

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»  
ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗВІТ  
для лабораторної роботи № 2  
з дисципліни  
«Спеціалізовані мови програмування»

Виконав:  
студент гр. ІТ-32  
Федик В.А

Прийняв:  
доц. каф. ІСМ  
Щербак С.С.

Львів-2023

**Мета роботи:** Розробка консольного калькулятора в об'єктно-орієнтованому стилі з використанням класів.

### **Хід виконання:**

#### **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:**

```
import math
```

```
class Calculator:
```

```
    def __init__(self):  
        self.first_value = None
```

```
self.operator = None
self.second_value = None
```

```
def run(self):
    while True:
        try:
            self.input()
            print("The result is {}".format(self.calculate()))

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

```
def input(self):
    try:
        self.__init__()
        self.first_value = float(input("Enter the first value: "))
        self.operator = str(input("Enter the operator ['+', '-', '*', '/', '**', '√', '%']: "))
        if self.__is_operator_correct(self.operator):
            if self.operator != "√":
                self.second_value = float(input("Enter the second value: "))
            else:
                raise RuntimeError("The input operator is incorrect")
        except ValueError as e:
            raise type(e)("The format of number is invalid")
```

```
@staticmethod
```

```
def __is_operator_correct(operator):
    valid_operators = ('+', '-', '*', '/', '**', '√', '%')
    return operator in valid_operators
```

```
def calculate(self):
    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 ZeroDivisionError("Impossible to divide")
        return self.first_value / self.second_value
    elif self.operator == "**":
        return self.first_value ** self.second_value
```

```
elif self.operator == "√":  
    if self.first_value < 0:  
        raise ArithmeticError("Number is negative, therefore it is impossible to calculate the  
square root")  
    return math.sqrt(self.first_value)  
elif self.operator == "%":  
    return self.first_value % self.second_value
```

```
c = Calculator()  
c.run()
```

**Посилання на GitHub-репозиторій:** <https://github.com/fedykvitalik2004/spl>

**Висновки:** Виконавши ці завдання, я розробив консольний калькулятор в об'єктно-орієнтованому стилі з використанням класів.