Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ІСМ



Звіт

про виконання лабораторної роботи № 2

«Введення в Python»

з дисципліни

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

Виконала:

Студентка групи ІТ-31,

Василик Ю. М.

Прийняв:

Щербак С.С

Львів

2024

**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.

**Хід виконання роботи:**

import math

class Calculator:

    def \_\_init\_\_(self):

        self.result = 0

    def add(self, x, y):

        self.result = x + y

        return self.result

    def subtract(self, x, y):

        self.result = x - y

        return self.result

    def multiply(self, x, y):

        self.result = x \* y

        return self.result

    def divide(self, x, y):

        try:

            self.result = x / y

            return self.result

        except ZeroDivisionError:

            return "Помилка: Неможливо поділити на нуль"

        except Exception as e:

            return f"Помилка: {str(e)}"

    def power(self, x, y):

        self.result = x \*\* y

        return self.result

    def square\_root(self, x):

        if x < 0:

            return "Помилка: Квадратний корінь від від'ємного числа"

        self.result = math.sqrt(x)

        return self.result

    def modulus(self, x, y):

        try:

            self.result = x % y

            return self.result

        except ZeroDivisionError:

            return "Помилка: Неможливо обчислити залишок від ділення на нуль"

        except Exception as e:

            return f"Помилка: {str(e)}"

    def get\_user\_input(self):

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

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

        if operator in ['+', '-', '\*', '/', '^', '√', '%']:

            if operator in ['+', '-', '\*', '/', '^', '%']:

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

                return x, operator, y

            else:

                return x, operator

        else:

            return "Помилка: Недійсний оператор"

    def perform\_calculation(self, x, operator, y=None):

        if operator == '+':

            return self.add(x, y)

        elif operator == '-':

            return self.subtract(x, y)

        elif operator == '\*':

            return self.multiply(x, y)

        elif operator == '/':

            return self.divide(x, y)

        elif operator == '^':

            return self.power(x, y)

        elif operator == '√':

            return self.square\_root(x)

        elif operator == '%':

            return self.modulus(x, y)

        else:

            return "Помилка: Недійсний оператор"

    def run\_calculator(self):

        print("Ласкаво просимо до калькулятора!")

        while True:

            user\_input = self.get\_user\_input()

            if isinstance(user\_input, tuple):

                x, operator, y = user\_input

                result = self.perform\_calculation(x, operator, y)

                print(f"Результат: {x} {operator} {y} = {result}")

            else:

                print(user\_input)

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

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

                break

calc = Calculator()

calc.run\_calculator()

**Висновки:** Виконавши ці завдання, ми перетворили консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проект допоможе нам вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.