Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторної роботи № 2

“Основи побудови об’єктно-орієнтованих додатків на Python”

з дисципліни **“ Спеціалізовані мови програмування”**

Виконала студентка групи ІТ-32
ДЕНИСЯК С.-М. Т.
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

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:

calculator.py

```
import math


class Calculator:
    def init(self):
        # Initialize the calculator with placeholders for values and operator.
        self.first_value = None
        self.operator = None
        self.second_value = None

    def run(self):
        while True:
            try:
                # Get user input and perform calculations.
                self.get_user_input()
                print("The result is", self.calculate())
            except Exception as e:
                # Handle exceptions and display error messages.
                print(str(e))

            response = input(
                "Would you like to continue? Enter 'Y' or 'y' if you do, or
anything else if you do not. Your response is ").lower()
            if response != "y":
                break

    def get_user_input(self):
        # Get input from the user for the first value and operator.
        self.first_value = float(input("Enter the first value: "))
        self.operator = input("Enter the operator ['+', '-', '*', '/', '^', '√',
'%']: ")

        if self.operator in ('+', '-', '*', '/', '^', '√', '%'):
            if self.operator != "√":
                # If the operator requires a second value, get it from the user.
                self.second_value = float(input("Enter the second value: "))
        else:
            # Raise a ValueError if the input operator is incorrect.
            raise ValueError("The input operator is incorrect")

    def calculate(self):
        # Perform calculations based on the operator chosen.
        if self.operator == "+":
            return self.first_value + self.second_value
        elif self.operator == "-":
            return self.first_value - self.second_value
        elif self.operator == "*":
            return self.first_value * self.second_value
        elif self.operator == "/":
            if self.second_value == 0:
                # Raise a ZeroDivisionError for division by zero.
                raise ZeroDivisionError("Impossible to divide")
            return self.first_value / self.second_value
        elif self.operator == "^":
            # Perform multiplication if the operator is an empty string.
            return self.first_value ** self.second_value
        elif self.operator == "√":
            if self.first_value < 0:
                # Raise a ValueError if trying to calculate the square root of a
negative number.
                raise ValueError("Number is negative, therefore it is impossible
to calculate the square root")
```

```
            return math.sqrt(self.first_value)
        elif self.operator == "%":
            # Calculate the remainder (modulo operation).
            return self.first_value % self.second_value
```

main.py

```
from calculator import Calculator

if __name__ == "__main__":
    calc = Calculator()
    calc.run()
```

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