МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра ІСМ

******

Звіт

до лабораторної роботи №2

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

З дисципліни “Спеціалізовані мови програмування”

*Виконав:*

*ст. гр. ІТ-31*

*Іштван Комоній*

*Прийняв:*

*Щербак С. С.*

*Львів - 2023*

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

**Хід роботи**

**Програмний код:**

/math\_operations/math\_operation

class MathOperationError(Exception):

pass

class MathOperation:

def \_\_init\_\_(self, num1, num2):

self.num1 = num1

self.num2 = num2

def execute(self):

pass

def error\_handler(self, error\_text):

raise MathOperationError(error\_text)

/math\_operations/math\_operations

import math

from math\_operations.math\_operation import MathOperation

class Addition(MathOperation):

def execute(self):

return self.num1 + self.num2

def error\_handler(self, error):

raise ValueError(f"Error during addition: {error}")

class Subtraction(MathOperation):

def execute(self):

return self.num1 - self.num2

def error\_handler(self, error):

raise ValueError(f"Error during subtraction: {error}")

class Multiplication(MathOperation):

def execute(self):

return self.num1 \* self.num2

def error\_handler(self, error):

raise ValueError(f"Error during multiplication: {error}")

class Division(MathOperation):

def execute(self):

if self.num2 == 0:

self.error\_handler('Division by zero is not allowed')

return self.num1 / self.num2

def error\_handler(self, error):

raise ValueError(f"Error during division: {error}")

class Power(MathOperation):

def execute(self):

return self.num1 \*\* self.num2

def error\_handler(self, error):

raise ValueError(f"Error during exponentiation: {error}")

class SquareRoot(MathOperation):

def execute(self):

if self.num1 < 0:

self.error\_handler(

'Square root of a negative number is not allowed')

return math.sqrt(self.num1)

def error\_handler(self, error):

raise ValueError(f"Error during square root: {error}")

class Modulo(MathOperation):

def execute(self):

if self.num2 == 0:

self.error\_handler('Division by zero is not allowed')

return self.num1 % self.num2

def error\_handler(self, error):

raise ValueError(f"Error during modulo operation: {error}")

/calculator/calculator

from math\_operations.math\_operations import Addition, Subtraction, Multiplication, Division, Power, SquareRoot, Modulo

class Calculator:

def \_\_init\_\_(self):

self.result = 0.0

def get\_user\_choice(self):

return input('Operation: [ +, -, \*, /, ^, √, % ]: ')

def get\_input\_num(self, prompt):

while True:

try:

num = float(input(prompt))

return num

except ValueError:

print('Please enter a valid number.')

def get\_input\_data(self):

choice = self.get\_user\_choice()

num1 = self.get\_input\_num('Enter the first number: ')

num2 = None

if choice != '√':

num2 = self.get\_input\_num('Enter the second number: ')

return choice, num1, num2

def perform\_operation(self, choice, num1, num2):

operation = None

if choice == '+':

operation = Addition(num1, num2)

elif choice == '-':

operation = Subtraction(num1, num2)

elif choice == '\*':

operation = Multiplication(num1, num2)

elif choice == '/':

operation = Division(num1, num2)

elif choice == '^':

operation = Power(num1, num2)

elif choice == '√':

operation = SquareRoot(num1, None)

elif choice == '%':

operation = Modulo(num1, num2)

if operation:

try:

self.result = operation.execute()

except ValueError as e:

print(e)

else:

print('Invalid choice. Please try again.')

@staticmethod

def ask\_user\_to\_continue():

result = input('Do you want to continue? (y/n) ')

return result == 'y'

def user\_interface(self):

while True:

try:

choice, num1, num2 = self.get\_input\_data()

self.perform\_operation(choice, num1, num2)

print(f'Result: {self.result}')

except:

print('Something went wrong. Please try again.')

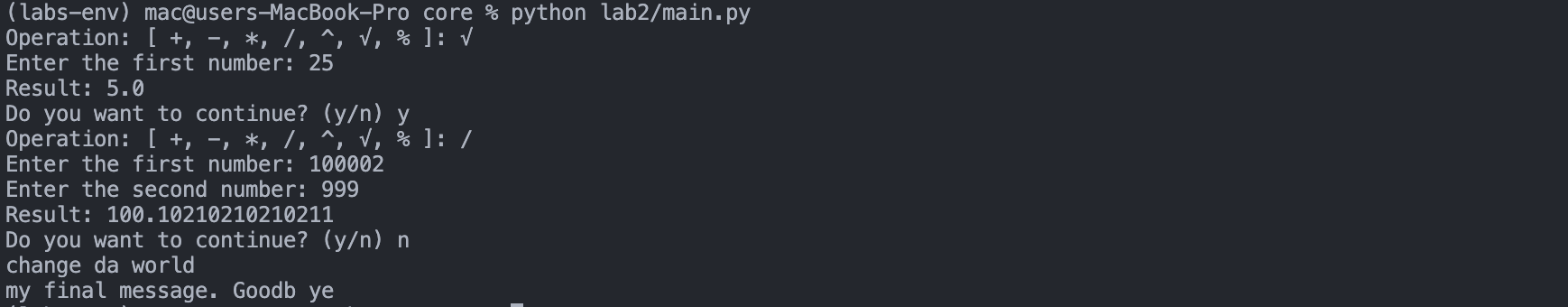
to\_continue = self.ask\_user\_to\_continue()

if not to\_continue:

print('change da world\nmy final message. Goodb ye')

break

Результат виконання програми:



**Висновок:** Під час виконання даної лабораторної роботи було створено консольний калькулятор в об’єктно орієнтованому стилі з використанням класів.