

Университет ИТМО

Кафедра ИПМ

Проектирование языков программирования и языков представления данных

Лабораторная работа №1

«Формальные грамматики и языки»

Выполнили студенты группы Р4117:

Пьянков Е.В., Данилов М.В.

Преподаватель: Лаздин А.В.

Санкт-Петербург

2018

Задание

Разработать лексический анализатор согласно варианту.

БНФ

```
<Программа> ::= <Объявление переменных> <Описание вычислений>.  
<Описание вычислений> ::= Begin <Список присваиваний> End  
<Объявление переменных> ::= Var <Список переменных> :Boolean;  
Var <Список переменных> :Boolean;| <Объявление переменных>  
<Объявление переменных> ::= Var <Список переменных> :Decimal;|  
Var <Список переменных> :Decimal; <Объявление переменных>  
<Список операторов> ::= <Оператор> | <Оператор> <Список операторов>  
<Оператор> ::= <Присваивание> | <Сложный оператор>  
<Присваивание> ::= <Идент> := <Выражение>  
<Выражение> ::= <Ун.оп.> <Подвыражение> | <Подвыражение>  
<Подвыражение> ::= ( <Выражение> ) | <Операнд> |  
<Подвыражение> > <Бин.оп.> <Подвыражение>  
<Ун.оп.> ::= "!"  
<Бин.оп.> ::= "&" | "|" | "^" | "-" | "+" | "*" | "/" | ">" | "<" | "=="  
<Операнд> ::= <Идент> | <Const>  
<Сложный оператор> ::= IF "(" <Выражение> ")" Оператор|  
IF "(" <Выражение> ")" <Оператор> ELSE <Оператор>  
<Операнд> ::= <Идент> | <Const>  
<Идент> ::= <Буква> <Идент> | <Буква>  
<Const> ::= <LConst> | <DConst>  
<DConst> > <Цифра> | <Цифра> <DConst>  
<LConst> ::= 0|1
```

Классы лексем:

ID - идентификатор

CONST - константа

KEY_WORD - ключевое слово

BINARY_OPERATOR - бинарный оператор

UNARY_OPERATOR - унарный оператор

SEPARATOR - разделитель

END_TOKEN - конечная лексема

UNDEFINED

Исходный код

Исходный код представлен здесь:

<https://github.com/EvgenyPyankov/itmo-projects/tree/master/src/main/java/lexing>

Примеры работы:

Пример 1:

Var

```
ab, c: Boolean;  
m, d, k: Decimal;
```

Begin

```
  If (k == d)  
  |  
  |   Begin  
  |   |   m := d + m  
  |   |   End  
  |  
  | Else  
  |   c := 0  
  |  
  | ab := 1  
  |  
End.
```

```
"C:\Program Files\Java\jdk1.8.0_151\bin\java" ...
```

Tokens.

<type, value> (line, row)

```
-----  
<KEY_WORD, Var> (0, 0)  
<ID, 0> (2, 0)  
<SEPARATOR, ,> (2, 2)  
<ID, 1> (2, 4)  
<SEPARATOR, :=> (2, 5)  
<KEY_WORD, Boolean> (2, 7)  
<SEPARATOR, ;> (2, 14)  
<ID, 2> (3, 0)  
<SEPARATOR, ,> (3, 1)  
<ID, 3> (3, 3)  
<SEPARATOR, ,> (3, 4)  
<ID, 4> (3, 6)  
<SEPARATOR, :=> (3, 7)  
<KEY_WORD, Decimal> (3, 9)  
<SEPARATOR, ;> (3, 16)  
<KEY_WORD, Begin> (5, 0)  
<KEY_WORD, If> (6, 1)  
<SEPARATOR, (> (6, 4)  
<ID, 4> (6, 5)  
<BINARY_OPERATOR, ==> (6, 7)  
<ID, 3> (6, 10)  
<SEPARATOR, )> (6, 11)  
<KEY_WORD, Begin> (7, 5)  
<ID, 2> (8, 5)  
<BINARY_OPERATOR, :=> (8, 7)  
<ID, 3> (8, 10)  
<BINARY_OPERATOR, +> (8, 12)  
<ID, 2> (8, 14)  
<KEY_WORD, End> (9, 5)  
<KEY_WORD, Else> (10, 1)  
<ID, 1> (11, 2)  
<BINARY_OPERATOR, :=> (11, 4)  
<CONST, 0> (11, 7)  
<ID, 0> (12, 1)  
<BINARY_OPERATOR, :=> (12, 4)  
<CONST, 1> (12, 7)  
<KEY_WORD, End> (13, 0)  
<END_TOKEN, .> (13, 3)
```

IDs table.

```
-----  
0 ab  
1 c  
2 m  
3 d  
4 k
```

Пример 2:

```
Var  
ab, c: Integer;  
Begin  
  a  
End.
```

```
"C:\Program Files\Java\jdk1.8.0_151\bin\java" ...  
Tokens.  
<type, value> (line, row)  
-----  
<KEY_WORD, Var> (0, 0)  
<ID, 0> (1, 0)  
<SEPARATOR, ,> (1, 2)  
<ID, 1> (1, 4)  
<SEPARATOR, :=> (1, 5)  
<ID, 2> (1, 7)  
<SEPARATOR, ;> (1, 14)  
<KEY_WORD, Begin> (2, 0)  
<ID, 3> (3, 4)  
<KEY_WORD, End> (4, 0)  
<END_TOKEN, .> (4, 3)  
  
IDs table.  
-----  
0 ab  
1 c  
2 Integer  
3 a  
  
Process finished with exit code 0
```

Вывод

В ходе выполнения лабораторной работы был разработан лексический анализатор.