

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Кафедра інформаційних систем та мереж



**Звіт**

Про виконання лабораторної роботи №2  
з дисципліни «Спеціалізовані мови програмування»  
на тему «Основи побудови об'єктно-орієнтованих додатків на Python»

**Виконала:**

Студентка гр. ПІ-21

Зузяк Л. Р.

**Прийняв:**

Щербак С.С.

Львів 2024

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

План роботи

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 ( $\sqrt{\phantom{x}}$ ) 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.

Виконання:

```
from .operations import Operations
from .memory import Memory
from .history import History
from .settings import Settings

class Calculator:
    def __init__(self, decimal_places=2):
        self.operations = Operations()
        self.memory = Memory()
        self.history = History()
        self.settings = Settings(decimal_places=decimal_places)

    def get_user_input(self):
        while True:
            try:
                num1 = float(input("Enter the first number: "))
                operator = input("Enter an operator (+, -, *, /, ^, √, %): ")
                num2 = None
                if operator not in ('√',):
                    num2 = float(input("Enter the second number: "))
                return num1, operator, num2
            except ValueError:
                print("Invalid input. Please enter numbers correctly.")

    def run(self):
        while True:
            num1, operator, num2 = self.get_user_input()
            if not self.operations.validate_operator(operator):
                continue

            result = self.operations.calculate(num1, operator, num2)

            if result is not None:
                result = self.settings.round_result(result)
                print(f"Result: {result}")
                self.history.add_entry(num1, operator, num2, result)
```

```
)
    save_memory = input("Do you want to save this result to memory? (y/n): ")

    if save_memory.lower() == 'y' and result is not None:
        self.memory.store(result)

    view_history = input("View calculation history? (y/n): ")
    if view_history.lower() == 'y':
        self.history.show()

    new_calc = input("Would you like to perform another calculation? (y/n): ")

    if new_calc.lower() != 'y':
        print("Goodbye!")
        break
```

Результат:

```
Enter the first number: 567
Enter an operator (+, -, *, /, ^, √, %): ^
Enter the second number: 3
Result: 182284263.0
Do you want to save this result to memory? (y/n): y
Value 182284263.0 stored in memory.
View calculation history? (y/n): y
Calculation History:
345.0 % 23.0 = 0.0
567.0 ^ 3.0 = 182284263.0
Would you like to perform another calculation? (y/n): n
Goodbye!
```

Покликання на виконану роботу в GitHub:

[https://github.com/Lilia427/Lab\\_2\\_p](https://github.com/Lilia427/Lab_2_p)

Висновок: в ході виконання лабораторної роботи я на практиці освоїла як перетворити консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проект допоміг мені вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.