

ЗВІТ
про виконання лабораторної роботи № 2
«Основи побудови об'єктно-орієнтованих додатків на Python»
з дисципліни
«Спеціалізовані мови програмування»
ст. групи РІ-31
Танечник Марічки

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

Умова 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.

Текст програми:

```
import math
from classes.FileManager import FileManager
from classes.HistoryManager import HistoryManager
from classes.Memory import Memory
from classes.Operation import Operation, Addition, Subtraction,
Multiplication, Division, SquareRoot
from functions.calculate import calculate
from functions.display_menu import display_menu
from AppSettings import get_decimal_places
```

```
class Calculator:
    def __init__(self):
        self._memory = Memory()
        self._history_manager =
HistoryManager("calculation_history.txt")
        self._operations = {
            '+': Addition(),
            '-': Subtraction(),
            '*': Multiplication(),
            '/': Division(),
            'sqrt': SquareRoot(),
        }
        self.last_result = None

    def perform_operation(self, operator, num1, num2=None):
        if operator not in self._operations:
            raise ValueError("Invalid operator!")
        result = self._operations[operator].calc(num1, num2)
        self.last_result = result
        return result

    def memory_operations(self):
        self._memory.perform_memory_operation(self.last_result)
```

```
def show_history(self):  
    self._history_manager.view_history()  
  
def add_to_history(self, operation):  
    self._history_manager.add_to_history(operation)  
  
def get_decimal_places(self):  
    return get_decimal_places()  
  
from classes.Calculator import Calculator  
from functions.menu import menu  
  
def main():  
    calculator = Calculator()  
    menu(calculator)  
  
if __name__ == "__main__":  
    main()
```

Висновки: виконуючи лабораторну роботу я перетворила консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python.