МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Лабораторна робота №1

з курсу

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконала студентка

групи ІТ-21сп

**Філіпчук М.С.**

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.

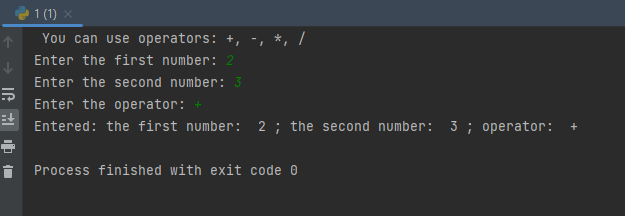
**План роботи**

Завдання 1. Введення користувача

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

print(" You can use operators: +, -, \*, /")  
a = int(input("Enter the first number: "))  
b = int(input("Enter the second number: "))  
oper = input("Enter the operator: ")  
print("Entered: the first number: ", a, "; the second number: ", b, "; operator: ", oper)

На рисунку 1 зображено код виконання програми.



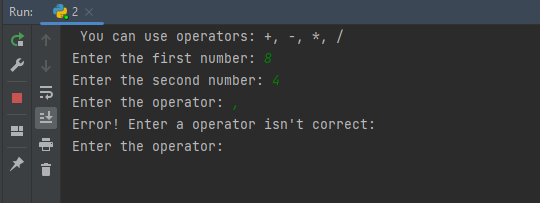
*Рис.1 Код виконання програми*

Завдання 2. Перевірка оператора.

Перевірити чи введений оператор є дійсним (тобто одним із +, -, \*, /). Якщо ні, відобразити повідомлення про помилку і попросити користувача ввести дійсний оператор.

print(" You can use operators: +, -, \*, /")  
a = int(input("Enter the first number: "))  
b = int(input("Enter the second number: "))  
while True:  
 oper = input("Enter the operator: ")  
 if oper in ('+', '-', '\*', '/'):  
 print("Entered: the first number: ", a, ";Entered: the second number: ", b, "; Operator: ", oper)  
 break  
 else:  
 print("Error! Enter a operator isn't correct: ")

На рисунку 2 зображено результат виконання задачі.



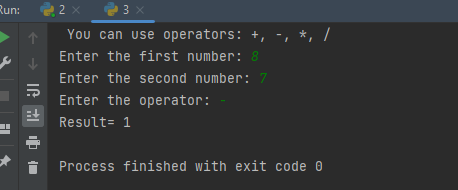
*Рис.2 Результат виконання задачі*

Завдання 3. Обчислення.

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

print(" You can use operators: +, -, \*, /")  
a = int(input("Enter the first number: "))  
b = int(input("Enter the second number: "))  
while True:  
 oper = input("Enter the operator: ")  
 if oper not in ('+', '-', '\*', '/'):  
 print("Error! Enter a operator isn't correct: ")  
 continue  
 if oper == '+':  
 res = a + b  
 elif oper == '-':  
 res = a - b  
 elif oper == '\*':  
 res = a \* b  
 elif oper == '/':  
 res = a / b  
 print("Result=", res)  
 break

На рисунку 3 зображено обчислення.



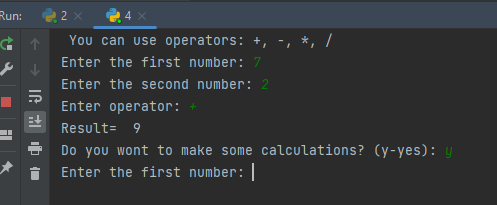
*Рис.3 Виконання програми*

Завдання 4. Повторення обчислень.

Запитати користувача, чи він хоче виконати ще одне обчислення. Якщо так, дозволити йому ввести нові числа і оператор. Якщо ні, вийти з програми.

print(" You can use operators: +, -, \*, /")  
while True:  
 try:  
 a = int(input("Enter the second number: "))  
 b = int(input("Enter the second number: "))  
 oper = input("Enter operator: ")  
 if oper not in ('+', '-', '\*', '/'):  
 print("Error! Please enter correct operator: ")  
 continue  
 if oper == '+':  
 res = a + b  
 elif oper == '-':  
 res = a - b  
 elif oper == '\*':  
 res = a \* b  
 elif oper == '/':  
 res = a / b  
 print("Result= ", res)  
 ques = input("Do you wont to make some calculations? (y-yes): ")  
 if ques.lower() != 'y':  
 break  
 except ValueError:  
 print("Error! Please try again.")

На рисунку 4 вивід роботи програми.



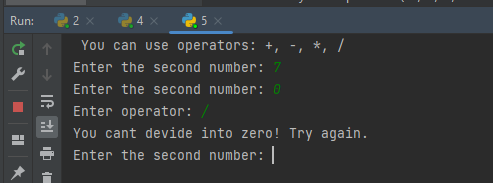
*Рис.4 Вивід роботи програми*

Завдання 5. Обробка помилок.

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

print(" You can use operators: +, -, \*, /")  
while True:  
 try:  
 a = int(input("Enter the second number: "))  
 b = int(input("Enter the second number: "))  
 oper = input("Enter operator: ")  
 if oper not in ('+', '-', '\*', '/'):  
 print("Error! Please enter correct operator: ")  
 continue  
  
 if oper == '+':  
 res = a + b  
 elif oper == '-':  
 res = a - b  
 elif oper == '\*':  
 res = a \* b  
 elif oper == '/':  
 if b == 0:  
 print("You cant devide into zero! Try again.")  
 continue  
 res = a / b  
  
 print("Result= ", res)  
 ques = input("Do you wont to make some calculations? (y-yes): ")  
 if ques.lower() != 'y':  
 break  
 except ValueError:  
 print("The symbol isn't a number, Try again!")

На рисунку 5 зображено результат програми.



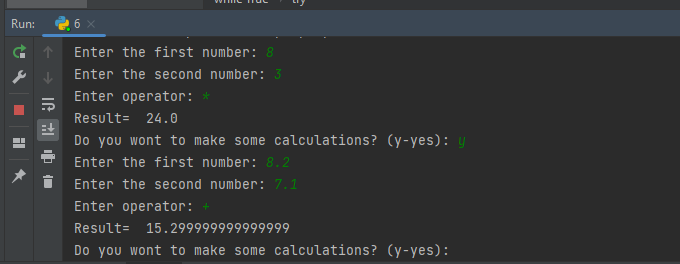
*Рис.5 Результат програм*

Завдання 6. Десяткові числа.

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

print(" You can use operators: +, -, \*, /")  
while True:  
 try:  
 a = float(input("Enter the second number: "))  
 b = float(input("Enter the second number: "))  
 oper = input("Enter operator: ")  
  
 if oper not in ('+', '-', '\*', '/'):  
 print("Error! Please enter correct operator: ")  
 continue  
  
 if oper == '+':  
 res = a + b  
 elif oper == '-':  
 res = a - b  
 elif oper == '\*':  
 res = a \* b  
 elif oper == '/':  
 if b == 0:  
 print("You cant devide into zero! Try again.")  
 continue  
 res = a / b  
 print("Result= ", res)  
 ques = input("Do you wont to make some calculations? (y-yes): ")  
 if ques.lower() != 'y':  
 break  
 except ValueError:  
 print("The symbol isn't a number, Try again!")

На рисунку 6 вивід виконаного завдання.



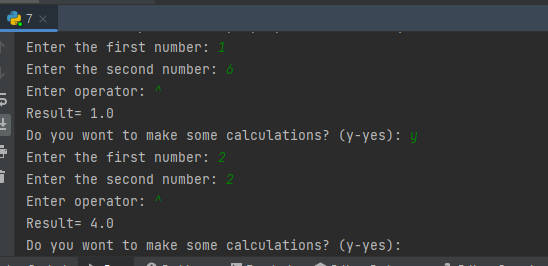
*Рис.6 Вивід виконаного завдання*

Завдання 7. Додаткові операції.

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

import math  
print(" You can use operators: +, -, \*, / Note: The square root is taken from the first entered number.")  
while True:  
 try:  
 a = float(input("Enter the second number: "))  
 b = float(input("Enter the second number: "))  
 oper = input("Enter operator: ")  
  
 if oper not in ('+', '-', '\*', '/', '%', '^','√'):  
 print("Error! Please enter correct operator: ")  
 continue  
  
 if oper == '+':  
 res = a + b  
 elif oper == '-':  
 res = a - b  
 elif oper == '\*':  
 res = a \* b  
 elif oper == '%':  
 res = a % b  
 elif oper == '^':  
 res = a \*\* b  
 elif oper == '/':  
 if b == 0:  
 print("The operation is impossible! Try again.")  
 continue  
 res = a / b  
 elif oper == '√':  
 if a < 0:  
 print("The operation is impossible! Try again.. ")  
 continue  
 res = math.sqrt(a)  
  
 print("Result=", res)  
  
 ques = input("Do you wont to make some calculations? (y-yes): ")  
 if ques.lower() != 'y':  
 break  
 except ValueError:  
 print("The symbol isn't a number, Try again!")

На рисунку 7 відображення результату програми.



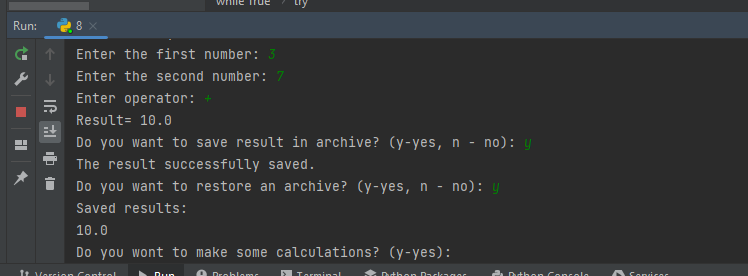
*Рис.7 Результат виконання програми*

Завдання 8. Функція пам'яті.

Реалізувати функцію пам'яті, яка дозволяє користувачам зберігати і відновлювати результати. Додати можливості для зберігання та отримання значень з пам'яті.

import math  
memoryres = []  
print("\nYou can use operators: +, -, \*, / ^, √, % \nNote: The square root is taken from the first entered number.")  
while True:  
 try:  
 a = float(input("Enter the second number: "))  
 b = float(input("Enter the second number: "))  
 oper = input("Enter operator: ")  
  
 if oper not in ('+', '-', '\*', '/', '%', '^', '√', 'M','R'):  
 print("Error! Enter valid operator: ")  
 continue  
  
 if oper == '+':  
 res = a + b  
 elif oper == '-':  
 res = a - b  
 elif oper == '\*':  
 res = a \* b  
 elif oper == '%':  
 res = a % b  
 elif oper == '^':  
 res = a \*\* b  
 elif oper == '/':  
 if b == 0:  
 print("The operation is impossible! Try again.")  
 continue  
 res = a / b  
 elif oper == '√':  
 if a < 0:  
 print("The operation is impossible! Try again. ")  
 continue  
 res = math.sqrt(a)  
  
 print("Result=", res)  
 ques3 = input("Do you want to save result in archive? (y-yes, n - no): ")  
 if ques3.lower() == 'y':  
 memoryres.append(res)  
 print("The result successfully saved.")  
  
 ques2 = input("Do you want to restore an archive? (y-yes, n - no): ")  
 if ques2.lower() == 'y':  
 if memoryres:  
 print("Saved results:")  
 for result in memoryres:  
 print(result)  
 else:  
 print("Archive is empty!")  
  
 ques1 = input("Do you wont to make some calculations? (y-yes): ")  
 if ques1.lower() != 'y':  
 break  
 except ValueError:  
 print("The symbol isn't a number, Try again!")

На рисунку 8 зображено виконання програми.



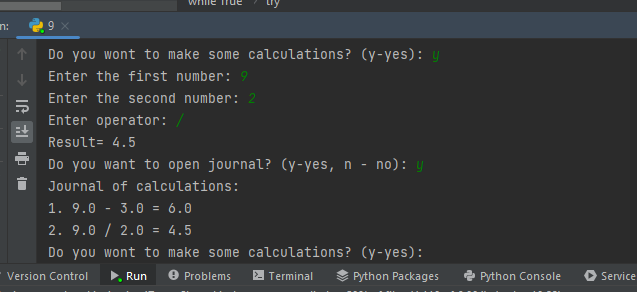
*Рис.8 Код виконання програми*

Завдання 9. Історія обчислень.

Створити журнал, який зберігає історію попередніх обчислень, включаючи вираз і результат. Дозволити користувачам переглядати історію своїх обчислень.

import math  
memoryres = []  
print("\nYou can use operators: +, -, \*, / ^, √, % "  
 "\nNote: The square root is taken from the first entered number.")  
while True:  
 try:  
 a = float(input("Enter the second number: "))  
 b = float(input("Enter the second number: "))  
 oper = input("Enter operator: ")  
  
 if oper not in ('+', '-', '\*', '/', '%', '^', '√', 'M','R'):  
 print("Error! Enter valid operator: ")  
 continue  
  
 if oper == '+':  
 res = a + b  
 elif oper == '-':  
 res = a - b  
 elif oper == '\*':  
 res = a \* b  
 elif oper == '%':  
 res = a % b  
 elif oper == '^':  
 res = a \*\* b  
 elif oper == '/':  
 if b == 0:  
 print("Operation is not correct. Try again.")  
 continue  
 res = a / b  
 elif oper == '√':  
 if a < 0:  
 print("The operation is impossible! Try again. ")  
 continue  
 res = math.sqrt(a)  
  
 expression = f"{a} {oper} {b}"  
 memoryres.append((expression, res))  
  
 print("Result=", res)  
  
 ques2 = input("Do you want to open journal? (y-yes, n - no): ")  
 if ques2.lower() == 'y':  
 if memoryres:  
 print("Journal of calculations:")  
 for i, (expression, result) in enumerate(memoryres, start=1):  
 print(f"{i}. {expression} = {result}")  
 else:  
 print("Journal is empty!")  
  
 ques1 = input("Do you wont to make some calculations? (y-yes): ")  
 if ques1.lower() != 'y':  
 break  
 except ValueError:  
 print("The symbol isn't a number, Try again!")

На рисунку 9 зображено роботу калькулятора.



*Рис.9 Вивід роботи програми*

Завдання 10. Налаштування користувача.

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

Було додано можливість користувачу обирати кількість десяткових розрядів та очищувати історію обчислень.

import math

memoryres = []

print(

"\nYou can use operators: +, -, \*, /, ^, √, % "

"\nNote: The square root is taken from the first entered number.")

while True:

try:

a = float(input("Enter the second number: "))

b = float(input("Enter the second number: "))

oper = input("Enter operator: ")

if oper not in ('+', '-', '\*', '/', '%', '^', '√', 'M', 'R'):

print("Error! Enter valid operator: ")

continue

if oper == '+':

res = a + b

elif oper == '-':

res = a - b

elif oper == '\*':

res = a \* b

elif oper == '%':

res = a % b

elif oper == '^':

res = a \*\* b

elif oper == '/':

if b == 0:

print("The operation is impossible! Try again.")

continue

res = a / b

elif oper == '√':

if a < 0:

print("The operation is impossible! Try again. ")

continue

res = math.sqrt(a)

expression = f"{a} {oper} {b}"

memoryres.append((expression, res))

decimal\_places = int(input("Enter the number of decimal places: ")) #кількість десяткових розрядів:

formatted\_res = "{:.{}f}".format(res, decimal\_places)

print(f"Result= {formatted\_res}")

ques3 = input("Do you want to clear a journal? (y-yes, n - no): ")

if ques3.lower() == 'y':

memoryres.clear()

print("Journal cleaned.")

ques2 = input("Do you wwant to open journal? (y-yes, n - no): ")

if ques2.lower() == 'y':

if memoryres:

print("Journal:")

for i, (expression, res) in enumerate(memoryres, start=1):

print(f"{i}. {expression} = {formatted\_res}")

else:

print("Archive is empty!")

ques1 = input("Do you wont to make some calculations? (y-yes): ")

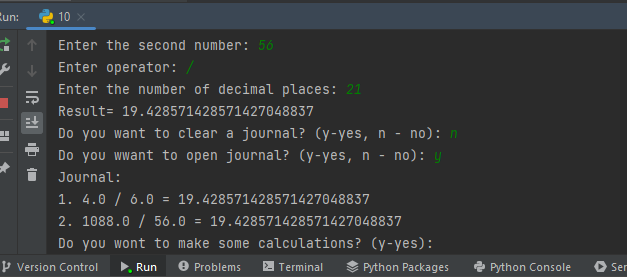
if ques1.lower() != 'y':

break

except ValueError:

print("The symbol isn't a number, Try again!")

На рисунку 10 вивід роботи програми.



*Рис.10 Виконання завдання*

**Висновок:** під час виконання лабораторної роботи я створила простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс. Цей проєкт допоміг мені вивчити основний синтаксис Python і концепції, такі як введення користувача, умовні оператори, цикли та обробка помилок.