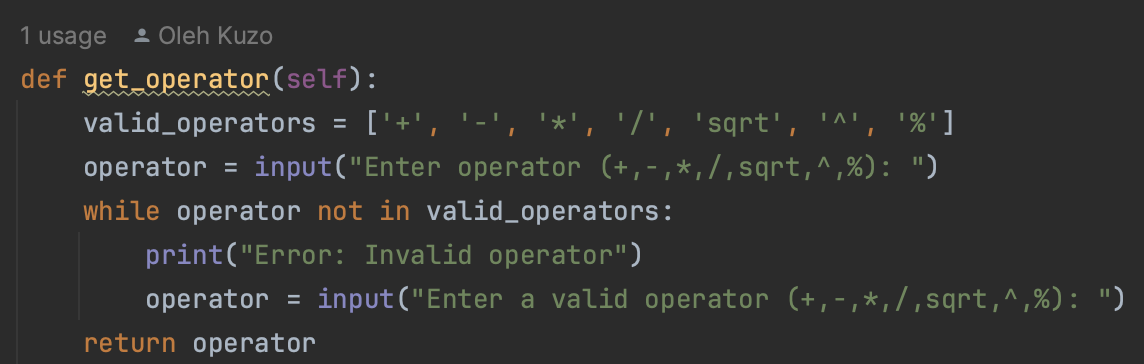


Рис.4. метод перевірки оператора

Завдання 5: Обчислення

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

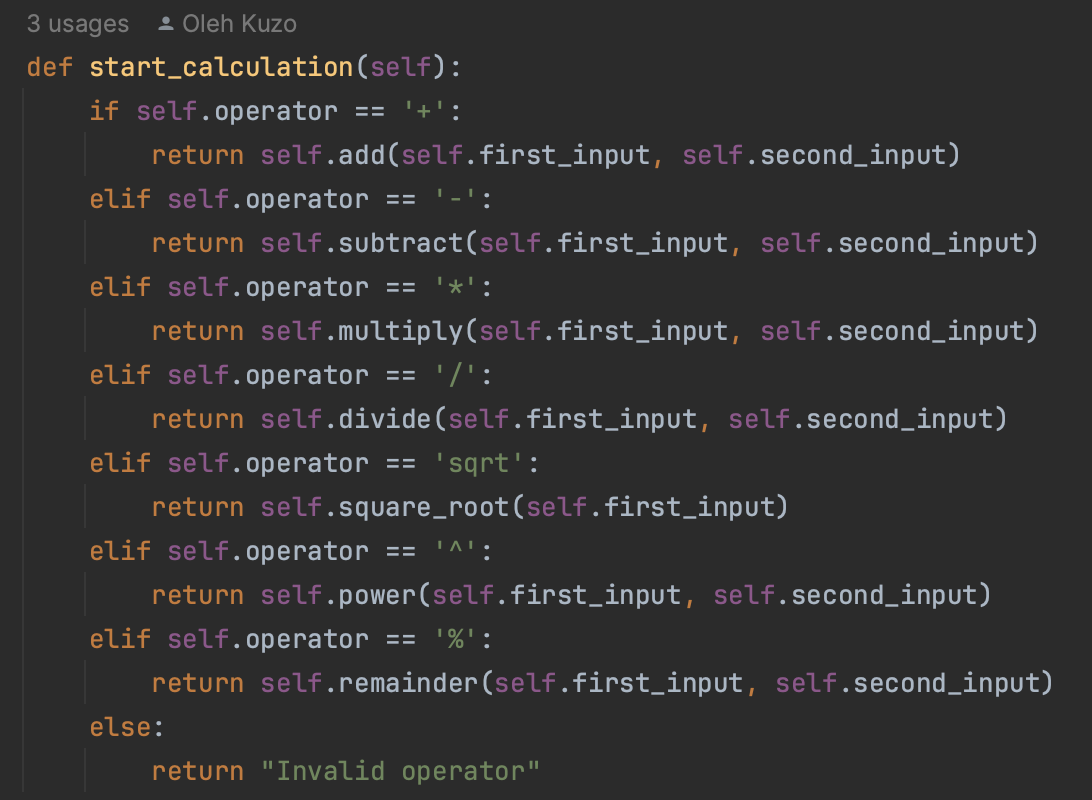


Рис.5. метод виконання обчислення на основі введення користувача

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.

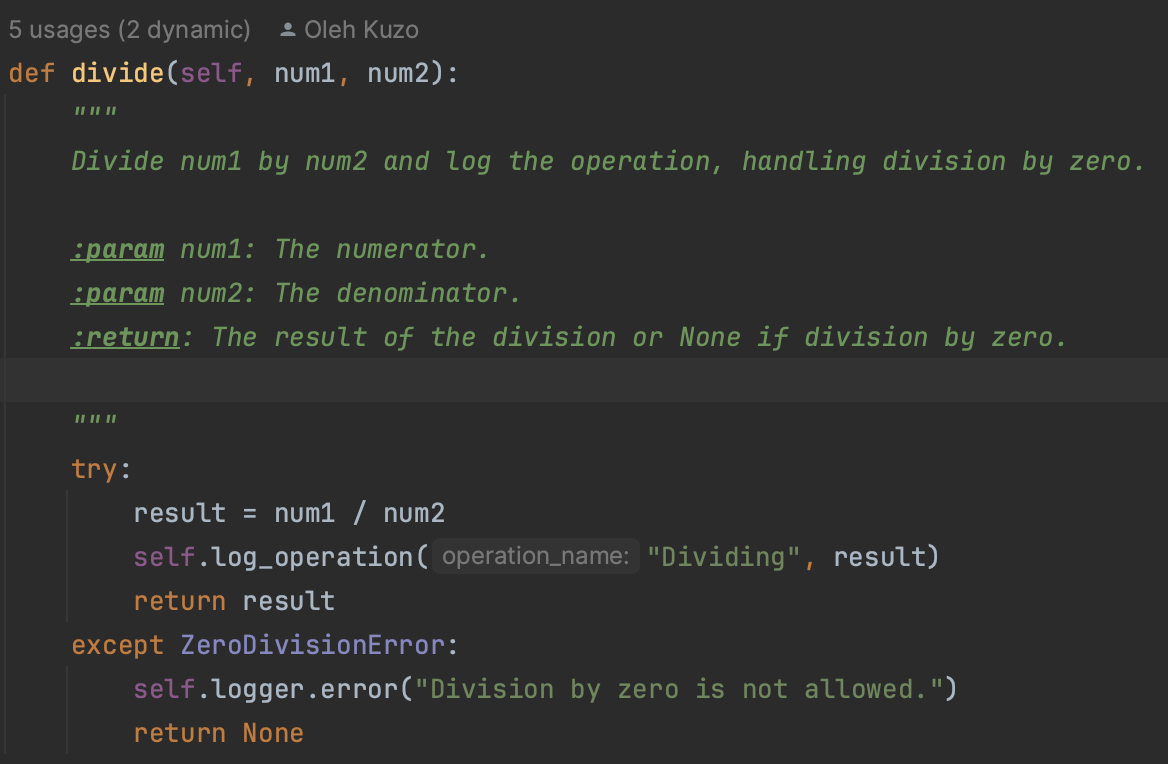


Рис.6. обробник помилок ділення на нуль

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.

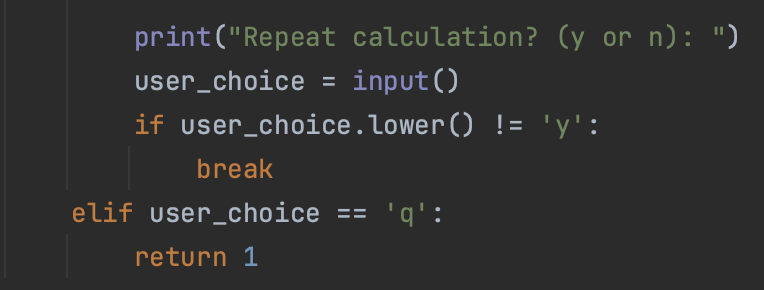


Рис.7. метод повторення обчислень

Завдання 8: Десяткові числа

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

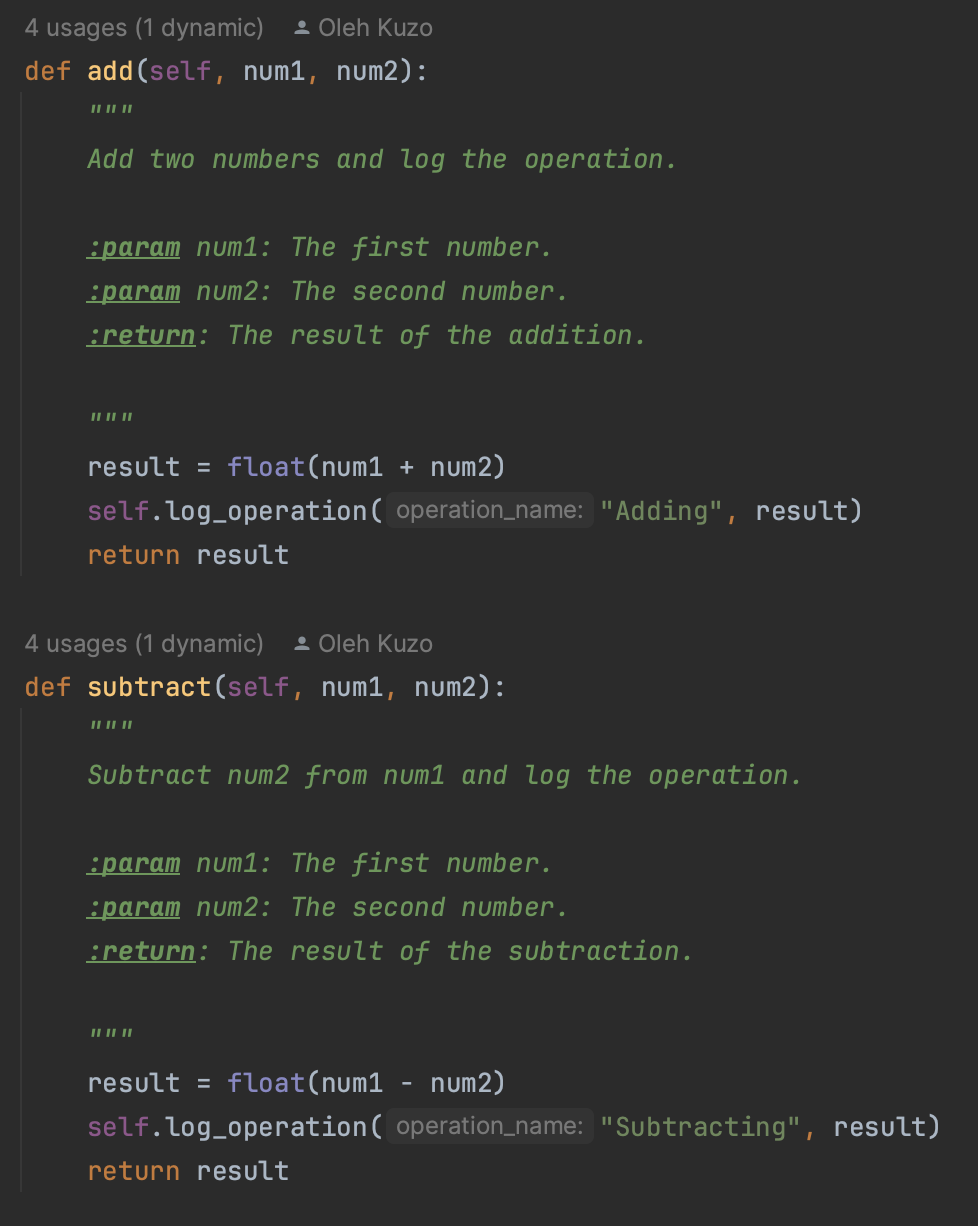


Рис.8. модифікація методів розрахунків для використання десяткових чисел

Завдання 9: Додаткові операції

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

def square\_root(self, num):

if num < 0:

self.logger.error("Square root of a negative number is undefined.")

return None

result = float(sqrt(num))

self.log\_operation("Calculating square root", result)

return result

def power(self, num1, num2):

result = float(pow(num1, num2))

self.log\_operation("Power", result)

return result

def remainder(self, num1, num2):

result = num1 % num2

self.log\_operation("Calculating remainder", result)

return result

Завдання 10: Інтерфейс, зрозумілий для користувача

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

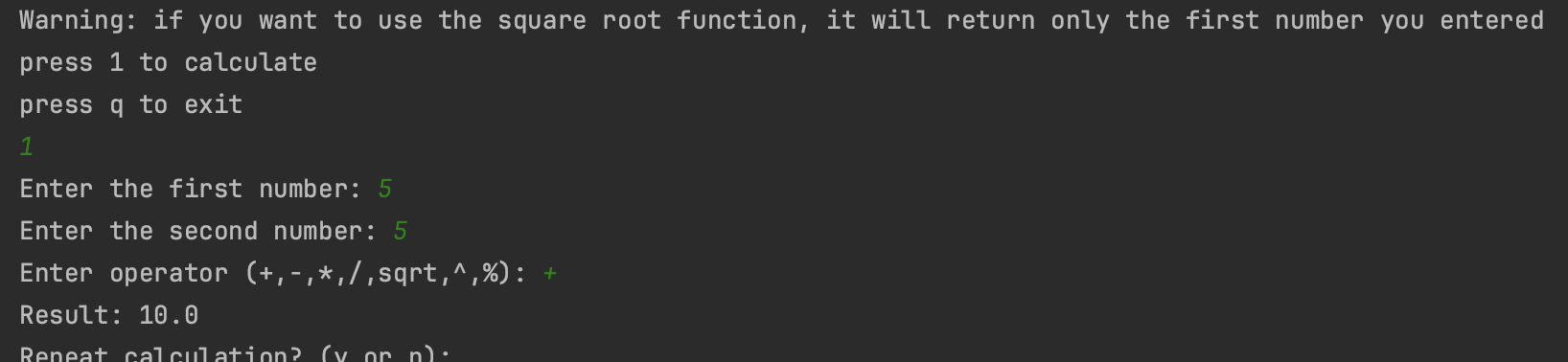


Рис.10. інтерфейс користувача

**Висновок:** під час опрацювання лабораторної роботи було розроблено консольний калькулятор в обʼєктно орнієнтованому стилі з використанням класів.