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:

ст. гр. РІ-21сп

Костецький І.С.

Прийняв:

Шербак С.С.

Львів-2024

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

**Завдання**

**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.

**Виконання роботи**

**Структура файлів:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Файли було поділено та переміщено у директорію Shared для спільного використання у лабораторній №6

**Вміст файлу cli.py:**

import sys

import os

sys.path.append(os.path.abspath(os.path.join(os.path.dirname(file), "..")))

sys.path.append(os.path.abspath(os.path.join(os.path.dirname(file), "..", "..")))

from Shared.Classes.Calculator import Calculator

def user\_input\_interface():

calc = Calculator()

while True:

print("\nМеню:")

print("1. Провести обчислення")

print("2. Вивести історію обчислень")

print("3. Записати число у памʼять")

print("4. Вивести число з памʼяті")

print("5. Поміняти налаштування (кількість десяткових знаків)")

print("6. Вийти з калькулятора")

choice = input("Оберіть опцію (1-6): ")

if choice == '1':

try:

calc.calculator()

except ValueError as e:

print(e)

except TypeError as e:

print(e)

except ZeroDivisionError as e:

print(e)

elif choice == '2':

try:

calc.history\_manager.display\_history()

except FileNotFoundError as e:

print(e)

elif choice == '3':

try:

custom\_value = float(input("Введіть число для збереження в пам'ять: "))

calc.input\_handler.memory.set\_memory(custom\_value)

print(f"Число {custom\_value} збережене в пам'ять (M).")

except ValueError as e:

print(e)

elif choice == '4':

print(f"Число збережене у пам'яті: {calc.input\_handler.memory.get\_memory()}")

elif choice == '5':

try:

calc.settings.change\_settings()

except ValueError as e:

print(e)

except TypeError as e:

print(e)

elif choice == '6':

print("Вихід з калькулятора.")

break

else:

print("Невірний вибір. Спробуйте ще раз.")

Результат виконання: а саме виконання обчислення на рисунку 1 та перегляд історії обчислень на рисунку 2.

A screenshot of a computer program

Description automatically generated

Рисунок 1 – Обчислення

A screen shot of a computer

Description automatically generated

Рисунок 2 – Перегляд історії калькулятора

**Висновок:** У ході виконання ЛР я перетворив консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Калькулятор виконує обчислення, зберігає результат у пам’яті та має зручний інтерфейс.