# Спецификация языка SmallVL

Этот документ описывает синтаксис и базовую семантику языка SmallVL.

## Линейная структура

SmallVL программа делиться на несколько логических строк.

## Логические линии

Конец логической строки обозначается токеном NEWLINE. Выражения не могут занимать несколько логических строк, если это не разрешено (далее) синтаксисом.

## Физические строки

Физическая строка – последовательность символов, которая заканчивается EOL последовательностью символов (\n); In source files, any of the standard platform line termination sequences can be used - the Unix form using ASCII LF (linefeed), the Windows form using the ASCII sequence CR LF (return followed by linefeed), or the old Macintosh form using the ASCII CR (return) character. All of these forms can be used equally, regardless of platform.

## Инструкции

Каждая инструкция должна начинаться с новой строки и оканчиваться символом ‘;’.

## Коментарии

Коментарии начинаются с символа // и заканчиваются в конце физической строки. Коментарии не являются частью программы и игнорируются лексическим анализатором.

Также поддерживаются многострочные комментарии. Весь текст между символами ‘/\*’ и ‘\*/’ будет воспринят как комментарий и не будет анализироваться.

### Типы

Все переменные представляют собой строки, но, в зависимости от контекста, могу использоваться использоваться как целочисленные, логические переменные. Также, как числа, с плавающей запятой.

## Переменные

Имена переменных состоят из разрешенных символов и должны начинаться с символа ‘$’. Так как SmalVL безтиповый язык, нет необходимости указывать тип при инициализации.

## Функции

Имя функции состоит из разрешенных символов, после которых следуют скобки. Внутри скобок через запятую задан список аргументов функции. Далее открывающая и закрывающая скобки ограничивают тело функции. При помощи ключевого слова ‘return’ функция может передать управление вызывающему коду вместе с любым возвращаемым значением.

## Операторы

Каждый из операторов для каждого типа операндов работает по-разному. При невозможности операции будет возвращено значение ‘undefined’

Оператор ‘+’ выполняет функцию суммирования 2х чисел либо конкатенацию строк. В иных случаях будет возвращено ‘undefined’

Оператор ‘-’ выполняет вычитание 2х чисел либо удаляет все вхождения 2й строки из 1й

Оператор ‘==’ возвращает ‘true’, если правое и левое значения одинаковы, иначе ‘false’

Оператор ‘>’ возвращает ‘true’, если левое значение больше правого, иначе ‘false’. Работает только для чисел. Иначе ‘undefined’

Оператор ‘<’ возвращает ‘true’, если левое значение меньше правого, иначе ‘false’. Работает только для чисел. Иначе ‘undefined’

Оператор ‘=’ выполняет операцию присваивания значения правой переменной левой.

## Массивы

Все массивы в SmallVL имеют динамический размер. Они объявляются как обычные переменные, но требуют особой инициализации.

Пример:

$size = 1024;

//dynamic resized array

$someArray = array[$size];

//or without size

$someArray = array[];

## Циклы

SmallVL поддерфивает 2 типа циклов ‘for’ и ‘while’

Пример синтаксиса объявления:

for($i = 0; $i < 0; $i++)

{

read($srcFile, $buffer);

write($dstFile, $buffer);

}

//Init boolean var

$conditionVal = true;

while($conditionValue)

{

if( rand() % 10 > 5 )

$conditionValue = false;

print(".");

}

## Ключевые слова

Ключевое слово ‘import’ выполняет подключение внешнего файла исходного кода и все объявленные там функции и переменные становятся видимыми в данном файле.