МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Лабораторна робота №2

з курсу

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконала студентка

групи ІТ-21сп

**Романишин.Д.О.**

Прийняв

**Щербак С.С.**

Львів - 2023

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

**План роботи**

**Завдання 1:** Створення класу Calculator.

Створити клас Calculator, який буде служити основою для додатка калькулятора.

**Завдання 2:** Ініціалізація калькулятора.

Реалізувати метод \_\_init\_\_ у класі Calculator для ініціалізації необхідних атрибутів або змінних.

**Завдання 3:** Введення користувача.

Перемістити функціональність введення користувача в метод у межах класу Calculator. Метод повинен приймати введення для двох чисел і оператора.

**Завдання 4:** Перевірка оператора.

Реалізувати метод у класі Calculator, щоб перевірити, чи введений оператор є дійсним (тобто одним із +, -, \*, /). Відобразити повідомлення про помилку, якщо він не є дійсним.

**Завдання 5:** Обчислення.

Створити метод у класі Calculator, який виконує обчислення на основі введення користувача (наприклад, додавання, віднімання, множення, ділення).

**Завдання 6:** Обробка помилок.

Реалізувати обробку помилок у межах класу Calculator для обробки ділення на нуль або інших потенційних помилок. Відобразити відповідні повідомлення про помилку.

**Завдання 7:** Повторення обчислень.

Додати метод до класу Calculator, щоб запитати користувача, чи він хоче виконати ще одне обчислення. Якщо так, дозволити йому ввести нові числа і оператор. Якщо ні, вийдіть з програми.

**Завдання 8:** Десяткові числа.

Модифікувати клас Calculator для обробки десяткових чисел (плаваюча кома) для більш точних обчислень.

**Завдання 9:** Додаткові операції.

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

**Завдання 10:** Інтерфейс, зрозумілий для користувача.

Покращити інтерфейс користувача у межах класу Calculator, надавши чіткі запити, повідомлення та форматування виводу для зручності читання.

Код виконаних завдань представлено нижче.

import math  
  
class Calculator:

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

self.result = None

def get\_user\_input(self):

try:

num1 = float(input("Enter the first number: "))

operator = input("Enter the operator (+, -, \*, /, ^, √, %): ")

num2 = float(input("Enter the second number: "))

return num1, operator, num2

except ValueError:

print("Invalid number format. Try again.")

return self.get\_user\_input()

def check\_operator(self, operator):

valid\_operators = ['+', '-', '\*', '/', '^', '√', '%']

if operator not in valid\_operators:

print("Error: Invalid operator entered. Please enter a valid operator.")

return False

return True

def calculate(self, num1, operator, num2):

if operator == '+':

self.result = num1 + num2

elif operator == '-':

self.result = num1 - num2

elif operator == '\*':

self.result = num1 \* num2

elif operator == '/':

try:

self.result = num1 / num2

except ZeroDivisionError:

print("You cannot divide by zero!")

elif operator == '^':

self.result = num1 \*\* num2

elif operator == '√':

self.result = num1 \*\* 0.5

elif operator == '%':

self.result = num1 % num2

def run\_calculator(self):

memoryres = []

decimal\_places = 2

while True:

num1, operator, num2 = self.get\_user\_input()

if self.check\_operator(operator):

self.calculate(num1, operator, num2)

if self.result is not None:

print(f"Answer: {self.result:.{decimal\_places}f}")

archive1 = input("Do you want to save the result to the archive? (yes/no): ")

if archive1.lower() == 'yes':

memoryres.append(self.result)

print("The result is saved in the archive")

archive2 = input("Do you want to restore the archive? (yes/no): ")

if archive2.lower() == 'yes':

if memoryres:

print("Archive of saved results:")

for i, saved\_result in enumerate(memoryres, start=1):

print(f"{i}. {saved\_result:.{decimal\_places}f}")

else:

print("The archive is empty!")

magazine = input("Open the magazine? (yes/no): ")

if magazine.lower() == 'yes':

if memoryres:

print("Magazine:")

for i, saved\_result in enumerate(memoryres, start=1):

print(f"{i}. Result: {saved\_result:.{decimal\_places}f}")

else:

print("The magazine is empty!")

settings = input("Do you want to configure calculator settings? (yes/no): ")

if settings.lower() == 'yes':

decimal\_places = int(input("Enter the number of decimal places to display: "))

povtor = input("Do you want to make a new calculation? (yes/no): ")

if povtor.lower() != 'yes':

break

class ScientificCalculator(Calculator):

def calculate(self, num1, operator, num2):

if operator == '√':

self.result = num1 \*\* (1 / num2)

else:

super().calculate(num1, operator, num2)

def main():

print("Welcome to the Calculator!")

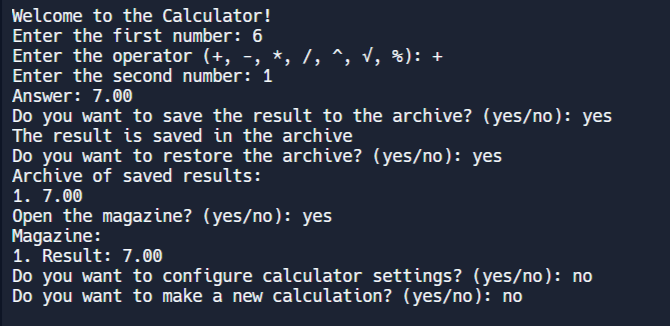
calculator = ScientificCalculator()

calculator.run\_calculator()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Результат виконання програми**



**Висновок:** під час виконання лабораторної роботи я навчився перетворювати консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проєкт допоміг мені вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.