

Пример синтаксиса для синтаксического анализатора

Будем определять составляющие синтаксиса двигаясь от абстракций верхнего уровня к абстракциям нижнего уровня:

Программа -> функции

Функции -> идентификатор, аргументы, инструкции

Инструкции -> присвоение, возвращение значения из функции, условие, цикл

Аргументы -> список идентификаторов

Идентификатор -> список символов

Выражения -> действие операторов по приоритету над числами, идентификаторами, вызовами функций

Числа -> список цифр

Формальный синтаксис:

```
<Program> ::= <Function> {<Function>}
<Function> ::= <Identifier> '(' {<Identifier> | ',' <Identifier>} ')' '{'
{<Instruction> <Separator>} '}'
<Instruction> ::= <Assignment> | <Return> | <if> | <while>
<Assignment> ::= <Identifier> '=' <Expression>
<Return> ::= 'return' <Expression>
<if> ::= 'if' <Expression> 'then' '{' {<Instruction> <Separator>} '}' 'else'
'{' {<Instruction> <Separator>} '}'
<while> ::= 'while' <Expression> '{' {<Instruction> <Separator>} '}'
<Identifier> ::= <Letter1> {<Letter1> | <Letter2> | <Simbol> | <Digit>}
<Letter1> ::= 'a' | ... | 'z'
<Letter2> ::= 'A' | ... | 'Z'
<Simbol> ::= '_'
<Separator> ::= ';'
<Expression> ::= <Term1> {<Operation1> <Term1>}
<Term1> ::= <Term2> {<Operatio2> <Term2>}
<Term2> ::= <Operatio3> <Term3> | <Term3>
<Term3> ::= <Term4> {<Operatio4> <Term4>}
<Term4> ::= <Term5> {<Operatio5> <Term5>}
<Term5> ::= <Term6> {<Operatio6> <Term6>}
<Term6> ::= <Term7> <Operation7> <Term6> | <Term7>
<Term7> ::= <Operatio8> <Term8> | <Term8>
<Term8> ::= <Expression> | <Number> | <Identifier> | <Identifier> '('
{<Identifier> | ',' <Identifier>} ')'
<Number> ::= '0' | <digit> {<digit> | '0'}
<digit> ::= '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
<Operation1> ::= '||'
<Operation2> ::= '&&'
<Operation3> ::= '--'
<Operation4> ::= '==' | '/=' | '<' | '<=' | '>' | '>='
<Operation5> ::= '+' | '-'
<Operation6> ::= '*' | '/'
<Operation7> ::= '**'
<Operation8> ::= '-'
```