

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 №10
Реализация парсеров на Java
Вариант №15

Выполнил
студент группы ИУ9-21Б
Лисов Алексей

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. Условие задачи, исходный код и пример работы программы необходимо прислать в формате \LaTeX .

2 Код решения

Файл LambdaParser.java

```
import java.util.*;

public class LambdaParser {
    private String input;
    private int index;

    public LambdaParser(String input) {
        this.input = input;
        this.index = 0;
    }

    public void parse() {
        clean();

        ArrayList<Node> res = new ArrayList<>();

        while (index < input.length()) {
            if (input.charAt(index) == ':') {
                res.add(parseApply());
            }
            else {
                res.add(parseLambda());
            }
        }

        if (index != input.length()) {
            throw new RuntimeException("Unexpected input at position " + index);
        }
    }
}
```

```

        for (Node i : res) {
            System.out.println(i.toString());
        }
    }

    private void clean() {
        while (index < input.length() && input.charAt(index) == ' ') {
            ++index;
        }
    }

    private Node parseLambda() {
        clean();

        Node result;
        if (Character.isDigit(input.charAt(index))) {
            int start = index;
            while (index < input.length() && Character.isDigit(input.charAt(index))) {
                index++;
            }
            int num = Integer.parseInt(input.substring(start, index));
            result = new Node("NUMBER", Integer.toString(num));
            result.addChild(parseApply());
        } else if (input.charAt(index) == '-') {
            index++;
            if (input.charAt(index) != '>') {
                throw new RuntimeException("Unexpected input at position " + index);
            }
            index++;
            result = new Node("->");
            Node some = parseLambda();
            result.addChild(some);
        } else if (input.charAt(index) == '(') {
            index++;
            result = parseLambda();
            if (input.charAt(index) != ')') {
                throw new RuntimeException("Unexpected input at position " + index);
            }
            index++;
        }
    }

```

```

    } else if (input.charAt(index) == ':') {
        result = parseApply();
    } else {
        throw new RuntimeException("Unexpected input at position " + index);
    }
    return result;
}

private Node parseApply() {
    clean();

    if (index < input.length() && input.charAt(index) == ':') {
        index++;
        Node result = new Node(":");
        result.addChild(parseLambda());
        result.addChild(parseApply());
        return result;
    } else {
        return new Node("");
    }
}

private static class Node {
    private String label;
    private String value;
    private List<Node> children;

    public Node(String label) {
        this.label = label;
        this.value = null;
        this.children = new ArrayList<>();
    }

    public Node(String label, String value) {
        this.label = label;
        this.value = value;
        this.children = new ArrayList<>();
    }

    public void addChild(Node child) {

```

```

        this.children.add(child);
    }

    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append(label);
        if (value != null) {
            sb.append("(").append(value).append(")");
        }
        for (Node child : children) {
            sb.append(" ").append(child.toString());
        }
        return sb.toString();
    }
}

```

Файл Main.java

```

public class Main {
    public static void main(String[] args) {
        // String input = "(-> -> 1): (-> 0: 0)";
        String input = "(-> -> 1)";
        LambdaParser parser = new LambdaParser(input);
        parser.parse();
    }
}

```

3 Скриншоты

```
        this.value = null;
        this.children = new ArrayList<>();
    }

    1 usage
    public Node(String label, String value) {
        this.label = label;
        this.value = value;
        this.children = new ArrayList<>();
    }

    4 usages
    public void addChild(Node child) {
        this.children.add(child);
    }

    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append(label);
        if (value != null) {
            sb.append("(").append(value).append(")");
        }
        for (Node child : children) {
            sb.append(" ").append(child.toString());
        }
        return sb.toString();
    }
}
```

Рис. 1: Вывод программы

```
no usages
1 ▶ public class Main {
    no usages
2 ▶   public static void main(String[] args) {
3       // String input = "(-> -> 1): (-> 0: 0)";
4       String input = "(-> -> 1)";
5       LambdaParser parser = new LambdaParser(input);
6       parser.parse();
7
8   }
9 }
```

Рис. 2: Вывод программы

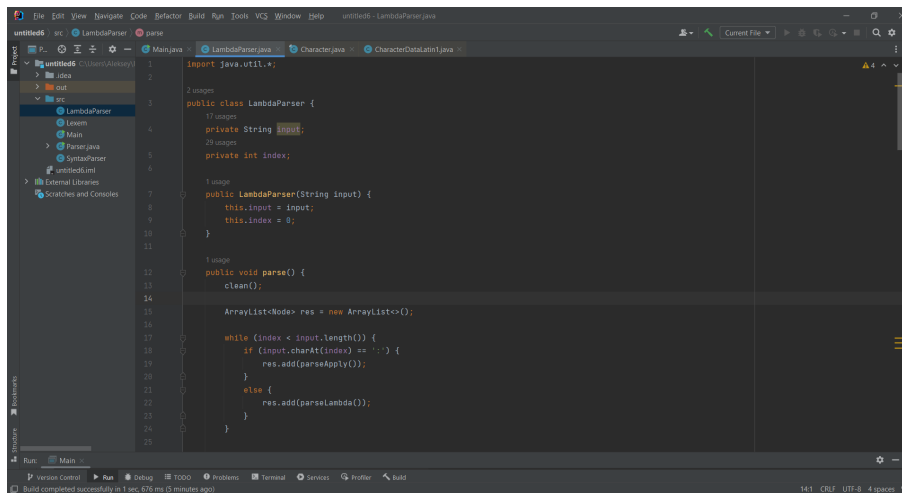


Рис. 3: Вывод программы