

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ имени Н.Э.БАУМАНА
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №11. Вариант №31.
Разработка парсеров на языке Java

Выполнил
студент группы ИУ9-21Б
Краев Степан

Москва, 2023

1 Условие

В ходе лабораторной работы нужно разработать программу, выполняющую синтаксический анализ текста по одной из LL(1)-грамматик, БНФ которых

приведены в таблицах 1–6. Текст может содержать символы перевода строки. В записи БНФ терминальные символы IDENT, NUMBER и STRING означают идентификаторы, числа и строки, соответственно. Идентификатор – это последовательность букв и цифр, начинающаяся с буквы. Число – это непустая последовательность десятичных цифр. Строка – это обрамлённая кавычками произвольная последовательность символов, не содержащая кавычек и символов перевода строки. Программа должна выводить в стандартный поток вывода последовательность правил грамматики, применение которых даёт левый вывод введённого из стандартного потока ввода текста. Если вывод не может быть построен, программа должна выводить сообщение «syntax error at (line, col)», где line и col – координаты ошибки в тексте. \LaTeX .

2 Код решения

Файл Main.java

```
public class Main {
    public static void main(String[] args) {
        String s;

        s="template < " +
            "template < typename T , int S > " +
            "class C , " +
            "typename T , int N " +
            ">";
        // System.out.println(s);
        System.out.println(new Parser(s).parse());

        System.out.println("ok");
        s="template < class C , typename T , int N >";
        //System.out.println(s);
        System.out.println(new Parser(s).parse());
    }
}
```

Файл Parser.java

```

public class Parser {
    private String[] arr;

    private String slovar("< , template int typename >");
    private int pos, len, in;
    private String[] res;
    public Parser(String s){
        s+=" ";
        System.out.println(s);
        this.arr=s.split(" ");
        this.pos=0;
        this.len=arr.length;
        this.res=new String[len];
        in=0;
    }
    public String parse(){
        //Tm();
        if (!Tm()){
            System.out.println("syntax error at (" +pos+" )");
        }
        String s="";
        for(int i=0;i<in;i++){
            s+=res[i]+" ";
        }
        return s;
    }
    private boolean Tm(){
        if (pos<len){
            res[in]="Tm";
            in++;
            if (arr[pos].equals("\n"))
                pos++;
            /*for (int el=0; el<arr.length; el++)
                System.out.println(arr[el]);*/
            //System.out.println(pos);
            if (arr[pos].equals("template")){
                pos++;
                //System.out.println(pos);
                if (arr[pos].equals("<")){
                    pos++;
                }
            }
        }
    }
}

```

```

while (pos < len) {
    if (arr[pos].equals("\n"))
        pos++;
    if (arr[pos].equals("template")) {
        pos++;
        if (!Tm())
            return false;
    }
    if (arr[pos].equals("typename")) {
        pos++;
        if (!Tpm())
            return false;
    }
    if (arr[pos].equals("class")) {
        pos++;
        if (!Class())
            return false;
    }
    if (arr[pos].equals("int")) {
        res[in] = "IDENT";
        in++;
        pos++;
        if (!Term())
            return false;
    }
    pos++;
}

}
if (arr[pos-1].equals(">")) {
    return true;
}
else
    return false;
}
}
return true;

```

```

}

private boolean Class(){
    //System.out.println(pos);
    res[in]="Class";
    in++;
    if (Term())
        return true;
    else
        return false;
}
private boolean Tpmn(){
    //System.out.println(pos);
    res[in]="Typename";
    in++;
    if (Term())
        return true;
    else
        return false;
}
private boolean Term(){
    //System.out.println(pos);
    if (slovar.indexOf(arr[pos])!=-1) {
        res[in]="E";
        in++;
        pos++;
        return true;
    }
    else
        return false;
}
}
}

```

3 Пример работы программы

```
C:\Users\Stepa\Downloads\jdk-19.0.2\bin\java.exe "-javaagent:C:\Users\Stepa\
template < template < typename T , int S > class C , typename T , int N >
Tm Tm Typename E IDENT E Class E Typename E IDENT E
ok
template < class C , typename T , int N >
Tm Class E Typename E IDENT E

Process finished with exit code 0
```

Рис. 1: Вывод программы