

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:

ст. гр. ІТ-32,
Троцько О. М.

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 ($\sqrt{}$) 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:

```
def __main__():
    try:
        user_history = []
        menu(user_history)

    except ZeroDivisionError as zde:
        print(f"Zero Division Error: {zde}")
    except Exception as e:
        print(f"An error occurred: {e}")
```

```

def is_integer(num):
    return int(num) == num

def get_parameters():
    parameter1 = float(input("Enter the first number: "))
    parameter2 = float(input("Enter the second number: "))
    if is_integer(parameter1) and is_integer(parameter2):
        parameter1 = int(parameter1)
        parameter2 = int(parameter2)

    return parameter1, parameter2

def get_opetaror():
    operator = input("Enter the operator ['+', '-', '*', '/', '**', '//', '%']: ")
    while operator not in ['+', '-', '*', '/', '**', '//', '%']:
        print("Invalid operator! Try again.")
        operator = input("Enter the operator: ")
    return operator

def count_sum(parameter1, parameter2):
    return parameter1 + parameter2

def count_difference(parameter1, parameter2):
    return parameter1 - parameter2

def count_product(parameter1, parameter2):
    return parameter1 * parameter2

def count_quotient(parameter1, parameter2):
    if parameter2 == 0:
        raise ZeroDivisionError("Division by zero is not allowed.")
    return parameter1 / parameter2

def count_power(parameter1, parameter2):
    return parameter1 ** parameter2

def count_square_root(parameter1, parameter2):
    if parameter1 < 0 or parameter2 < 0:
        raise ValueError("Square root of negative number is not allowed.")
    return parameter1 ** 0.5, parameter2 ** 0.5

```

```
def count_remainder(parameter1, parameter2):  
    if parameter2 == 0:  
        raise ZeroDivisionError("Division by zero is not allowed.")  
    return parameter1 % parameter2
```

```
def make_calculations(parameter1, parameter2, operator):  
    match operator:  
        case "+":  
            return count_sum(parameter1, parameter2)  
        case "-":  
            return count_difference(parameter1, parameter2)  
        case "*":  
            return count_product(parameter1, parameter2)  
        case "/":  
            return count_quotient(parameter1, parameter2)  
        case "**":  
            return count_power(parameter1, parameter2)  
        case "//":  
            return count_square_root(parameter1, parameter2)  
        case "%":  
            return count_remainder(parameter1, parameter2)
```

```
def add_to_history(user_history, result):  
    user_history.append(result)  
    return user_history
```

```
def menu(user_history):  
    decimal_places = 2  
  
    while True:  
        print("Menu:")  
        print("1 - Settings")  
        print("2 - Make Calculations")  
        print("3 - View History")  
        print("0 - Exit")  
  
        choice = str(input("Enter your choice: "))  
        match choice:  
            case "1":
```

```

print("Settings:")
print("1 - Change Decimal Places (default 2)")
print("2 - Clear History")
sub_choice = input("Enter your setting choice: ")

if sub_choice == "1":
    decimal_places = int(input("Enter the number of decimal places:
"))

elif sub_choice == "2":
    user_history = []
case "2":
    parameter1, parameter2 = get_parameters()
    operator = get_opetaror()
    calculation_result = make_calculations(parameter1, parameter2,
operator)

    if operator == "//":
        formatted_calculation_result1 = f"{calculation_result[0]:.
{decimal_places}f}"
        formatted_calculation_result2 = f"{calculation_result[1]:.
{decimal_places}f}"
        result = f"Square root of {parameter1} =
{formatted_calculation_result1}, Square root of {parameter2} =
{formatted_calculation_result2}"
    else:
        formatted_calculation_result = f"{calculation_result:.
{decimal_places}f}"
        result = f"{parameter1} {operator} {parameter2} =
{formatted_calculation_result}"

    print(result)

sub_choice = str(input("Do you want to save the result? [Y/n] "))
if sub_choice == "y" or sub_choice == "Y":
    user_history = add_to_history(user_history, result)
else:
    print("The result wasn't saved")
case "3":
    if user_history:
        print(user_history)

```

```
        else:
            print("No calculation history yet.")
    case "0":
        break
    case _:
        print("Invalid option! Try again!")

if __name__ == "__main__":
    __main__()
```

GitHub Repository: <https://github.com/trolchiha/SPL-labs.git>

Висновок: під час виконання лабораторної роботи навчилася створювати простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс. Вивчила основний синтаксис Python і концепції, такі як введення користувача, умовні оператори, цикли та обробка помилок.