**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**

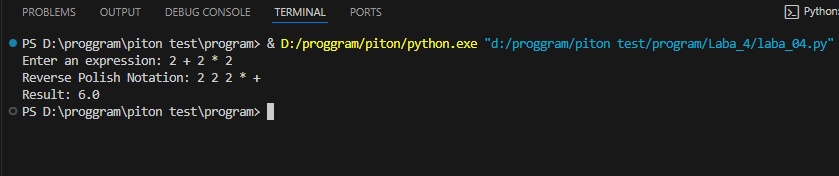
**Створено клас ReversePolishNotation для роботи з оберненою польською нотацією (ОПН):**

* to\_rpn: перетворює інфіксний вираз у ОПН.
* evaluate\_rpn: обчислює значення виразу в ОПН.

**Робота програми:**

* Користувач вводить математичний вираз.
* Вираз перетворюється в ОПН.
* Обчислюється результат і виводиться.

**Приклад:**



Текст програми:

|  |
| --- |
| class ReversePolishNotation:  def \_\_init\_\_(self):  self.operators = {'+': 1, '-': 1, '\*': 2, '/': 2, '^': 3}  def to\_rpn(self, expression):  output = []  stack = []  for token in expression.split():  if token.isnumeric():  output.append(token)  elif token == '(':  stack.append(token)  elif token == ')':  while stack and stack[-1] != '(':  output.append(stack.pop())  if stack and stack[-1] == '(':  stack.pop()  else:  while (stack and stack[-1] != '(' and  self.operators.get(stack[-1], 0) >= self.operators.get(token, 0)):  output.append(stack.pop())  stack.append(token)  while stack:  output.append(stack.pop())  return output  def evaluate\_rpn(self, rpn\_expression):  stack = []  for token in rpn\_expression:  if token.isnumeric():  stack.append(float(token))  else:  b = stack.pop()  a = stack.pop()  if token == '+':  stack.append(a + b)  elif token == '-':  stack.append(a - b)  elif token == '\*':  stack.append(a \* b)  elif token == '/':  stack.append(a / b)  elif token == '^':  stack.append(a \*\* b)  return stack[0]  if \_\_name\_\_ == "\_\_main\_\_":  expression = input("Enter an expression: ")  rpn\_converter = ReversePolishNotation()  rpn = rpn\_converter.to\_rpn(expression)  print("Reverse Polish Notation:", " ".join(rpn))  result = rpn\_converter.evaluate\_rpn(rpn)  print("Result:", result) |

**Висновок:** Під час виконання роботи використовуючи теоретичне підґрунтя про зворотний польський запис зроблено програму на вхід якої подається математичний вираз, що має довільний набір операндів, операторів та дужок, на виході програма обчислює результат математичного виразу.