

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 №6

По Теория автоматов и формальных языков

Руководитель

Кульков Я.Ю.

(фамилия, инициалы)

(подпись)

(дата)

Студент _____ ПИН - 121 _____

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись)

(дата)

Муром 2023

Лабораторная работа №6

Тема: Реализация восходящего распознавателя.

Цель работы: Получить навык в программировании синтаксических анализаторов.

Задание:

1. Реализовать синтаксический анализатор на основе LR(k)-грамматики.
2. В каждом из состояний 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 символ и набор лексем, ожидаемых в данном состоянии.

					МИ ВлГУ 09.03.04							
Изм.	Лист	№ докум.	Подпись	Дата				Лит.	Лист	Листов		
Разраб.									2	6		
Провер.												
Реценз.								ПИН-121				
Н. Контр.												
Утверд.												

Код по заданию

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace work
{
    public class LR
    {
        List<Token> tokens;

        Token _Curent()
        {
            Error1();
            return Curent();
        }

        Token _CurentUp()
        {
            i++;
            Error1();
            return Curent();
        }

        Token? Curent() => i < tokens.Count ? tokens[i] : null;
        Token? Previe() => i > 0 ? tokens[i - 1] : null;
        Token? Next() => i < tokens.Count - 1 ? tokens[i + 1] : null;

        public LR(Token[] Tokens)
        {
            tokens = new List<Token>();
            foreach (Token token in Tokens)
            {
                if (token.Type != TokenType.TOKEN_ERROR)
                {
                    tokens.Add(token);
                }
                else
                {
                    throw new Exception(token.ToString());
                }
            }
        }

        int i;
        public void Check()
        {
            i = 0;
            while (i != -1)
                _1();
        }

        void _1()
        {
            if(i != -1)
            {
                if(Curent() == null)
                {
                    i = -1;
                }
                else
                {
                    switch(Curent().Type)
                    {
                        case TokenType.DIM:
                            _16();
                    }
                }
            }
        }
    }
}
```

					МИ ВлГУ 09.03.04	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

```

        break;
    case TokenType.IDENTIFIER:
        _4();
        break;
    case TokenType.DO:
        _25();
        break;
    case TokenType.ENDLINE:
        i++;
        break;
    default:
        Error2();
        break;
    }
}

void _4()
{
    _CurentUp();
    if(Curent().Type == TokenType.EQUAL)
        _5();
    else if (Curent().Type == TokenType.PLUS ||
             Curent().Type == TokenType.MINUS ||
             Curent().Type == TokenType.MULTIPLY ||
             Curent().Type == TokenType.DIVIDE)
    {
        _CurentUp();
        if (Curent().Type == TokenType.EQUAL)
            _5();
        else Error2();
    } else Error2();
}

void _5()
{
    while(Curent() != null && Curent().Type != TokenType.ENDLINE)
        i++;
}

void _16()
{
    _CurentUp();
    if(Curent().Type == TokenType.IDENTIFIER)
    {
        _CurentUp();
        if (Curent().Type == TokenType.COMMA)
            _16();
        else if (Curent().Type == TokenType.AS)
            _18();
        else Error2();
    } else Error2();
}

void _18()
{
    _CurentUp();
    if (Curent().Type == TokenType.INTEGER ||
        Curent().Type == TokenType.DOUBLE ||
        Curent().Type == TokenType.STRING)
        i++;
    else Error2();
}

void _25()
{
    if(_CurentUp().Type == TokenType.WHILE)
    {
        _5();
    }
}

```

```

        _CurentUp();
        while (Curent() != null && Curent().Type != TokenType.LOOP)
            _1();
        if (Curent() == null)
            Error1();
        i++;
    } else Error2();
}

private void Error1()
{
    if (Curent() == null)
        throw new Exception("Ожидалось наличие " + (1 + i) + "-ого элемента");
}
private void Error2() => Error2(i);
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();
    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();
            throw new Exception((1 + I) + " и " + (2 + I) + " элементов (" +
Curent() + " и " + Curent() + "), ожидалось " + err);
        }
        throw new Exception("Вместо " + (1 + I) + "-ого элемента (" + Curent()
+ "), ожидалось " + err);
    }
    if(next)
    {
        Error1();
        throw new Exception((1 + I) + " и " + (2 + I) + " элемент (" + Curent()
+ " и " + Curent() + "), не ожидался");
    }
    throw new Exception((1 + I) + "-ый элемент (" + Curent() + "), не
ожидался");
}
}
}

```

Form1

Открыть файл

Провести операцию

Проверить

```

Dim a, b as integer
b = 12 * 32
do while a > 12
  Dim c as double
  c = a - b
  do while b > 12
    a += 12
    b -= 1
  loop
  c += a
loop

```

Проверено успешно

Рис 1 - пример работы программы