Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**ЗВІТ**

про виконання лабораторної роботи № 3

з курсу:

«Спеціалізовані мови програмування»

**Виконала:**

студент гр. ІТ-31

Максим ПОЛАПА

**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**

import math

import gettext

import locale

# Memory class for storing user inputs and results

class Memory:

    def \_\_init\_\_(self):

        self.memory = []

    def store(self, expression, result):

        self.memory.append((expression, result))

    def recall(self):

        return self.memory

    def clear(self):

        self.memory = []

# Calculator class for performing calculations

class Calculator:

    OPERATORS = {

        '+': lambda num1, num2: num1 + num2,

        '-': lambda num1, num2: num1 - num2,

        '\*': lambda num1, num2: num1 \* num2,

        '/': lambda num1, num2: num1 / num2 if num2 != 0 else \_("Error: division on zero"),

        '^': lambda num1, num2: num1 \*\* num2,

        '√': lambda num1, \_: math.sqrt(num1) if num1 >= 0 else \_("Error: negative number under the root"),

        '%': lambda num1, num2: num1 % num2 if num2 != 0 else \_("Error: division on zero")

    }

    def \_\_init\_\_(self, locale):

        self.memory = Memory()

        self.locale = locale

    def get\_float\_input(self, prompt\_key):

        while True:

            try:

                return float(input(questions[self.locale][prompt\_key]))

            except ValueError:

                print(\_("Error: Invalid input, please enter a valid number."))

    def run(self):

        while True:

            num1 = self.get\_float\_input('first\_number')

            num2 = self.get\_float\_input('second\_number')

            operator = input(questions[self.locale]['operator'])

            if operator in Calculator.OPERATORS:

                result = Calculator.OPERATORS[operator](num1, num2)

                print(\_("Result:"), result)

                expression = f"{num1} {operator} {num2}"

                self.memory.store(expression, result)

            else:

                print(\_("Error: Invalid operator. Please choose one of these operators: '+, -, \*, /, ^, √, %' "))

            memory\_values = self.memory.recall()

            print(\_("Memory operator:"))

            for expression, result in memory\_values:

                print(f"{expression} = {result}")

            repeat = input(questions[self.locale]['continue']).strip().lower()

            if repeat != 'так' and repeat != 'yes':

                break

# Set the default language to Ukrainian

locale.setlocale(locale.LC\_ALL, 'uk\_UA.utf8')

# Function to choose the language

def choose\_language():

    language = input("Оберіть мову | Choose language (укр/eng): ").strip().lower()

    if language == "eng":

        return 'en\_US'

    elif language == "укр":

        return 'uk\_UA'

    else:

        print("Непідтримувана мова, обрана українська.")

        return 'uk\_UA'

# Localization function based on the selected language

selected\_locale = choose\_language()

\_ = gettext.translation('calculator', localedir='locales', languages=[selected\_locale], fallback=True).gettext

# Dictionary for localized prompts and messages

questions = {

    'en\_US': {

        'first\_number': "First number: ",

        'second\_number': "Second number: ",

        'operator': "Choose operator (+, -, \*, /, ^, √, %): ",

        'continue': "Do you want to continue? (yes/no): "

    },

    'uk\_UA': {

        'first\_number': "Перше число: ",

        'second\_number': "Друге число: ",

        'operator': "Оберіть операцію (+, -, \*, /, ^, √, %): ",

        'continue': "Бажаєте продовжити? (так/ні): "

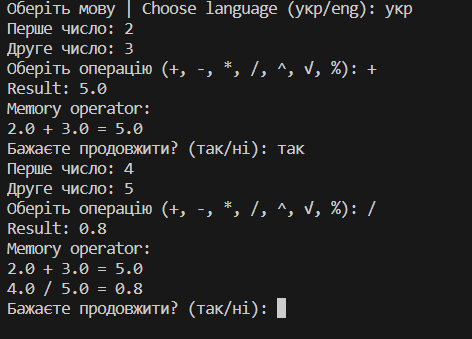
    }

}

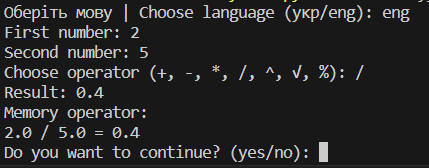
if \_\_name\_\_ == "\_\_main\_\_":

    calculator = Calculator(selected\_locale)

    calculator.run()



*Рис.1 – Результат виконання для української мови*

**

*Рис.2 – Результат виконання для англійської мови*

**Висновок:** на даній лабораторній роботі я перетворив 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.