Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторної роботи № 2

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

з дисципліни **“** **Спеціалізовані мови програмування”**

Виконав:

студент групи ІТ-32

ОЛЕНЮК О. М.

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

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.

**Код програми:**

calculator.py

import math  
  
  
class Calculator:  
 def \_\_init\_\_(self):  
 self.calculations\_history = [] # List to store calculation history  
 self.decimal\_places = 2 # Default decimal places  
  
 def view\_history(self):  
 if not self.calculations\_history:  
 return "The history is empty"  
 else:  
 history = "Calculation history:\n"  
 for item in self.calculations\_history:  
 history += " ".join(map(str, item)) + "\n"  
 return history  
  
 def view\_settings(self):  
 return f"Settings:\nDecimal places are {self.decimal\_places}"  
  
 def change\_decimal\_places(self, value):  
 if value <= 0:  
 raise ValueError("Decimal places must be greater than zero")  
 self.decimal\_places = value  
  
 def calculate(self, operator, num1, num2):  
 try:  
 if operator == "+":  
 result = num1 + num2  
 elif operator == "-":  
 result = num1 - num2  
 elif operator == "\*":  
 result = num1 \* num2  
 elif operator == "/":  
 if num2 == 0:  
 raise ValueError("Division by zero is not allowed")  
 result = num1 / num2  
 elif operator == "^":  
 result = num1 \*\* num2  
 elif operator == "%":  
 result = num1 % num2  
 elif operator == "√":  
 if num1 < 0:  
 raise ValueError("Square root of a negative number is not allowed")  
 result = math.sqrt(num1)  
 else:  
 raise ValueError("Invalid operator")  
  
 calculation = (num1, operator, num2, "=", round(result, self.decimal\_places))  
 self.calculations\_history.append(calculation)  
 return calculation[-1]  
 except (ValueError, ZeroDivisionError) as e:  
 return str(e)

console\_output.py

class ConsoleOutput:  
 @staticmethod  
 def show\_message(message):  
 print(message)  
  
 @staticmethod  
 def get\_user\_input(prompt):  
 return input(prompt)

runner.py

from calculator import Calculator  
from console\_output import ConsoleOutput  
  
  
def main():  
 calculator = Calculator()  
 output = ConsoleOutput()  
  
 while True:  
 # Display available operations  
 output.show\_message("""  
 Available operations:  
 + : Sum of numbers  
 - : Difference of numbers  
 \* : Product of numbers  
 / : Division of numbers  
 ^ : Exponentiation of a number  
 % : Remainder from division  
 √ : Calculation of the square root  
 0 : Exit  
 1 : View history  
 2 : Open settings  
 """)  
  
 input\_action = output.get\_user\_input("Enter the action you want to perform: ")  
  
 if input\_action == "0":  
 break  
 elif input\_action in ("+", "-", "\*", "/", "^", "%"):  
 first\_number = float(output.get\_user\_input("Enter the first number: "))  
 second\_number = float(output.get\_user\_input("Enter the second number: "))  
 result = calculator.calculate(input\_action, first\_number, second\_number)  
 output.show\_message(f"Result: {result}")  
 elif input\_action == "√":  
 number = float(output.get\_user\_input("Enter a number: "))  
 result = calculator.calculate(input\_action, number, 0) # Pass 0 as a placeholder for the second number  
 output.show\_message(f"Result: {result}")  
 elif input\_action == "1":  
 history = calculator.view\_history()  
 output.show\_message(history)  
 elif input\_action == "2":  
 output.show\_message(calculator.view\_settings())  
 inner\_input\_value = output.get\_user\_input("""Available options:  
 0: Exit"  
 1: Change decimal places   
 2: Clear history   
 Your option is """)  
 if inner\_input\_value == "0":  
 pass  
 elif inner\_input\_value == "1":  
 new\_value = int(output.get\_user\_input("Enter a new value for decimal places: "))  
 calculator.change\_decimal\_places(new\_value)  
 output.show\_message(f"Decimal places set to {new\_value}")  
 elif inner\_input\_value == "2":  
 calculator.calculations\_history.clear()  
 output.show\_message("History cleared")  
 else:  
 output.show\_message("Invalid option")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Результат програми:**

На рис.1, 2, 3, 4, 5 зобрадений результат виконання операцій у програмі калькулятор.

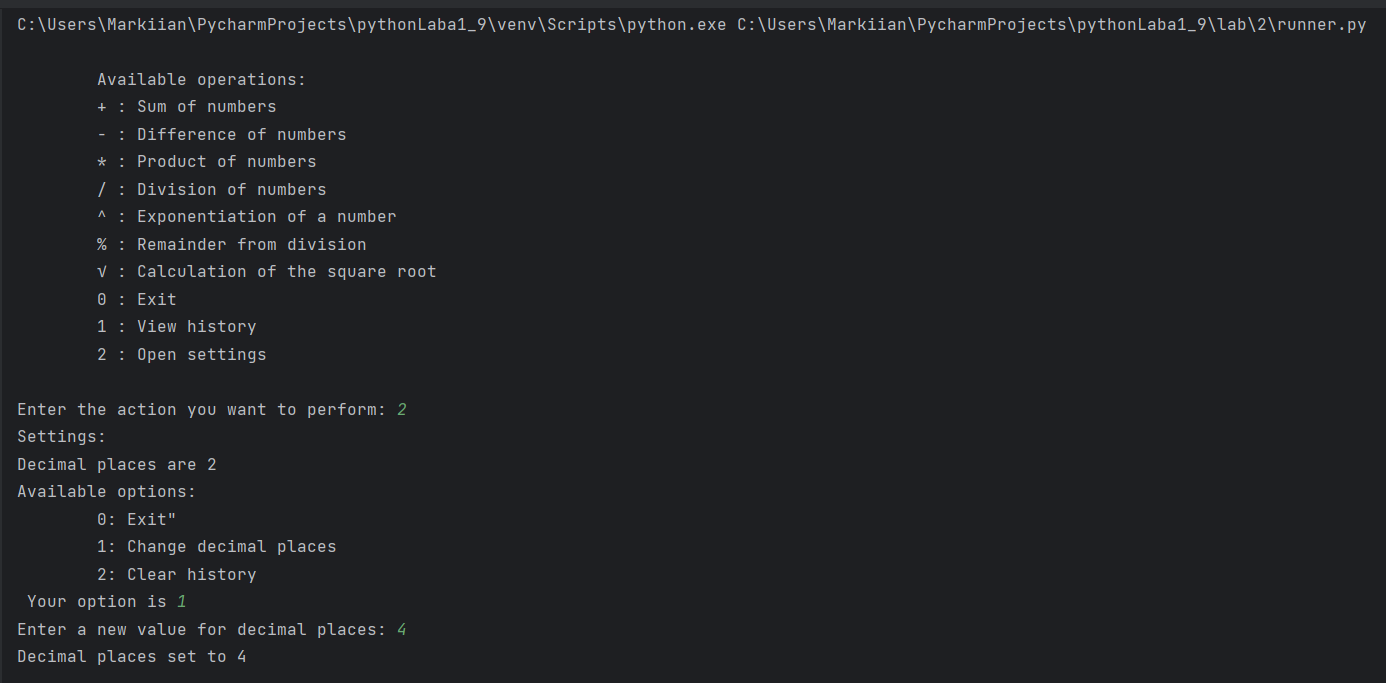


Рис.1

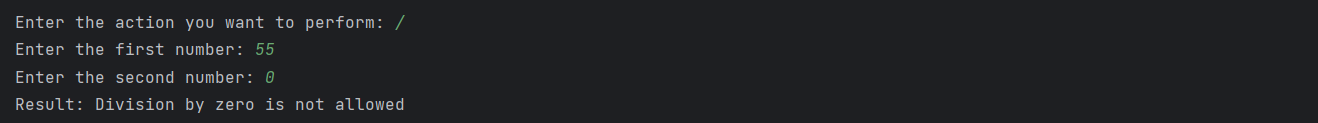


Рис.2

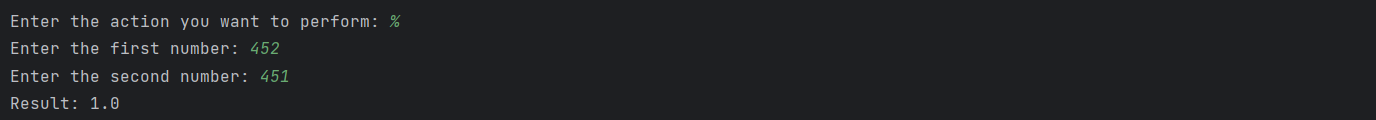


Рис.3

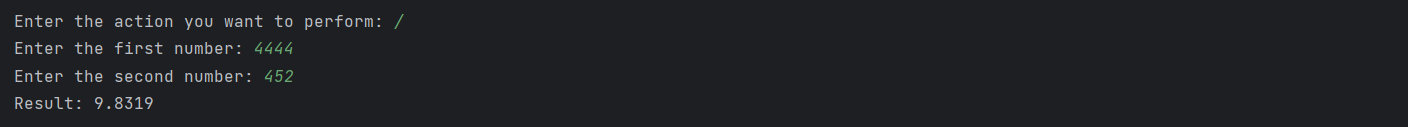


Рис.4



Рис.5

**Посилання на репозиторій**: <https://github.com/Oleksandr2004Oleniuk/SMP.git>

**Висновок**. у цій лабораторній роботі 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.