ЗВІТ

про виконання лабораторної роботи № 2

«Основи побудови об’єктно-орієнтованих додатків на Python»

з дисципліни

«Спеціалізовані мови програмування»

студента групи РІ-31

Висоцького Володимира Володимировича

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

**Умова завдання:**

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.

**Текст програми:**

class Calculator:

    def \_\_init\_\_(self):

        self.memory\_value = global\_value.memory\_value

    def get\_input(self):

        while True:

            try:

                user\_input = input('Input first operand (or MR for memory recall): ').upper()

                if user\_input == 'MR':

                    first\_operand = self.memory\_value

                    print(f"Recalled from memory: {first\_operand}")

                else:

                    first\_operand = float(user\_input)

                operator = input("Input operator (+, -, \*, /, ^, %, sq): ").strip()

                if not self.validate\_operator(operator):

                    print("Invalid operator. Try again.")

                    continue

                while True:

                    user\_input = input('Input second operand (or MR for memory recall): ').upper()

                    if user\_input == 'MR':

                        second\_operand = self.memory\_value

                        print(f"Recalled from memory: {second\_operand}")

                    else:

                        second\_operand = float(user\_input)

                    # Якщо обраний оператор - це ділення, перевіряємо, чи другий операнд не дорівнює нулю

                    if operator == '/' and second\_operand == 0:

                        print("Error: Division by zero is not allowed. Please enter a non-zero second operand.")

                    else:

                        break  # Вихід з циклу, якщо другий операнд правильний

                return first\_operand, second\_operand, operator

            except ValueError:

                print("Invalid number format. Try again.")

    def validate\_operator(self, operator):

            valid\_operators = ['+', '-', '\*', '/', '^', '%', 'sq']

            return operator in valid\_operators

    def calculate(self, first\_operand, second\_operand, operator):

        try:

            match operator:

                case '+':

                    result = functions.addition(first\_operand, second\_operand)

                case '-':

                    result = functions.subtraction(first\_operand, second\_operand)

                case '/':

                    if second\_operand == 0 :

                        raise ZeroDivisionError("Error: Division by zero is not possible.")

                    result = functions.division(first\_operand, second\_operand)

                case '\*':

                    result = functions.multiplication(first\_operand, second\_operand)

                case '^':

                    result = functions.power(first\_operand, second\_operand)

                case 'sq':

                    if first\_operand < 0:

                            raise ValueError("Error: Negative number under the root.")

                    result = functions.square\_root(first\_operand, second\_operand)

                case '%':

                    result = functions.modulus(first\_operand, second\_operand)

            functions.log\_history(first\_operand, operator, second\_operand, round(result,global\_value.round\_number))

            return round(result,global\_value.round\_number)

        except (ZeroDivisionError, ValueError) as e:

            print(e)

    def run(self):

        while True:

            first\_operand, second\_operand , operator = self.get\_input()

            result = self.calculate(first\_operand, second\_operand, operator)

            if result is not None:

                print(f"Результат: {result}")

            choice\_memory = input('Would you like to store result in memory (MS), add to memory (M+), clear memory (MC), or skip? ').upper()

            match choice\_memory:

                case 'MS':

                    memory\_value = result

                    print(f"Stored {result} in memory.")

                case 'M+':

                    memory\_value += result

                    print(f"Added {result} to memory. New memory value: {memory\_value}.")

                case 'MC':

                    memory\_value = 0

                    print("Memory cleared.")

            if input("Do you want to view history? (yes/no): ").strip().lower() == 'yes':

                print(functions.show\_history())

            if input('Do you want to make another calculation? (yes/no): ').lower() !='yes':

                break

**Висновки:** виконуючи лабораторну робото я перетворив консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python..