



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
по курсу «Языки и методы программирования»
«Разработка парсеров на языке Java»

Студент группы ИУ9-22Б Жук А. Д.

Преподаватель Посевин Д. П.

Москва 2023

1 Цель работы

Овладение методом рекурсивного спуска для разработки парсеров по грамматике некоторого формального языка.

2 Задание

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.

```
33 <Expr> ::= <Term> <Add>      a + 10 - (x - 1)
    <Add> ::= + <Expr> | - <Expr> | ε
    <Term> ::= IDENT | NUMBER
    | ( <Expr> )
```

Рис. 1 — Вариант 33

3 Результаты

Для проверки распарсим строку из примера (a + 10 - (x - 1)) Получим:
IDENT

+ <Expr>
 NUMBER
 - <Expr>
 (<Expr>)
 IDENT
 - <Expr>
 NUMBER
 epsilon
 epsilon

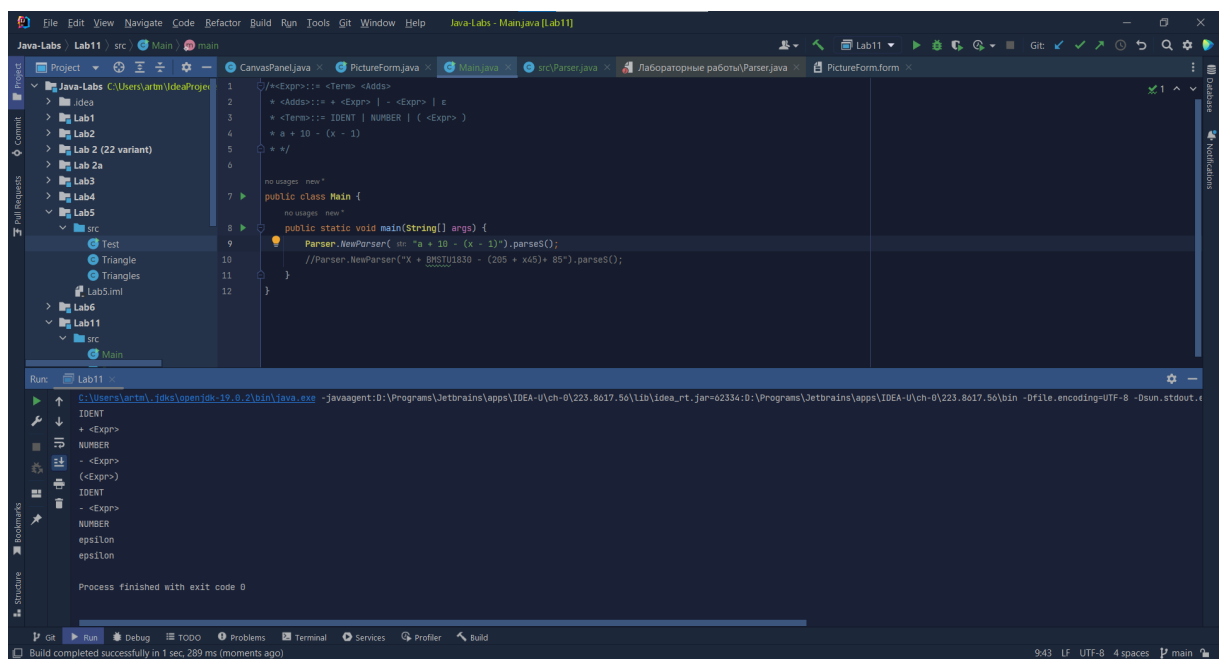


Рис. 2 — Строка из примера

Далее попробуем распарсить заведомо неправильную строку (X + BMSTU1830! - (205 + x45)+ 85)

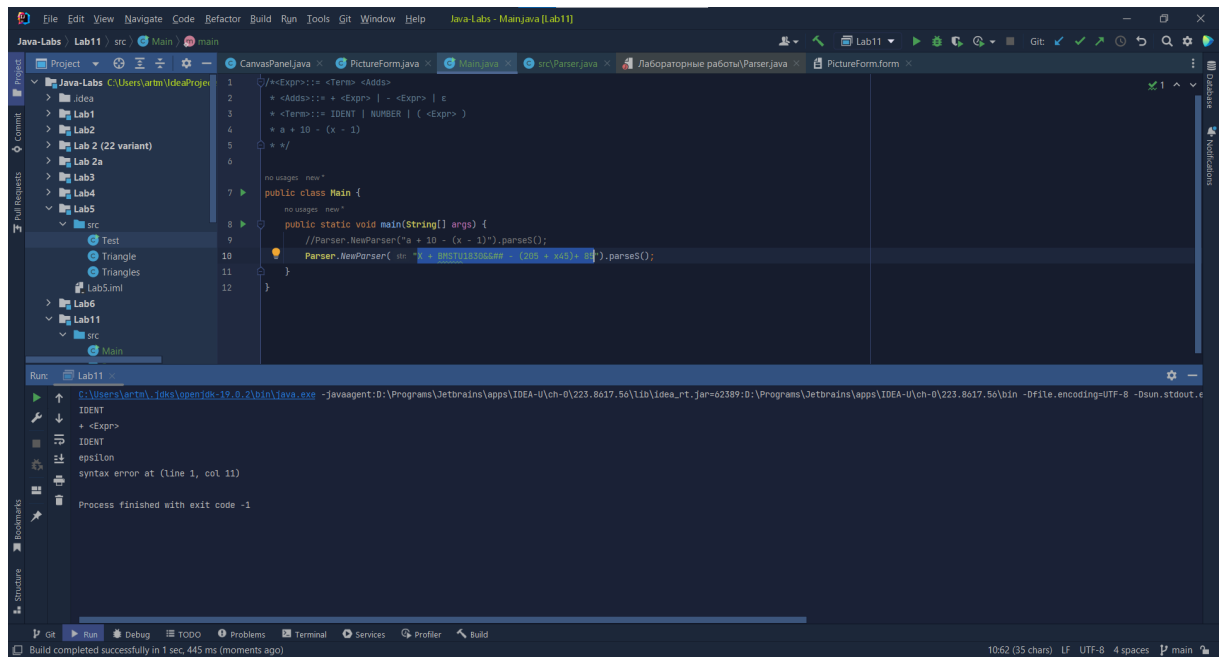


Рис. 3 — Скриншот ошибки

Далее попробуем распарсить строку из 35го варианта (которая заведомо не подходит под грамматику моего варианта)

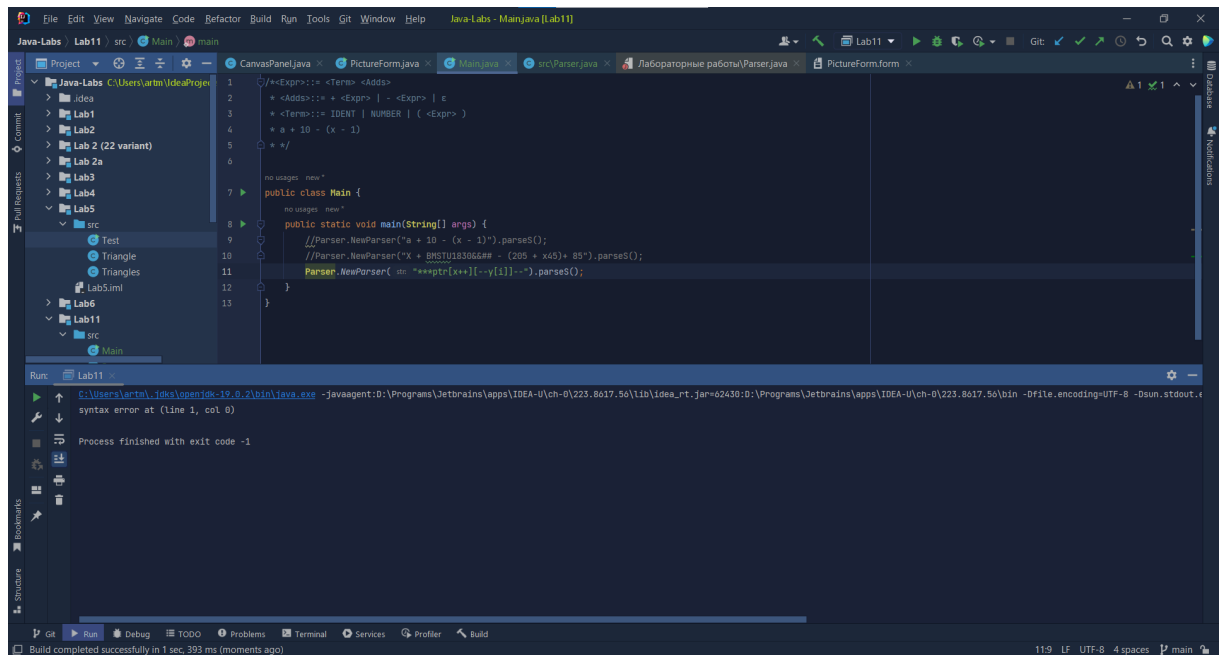


Рис. 4 — Скриншот ошибки

Листинг 1 — Код Main.java

```
1 public class Main {  
2     public static void main(String[] args) {  
3         //Parser.NewParser("a + 10 - (x - 1)").parseS();  
4         //Parser.NewParser("X + BMSTU1830&&## - (205 + x45)+ 85").parseS  
        ();  
5         Parser.NewParser("***ptr[x++][--y[i]]--").parseS();  
6     }  
7 }
```

Листинг 2 — Код Parser.java

```

1 public class Parser {
2     private int currentIndex;
3     private String StringToParse;
4     Parser(String str) {
5         this.StringToParse = str.replaceAll("\\s+", "");
6         this.currentIndex = 0;}
7     public static Parser NewParser(String str) {
8         return new Parser(str);}
9     void parseS() {
10        parseExpr();
11        if (currentIndex != StringToParse.length()) {
12            System.out.println("syntax error at (line 1, col " +
currentCharIndex + ")");
13            System.exit(-1);}
14        private void parseExpr() {
15            parseTerm();
16            parseAdds();}
17        private void parseAdds() {
18            if (currentIndex == StringToParse.length()) {
19                System.out.println("epsilon");
20                return;}
21            if (StringToParse.charAt(currentIndex) == '+') {
22                System.out.println("+ <Expr>");
23                currentIndex++;
24                parseExpr();
25            } else if (StringToParse.charAt(currentIndex) == '-') {
26                System.out.println("- <Expr>");
27                currentIndex++;
28                parseExpr();
29            } else {
30                System.out.println("epsilon");
31                return;}
32        private void parseTerm() {
33            if (Character.isLetter(StringToParse.charAt(currentIndex)))
34            {
35                System.out.println("IDENT");
36                while (currentIndex < StringToParse.length() && (
Character.isLetter(StringToParse.charAt(currentIndex)) ||
37                    Character.isDigit(StringToParse.charAt(
currentCharIndex)))) {
38                    currentIndex++;}
39                } else if (Character.isDigit(StringToParse.charAt(
currentCharIndex))) {
40                    System.out.println("NUMBER");
41                    while (currentIndex < StringToParse.length() &&
Character.isDigit(StringToParse.charAt(currentCharIndex))) {
42                        currentIndex++;}
43                } else if (StringToParse.charAt(currentCharIndex) == '(') {
44                    System.out.println("<Expr>");
45                    currentIndex++;
46                    parseExpr();
47                    if (StringToParse.charAt(currentCharIndex) == ')') {
48                        currentIndex++;
49                    } else {
50                        System.out.println("error");
51                        System.exit(-1);}
52                } else {
53                    System.out.println("syntax error at (line 1, col " +
currentCharIndex + ")");
54                    System.exit(-1);}}}

```