Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Лабораторна роботи №1

з дисципліни «Спеціалізовані мови програмування»

на тему

«Введення в Python»

Виконав:

Бабич О. Ю.

Перевірив:

Щербак С. С.

Львів 2024

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

**План 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.  
  
**Реалізація:**

**Папка AppSettings, файл AppSettings.py:**  
decimal\_places = 2

**Папка functions, файл functions.py:**import math

from tools.tools import get\_from\_memory

def get\_numbers():

choice = input("Використати значення з пам'яті? (так/ні): ").lower()

if choice == 'так':

num1 = get\_from\_memory()

if num1 is None:

return get\_numbers()

print(f"Використано значення з пам'яті: {num1}")

else:

try:

num1 = float(input("Введіть перше число: "))

except ValueError:

print("Помилка: потрібно ввести дійсне число.")

return get\_numbers()

try:

num2 = float(input("Введіть друге число: "))

return num1, num2

except ValueError:

print("Помилка: потрібно ввести дійсне число.")

return get\_numbers()

def get\_operator():

operator = input("Введіть оператор (+, -, \*, /, %, ^, √): ")

if operator in ['+', '-', '\*', '/', '%', '^', '√']:

return operator

else:

print("Помилка: некоректний оператор.")

return get\_operator()

def calculate(num1, num2, operator):

try:

if operator == '+':

return num1 + num2

elif operator == '-':

return num1 - num2

elif operator == '\*':

return num1 \* num2

elif operator == '/':

if num2 == 0:

raise ZeroDivisionError

return num1 / num2

elif operator == '%':

return num1 % num2

elif operator == '^':

return num1 \*\* num2

elif operator == '√':

if num1 < 0:

raise ValueError("Корінь з від'ємного числа не існує.")

return math.sqrt(num1)

except ZeroDivisionError:

print("Помилка: ділення на нуль.")

except ValueError as e:

print(f"Помилка: {e}")

**Папка GlobalVariables, файл GlobalVariables.py:**

memory = None

history = []

decimal\_places = 2

**Папка interface, файл interface.py:**

def get\_user\_input(prompt):

return input(prompt)

def display\_output(message):

print(message)

**Папка logs, файл logs.py:**

def log\_error(error\_message):

with open("error\_log.txt", "a") as log\_file:

log\_file.write(f"Помилка: {error\_message}\n")

def log\_calculation(expression, result):

with open("calculation\_log.txt", "a") as log\_file:

log\_file.write(f"Обчислення: {expression} = {result}\n")

**Папка tools, файл tools.py:**

from GlobalVariables.GlobalVariables import memory, history

def save\_to\_memory(result):

global memory

memory = result

print(f"Збережено у пам'яті: {memory}")

def get\_from\_memory():

if memory is None:

print("Помилка: пам'ять порожня.")

return None

return memory

def show\_history():

for i, entry in enumerate(history, 1):

print(f"{i}: {entry}")

def customize():

global decimal\_places

try:

decimal\_places = int(input("Введіть кількість десяткових розрядів для результатів (за замовчуванням 2): "))

except ValueError:

print("Помилка: потрібно ввести ціле число.")

customize()

**Файл main.py:**

from functions.functions import get\_operator, get\_numbers, calculate

from GlobalVariables.GlobalVariables import memory, history, decimal\_places

from tools.tools import save\_to\_memory, get\_from\_memory, show\_history, customize

def run\_calculator():

global memory, history, decimal\_places

while True:

operator = get\_operator()

if operator == '√':

choice = input("Використати значення з пам'яті? (так/ні): ").lower()

if choice == 'так':

num1 = get\_from\_memory()

if num1 is None:

continue

else:

num1 = float(input("Введіть число для операції √: "))

num2 = 0

else:

num1, num2 = get\_numbers()

result = calculate(num1, num2, operator)

if result is not None:

result = round(result, decimal\_places)

print(f"Результат: {result}")

history.append(f"{num1} {operator} {num2} = {result}" if operator != '√' else f"√{num1} = {result}")

save\_choice = input("Зберегти результат у пам'ять? (так/ні): ").lower()

if save\_choice == 'так':

save\_to\_memory(result)

choice = input("Чи хочете виконати ще одне обчислення? (так/ні): ").lower()

if choice != 'так':

break

history\_choice = input("Бажаєте переглянути історію? (так/ні): ").lower()

if history\_choice == 'так':

show\_history()

customize\_choice = input("Бажаєте налаштувати калькулятор? (так/ні): ").lower()

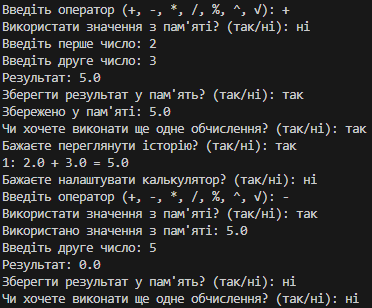
if customize\_choice == 'так':

customize()

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

run\_calculator()

**Результат виконання:**

****

**Висновок:** на цій лабораторній роботі я створив простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс.