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

Шептицький Р.М

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.**

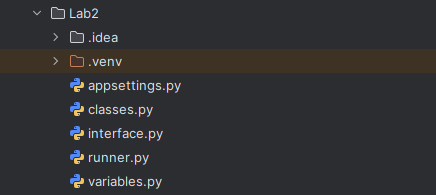
Розширте клас Calculator, щоб підтримувати додаткові операції, такі як піднесення до степеня (^), квадратний корінь (√) та залишок від ділення (%).

**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:**



**Вміст файлу interface.py:**

from Lab2.classes import Calculator  
  
def main():  
 try:  
 calc = Calculator()  
 except TypeError as e:  
 print(e)  
 exit(1)  
  
 while True:  
 print("\nГоловне меню:")  
 print("1. Виконати обчислення")  
 print("2. Переглянути історію")  
 print("3. Отримати результат з пам'яті")  
 print("4. Очистити результат з пам'яті")  
 print("5. Налаштувати кількість знаків після коми")  
 print("6. Вийти")  
  
 choice = input("Оберіть дію: ")  
  
 match choice:  
 case '1':  
 try:  
 calc.calculate()  
 except ZeroDivisionError as e:  
 print(e)  
 except ValueError as e:  
 print(e)  
 case '2':  
 calc.history.show\_history()  
 case '3':  
 print(f"Збережене значення з пам'яті: {calc.memory.get\_memory()}")  
 case '4':  
 calc.memory.clear\_memory()  
 print("Збережене значення очищено")  
 case '5':  
 try:  
 calc.settings.set\_decimal\_places()  
 except ValueError as e:  
 print(e)  
 except TypeError as e:  
 print(e)  
 case '6':  
 print("Вихід із програми.")  
 break  
 case \_:  
 print("Невірний вибір. Спробуйте ще раз.")

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

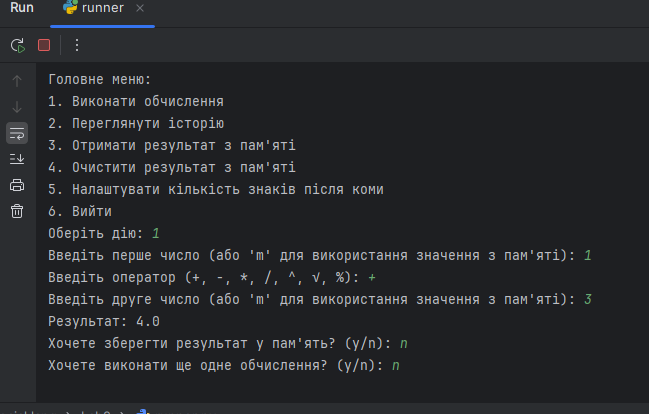


Рисунок 1 – Обчислення

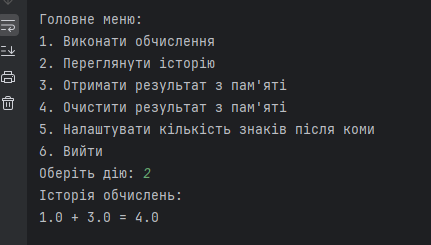


Рисунок 2 – Перегляд історії калькулятора

**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.