

1. ГРАММАТИКА ЯЗЫКА L

Здесь описана грамматика языка L. Степень формальности и строгости немного опущена для простоты изложения. Определения идентификаторов, литералов, бинарных операторов и т.п. взяты из домашнего задания про лексер.

Основными понятиями в грамматике языка L являются объявление функции (function definition), вызов функции (function call), инструкция (statement) и выражение (expression).

Программа на языке L представляет собой набор из нескольких объявлений функций (возможно, ни одного) и инструкции.

1.1. Объявления функций. Объявление функции выглядит так:

```
<function definition> : <name> '(' (<arg> (',' <arg>)*)? ')' '{' <statement>? '}'
```

1.2. Вызовы функций. Вызов функции выглядит так:

```
<function call> : <name> '(' (<arg expr> (',' <arg expr>)*)? ')'
```

1.3. Инструкции. Инструкция может быть одной из следующих:

- Присваивание
- Вызов функции
- Несколько инструкций, разделенных точкой с запятой
- Вызов функции
- Вывод в выходной поток
- Чтение из входного потока
- Цикл "пока"
- Условное ветвление

```
<statement> : <variable name> ':=' <expression>
              | <function call>
              | <statement> ';' <statement>+
              | 'write' <expression>
              | 'read' <variable name>
              | 'while' <expression> '{' <statement>? '}'
              | 'if' <expression> '{' <statement>? '}' ('else' '{' <statement>? '}')?
```

1.4. Выражения. Выражение может быть одним из следующих:

- Выражение, взятое в скобки
- Вызов функции
- Взятие значения переменной
- Целое число
- Число с плавающей точкой
- Применение бинарной операции к двум выражениям (правила приведены в порядке уменьшения приоритета операций)

```
<expression> : '(' <expression> ')'
              | <function call>
              | <variable name>
              | <decimal integer literal>
              | <decimal floating point literal>
              | <expression> ('*' | '/' | '%') <expression>
              | <expression> ('+' | '-') <expression>
              | <expression> ('<' | '>' | '<=' | '>=') <expression>
              | <expression> ('==' | '!=') <expression>
              | <expression> '&&' <expression>
              | <expression> '||' <expression>
```