мІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

нАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «лЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ

|  |
| --- |
|  |



ЗВІТ

Про виконання лабораторної роботи № 2

З дисципліни

«Спеціалізовані мови програмування»

Виконала:

студентка гр. ІТ-31

Катерина ПОПОВА

Прийняв:

Сергій ЩЕРБАК

Львів – 2023

**Тема роботи :** Основи побудови об’єктно-орієнтованих додатків на Python **Мета роботи:** Розробка консольного калькулятора в об’єктно орієнтованому стилі з використанням класів

**План 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**

class Calculator:

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

self.memory = []

def get\_user\_input(self):

try:

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

operator = input("Введіть оператор (+, -, \*, /, ^, sq, %): ")

if operator not in ['+', '-', '\*', '/', '^', 'sq', '%']:

raise ValueError("Невідомий оператор")

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

return num1, num2, operator

except (ValueError, ZeroDivisionError) as error:

raise ValueError(f"Помилка: {error}")

def perform\_calculation(self, num1, num2, operator):

try:

if operator == '+':

result = num1 + num2

elif operator == '-':

result = num1 - num2

elif operator == '\*':

result = num1 \* num2

elif operator == '/':

if num2 == 0:

raise ZeroDivisionError("Ділення на нуль неможливе")

result = num1 / num2

elif operator == '^':

result = num1 \*\* num2

elif operator == 'sq':

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

elif operator == '%':

result = num1 % num2

else:

raise ValueError("Невідомий оператор")

self.memory.append(result)

print(f"Результат: {result}")

except (ValueError, ZeroDivisionError) as error:

print(f"Помилка: {error}")

def run\_calculator(self):

while True:

try:

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

self.perform\_calculation(num1, num2, operator)

except ValueError as error:

print(f"Помилка: {error}")

finally:

another\_calculation = input("Бажаєте виконати ще одне обчислення? (так/ні): ")

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

self.show\_calculation\_history()

break

def show\_calculation\_history(self):

print("Історія обчислень:")

for idx, item in enumerate(self.memory, start=1):

print(f"{idx}: {item}")

# Створюємо екземпляр класу Calculator та викликаємо метод run\_calculator()

calc = Calculator()

calc.run\_calculator()

**Результат виконання**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

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