МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж



Лабораторна робота №1

з дисципліни

Спеціалізовані мови програмування

на тему:

Введення в Python

Виконав:

ст. гр. ІТ-21сп

Олександр КОЗАК

Прийняв

доцент каф. ІСМ:

Сергій ЩЕРБАК

|  |  |
| --- | --- |
| **Балів** | **Дата** |
|  |  |

Львів-2023

**Мета:** створення консольної програми-калькулятора за допомогою основних синтаксичних конструкцій Python, з іншим завданням на заміну тестуванню та валідації.

**Хід роботи:**

**Завдання 1:** **Введення користувача**

Створіть Python-програму, яка приймає введення користувача для двох чисел і оператора (наприклад, +, -, \*, /).

**Завдання 2:** **Перевірка оператора**

Перевірте чи введений оператор є дійсним (тобто одним із +, -, \*, /). Якщо ні, відобразіть повідомлення про помилку і попросіть користувача ввести дійсний оператор.

**Завдання 3:** **Обчислення**

Виконайте обчислення на основі введення користувача (наприклад, додавання, віднімання, множення, ділення) і відобразіть результат.

**Завдання 4: Повторення обчислень**

Запитайте користувача, чи він хоче виконати ще одне обчислення. Якщо так, дозвольте йому ввести нові числа і оператор. Якщо ні, вийдіть з програми.

**Завдання 5:** **Обробка помилок**

Реалізуйте обробку помилок для обробки ділення на нуль або інших потенційних помилок. Відобразіть відповідне повідомлення про помилку, якщо виникає помилка.

**Завдання 6:** **Десяткові числа**

Змініть калькулятор так, щоб він обробляв десяткові числа (плаваючу кому) для більш точних обчислень.

**Завдання 7:** **Додаткові операції**

Додайте підтримку додаткових операцій, таких як піднесення до степеня (^), квадратний корінь (√) і залишок від ділення (%).

**Завдання 8:** **Функція пам'яті**

Реалізуйте функцію пам'яті, яка дозволяє користувачам зберігати і відновлювати результати. Додайте можливості для зберігання та отримання значень з пам'яті.

**Завдання 9:** **Історія обчислень**

Створіть журнал, який зберігає історію попередніх обчислень, включаючи вираз і результат. Дозвольте користувачам переглядати історію своїх обчислень.

**Завдання 10:** **Налаштування користувача**

Надайте користувачам можливість налаштувати поведінку калькулятора, таку як зміну кількості десяткових розрядів, які відображаються, або налаштування функцій пам'яті.

**Main.py**

from calculator import Calculator

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

calc = Calculator()

calc.runCalculator()

**Calculator.py**

import math

class Calculator:

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

self.settings = {

"decimalPlaces": 2,

"useMemory": True

}

self.memory = None # Словник для збереження значень в пам'яті

self.history = [] # Список для збереження історії обчислень

# Метод для зберігання значення в пам'яті

def storeInMemory(self, value):

self.memory = value

# Метод для отримання значення з пам'яті

def recallFromMemory(self):

return self.memory

# Метод для додавання запису в історію обчислень

def addToHistory(self, expression, result):

historyEntry = f'Вираз:{expression} Результат:{result}'

self.history.append(historyEntry)

# Метод для перегляду історії обчислень

def viewHistory(self):

if not self.history:

print("Даних в істрії ще немає!")

for historyEntry in self.history:

print(historyEntry)

def changeSettings(self):

print("1. Зміна знаків після коми")

print("2. Налаштування функції пам'яті.")

choice = input("Введіть ваш вибір: ")

if choice == '1':

try:

places = int(input("Введіть читсло скільки знаків після коми повино бути (0-20): "))

if 0 <= places <= 20:

self.settings["decimalPlaces"] = places

print(f"Встановлено {places} знаків після коми!")

else:

print("Кількість знаків що ви ввели виходить за межі дозволеного.")

except ValueError:

print("Некоректний ввід.")

elif choice == '2':

self.settings["useMemory"] = not self.settings["useMemory"]

status = "enabled" if self.settings["useMemory"] else "disabled"

print(f"Статус функції пам'яті {status}.")

def calculateValue (self,firstNumber, secondNumber,operator):

try:

if operator == '-':

return firstNumber - secondNumber

if operator == '+':

return firstNumber + secondNumber

if operator == '\*':

return firstNumber \* secondNumber

if operator == '%':

return firstNumber % secondNumber

if operator == '^':

return firstNumber \*\* secondNumber

if operator == '√':

return math.sqrt(firstNumber)

if operator == '/':

if secondNumber == 0: # Перевірка чи значення не нуль

print("Помилка: Ділення на нуль!")

return None

return firstNumber / secondNumber

except Exception as e:

print(f"Виникла помилка: {e}")

return None

def getParamValue(self, inputMessage):

while True:

try:

param = float(input(inputMessage))

return param

except ValueError:

print("Невіртний формат числа, Будь ласк, ввдедіть число знову")

def getOperator(self, inputOperatorMasseg):

while True:

operator = input(inputOperatorMasseg)

if operator in ('+', '-', '\*', '/', '%', '^', '√'):

return operator

else:

print("Неправильний оператор. Спробуйте ще раз.")

def userMenu(self):

print("\nМеню калькулятора:")

print("1. Перейти до обчислення виразу.")

print("2. Змінити налаштування.")

print("3. Переглянути історію.")

print("4. Вихід.")

def runCalculator(self):

while True:

self.userMenu()

mainMenuChoice = input("Введіть ваш вибір: ")

if mainMenuChoice == "1":

if self.memory is not None:

print(f"Число в пам'яті {self.memory}")

getParamFromMemory = input("Взяти читсло з пам'яті? Y/N \n").strip().lower()

if getParamFromMemory == 'y':

firstParam = self.memory

else:

firstParam = self.getParamValue("Введіть перше число:\n")

else:

firstParam = self.getParamValue("Введіть перше число:\n")

operator = self.getOperator("Введіть математичний оператор (+, -, \*, /, %, ^, √): ")

if operator == '√':

resultOfCalculation = round(self.calculateValue(firstParam, None, operator),self.settings["decimalPlaces"])

expression = f" {operator}{firstParam}"

print("Результат обчислення виразу:\n",operator, firstParam, "=",resultOfCalculation)

else:

secondParam = self.getParamValue("Введіть друге число:\n")

resultOfCalculation = round(self.calculateValue(firstParam, secondParam, operator),self.settings["decimalPlaces"])

expression = f"{firstParam} {operator} {secondParam}"

print("Результат обчислення виразу:\n",firstParam, operator, secondParam, "=",resultOfCalculation)

# Додавання результату до історії обчислень

self.addToHistory(expression, resultOfCalculation)

if self.settings["useMemory"]:

self.storeInMemory(resultOfCalculation)

elif mainMenuChoice == '2':

self.changeSettings()

elif mainMenuChoice == '3':

self.viewHistory()

elif mainMenuChoice == '4':

break

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

*Рис. 1 Результат виконання програми*

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

*Рис. 2 Результат виконання програми*

**Висновок:** Виконавши ці завдання, я створив простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс. Цей проект допоміг мені вивчити основний синтаксис Python і концепції, такі як введення користувача, умовні оператори, цикли та обробка помилок.