Технології програмування

Лабораторна робота №4

**Зворотний польський запис та обчислення математичного виразу**

**Мета роботи**: 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.

Хід роботи

Ознайомившись із теоретичними відомостями та завданням, розпочав виконання лабораторної роботи. Створив новий файл Lab\_04.py і написав код для реалізації двох основних функцій:

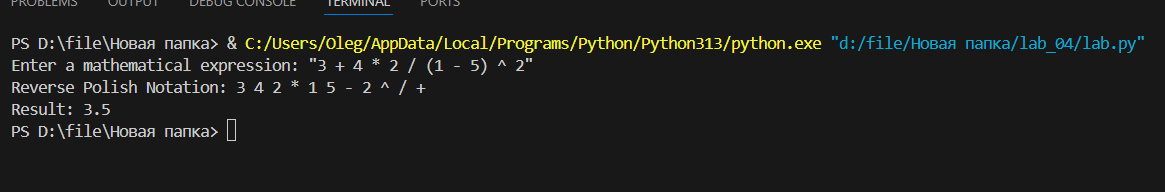
rpn() – перетворює рядок у список для обчислення в форматі зворотного польського запису (ЗПЗ).

calculate() – обчислює результат математичного виразу за допомогою ЗПЗ.

Ось реалізація:

|  |
| --- |
| def rpn(expression):      result = []      stack = []      precedence = {'^': 3, '\*': 2, '/': 2, '+': 1, '-': 1}      current\_number = ""      for char in expression:          if char.isspace():              continue          if char.isdigit():              current\_number += char          else:              if current\_number:                  result.append(int(current\_number))                  current\_number = ""              if char in precedence:                  while stack and stack[-1] != '(' and precedence.get(stack[-1], 0) >= precedence[char]:                      result.append(stack.pop())                  stack.append(char)              elif char == '(':                  stack.append(char)              elif char == ')':                  while stack and stack[-1] != '(':                      result.append(stack.pop())                  if stack and stack[-1] == '(':                      stack.pop()      if current\_number:          result.append(int(current\_number))      while stack:          result.append(stack.pop())      return result  def calculate(rpn\_list):      stacks = []      for token in rpn\_list:          if isinstance(token, int):              stacks.append(token)          else:              if len(stacks) < 2:                  raise ValueError("Invalid expression: insufficient operands for the operator.")              right = stacks.pop()              left = stacks.pop()              if token == '+':                  stacks.append(left + right)              elif token == '-':                  stacks.append(left - right)              elif token == '\*':                  stacks.append(left \* right)              elif token == '/':                  if right == 0:                      raise ZeroDivisionError("Division by zero is not allowed.")                  stacks.append(left / right)              elif token == '^':                  stacks.append(left \*\* right)      if len(stacks) != 1:          raise ValueError("Invalid expression: extra operands remain.")      return stacks[0]  def main():      try:          user\_expression = input("Enter a mathematical expression: ")          rpn\_result = rpn(user\_expression)          print(f"Reverse Polish Notation: {' '.join(map(str, rpn\_result))}")          print(f"Result: {calculate(rpn\_result)}")      except Exception as error:          print(f"Error: {error}")  if \_\_name\_\_ == "\_\_main\_\_":      main() |

Результат виконання програми:



Висновок: У рамках виконання цієї лабораторної роботи, спираючись на теоретичні основи зворотного польського запису, я розробив програму, яка приймає математичний вираз з будь-яким набором операндів, операторів та дужок. Програма обчислює результат цього виразу, використовуючи алгоритм перетворення виразу в зворотний польський запис та його подальше оцінювання.