

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №2

з дисципліни

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

на тему

ОСНОВИ ПОБУДОВИ ОБ'ЄКТНО-ОРІЄНТОВАНИХ ДОДАТКІВ НА PYTHON

Виконав:

ст. гр. IT-32

Шоха А.А.

Прийняв:

Щербак С.С.

Львів-2023

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

Завдання на лабораторну роботу

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 роботи

Клас калькулятор:

Calculator.py

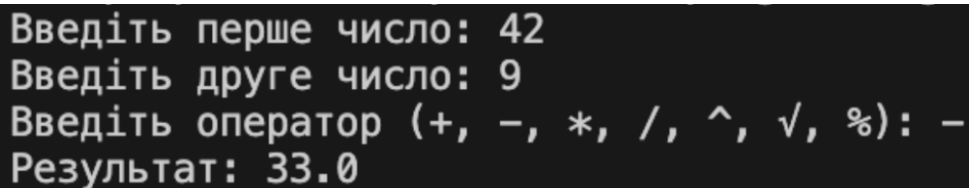
```
import sys
from constants import home_path
sys.path.append(home_path + 'source/lab1')

#import functions from lab1 to handle user input and perform
calculations
from settings_funcs import ask_if_repeat, call_menu,
get_user_input
from calculator_funcs import *

class Calculator:
    def __init__(self):
        self.result = None
        self.history = []
        self.repeat = True
        self.rounding_value = 50
        self.operations = {'+':add, '-':sub, '*':mul,
'/:div, '^':exp, '√':root, '%':mod}
        self.num1 = 0
        self.num2 = 0

    def calculate(self):
        while self.repeat:
            num1, num2, operator = get_user_input()
            result =
round(self.operations[operator](float(num1), float(num2)),
self.rounding_value)
            self.history.append((num1, num2, operator,
result))
            self.repeat = ask_if_repeat(self.history)
            call_menu(self.history)
```

Приклад використання:



```
Введіть перше число: 42
Введіть друге число: 9
Введіть оператор (+, -, *, /, ^, √, %): +
Результат: 33.0
```

Рис. 1. Приклад виконання обчислень

Збереження історії:

history.txt

1) $6 - 8 = -2.0$

1) $6 - 8 = -2.0$

2) $7 - 9 = -2.0$

3) $4 - 5 = -1.0$

1) $5 * 6 = 30.0$

1) $5 * 6 = 30.0$

1) $5 + 6 = 11.0$

1) $5 + 6 = 11.0$

Висновок: під час виконання лабораторної роботи було розроблено консольний File display кулятор в об'єктно орієнтованому стилі з використанням класів.