мІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

нАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «лЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗВІТ

для лабораторної роботи № 2

з дисципліни

«Спеціалізовані мови програмування»

Виконав:

студент гр. ІТ-32

Сташкевич Владислав

Прийняв:

доц. каф. ІСМ

Щербак С.С.

Львів-2023

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

**Хід виконання:**

**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:**

**main.py**

from lab2.calculator import Calculator

calc = Calculator(0, 0, "+")

calc.start()

**calculator.py**

class Calculator:

def \_\_init\_\_(self, number1: float, number2: float, operator: object):

self.number1 = number1

self.number2 = number2

self.operator = operator

def input(self):

self.number1 = float(input("enter first number: "))

self.number2 = float(input("enter second number: "))

self.operator = input("enter operation to perform: ")

if not self.verify():

self.input()

def check\_operator(self):

return (self.operator == "+" or

self.operator == "-" or

self.operator == "\*" or

self.operator == "/" or

self.operator == "^" or

self.operator == "%")

def calculate(self) -> float:

num1 = float(self.number1)

num2 = float(self.number2)

if self.operator == "+":

return num1 + num2

elif self.operator == "-":

return num1 - num2

elif self.operator == "\*":

return num1 \* num2

elif self.operator == "/":

if num2 != 0:

return num1 / num2

elif self.operator == "^":

return pow(num1, num2)

elif self.operator == "%":

return num1 % num2

def verify(self):

if not self.check\_operator():

print("you've entered the wrong operator, please try again.")

return False

if self.operator == "/" and float(self.number2) == 0:

print("number cannot be divided by zero, please try again.")

return False

return True

def output(self, result: float):

print(self.number1, self.operator, self.number2, "=", result)

def ask\_to\_repeat(self):

choice = input("For one more calculation, type Y: ")

if choice == "Y" or choice == "y":

self.start()

def start(self):

self.input()

result = self.calculate()

self.output(result)

self.ask\_to\_repeat()

**Посилання на GitHub-репозиторій:** <https://github.com/VladyslavStashkevych/SMP>

**Висновки:** Виконавши ці завдання, 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.