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. ІТ-32

Ольга ЧИГИРИК

Прийняв

доцент каф. ІСМ:

Сергій ЩЕРБАК

|  |  |
| --- | --- |
| **Балів** | **Дата** |
|  |  |

Львів-2023

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

**Хід 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.

**Main.py**

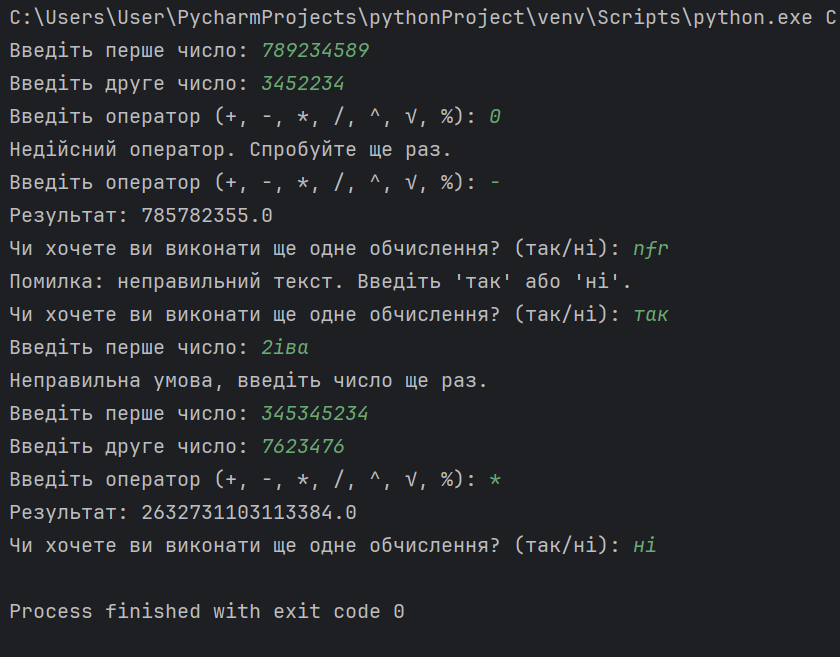
#import console2  
from Scripts.all\_python\_labs.shared\_lib.console\_ui import ConsoleUI  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 console\_ui = ConsoleUI()  
 console\_ui.run\_calculator()  
 #console2.run\_calculator()

**Functions.py**

class Calculator:  
 def \_\_init\_\_(self):  
 self.num1 = 0  
 self.num2 = 0  
 self.operator = ""  
 self.valid\_operators = ['+', '-', '\*', '/', '^', '√', '%']  
  
 def get\_input(self):  
 while True:  
 try:  
 value = input("Введіть перше число: ")  
 if not value.replace('.', '', 1).isdigit():  
 raise ValueError  
 self.num1 = float(value)  
 break  
 except ValueError:  
 print("Неправильна умова, введіть число ще раз.")  
  
 while True:  
 try:  
 value = input("Введіть друге число: ")  
 if not value.replace('.', '', 1).isdigit():  
 raise ValueError  
 self.num2 = float(value)  
 break  
 except ValueError:  
 print("Неправильна умова, введіть число ще раз.")  
  
 self.operator = input("Введіть оператор (+, -, \*, /, ^, √, %): ")  
  
 def is\_valid\_operator(self):  
 return self.operator in self.valid\_operators  
  
 def calculate(self):  
 if self.operator not in self.valid\_operators:  
 raise ValueError("Недійсний оператор")  
 if self.operator == '+':  
 return self.num1 + self.num2  
 elif self.operator == '-':  
 return self.num1 - self.num2  
 elif self.operator == '\*':  
 return self.num1 \* self.num2  
 elif self.operator == '/':  
 if self.num2 == 0:  
 raise ZeroDivisionError("Помилка: ділення на нуль!")  
 return self.num1 / self.num2  
 elif self.operator == '^':  
 return self.num1 \*\* self.num2  
 elif self.operator == '√':  
 if self.num1 < 0:  
 raise ValueError("Корінь з від'ємного числа не дозволений")  
 return self.num1 \*\* (1/self.num2)  
 elif self.operator == '%':  
 return self.num1 % self.num2  
  
 def safe\_calculate(self):  
 try:  
 return self.calculate()  
 except ZeroDivisionError as e:  
 return f"Помилка: {e}"  
 except ValueError as e:  
 return f"Помилка: {e}"  
 except Exception as e:  
 return f"Помилка: {e}"  
  
 def repeat\_calculation(self):  
 while True:  
 repeat = input("Чи хочете ви виконати ще одне обчислення? (так/ні): ").lower()  
 if repeat == "так":  
 return True  
 elif repeat == "ні":  
 exit()  
 else:  
 print("Помилка: неправильний текст. Введіть 'так' або 'ні'.")

**Console.py**

from functions2 import Calculator  
  
def run\_calculator():  
 calc = Calculator()  
 while True:  
 calc.get\_input()  
 while not calc.is\_valid\_operator():  
 print("Недійсний оператор. Спробуйте ще раз.")  
 calc.operator = input("Введіть оператор (+, -, \*, /, ^, √, %): ")  
 result = calc.safe\_calculate()  
 print(f"Результат: {result}")  
 if not calc.repeat\_calculation():  
 break



*Рис. 1 Результат виконання програми*

**Висновок:** Виконавши ці завдання, я перетворила консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проект допоміг вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.