Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**ЗВІТ**

про виконання лабораторної роботи № 2

з курсу:

«Спеціалізовані мови програмування»

**Виконала:**

студентка гр. ІТ-31

Ірина ПЕРХУН

**Прийняв:**

Сергій ЩЕРБАК

Львів 2023

**Тема:** Основи побудови об’єктно-орієнтованих додатків на Python

**Мета роботи:** Розробка консольного калькулятора в об’єктно орієнтованому стилі з використанням класів

**Хід 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 (√) 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.

**main.py**

*from* calculator *import* Calculator

*if* \_\_name\_\_ == "\_\_main\_\_":

    language\_choice = input("Choose language (English/Ukrainian): ").strip().lower()

*# strip - removes all spaces (and other whitespace characters) from the beginning and end of the string.*

*# lower - converts to lowercase.*

*if* language\_choice == 'english':

*import* en\_localization *as* localization

*elif* language\_choice == 'ukrainian':

*import* ua\_localization *as* localization

*else*:

        print("Invalid choice. Using English as default.")

*import* en\_localization *as* localization

    calculator = Calculator(localization)

    calculator.run\_calculator()

**calculator.py**

*import* math

class Calculator:

    def \_\_init\_\_(self, localization):

*# Constructor initializes the Calculator object with localization*

*self*.memory = None  *# Memory for storing results of previous calculations*

*self*.calculation\_records = []  *# List to store calculation records*

*self*.localization = localization  *# Stores the localization object for displaying texts*

    def is\_valid\_float(self, data):

*# Checks if the text can be converted to a floating-point number*

*try*:

            float(data)

*return* True

*except* ValueError:

*return* False

    def get\_number\_from\_user(self, prompt):

*# Prompts the user to enter a number and returns its value*

*while* True:

            num\_input = input(prompt)

*if* *self*.is\_valid\_float(num\_input):

*return* float(num\_input)

*else*:

                print(*self*.localization.INVALID\_INPUT)

    def get\_operator\_from\_user(self):

*# Prompts the user to enter an operator and returns it*

        valid\_operators = ['+', '-', '\*', '/', '^', '√', '%']

*while* True:

            operator = input(*self*.localization.PROMPT\_CHOOSE\_OPERATOR)

*if* operator in valid\_operators:

*return* operator

*else*:

                print(*self*.localization.INVALID\_OPERATOR)

    def perform\_calculation(self, number1, number2, operator, decimal\_places):

*# Performs calculations based on the entered numbers and operator*

        result = None

*try*:

*if* operator == '+':

                result = number1 + number2

*elif* operator == '-':

                result = number1 - number2

*elif* operator == '\*':

                result = number1 \* number2

*elif* operator == '/':

*if* number2 == 0:

*raise* ZeroDivisionError

*else*:

                    result = number1 / number2

*elif* operator == '^':

                result = number1 \*\* number2

*elif* operator == '√':

                result = math.sqrt(number1)

*elif* operator == '%':

                result = number1 % number2

*return* result

*except* ZeroDivisionError:

            print(*self*.localization.BAN\_FOR\_ZERO)

*return* None

    def get\_continue\_input(self):

*# Asks the user if they want to continue calculations*

*return* input(*self*.localization.CONTINUE\_CALCULATION\_PROMPT).lower()

    def run\_calculator(self):

*# Main method for interacting with the user and performing calculations*

*while* True:

            number1 = *self*.get\_number\_from\_user(*self*.localization.PROMPT\_ENTER\_FIRST\_NUMBER)

            number2 = *self*.get\_number\_from\_user(*self*.localization.PROMPT\_ENTER\_SECOND\_NUMBER)

            operator = *self*.get\_operator\_from\_user()

            decimal\_places = int(input(*self*.localization.PROMPT\_ENTER\_DECIMAL\_PLACES))

            result = *self*.perform\_calculation(number1, number2, operator, decimal\_places)

*if* result is not None:

*self*.memory = result

                expression = f"{number1:.{decimal\_places}f} {operator} {number2:.{decimal\_places}f} = {result:.{decimal\_places}f}"

*self*.calculation\_records.append(expression)

                formatted\_result = *self*.localization.RESULT\_FORMAT.format(result=result, decimal\_places=decimal\_places)

                print(formatted\_result)

                history\_str = "\n".join(*self*.calculation\_records)

*# join - joins all strings from the self.calculation\_records list into one string, separating them with a newline character "\n".*

                print(f"{*self*.localization.MEMORY\_OPERATOR\_HEADER}\n{history\_str}")

*else*:

*pass*

            another\_calculation = *self*.get\_continue\_input()

*if* another\_calculation != *self*.localization.CONTINUE\_CALCULATION:

*break*

**en\_localization.py**

*# en\_localization.py*

PROMPT\_ENTER\_FIRST\_NUMBER = "Enter the first number: "

PROMPT\_ENTER\_SECOND\_NUMBER = "Enter the second number: "

INVALID\_INPUT = "Invalid input. Please enter a valid number."

PROMPT\_CHOOSE\_OPERATOR = "Choose operator (+, -, \*, /, ^, √, %): "

INVALID\_OPERATOR = "Invalid operator. Please enter a valid operator from the list (+, -, \*, /, ^, √, %): "

PROMPT\_ENTER\_DECIMAL\_PLACES = "Enter the number of decimal places: "

RESULT\_FORMAT = "Result: {result:.{decimal\_places}f}"

MEMORY\_OPERATOR\_HEADER = "Memory operator:"

CONTINUE\_CALCULATION\_PROMPT = "Do you want to perform another calculation? (yes/no): "

CONTINUE\_CALCULATION = "yes"

BAN\_FOR\_ZERO = "Error: Division by zero is not allowed"

**ua\_localization.py**

*# ua\_localization.py*

PROMPT\_ENTER\_FIRST\_NUMBER = "Введіть перше число: "

PROMPT\_ENTER\_SECOND\_NUMBER = "Введіть друге число: "

INVALID\_INPUT = "Неправильний ввід. Введіть правильне число."

PROMPT\_CHOOSE\_OPERATOR = "Виберіть оператор (+, -, \*, /, ^, √, %): "

INVALID\_OPERATOR = "Неправильний оператор. Виберіть із списку (+, -, \*, /, ^, √, %): "

PROMPT\_ENTER\_DECIMAL\_PLACES = "Введіть кількість десяткових знаків: "

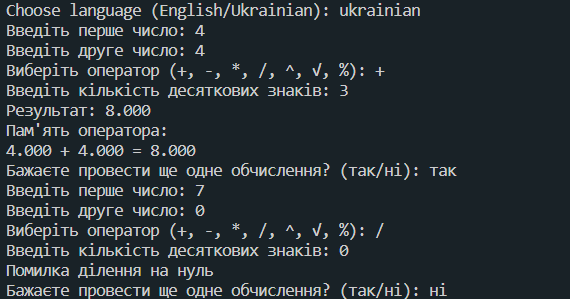
RESULT\_FORMAT = "Результат: {result:.{decimal\_places}f}"

MEMORY\_OPERATOR\_HEADER = "Пам'ять оператора:"

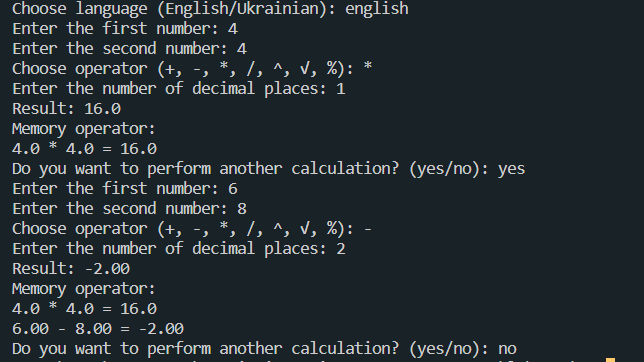
CONTINUE\_CALCULATION\_PROMPT = "Бажаєте провести ще одне обчислення? (так/ні): "

CONTINUE\_CALCULATION = "так"

BAN\_FOR\_ZERO = "Помилка ділення на нуль"



*Рис.1 – Результат виконання для української мови*

**

*Рис.2 – Результат виконання для англійської мови*

**Висновок:** на даній лабораторній роботі я перетворила консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проект допоміг вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.