

Название языка

Do

Грамматика языка

```
<program> ::= <main function> <function declaration>*  
<main function> ::= "[func] main {" <statement>* "}"
```

```
<function declaration> ::= "[func" <argument type>* ("=>" <return type>){0,1} "]" <identifier> "{" <function arguments>{0,1}  
<statement>* "}"
```

```
<argument type> ::= <type> | "array"<type>
```

```
<return type> ::= <type> | "array"<type>
```

```
<function arguments> ::= "[args]" <identifier> ("," <identifier>)*
```

```
<statement> ::= <variable declaration> |  
               <constant declaration> |  
               <array declaration> |  
               "{" <statement>* "}" |  
               <lvalue> "=" <expression> |  
               "if (" <condition> ")" <statement> ("else" <statement>){0,1} |  
               "while (" <condition> ")" <statement> |  
               "for (" <counter declaration>{0,1} "," <condition> "," <expression> ")" <statement>*  
               "print (" <expression> ")" |  
               ">" <expression> |  
               <expression>
```

```
<expression> ::= <binary operation> | <atom> | "!" <expression>
```

```
<binary operation> ::= <logical or>
```

```
<logical or> ::= <logical and> ("||" <logical and>)*
```

```
<logical and> ::= <equality> ("&&" <equality>)*
```

```
<equality> ::= <comparison> ("=="|"!=" <comparison>)*
```

```
<comparison> ::= <add> ("<"|">"|"<="|">=" <add>)*
```

```
<add> ::= <mult> ("+"|"-" <mult>)*
```

```
<mult> ::= <atom> ("*"|"/"|"%" <atom>){1,}
```

```
<atom> ::= <number> |
```

```
           <identifier> |
```

```
           <array identifier> "." <number> |
```

```
<array identifier> ".length" |  
<array identifier> ".push(" <expression> ")" |  
<array identifier> ".pop()" |  
<function invocation> |  
"(" <expression> ")"
```

```
<lvalue> ::= <identifier> |  
    <array identifier> "." <number> ("+"|"-"<expression>)*  
<function invocation> ::= <identifier> "(" <identifier>* ")"
```

```
<variable declaration> ::= "[var" <type> "]" <identifier> (<expression>){0,1}  
<constant declaration> ::= "[const" <type> "]" <identifier> <expression>  
<array declaration> ::= "[array" <type> "]" <array identifier> (<expression> ("," <expression>)*){0,1}  
<counter declaration> ::= <identifier> | <identifier> "=" <expression>  
<condition> ::= <expression>  
<array identifier> ::= <identifier>
```

```
<type> ::= "<int>" | "<float>" | "<string>" | "<boolean>"
```

Особенности

Глобальные переменные запрещены.

Ключевые слова

```
print
```

Примеры кода

Функция main

```
[func] main {  
  
}
```

Функция Foo с двумя параметрами типа string, float и Возвращаемым значением типа int

```
[func <string> <float> => <int>] Foo {  
    [args] arg1, arg2 // Получение аргументов, переданных в функцию  
  
}
```

Вызов функции Foo

```
Foo("string" 1.25)
```

Объявление массива элементов типа float

```
[array <float>] someArray
```

Объявление массива типа int с инициализацией

```
[array <int>] intArray 1, 2, 3, 4, 5
```

Объявление переменной типа int

```
[var <int>] boo
```

Объявление переменной типа float с инициализацией

```
[var <float>] foo 1.0
```

Объявление константы типа string

```
[const <string>] someString "Hello, World!"
```

Длина массива

```
someArray.length
```

Доступ к 1-му элементу массива

```
someArray.0
```

Доступ к 3-му элементу массива

```
someArray.2
```

Примеры программ

Hello, world!

```
[func] main {  
    [const <string>] helloWorld "Hello, World!"  
    print(helloWorld)  
}
```

Bubble sort with print of result

```
[func] main {  
    [array <int>] array 5, 1, 4, 2, 8  
    array = BubbleSort(array)  
    print(array)  
}
```

```
[func array<int> => array<int>] BubbleSort {
```

[args] array

```
for (i = 0, i < array.length, i = i + 1) {  
    for (j = 0, j < array.length - 1, j = j + 1) {  
        if (array.j > array.j + 1) {  
            [var <int>] temp = array.j  
            array.j = array.j + 1  
            array.j + 1 = temp  
        }  
    }  
}
```

=> array

}