Министерство науки и высшего образования Российской Федерации

**Муромский институт (филиал)**

Федерального государственного бюджетного образовательного учреждения высшего образования

**«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»**

Факультет ИТР

Кафедра ПИн

*ЛАБОРАТОРНАЯ*

*РАБОТА №7*

# По Теория автоматов и формальных языков

Тема Трансляция арифметических выражений

Руководитель

Кульков Я.Ю.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИН - 120

(группа)

Комаров Р.Д.

(фамилия, инициалы)

(подпись) (дата)

Муром 2022

**Лабораторная работа № 7**

**Тема:** Трансляция арифметических выражений

**Цели и задачи:** Изучение методов трансляции арифметических и логических выражении.

**Ход работы:**

**Реализация трансляции арифметического выражения:**

public static Stack<string> E = new Stack<string>(); // для хранения операндов

public static Stack<string> T = new Stack<string>(); // для хранения знаков

public static void kID(ListWithDuplicates keyValuePairs, List<string> keyWords, List<string> separators, List<string> literals, List<string> variables)

{

E.Push(lexeme);

Next(keyValuePairs, keyWords, separators, literals, variables);

}

public static List<string> listMatrix = new List<string>();

public static bool kOP(ListWithDuplicates keyValuePairs, List<string> keyWords, List<string> separators, List<string> literals, List<string> variables)

{

if(E.Count > 1)

{

string operand = T.Pop();

string firstOp = E.Pop();

string secondOp = E.Pop();

string matrix = $"Операция: {listMatrix.Count +1}";

listMatrix.Add($"{matrix}: {operand} {secondOp} {firstOp}");

E.Push(matrix);

return true;

}

else

{

MessageBox.Show("Ошибка выражения","Анализ сложных выражений", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

return false;

}

}

public static void D1(ListWithDuplicates keyValuePairs, List<string> keyWords, List<string> separators, List<string> literals, List<string> variables)

{

T.Push(lexeme);

Next(keyValuePairs, keyWords, separators, literals, variables);

}

public static bool D2(ListWithDuplicates keyValuePairs, List<string> keyWords, List<string> separators, List<string> literals, List<string> variables)

{

if (!kOP(keyValuePairs, keyWords, separators, literals, variables))

return false;

else

{

T.Push(lexeme);

Next(keyValuePairs, keyWords, separators, literals, variables);

return true;

}

}

public static void D3(ListWithDuplicates keyValuePairs, List<string> keyWords, List<string> separators, List<string> literals, List<string> variables)

{

T.Pop();

Next(keyValuePairs, keyWords, separators, literals, variables);

}

public static bool D4(ListWithDuplicates keyValuePairs, List<string> keyWords, List<string> separators, List<string> literals, List<string> variables)

{

if (!kOP(keyValuePairs, keyWords, separators, literals, variables)){

return false;

}

else

{

return true;

}

}

public static bool D5()

{

MessageBox.Show("Ошибка выражения", "Анализ сложных выражений", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

return false;

}

public static bool D6()

{

MessageBox.Show("Успех");

foreach(string item in listMatrix)

{

Form1.rMatrixText.Text += $"{item}\n";

}

return true;

}

public static bool expr(ListWithDuplicates keyValuePairs, int k, List<string> keyWords, List<string> separators, List<string> literals, List<string> variables)

{

if (number < keyValuePairs.Count)

{

if (variables.Contains(lexeme) || literals.Contains(lexeme))

{

kID(keyValuePairs, keyWords, separators, literals, variables);

if(!expr(keyValuePairs, number, keyWords, separators, literals, variables))

{

return false;

}

}

else if (lexeme == "\*" || lexeme == "/" || lexeme == "-" || lexeme == "+" || lexeme == "(" || lexeme == ")" || lexeme == ";")

{

if (lexeme == "(")

{

D1(keyValuePairs, keyWords, separators, literals, variables);

if (!expr(keyValuePairs, number, keyWords, separators, literals, variables))

{

return false;

}

}

else if(lexeme == "-" || lexeme == "+")

{

if(T.Count == 0 || T.Peek() == "(")

{

D1(keyValuePairs, keyWords, separators, literals, variables);

if (!expr(keyValuePairs, number, keyWords, separators, literals, variables))

{

return false;

}

}

else if (T.Peek() == "+" || T.Peek() == "-")

{

if(!D2(keyValuePairs, keyWords, separators, literals, variables)){

return false;

}

if (!expr(keyValuePairs, number, keyWords, separators, literals, variables))

{

return false;

}

}

else if (T.Peek() == "\*" || T.Peek() == "/")

{

if (!D4(keyValuePairs, keyWords, separators, literals, variables))

return false;

if (!expr(keyValuePairs, number, keyWords, separators, literals, variables))

{

return false;

}

}

else

{

return false;

}

}

else if(lexeme == "\*" || lexeme == "/")

{

if (T.Count == 0 || T.Peek() == "(" || T.Peek() == "+" || T.Peek() == "-")

{

D1(keyValuePairs, keyWords, separators, literals, variables);

if (!expr(keyValuePairs, number, keyWords, separators, literals, variables))

{

return false;

}

}

else if (T.Peek() == "\*" || T.Peek() == "/")

{

if (!D2(keyValuePairs, keyWords, separators, literals, variables))

return false;

if (!expr(keyValuePairs, number, keyWords, separators, literals, variables))

{

return false;

}

}

else

{

return false;

}

}

else if (lexeme == ")")

{

if(T.Count == 0)

{

if (!D5())

{

return false;

}

}

else if(T.Peek() == "(")

{

D3(keyValuePairs, keyWords, separators, literals, variables);

if (!expr(keyValuePairs, number, keyWords, separators, literals, variables))

{

return false;

}

}

else if (T.Peek() == "+" || T.Peek() == "-" || T.Peek() == "\*" || T.Peek() == "/")

{

if (!D4(keyValuePairs, keyWords, separators, literals, variables))

return false;

if (!expr(keyValuePairs, number, keyWords, separators, literals, variables))

{

return false;

}

}

}

else if (lexeme == ";")

{

if(T.Count == 0)

{

if (D6())

{

return true;

}

}

else if(lexeme == "(")

{

if (!D5())

{

return false;

}

}

else if(T.Peek() == "+" || T.Peek() == "-" || T.Peek() == "\*" || T.Peek() == "/")

{

if (!D4(keyValuePairs, keyWords, separators, literals, variables))

return false;

if (!expr(keyValuePairs, number, keyWords, separators, literals, variables))

{

return false;

}

}

else

{

if (!D5())

{

return false;

}

}

}

else

{

return false;

}

}

listMatrix.Clear();

}

else

{

Errors.errorSintax();

return false;

}

if (true)

{

return true;

}

else

{

return false;

}

}

Работа программы:

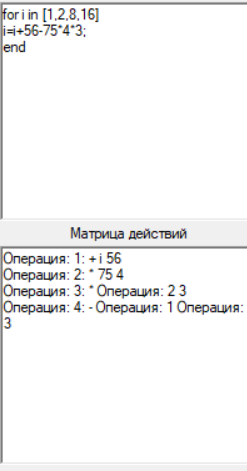


Рисунок 1 – работа программы

Вывод: в ходе лабораторной работы, были изучены методы трансляции арифметических и логических выражении.