МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Лабораторна робота №2

з курсу

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконав студент

групи ІТ-21сп

**Петровський Т.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.

class Calculator:

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

# Завдання 2: Ініціалізація калькулятора

self.result = 0

def get\_user\_input(self):

# Завдання 3: Введення користувача

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

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

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

def check\_operator(self):

# Завдання 4: Перевірка оператора

if self.operator not in ('+', '-', '\*', '/', '^', '√', '%'):

print("Помилка: Введено недійсний оператор.")

return False

return True

def calculate(self):

# Завдання 5: Обчислення

if self.operator == '+':

self.result = self.num1 + self.num2

elif self.operator == '-':

self.result = self.num1 - self.num2

elif self.operator == '\*':

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

elif self.operator == '/':

# Завдання 6: Обробка помилок

if self.num2 == 0:

print("Помилка: Ділення на нуль.")

return

self.result = self.num1 / self.num2

elif self.operator == '^':

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

elif self.operator == '√':

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

elif self.operator == '%':

self.result = self.num1 % self.num2

def repeat\_calculation(self):

# Завдання 7: Повторення обчислень

another = input("Виконати ще одне обчислення? (так/ні): ")

return another.lower() == 'так'

def run(self):

while True:

self.get\_user\_input()

if not self.check\_operator():

continue

self.calculate()

print("Результат:", self.result)

if not self.repeat\_calculation():

break

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

calc = Calculator()

calc.run()

**Висновки:** 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.