Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра інформаційних систем та мереж

******

**ЗВІТ**

**ДО ЛАБОРАТОРНОЇ РОБОТИ №1  
з дисципліни  
«Спеціалізовані мови програмування»  
на тему:**  
«Введення в Python»

***Виконала:*** *студентка групи РІ-31  
 Діана ВІЙЧУК*

***Прийняв:****к. т. н., доцент* *Сергій ЩЕРБАК*

## МЕТА РОБОТИ

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

## ЗАВДАННЯ

*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.

## ПРОГРАМНА РЕАЛІЗАЦІЯ

Файл */constants/globalVariables.py* .

memory\_value = 0

decimal\_places = 0

history = []

Файл */functions/validation.py* .

VALID\_OPERATORS = ['+', '-', '\*', '/', '^', '%', 'sq']

def validate\_operator(operator):

return operator in VALID\_OPERATORS

def validate\_number\_input(input\_str):

try:

return float(input\_str)

except ValueError:

print("Помилка: введіть дійсне число.")

return validate\_number\_input(input("Введіть число: "))

Файл */functions/history.py* .

from constants.globalVariables import history

def add\_to\_history(operand1, operator, operand2, result):

if operand2 is None:

history.append(f"{operator}({operand1}) = {result}")

else:

history.append(f"{operand1} {operator} {operand2} = {result}")

def show\_history():

if history:

print("\nІсторія обчислень:")

for entry in history:

print(entry)

else:

print("Історія порожня.")

def clear\_history():

history.clear()

print("Історію очищено.")

Файл */functions/memory.py* .

from constants.globalVariables import memory\_value

def save\_result(result):

global memory\_value

memory\_value = result

print(f"Збережено {result} в пам'ять")

def get\_result():

if memory\_value is not None:

return memory\_value

else:

print("Немає збереженого значення в пам'яті.")

return None

def has\_memory():

return memory\_value is not None

Файл */functions/operations.py* .

import math

def addition(operand1, operand2):

return operand1 + operand2

def subtraction(operand1, operand2):

return operand1 - operand2

def multiplication(operand1, operand2):

return operand1 \* operand2

def division(operand1, operand2):

if operand2 == 0:

return "Ділення на нуль неможливе!"

return operand1 / operand2

def power(operand1, operand2):

return pow(operand1,operand2)

def square\_root(operand1):

if operand1 >= 0:

return math.sqrt(operand1)

else:

return "Неможливо обчислити квадратний корінь із від'ємного числа"

def find\_remainder(operand1, operand2):

return operand1 % operand2

Файл *AppSettings.py* .

from constants.globalVariables import decimal\_places

def update\_settings():

global decimal\_places

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

if not decimal\_places.isdigit():

decimal\_places = 2

else:

decimal\_places = int(decimal\_places)

return {

"decimal\_places": decimal\_places

}

Файл *main.py* .

from functions.operations import addition, subtraction, division, multiplication, power, square\_root, find\_remainder

from functions.memory import save\_result, get\_result, has\_memory

from functions.validation import validate\_operator, validate\_number\_input

from functions.history import add\_to\_history, show\_history, clear\_history

from AppSettings import update\_settings

def main():

change\_settings = input("Хочете змінити налаштування калькулятора? (y/n): ").lower()

if change\_settings == 'y':

settings = update\_settings()

decimal\_places = settings["decimal\_places"]

print(f"Налаштування змінено: кількість десяткових розрядів - {decimal\_places}")

else:

decimal\_places = 2

while True:

try:

use\_memory = input("Введіть перше число (або M для значення з пам'яті): ").upper()

if use\_memory == 'M':

if has\_memory():

operand1 = get\_result()

print(f"Число з пам'яті: {operand1}")

else:

print("Немає збереженого значення в пам'яті.")

continue

else:

operand1 = validate\_number\_input(use\_memory)

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

while not validate\_operator(operator):

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

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

result = 0

if operator == "sq":

result = square\_root(operand1)

else:

operand2 = validate\_number\_input(input("Введіть друге число: "))

if operator == '+':

result = addition(operand1, operand2)

elif operator == '-':

result = subtraction(operand1, operand2)

elif operator == '\*':

result = multiplication(operand1, operand2)

elif operator == '/':

while True:

result = division(operand1, operand2)

if result == "Ділення на нуль неможливе!":

print(result)

operand2 = validate\_number\_input(input("Введіть інше число для ділення: "))

else:

break

elif operator == '^':

result = power(operand1, operand2)

elif operator == '%':

result = find\_remainder(operand1, operand2)

if result is not None:

result = round(result, decimal\_places)

print(f"Результат: {result}")

save\_result(result)

add\_to\_history(operand1, operator, operand2, result)

repeat = input("Хочете повторити обчислення? (y/n): ").lower()

if repeat == 'n':

if input("Показати історію обчислень? (y/n): ").lower() == 'y':

show\_history()

break

except ValueError as e:

print(f"Error: {e}")

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

main()

## ВИСНОВКИ

На цій лабораторній роботі я створила простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс. Також ознайомилась з основним синтаксисом Python і концепціями, такими як введення користувача, умовні оператори, цикли та обробка помилок.

Посилання на GitHub: <https://github.com/Dinasi4ka/Calculator.git>