МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж



Лабораторна робота №2

з дисципліни Спеціалізовані мови програмування

на тему

Основи побудови об’єктно-орієнтованих додатків на Python

Виконав:

студент групи РІ-21сп

Данило Вергун

Львів – 2024

**Мета виконання лабораторної роботи:** 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.

**Текст програмної реалізації:  
calculator.py:**import math

from ..dal.memory import Memory

class Calculator:

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

self.memory = Memory()

self.history = []

self.decimal\_places = 2

def perform\_calculation(self, num1, num2, operator):

match operator:

case '+':

return num1 + num2

case '-':

return num1 - num2

case '\*':

return num1 \* num2

case '/':

if num2 == 0:

raise ValueError("Ділення на нуль неможливе")

return num1 / num2

case '^':

return num1 \*\* num2

case '√':

if num1 < 0:

raise ValueError("Неможливо обчислити квадратний корінь від від'ємного числа")

return math.sqrt(num1)

case '%':

return num1 % num2

case \_:

raise ValueError("Невідомий оператор")

def add\_to\_history(self, expression, result):

self.history.append(f"{expression} = {result}")

def show\_history(self):

return self.history

def set\_decimal\_places(self, places):

self.decimal\_places = places

**Результати тестування:**



Рис. 1. Результати віднімання



Рис. 2. Результати ділення



Рис. 3. Виведення історії

**Висновки:** На цій лабораторній роботі я навчився розробляти консольний калькулятор в об’єктно орієнтованому стилі з використанням класів