

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
Нюл В. С.

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:

Calculator.py:

```
import math
```

```
from Menu import Menu
```

```
class Calculator:
    def __init__(self):
        self.history = []
        self.memory = []
        self.result = None
        self.choose = 0

    def run(self):
        menu_items = ['Виконати розрахунок', "Змінити пам'ять результатів",
                      'Показати історію розрахунків', 'Вийти з програми']
        menu = Menu(menu_items)

        while True:
            try:
                menu.display()
                self.choose = menu.choose_option()

            match self.choose:
                case 1:
                    while True:
                        num1, operator, num2, decimal_places = self.user_input()
                        self.check_operator(operator)
                        self.result = self.calculate(num1, num2, operator, decimal_places)
                        print(f'Результат: {self.result}')
                        repeat = self.ask_about_another_calculation()
                        if repeat.lower() == 'так':
                            continue
                        else:
                            break
                case 2:
                    self.memory_change()
                case 3:
                    self.show_history()
                case 4:
                    break
```

```
except ValueError as e:
    print(f'Помилка: {e}')
except ZeroDivisionError:
    print('Помилка: Ділення на нуль')
```

```
def user_input(self):
    num1 = float(input('Введіть перше число: '))
    operator = input('Введіть оператор (+, -, *, /, ^, √, %): ')
    num2 = None
    if operator != '√':
        num2 = float(input("Введіть друге число: "))

    decimal_places = int(input('Кількість десяткових розрядів для результату: '))
    if decimal_places < 0:
        raise ValueError('Непідтримувана кількість десяткових розрядів. Спробуйте ще раз.')

    return num1, operator, num2, decimal_places
```

```
def check_operator(self, operator):
    if operator not in ('+', '-', '*', '/', '^', '√', '%'):
        raise ValueError('Непідтримуваний оператор. Введіть оператор, який підтримується (+, -, *, /, ^, √, %)')
```

```
def calculate(self, num1, num2, operator, decimal_places):
    operator_functions = {
        '+': lambda x, y: x + y,
        '-': lambda x, y: x - y,
        '*': lambda x, y: x * y,
        '/': lambda x, y: x / y if y != 0 else None,
        '^': lambda x, y: x ** y,
        '√': lambda x, _: math.sqrt(x),
        '%': lambda x, y: x % y
    }
```

```
result = operator_functions[operator](num1, num2)
```

```
if result is None:
    raise ZeroDivisionError
```

```
if decimal_places == 0:
    result = int(result)
else:
    result = round(result, decimal_places)
```

```
self.push_to_history(num1, num2, operator, result)
```

```
return result
```

```
def push_to_history(self, num1, num2, operator, result):
    if operator == '√':
        self.history.append((operator, num1, result))
    else:
        self.history.append((num1, operator, num2, result))
```

```
def memory_change(self):
    if self.result is None:
        print("Спочатку зробіть розрахунок")
        return
    memory_option = input("Оновити пам'ять калькулятора? (так/ні): ")
    if memory_option.lower() != 'так':
        return
```

```
memory_option = input('Зберегти попередній результат? (так/ні): ')
if memory_option.lower() == 'так':
    self.memory.append(self.result)
```

```
if self.memory:
    memory_option = input("Отримати результат з пам'яті? (так/ні): ")
    if memory_option.lower() == 'так':
        selected_position = int(
            input(f'Оберіть порядковий номер результату, який бажаєте отримати(макс:
{len(self.memory)}): '))
```

```

    if 1 <= selected_position <= len(self.memory):
        value = self.memory[selected_position - 1]
        print(f"Значення з пам'яті: {value}")
    else:
        raise ValueError('Порядковий номер не дійсний.')

```

```

def show_history(self):
    print("\nІсторія обчислень:")
    for entry in self.history:
        if len(entry) == 3:
            operator, num1, result = entry
            print(f'{operator} {num1} = {result}')
        else:
            num1, operator, num2, result = entry
            print(f'{num1} {operator} {num2} = {result}')

```

```

def ask_about_another_calculation(self):
    return input('Виконати ще одне обчислення? (так/ні): ')

```

```

if __name__ == '__main__':
    calc = Calculator()
    calc.run()

```

Menu.py:

```

class Menu:
    def __init__(self, items):
        self.items = items

    def display(self):
        for i, item in enumerate(self.items, 1):
            print(f'{i}. {item}')

    def choose_option(self):
        try:
            choice = int(input("Оберіть пункт меню: "))
            if 1 <= choice <= len(self.items):

```

```
        return choice
    else:
        print("Неправильний вибір. Будь ласка, введіть номер зі списку.")
except ValueError:
    print("Неправильний ввід. Введіть номер зі списку.")
```

Посилання на GitHub-репозиторій: <https://github.com/VasylNyul/python-labs>

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