

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

КАФЕДРА ІСМ



ЗВІТ

про виконання лабораторної роботи №2

з дисципліни

«Спеціалізовані мови програмування»

студента групи ІТ-32

Ткачишина Юрія

Прийняв Щербак С. С.

Львів - 2023

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

Індивідуальне завдання

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:

Код:

Файл `calculator.py`:

```
import math
```

```
class Calculator:
```

```
    def __init__(self):
```

```

self.num1 = None

self.num2 = None

self.operator = None


def get_user_input(self):
    while True:
        try:
            self.num1 = float(input("Введіть перше число: "))
            self.operator = input("Введіть оператор (+, -, *, /, ^, sqrt, %): ")
            if self.operator != 'sqrt':
                if self.operator not in ['+', '-', '*', '/', '^', 'sqrt', '%']:
                    print("Помилка: недійсний оператор.")
                    continue # Ask for operator again if it's invalid
            self.num2 = float(input("Введіть друге число: "))
            break # Exit the loop when valid input is provided
        except ValueError:
            print("Помилка: введіть дійсне число.")
        except KeyboardInterrupt:
            print("\nДо побачення!")
            exit(0)


def calculate_result(self):
    raise NotImplementedError("Метод calculate_result повинен бути реалізований у підкласі.")


def repeat_calculation(self):
    while True:

```

```
print("Підтримувані операції: +, -, *, /, ^ (піднесення до степеня), sqrt  
(квадратний корінь), % (залишок від ділення)")
```

```
self.get_user_input()
```

```
result = self.calculate_result()
```

```
if result is not None:
```

```
    print(f"Результат: {result:.2f}")
```

```
another_calculation = input("Виконати ще одне обчислення? (y/n): ")
```

```
if another_calculation.lower() != 'y':
```

```
    break
```

Код файлу:

```
from calculator import Calculator # Імпортуємо базовий клас
```

```
import math
```

```
class PolymorphicCalculator(Calculator):
```

```
    def calculate_result(self):
```

```
        if self.operator == '+':
```

```
            return self.num1 + self.num2
```

```
        elif self.operator == '-':
```

```
            return self.num1 - self.num2
```

```
        elif self.operator == '*':
```

```
            return self.num1 * self.num2
```

```
        elif self.operator == '/':
```

```
            try:
```

```
                if self.num2 == 0:
```

```
                    raise ZeroDivisionError
```

```
                return self.num1 / self.num2
```

```
except ZeroDivisionError:

    print("Помилка: ділення на 0.")

    return None

elif self.operator == '^':

    return self.num1 ** self.num2

elif self.operator == 'sqrt':

    if self.num1 < 0:

        print("Помилка: неможливо взяти корінь з від'ємного числа.")

        return None

    return math.sqrt(self.num1)

elif self.operator == '%':

    return self.num1 % self.num2
```

Тестування:

```
if __name__ == "__main__":

    calc = PolymorphicCalculator()

    calc.repeat_calculation()
```

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