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

Виконала:

ст. гр. РІ-31

Оксана ЛЕСЮК

Прийняв:

Сергій Сергійович ЩЕРБАК

Львів-2024

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

**Завдання:**

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.

Виконавши ці завдання, ви перетворите консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проект допоможе вам вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.

**Хід роботи**

class Operation(ABC):

"""Abstract base class for mathematical operations."""

@abstractmethod

def execute(self, a: float, b: Optional[float] = None) -> float:

pass

class Multiplication(Operation):

def execute(self, a: float, b: float) -> float:

return a \* b

class Division(Operation):

def execute(self, a: float, b: float) -> float:

if b == 0:

raise ZeroDivisionError("Division by zero is not allowed.")

return a / b

class Addition(Operation):

def execute(self, a: float, b: float) -> float:

return a + b

class Power(Operation):

def execute(self, a: float, b: float) -> float:

return a \*\* b

class SquareRoot(Operation):

def execute(self, a: float, b: Optional[float] = None) -> float:

if a < 0:

raise ValueError("Square root of negative number is not allowed.")

return math.sqrt(a)

class Subtraction(Operation):

def execute(self, a: float, b: float) -> float:

return a - b

class InputHandler(ABC):

"""Abstract base class for handling input."""

@abstractmethod

def get\_numbers(self):

pass

@abstractmethod

def get\_operator(self):

pass

class CLIInputHandler(InputHandler):

"""Handles input from the command line interface."""

VALID\_OPERATORS = ['+', '-', '\*', '/', '^', '√', '%']

def get\_numbers(self):

"""Get numbers from user."""

try:

a = float(input("Enter the first number: "))

b = None

operator = self.get\_operator()

if operator not in ('√',):

b = float(input("Enter the second number: "))

return a, operator, b

except ValueError:

print("Invalid input. Please enter numeric values.")

return self.get\_numbers()

def get\_operator(self):

"""Get operator from user."""

operator = input(f"Enter an operator ({', '.join(self.VALID\_OPERATORS)}): ")

if operator not in self.VALID\_OPERATORS:

print("Invalid operator. Please try again.")

return self.get\_operator()

return operator

**Посилання на гіт-репозиторій:**

https://github.com/lesiukoksana/SMP-labs

**Висновок:**

У рамках лабораторної роботи було реалізовано об’єктно-орієнтований консольний калькулятор. Проєкт розділено на класи, що відповідають за різні аспекти функціональності, включаючи виконання математичних операцій, обробку вводу користувача, та основний алгоритм роботи калькулятора.