МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

**Лабораторна робота №2**

**з дисципліни**

**СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ**

**на тему**

**«Основи побудови об’єктно-орієнтованих додатків на Python»**

Виконав:

ст. гр. ІТ-21сп

Одноріг Д.І.

Прийняв:

Щербак С.С.

Львів-2023

**Мета роботи:** 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.

Код програми:

import math

#Завдання 1. Створення класу.

class Calculator:

#Завдання 2. Ініціалізація калькулятора.

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

self.result = None

#Завдання 3: Введення користувача

def get\_user\_input(self):

try:

#Завдання 8. Десяткові числа (float)

self.num1 = float(input("Введіть перше число: "))

#Завдання 9. Додаткові операції

self.operator = input("Введіть операцію (+, -, \*, /, sqrt, ^, %): ")

except ValueError:

print("Помилка: Введене значення має бути числом.")

return False

return True

#Завдання 4. Валідність оператора

def is\_valid\_operator(self):

valid\_operators = ('+', '-', '\*', '/', 'sqrt', '^', '%')

if self.operator in valid\_operators:

return True

else:

print("Помилка: Введена операція недійсна. Допустимі операції: +, -, \*, /, sqrt, ^, %.")

return False

#Завдання 5. Обчислення

def calculate(self):

if self.is\_valid\_operator():

if self.operator == '+':

self.num2 = float(input("Введіть друге число: "))

self.result = self.num1 + self.num2

elif self.operator == '-':

self.num2 = float(input("Введіть друге число: "))

self.result = self.num1 - self.num2

elif self.operator == '\*':

self.num2 = float(input("Введіть друге число: "))

self.result = self.num1 \* self.num2

elif self.operator == '/':

self.num2 = float(input("Введіть друге число: "))

#Завдання 6.1 Обробка помилок

if self.num2 == 0:

print("Помилка: Ділення на нуль неможливе.")

else:

self.result = self.num1 / self.num2

elif self.operator == '^':

self.num2 = float(input("Введіть ступінь: "))

self.result = self.num1 \*\* self.num2

elif self.operator == 'sqrt':

#Завдання 6.2 Обробка помилок

if self.num1 < 0:

print("Помилка: Корінь з від'ємного числа неможливий.")

else:

self.result = math.sqrt(self.num1)

elif self.operator == '%':

self.num2 = float(input("Введіть число для відсотків: "))

self.result = (self.num1 \* self.num2) / 100

return True

return False

#Завдання 7. Повторення обчислень

#Завдання 10. зрозумілий інтерфейс

def run\_calculator(self):

print("Вітаю в калькуляторі!")

while True:

if self.get\_user\_input():

if self.calculate():

print("Результат: {:.2f}".format(self.result))

another\_calculation = input("Виконати ще одну операцію? (Так/Ні): ")

if another\_calculation.lower() != "так":

print("Дякую за користуванням калькулятором!")

break

calc = Calculator()

calc.run\_calculator()

На рис. 1-3 зображено демонстрацію роботи програми.

*Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис*

*Рис. 1.* Виконання операції множення із плаваючою комою

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

*Рис. 2.* Виконання операції піднесення до степеня

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

*Рис. 3.* Виконання повторного обчислення

**Висновок:** Виконуючи дану лабораторну роботу, я успішно розробив консольний калькулятор в об’єктно орієнтованому стилі з використанням класів.