МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

**Звіт**

До лабораторної роботи № 1

**Введення в Python**

Виконав:

Студент гр. ІТ 21-сп

Олександр Вавренчук

Прийняв:

C.C. ЩЕРБАК

Львів – 2023

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

**Хід роботи:**

**Завдання 1: Введення користувача.**

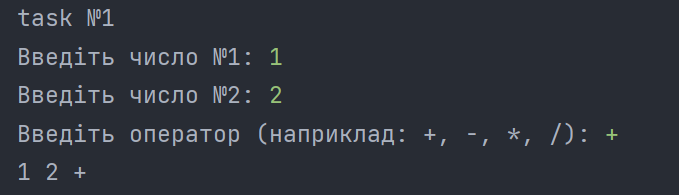
Створіть Python-програму, яка приймає введення користувача для двох чисел і оператора (наприклад, +, -, \*, /).

number\_1 = int(input("Введіть число №1: "))

number\_2 = int(input("Введіть число №2: "))

operator = str(input("Введіть оператор (наприклад: +, -, \*, /): "))

print(number\_1, number\_2, operator)



**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.

lst = ['+', '-', '\*', '/']

while True:

if operator in lst:

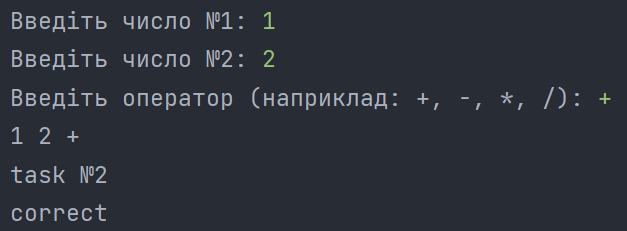
print("correct")

break

else:

print("Помилка при введені оператора, введіть ще раз")

operator = str(input("Введіть оператор (наприклад: +, -, \*, /): "))

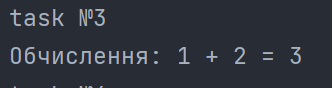


**Завдання 3: Обчислення.**

Виконайте обчислення на основі введення користувача (наприклад, додавання, віднімання, множення, ділення) і відобразіть результат.

result = eval(f"{number\_1} {operator} {number\_2}")

print(f"Обчислення: {number\_1} {operator} {number\_2} = {result}")



**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.

while True:

check = str(input("Виконати ще одне обчислення?(так/ні)"))

if check == 'так':

number\_1 = int(input("Введіть число №1: "))

number\_2 = int(input("Введіть число №2: "))

operator = str(input("Введіть оператор (наприклад: +, -, \*, /): "))

result = eval(f"{number\_1} {operator} {number\_2}")

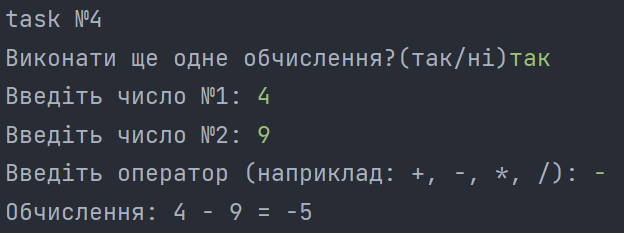
print(f"Обчислення: {number\_1} {operator} {number\_2} = {result}")

elif check == 'ні':

break

else:

print("Помилка введення, повторіть дію")



**Завдання 5: Обробка помилок.**

Реалізуйте обробку помилок для обробки ділення на нуль або інших потенційних помилок. Відобразіть відповідне повідомлення про помилку, якщо виникає помилка.

while True:

check = str(input("Виконати ще одне обчислення?(так/ні)"))

if check == 'так':

try:

number\_1 = int(input("Введіть число №1: "))

number\_2 = int(input("Введіть число №2: "))

operator = str(input("Введіть оператор (наприклад: +, -, \*, /): "))

task\_2(operator)

except ValueError:

print("Помилка вводу одного з чисел")

continue

try:

result = eval(f"{number\_1} {operator} {number\_2}")

except ZeroDivisionError:

print("Ділення на нуль неможливо")

continue

except SyntaxError:

print("Помилка арифметичної операції")

continue

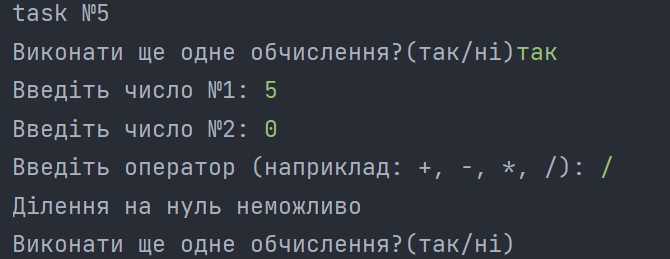
print(f"Обчислення: {number\_1} {operator} {number\_2} = {result}")

elif check == 'ні':

break

else:

print("Помилка введення, повторіть дію")



**Завдання 6: Десяткові числа.**

Змініть калькулятор так, щоб він обробляв десяткові числа (плаваючу кому) для більш точних обчислень.

while True:

check = str(input("Виконати ще одне обчислення?(так/ні)"))

if check == 'так':

try:

number\_1 = int(input("Введіть число №1: "))

number\_2 = int(input("Введіть число №2: "))

operator = str(input("Введіть оператор (наприклад: +, -, \*, /): "))

task\_2(operator)

except ValueError:

print("Помилка вводу одного з чисел")

continue

try:

result = round(eval(f"{number\_1} {operator} {number\_2}"), 2)

except ZeroDivisionError:

print("Ділення на нуль неможливо")

continue

except SyntaxError:

print("Помилка арифметичної операції")

continue

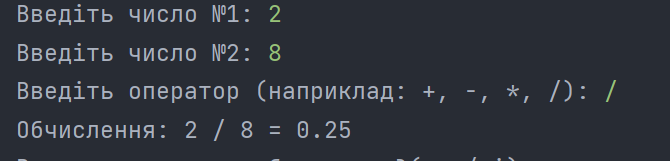
print(f"Обчислення: {number\_1} {operator} {number\_2} = {result}")

elif check == 'ні':

break

else:

print("Помилка введення, повторіть дію")



**Завдання 7: Додаткові операції**

Додайте підтримку додаткових операцій, таких як піднесення до степеня (^), квадратний корінь (√) і залишок від ділення (%).

while True:

check = str(input("Виконати ще одне обчислення?(так/ні)"))

if check == 'так':

try:

number\_1 = int(input("Введіть число №1: "))

number\_2 = int(input("Введіть число №2: "))

operator = str(input("Введіть оператор (наприклад: +, -, \*, /, ^, √, %"

"): "))

lst = ['+', '-', '\*', '/', '^', '√', '%']

while True:

if operator in lst:

break

else:

print("Помилка при введені оператора, введіть ще раз")

operator = str(input("Введіть оператор (наприклад: +, -, \*, /, ^, √, %"

"): "))

except ValueError:

print("Помилка вводу одного з чисел")

continue

try:

if operator == '^':

operator = '\*\*'

if operator != '√':

result = round(eval(f"{number\_1} {operator} {number\_2}"), 2)

message = f"Обчислення: {number\_1} {operator} {number\_2} = {result}"

else:

import math

result = (round(math.sqrt(number\_1), 2))

message = f"Обчислення: {operator} {number\_1} = {result}"

except ZeroDivisionError:

print("Ділення на нуль неможливо")

continue

except SyntaxError:

print("Помилка арифметичної операції")

continue

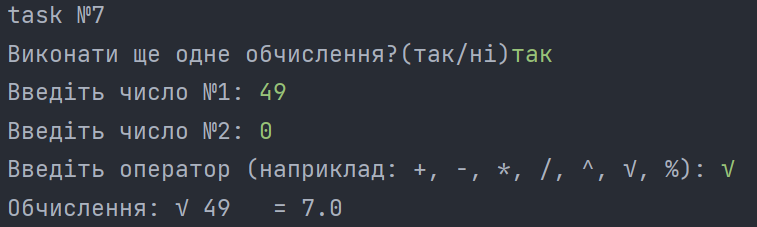
print(message)

elif check == 'ні':

break

else:

print("Помилка введення, повторіть дію")



**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.

history = []

while True:

check = str(input("Виконати ще одне обчислення?(так/ні)"))

if check == 'так':

try:

number\_1 = int(input("Введіть число №1: "))

number\_2 = int(input("Введіть число №2: "))

operator = str(input("Введіть оператор (наприклад: +, -, \*, /, ^, √, %"

"): "))

lst = ['+', '-', '\*', '/', '^', '√', '%']

while True:

if operator in lst:

if operator == '^':

operator = '\*\*'

break

else:

print("Помилка при введені оператора, введіть ще раз")

operator = str(input("Введіть оператор (наприклад: +, -, \*, /, ^, √, %"

"): "))

except ValueError:

print("Помилка вводу одного з чисел")

continue

try:

if operator != '√':

result = round(eval(f"{number\_1} {operator} {number\_2}"), 2)

message = f"Обчислення: {number\_1} {operator} {number\_2} = {result}"

else:

import math

result = (round(math.sqrt(number\_1), 2))

message = f"Обчислення: {operator} {number\_1} = {result}"

except ZeroDivisionError:

print("Ділення на нуль неможливо")

continue

except SyntaxError:

print("Помилка арифметичної операції")

continue

print(message)

save = input("Зберегти результат:(введіть так, якщо бажаєте) ")

if save == 'так':

history.append(result)

else:

continue

output = input("Показати результати:(введіть так, якщо бажаєте) ")

if output == 'так':

print([result for result in history])

else:

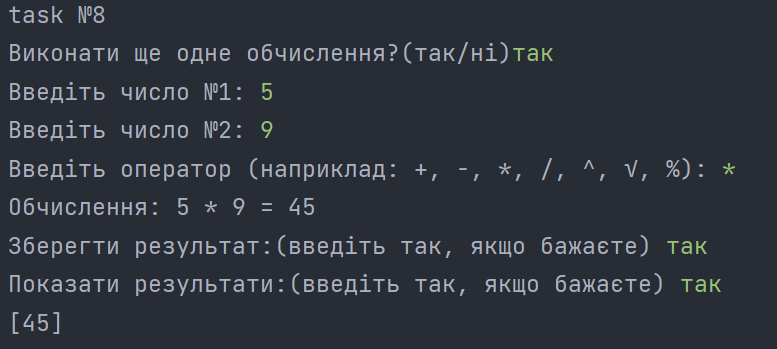
continue

elif check == 'ні':

break

else:

print("Помилка введення, повторіть дію")



**Завдання 9: Історія обчислень.**

Створіть журнал, який зберігає історію попередніх обчислень, включаючи вираз і результат. Дозвольте користувачам переглядати історію своїх обчислень.

history = []

while True:

check = str(input("Виконати ще одне обчислення?(так/ні)"))

if check == 'так':

try:

number\_1 = int(input("Введіть число №1: "))

number\_2 = int(input("Введіть число №2: "))

operator = str(input("Введіть оператор (наприклад: +, -, \*, /, ^, √, %"

"): "))

lst = ['+', '-', '\*', '/', '^', '√', '%']

while True:

if operator in lst:

if operator == '^':

operator = '\*\*'

break

else:

print("Помилка при введені оператора, введіть ще раз")

operator = str(input("Введіть оператор (наприклад: +, -, \*, /, ^, √, %"

"): "))

except ValueError:

print("Помилка вводу одного з чисел")

continue

try:

if operator != '√':

result = round(eval(f"{number\_1} {operator} {number\_2}"), 2)

message = f"Обчислення: {number\_1} {operator} {number\_2} = {result}"

else:

import math

result = (round(math.sqrt(number\_1), 2))

message = f"Обчислення: {operator} {number\_1} = {result}"

except ZeroDivisionError:

print("Ділення на нуль неможливо")

continue

except SyntaxError:

print("Помилка арифметичної операції")

continue

print(message)

history.append(message)

output = input("Показати історію:(введіть так, якщо бажаєте) ")

if output == 'так':

print([result for result in history])

else:

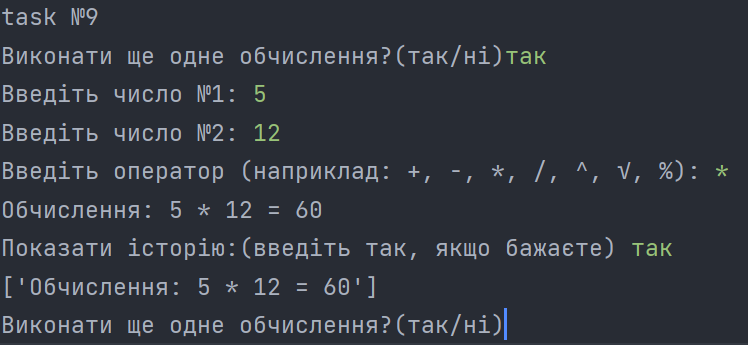
continue

elif check == 'ні':

break

else:

print("Помилка введення, повторіть дію")



**Завдання 10: Налаштування користувача.**

Надайте користувачам можливість налаштувати поведінку калькулятора, таку як зміну кількості десяткових розрядів, які відображаються, або налаштування функцій пам'яті.

history = []

while True:

check = str(input("Виконати ще одне обчислення?(так/ні)"))

if check == 'так':

try:

count = int(input("Введіть кількість знаків після коми: "))

number\_1 = int(input("Введіть число №1: "))

number\_2 = int(input("Введіть число №2: "))

operator = str(input("Введіть оператор (наприклад: +, -, \*, /, ^, √, %"

"): "))

lst = ['+', '-', '\*', '/', '^', '√', '%']

while True:

if operator in lst:

if operator == '^':

operator = '\*\*'

break

else:

print("Помилка при введені оператора, введіть ще раз")

operator = str(input("Введіть оператор (наприклад: +, -, \*, /, ^, √, %"

"): "))

except ValueError:

print("Помилка вводу одного з чисел")

continue

try:

if operator != '√':

result = round(eval(f"{number\_1} {operator} {number\_2}"), count)

message = f"Обчислення: {number\_1} {operator} {number\_2} = {result}"

else:

import math

result = (round(math.sqrt(number\_1), 2))

message = f"Обчислення: {operator} {number\_1} = {result}"

except ZeroDivisionError:

print("Ділення на нуль неможливо")

continue

except SyntaxError:

print("Помилка арифметичної операції")

continue

print(message)

history.append(message)

output = input("Показати історію:(введіть так, якщо бажаєте) ")

if output == 'так':

print([result for result in history])

else:

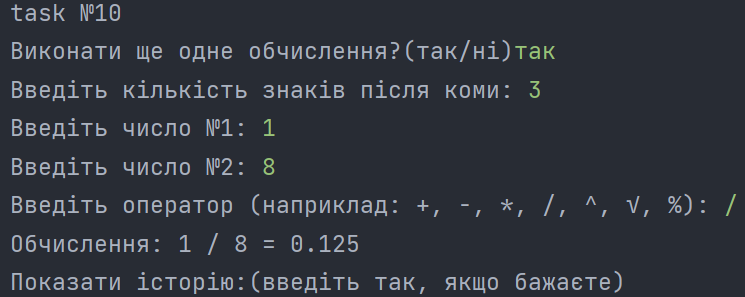
continue

elif check == 'ні':

break

else:

print("Помилка введення, повторіть дію")



**Висновок:** На даній лабораторній роботі я cтворив консольну програму-калькулятор за допомогою основних синтаксичних конструкцій Python, з іншим завданням на заміну тестуванню та валідації.