

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»  
ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗВІТ  
для лабораторної роботи № 1  
з дисципліни  
«Спеціалізовані мови програмування»

Виконав:  
студент гр. ІТ-32  
Нюл В. С.

Прийняв:  
доц. каф. ІСМ  
Щербак С.С.

Львів-2023

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

**Хід виконання:**

**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:**

```
import math

history = []
memory = []

def main():
    while True:
        try:
            num1 = float(input('Введіть перше число: '))
            operator = input('Введіть оператор (+, -, *, /, ^, √, %): ')
            num2 = None
```

```

if operator != '√':
    num2 = float(input("Введіть друге число: "))

decimal_places = int(input("Кількість десяткових розрядів для результату: "))
if decimal_places < 0:
    raise ValueError('Непідтримувана кількість десяткових розрядів. Спробуйте ще раз.')

result = calculate(num1, num2, operator, decimal_places)

print(f'Результат: {result}')

memory_change(result)
show_history()

repeat = input('Виконати ще одне обчислення? (так/ні): ')
if repeat.lower() != 'так':
    break

except ValueError as e:
    print(f'Помилка: {e}')
except ZeroDivisionError:
    print('Помилка: Ділення на нуль')

def memory_change(result):
    global memory
    memory_option = input("Зберегти результат? (так/ні): ")
    if memory_option.lower() == 'так':
        memory.append(result)

    if memory:
        memory_option = input("Отримати результат з пам'яті? (так/ні): ")
        if memory_option.lower() == 'так':
            selected_position = int(
                input(f'Оберіть порядковий номер результату, який бажаєте отримати(макс: {len(memory)}): ')
            )
            if 1 <= selected_position <= len(memory):
                value = memory[selected_position-1]
                print(f'Значення з пам'яті: {value}')
            else:
                raise ValueError('Порядковий номер не дійсний.')

def show_history():
    show = input('Показати історію обчислень? (так/ні): ')
    if show.lower() == 'так':
        print("\nІсторія обчислень:")
        for entry in history:
            num1, operator, num2, result = entry
            if num2 is not None:
                print(f'{num1} {operator} {num2} = {result}')
            else:

```

```

print(f'{operator} {num1} = {result}')

def calculate(num1, num2, operator, decimal_places):
    operator_functions = {
        '+': lambda x, y: x + y,
        '-': lambda x, y: x - y,
        '*': lambda x, y: x * y,
        '/': lambda x, y: x / y if y != 0 else None,
        '^': lambda x, y: x ** y,
        '√': lambda x, _: math.sqrt(x),
        '%': lambda x, y: x % y
    }

    if operator in operator_functions:
        result = operator_functions[operator](num1, num2)
        if result is None:
            raise ZeroDivisionError
        else:
            raise ValueError('Непідтримуваний оператор. Введіть оператор, який підтримується (+, -, *, /, ^, √, %)')

    if decimal_places == 0:
        result = int(result)
    else:
        result = round(result, decimal_places)
    push_to_history(num1, num2, operator, result)

    return result

def push_to_history(num1, num2, operator, result):
    if operator == '√':
        history.append((operator, num1, result))
    else:
        history.append((num1, operator, num2, result))

if __name__ == '__main__':
    main()

```

**Посилання на GitHub-репозиторій:** <https://github.com/VasyINyul/python-labs>

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