**Лабораторна робота №4**

**Тема:** Зворотний польський запис та обчислення математичного виразу.

**Мета:** Використовуючи теоретичне підґрунтя про зворотний польський запис розробити програму на вхід якої подається математичний вираз, що має довільний набір операндів, операторів та дужок, на виході програма обчислює результат математичного виразу.

**Завдання до лабораторної роботи**

Використовуючи теоретичне відомості розробити програму яка на вхід отримує математичний вираз з довільною кількістю операндів, операторів та дужок. В першу чергу сформувати послідовність символів у ЗПН. На другому етапі виконання лабораторної роботи вирахувати результат послідовності, що була сформована, використовуючи алгоритм запису математичного виразу у ЗПН.

**Хід роботи**

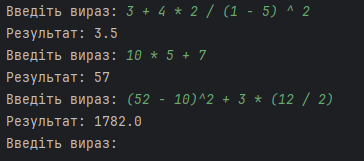
Ознайомившись з теоретичними відомостями та завданням розпочав виконувати лабораторну роботу.

Створив новий файл Lab\_04.py та почав писати код.

Реалізував 2 основні функції handleExpression() – для створення потрібного списку для обчислення із строки та calculate() – для обчислення виразу:

|  |
| --- |
| def handleExpression(expression):  output = []  stack = []   priority = {'^': 3, '\*': 2, '/': 2, '+': 1, '-': 1}   number = ""   for c in expression:  if c == ' ':  continue   if c.isdigit():  number += c  else:   if number != "":  output.append(int(number))  number = ""   if c in priority:  while len(stack) > 0 and stack[-1] != '(' and priority[stack[-1]] >= priority[c]:  output.append(stack.pop())  stack.append(c)   elif c == '(':  stack.append(c)  elif c == ')':  while len(stack) > 0 and stack[-1] != '(':  output.append(stack.pop())  stack.pop()    if number != "":  output.append(int(number))   while len(stack) > 0:  output.append(stack.pop())   return output  def calculate(output):  stack = []  for elem in output:  if elem not in ["+", "-", "\*", "/", "^"]:  stack.append(elem)  continue   second = stack.pop()  first = stack.pop()   match (elem):  case "+":  stack.append(first + second)  case "-":  stack.append(first - second)  case "\*":  stack.append(first \* second)  case "/":  stack.append(first / second)  case "^":  stack.append(first \*\* second)   return stack[0]  def main():  while True:  expression = input("Введіть вираз: ")   output = handleExpression(expression)   print("Результат:", calculate(output))   main() |

Результат виконання програми:



Обчислення виконані правильно, отже програма працює коректно.

**Висновок:** На цій лабораторній роботі я використовуючи теоретичне підґрунтя про зворотний польський запис розробив програму на вхід якої подається математичний вираз, що має довільний набір операндів, операторів та дужок, на виході програма обчислює результат математичного виразу.