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 №4

По Теория автоматов и формальных языков

Руководитель				
Кульков 2				
(фамилия, и	інициалы)			
(подпись)	(дата)			
Студент <u>ПИн - 121</u> (группа)				
Ермилов М.В.				
(фамилия, и	інициалы)			
(подпись)	(дата)			

Лабораторная работа №4

Тема: Реализация нисходящего распознавателя.

Цель работы: Получить навык в программировании синтаксических анализаторов.

Задание:

- 1. Реализовать синтаксический анализатор на основе LL(k)-грамматики.
- 2. В каждой из подпрограмм необходимо проверять текущую лексему в соответствии с решающей таблицей. Если не совпадает ни с одной из ожидаемых, то зафиксировать ошибку и завершить работу анализатора. При этом, будет известен текущий анализируемый (ошибочный) символ и набор лексем, ожидаемых в данном состоянии. Эту информацию нужно вывести пользователю. Например «Ожидался int или char, а встретился for» или «Ожидался идентификатор, а встретился =».
- 3. Привести результаты тестирования на 3х примерах с использованием различных вариантов расположения операторов языка (включая вложенные).

					МИ ВлГУ (9.03.0	4	
Изм.	Лист	№ докум.	Подпись	Дата			•	
Разр	аб.					Лит.	Лист	Листов
Проє	вер.						2	11
Реце	:нз.							
Н. Ко	онтр.						ПИн-1	21
Утв	∍рд.							

```
Листинг:
Form1.cs
using System. Windows. Forms;
namespace work
  public partial class Form1: Form
    public Form1()
       InitializeComponent();
    private void OpenFile(object sender, EventArgs e)
       OpenFileDialog fileDialog = new OpenFileDialog();
       fileDialog.Filter = "Файлы txt (*.txt)|*.txt";
       if (fileDialog.ShowDialog() == DialogResult.OK)
         StreamReader rdr = new StreamReader(fileDialog.FileName);
         string line = rdr.ReadToEnd();
         rdr.Close();
         richTextBox1.Text = line;
         Tokens(null, null);
       }
     }
    private void Tokens(object sender, EventArgs e)
       code(richTextBox1.Text);
    void code(string code)
```

Лист

```
listBox1.Items.Clear();
       Token[] tokens = generation. Tokens(generation. Split(code));
       foreach (Token token in tokens)
         listBox1.Items.Add(token);
       }
    private void button3_Click(object sender, EventArgs e)
       Token[] tokens = generation.Tokens(generation.Split(richTextBox1.Text));
       try
         LL a = new LL(tokens);
         listBox1.Items.Clear();
         a.Check();
         listBox1.Items.Add("Успешная проверка");
       catch(Exception E)
         listBox1.Items.Add(E.Message);
LL.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System. Threading. Tasks;
namespace work
```

Изм.	Лист	№ докум.	Подпись	Дата

```
public class LL
  List<Token> tokens;
  private List<Token> id;
  Token? Curent(int I) => I < tokens.Count ? tokens[I] : null;
  Token? Previe(int I) \Rightarrow I \Rightarrow 0 ? tokens[I - 1] : null;
  Token? Next(int I) \Rightarrow I < tokens.Count - 1? tokens[I + 1] : null;
  public LL(Token[] Tokens)
     tokens = new List<Token>();
    id = new List<Token>();
     foreach (Token token in Tokens)
       if (token.Type != TokenType.TOKEN ERROR)
          tokens.Add(token);
       }
       else
          throw new Exception(token.ToString());
  public bool Check()
     return true;
  public int действие(int I)
    Error1(I);
    int r = 0;
```

Лист

№ докум.

Подпись

```
if (операнд(I))
  if (isEndLine(I))
     return 0;
  I++;
  int sign = 3нак(I);
  if (sign \ge 0)//устраняет любые символы кроме * / - = + < > < >
     Error2(I, sign == 2);
  if (sign == -2)
     I++;
  if (sign == -1)
     r = 1;
  if (isEndLine(I))
     return r;
  return r + действие(I + 1);
}
else
  int sign = 3нак(I);
  if(sign == -1)
  {
     r = 1;
     if(isEndLine(I))
       return r;
  else if(isEndLine(I))
     Error2(I);
  else if(sign == 1)
     r += 1;
```

```
else if (Curent(I).Type != TokenType.MINUS && Curent(I).Type != TokenType.PLUS)
          TokenType previe = Previe(I).Type;
          TokenType next = Next(I).Type;
          if ((previe == TokenType.RPAR || операнд(I - 1)) && (next == TokenType.LPAR ||
операнд(I + 1) \parallel next == TokenType.PLUS \parallel next == TokenType.MINUS));
          else
            Error2(I);
        }
        I++;
        return r + действие(I);
    private bool isEndLine(int I)
      if (Next(I) == null || Next(I).Type == TokenType.ENDLINE)
        return true;
      return false;
    private int знак(int I)
      Error1(I);
      if (Curent(I).Type == TokenType.MULTIPLY ||
        Curent(I).Type == TokenType.DIVIDE |
        Curent(I).Type == TokenType.MINUS |
        Curent(I).Type == TokenType.PLUS |
        Curent(I).Type == TokenType.MORE \parallel
        Curent(I).Type == TokenType.LESS |
        Curent(I).Type == TokenType.EQUAL)
        if (Next(I) != null)
          if (Curent(I).Type == TokenType.LESS && Next(I).Type == TokenType.MORE) // <>
            return -2; // <>
          || // +=
```

```
|| // -=
             || // /=
             (Curent(I).Type == TokenType.MULTIPLY && Next(I).Type ==
TokenType.EQUAL) | // *=
             return 2; // += -= *= /=
        else
          return -3;
       else
        return -3; // + - */ <> =
     if (Curent(I).Type == TokenType.LPAR)
       return 1; // (
     if (Curent(I).Type == TokenType.RPAR)
       return -1; // )
     return 0;
   private bool операнд(int I)
     Error1(I);
     if (Curent(I).Type == TokenType.IDENTIFIER || Curent(I).Type == TokenType.LITERAL)
       return true;
     return false;
   private bool тип(int I)
     Error1(I);
     if(Curent(I) == null)
       return false;
     if (Curent(I).Type == TokenType.DOUBLE ||
       Curent(I).Type == TokenType.INTEGER \parallel
       Curent(I).Type == TokenType.STRING)
       return true;
     return false;
```

Изм.	Лист	№ докум.	Подпись	Дата

```
private void Error1(int I)
       if(Curent(I) == null)
          throw new Exception("Ожидалось наличие" +(1 + I) + "-ого элемента");
     }
     private void Error2(int I) => Error2(I, false, null);
     private void Error2(int I, bool next) => Error2(I, next, null);
     private void Error2(int I, TokenType[] TokenTypes) => Error2(I, false, TokenTypes);
     private void Error2(int I, bool next, TokenType[] TokenTypes)
       Error1(I);
       if(TokenTypes != null && TokenTypes.Length != 0)
          string err = "(";
          for (int i = 0; i < TokenTypes.Length; i++)
            err += TokenTypes[i].ToString();
            if (i + 1 < TokenTypes.Length)
               err += ", ";
          }
          err += ")";
          if (next)
            Error1(I + 1);
            throw new Exception((1 + I) + " \mu " + (2 + I) + " элементов (" + Curent(I) + " \mu " +
Curent(I + 1) + "), ожидалось " + err);
          throw new Exception("Вместо" + (1 + I) + "-ого элемента (" + Curent(I) + "), ожидалось "
+ err);
       if(next)
```

Изм.	Лист	№ докум.	Подпись	Дата

```
Error1(I+1);
           throw new Exception((1 + I) + " и " + (2 + I) + " элемент (" + Curent(I) + " и " + Curent(I+1)
+ "), не ожидался");
        throw new Exception((1 + I) + "-ый элемент (" + Curent(I) + "), не ожидался");
 ■ Form1
                                                                                                           DIM
IDENTIFIER - a
                         Dim a, b as integer
       Открыть файл
                          b = 12 * 32
                                                                         COMMA
    Провести операцию
                                                                         IDENTIFIER - b
                          do while a > 12
                                                                         AS
                                 dim c as double
                                                                         INTEGER
                                 c = a - b
                                                                         ENDLINE
                                 do while b > 12
                                                                         ENDLINE
                                        a += 12
                                                                         IDENTIFIER - b
                                        b -= 1
                                                                         EQUAL
                                loop
                                                                         LITERAL - 12
        button3
                                c += a
                                                                         MULTIPLY
                          loop
                                                                         LITERAL - 32
                                                                         ENDLINE
```

Рис.1 Пример работы программы

ENDLINE
DO
WHILE
IDENTIFIER - a
MORE
LITERAL - 12
ENDLINE
IDENTIFIER - dim
IDENTIFIER - c
AS
DOUBLE
ENDLINE
IDENTIFIER - c
EQUAL

Изм.	Лист	№ докум.	Подпись	Дата

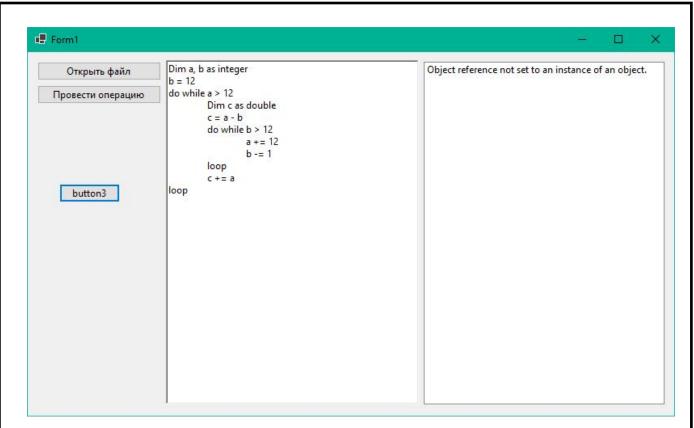


Рис.2 пример работы программы

Вывод: получены навыки в программировании синтаксических анализаторов.

Изм.	Лист	№ докум.	Подпись	Дата