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. РІ-21сп

Павлусь В.П

Прийняв:

Шербак С.С.

Львів-2024

**Мета лабораторної роботи:** 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.

**Виконання роботи**

**Вміст файлу functions.py:**

import math

from variables import memory, history, decimal\_point

def set\_memory(value):

global memory

memory = value

def get\_memory():

return memory

def set\_decimal\_point():

global decimal\_point

point = input("Enter the number of symbols after the decimal point: ")

if point.isalpha():

raise TypeError("Please, enter a positive integer!")

elif int(point) >= 0:

decimal\_point = int(point)

print(f"Number of symbols after decimal point - {decimal\_point}.")

else:

raise ValueError("Please, try again. Number of symbols must not be negated!")

def get\_decimal\_point():

return decimal\_point

def get\_input():

try:

num1\_input = input("Enter first number or 'm' for use value from memory: ")

num1 = get\_memory() if num1\_input.lower() == 'm' else float(num1\_input)

operator = get\_operator()

num2 = None

if operator != '√':

while True:

try:

num2\_input = input("Enter second number or 'm' for use value from memory: ")

num2 = get\_memory() if num2\_input.lower() == 'm' else float(num2\_input)

break

except ValueError:

print("Incorrect input! Please, try again!")

return num1, operator, num2

except ValueError:

print("Incorrect input! Please, try again!")

return get\_input()

def check\_operator(operator):

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

def get\_operator():

while True:

operator = input("Choose one of the operators (+, -, \*, /, ^, √, %): ")

if check\_operator(operator):

return operator

else:

print("Incorrect operator! Please, try again!")

def calculate(num1, operator, num2):

if operator == '+':

return round(num1 + num2, get\_decimal\_point())

elif operator == '-':

return round(num1 - num2, get\_decimal\_point())

elif operator == '\*':

return round(num1 \* num2, get\_decimal\_point())

elif operator == '/':

if num2 == 0:

raise ZeroDivisionError("You can't divide by zero!")

return round(num1 / num2, get\_decimal\_point())

elif operator == '^':

return round(math.pow(num1, num2), get\_decimal\_point())

elif operator == '√':

if num1 < 0:

raise ValueError("You can't get a square root from a negative number!")

return round(math.sqrt(num1), get\_decimal\_point())

elif operator == '%':

if num2 == 0:

raise ZeroDivisionError("You can't get a percentage from zero!")

return round(num1 % num2, get\_decimal\_point())

def add\_to\_history(expression, result):

history.append(f"{expression} = {result}")

def show\_history():

if history:

print("History of calculations:")

for entry in history:

print(entry)

else:

print("History is empty.")

def calculator():

while True:

num1, operator, num2 = get\_input()

result = calculate(num1, operator, num2)

if result is not None:

print(f"Result of calculations: {result}")

add\_to\_history(f"{num1} {operator} {num2 if num2 is not None else ''}", result)

if input("Do you want to save result? (y/n)").lower() == 'y':

set\_memory(result)

print(f"Result {result} was saved.")

if input("Do you want to do new calculation? (y/n)").lower() == 'n':

break

**Вміст файлу variables.py:**

memory = 0

history = []

decimal\_point = 2

**Вміст файлу interface.py:**

from functions import set\_memory, get\_memory, set\_decimal\_point, calculator, show\_history

def main():

while True:

print("\nCalculator:")

print("1. Perform calculations.")

print("2. Show history.")

print("3. Get result from memory.")

print("4. Clean memory.")

print("5. Change the number of symbols after the decimal point.")

print("6. Exit.")

choice = input("Select option: ")

if choice == '1':

try:

calculator()

except ZeroDivisionError as e:

print(e)

except ValueError as e:

print(e)

elif choice == '2':

show\_history()

elif choice == '3':

print(f"Saved value: {get\_memory()}")

elif choice == '4':

set\_memory(0)

print("Saved value cleared.")

elif choice == '5':

try:

set\_decimal\_point()

except ValueError as e:

print(e)

except TypeError as e:

print(e)

elif choice == '6':

print("Exit from calculator.")

break

else:

print("Wrong choice! Please try again!")

**Вміст файлу runner.py:**

from interface import main

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

main()

Результатом виконання є калькулятор, дію якого зображено на рис. 1.

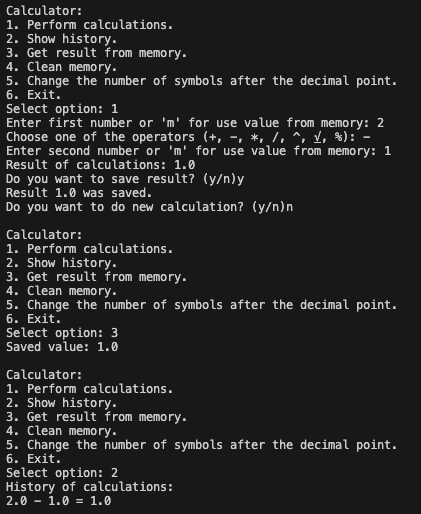


Рис. 1 – Результат роботи калькулятора

**Висновок:** У ході виконання ЛР я створив простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс.