

Міністерство освіти і науки України  
Національний університет “Львівська політехніка”  
Кафедра інформаційних систем та мереж

ЗВІТ  
про виконання лабораторної роботи № 1  
**“ Введення в Python.”**  
з дисципліни **“ Спеціалізовані мови програмування”**

Виконала студентка групи ІТ-32

ДЕНИСЯК С.-М. Т.

Прийняв:

ЩЕРБАК С. С.

**Львів – 2023**

**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.

### Код програми:

```
import math
```

```
def add(num1, num2):  
    return num1 + num2
```

```
def subtract(num1, num2):  
    return num1 - num2
```

```
def multiple(num1, num2):  
    return num1 * num2
```

```
def divide(num1, num2):  
    if num2 == 0:  
        raise ZeroDivisionError("Impossible to divide")  
    return num1 / num2
```

```
def raise_to_a_power(num1, num2):  
    return num1 ** num2
```

```
def compute_square_root(num):  
    if num < 0:  
        raise ArithmeticError("Number is negative, therefore it is impossible to calculate the square root")  
    return math.sqrt(num)
```

```
def divide_by_modulo(num1, num2):
```

```
return num1 % num2
```

```
def view_history(history):
```

```
    if not history:
```

```
        print("There is nothing in history")
```

```
    else:
```

```
        print("History of results:")
```

```
        for calculation in history:
```

```
            operands, operator, result = calculation
```

```
            formatted_result = "{:.{}}f".format(result, decimal_places)
```

```
            print(f"{operands} {operator} = {formatted_result}")
```

```
def view_settings(decimal_places):
```

```
    print("\tSettings:")
```

```
    print("\tDecimal places are " + str(decimal_places))
```

```
def change_decimal_places(value):
```

```
    if value <= 0:
```

```
        raise ArithmeticError("Decimal digits can't be negative or 0")
```

```
    return value
```

```
history_of_calculations = []
```

```
decimal_places = 2
```

```
while True:
```

```
    print("Options: ")
```

```
    print("1. Add numbers")
```

```
    print("2. Subtract numbers")
```

```
    print("3. Multiply numbers")
```

```
    print("4. Divide numbers")
```

```

print("5. Raise to a power")
print("6. Divide by modulo")
print("7. Compute the square root")
print("8. View history")
print("9. Open settings")
print("0. Exit")

input_value = input("Your option is ")

if input_value in {"1", "2", "3", "4", "5", "6"}:
    first_number = float(input("Enter first number: "))
    second_number = float(input("Enter second number: "))

    try:
        operators = {"1": "+", "2": "-", "3": "*", "4": "/", "5": "**", "6": "%"}
        operator = operators[input_value]
        operation_result = None

        if input_value == "1":
            operation_result = add(first_number, second_number)
        elif input_value == "2":
            operation_result = subtract(first_number, second_number)
        elif input_value == "3":
            operation_result = multiple(first_number, second_number)
        elif input_value == "4":
            operation_result = divide(first_number, second_number)
        elif input_value == "5":
            operation_result = raise_to_a_power(first_number, second_number)
        elif input_value == "6":
            operation_result = divide_by_modulo(first_number, second_number)

        history_of_calculations.append((f"{first_number} {operator} {second_number}", "=",
operation_result))

        formatted_result = "{:.{}}f".format(operation_result, decimal_places)

```

```

        print("Result is " + formatted_result + "\n")
    except ZeroDivisionError as e:
        print(str(e) + "\n")
elif input_value == "7":
    try:
        number = float(input("Enter number: "))
        result = compute_square_root(number)
        history_of_calculations.append(("v" + str(number), "=", result))
        formatted_result = "{:.{}f}".format(result, decimal_places)
        print("Result is " + formatted_result + "\n")
    except ArithmeticError as e:
        print(str(e) + "\n")
elif input_value == "8":
    view_history(history_of_calculations)
    print()
elif input_value == "9":
    while True:
        print("\tSettings options:")
        print("\t1. View settings")
        print("\t2. Change decimal places")
        print("\t3. Clean all records")
        print("\t0. Exit from the settings mode")

        inner_input_value = input("\tYour option is ")

        if inner_input_value == "1":
            view_settings(decimal_places)
            print()
        elif inner_input_value == "2":
            new_value = int(input("\tEnter a new value for decimal places: "))

            try:
                decimal_places = change_decimal_places(new_value)
                print()

```

```
except ArithmeticError as e:
    print("\t" + str(e) + "\n")
elif inner_input_value == "3":
    history_of_calculations.clear()
    print()
elif inner_input_value == "0":
    print()
    break
else:
    print("\tYou have just entered a wrong option\n")
elif input_value == "0":
    break
else:
    print("You have just entered a wrong option\n")
```

**Висновки.** Виконавши ці завдання, ви створите простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс. Цей проект допоможе вам вивчити основний синтаксис Python і концепції, такі як введення користувача, умовні оператори, цикли та обробка помилок.