ScribbleVM Specification

Registers

ScribbleVM is a register based virtual machine. Registers are fixed size (64 bit) areas in memory used to store values. In the case of ScribbleVM a large number of registers (Up to 255) are used. Most of these registers are used to store the values of variables however there are also several registers reserved for storing the results of operations.

Primitives

The primitives in ScribbleVM are byte, short, int, long which are 1, 2, 4 and 8 bytes wide accordingly. When in registers they are all seen as 8 bytes wide (So the opcodes do not have to differentiate between primitive type before executing) and these sizes only apply to values within heap memory or in the constant region.

Instructions

Instructions within ScribbleVM will all be fixed size (8 bytes). The first byte of the instruction will used to identify what operation the instruction is to perform and the remaining bytes will be used to carry the data necessary to perform that operation.

Instruction Set

The instruction set in ScribbleVM is split up into two regions, the instructions list (an array of fixed size instructions which ScribbleVM can execute) and the constant region a variable sized region that contains all of the constants used in the functions.

Namespaces, Functions & Types

Functions and types are registered to namespaces within the VM before it is executed. This registration allows them to be identified by the VM at runtime.

Namespaces can also have namespaces as children, so the root namespace may contain the namespaces 'sys' and 'main'.

Calls to entries in different namespaces are resolved at runtime using '.' as a delimiter (So a call to hello.World would look for the entry World in the hello namespace).

Heap

The VM heap is garbage collected and very heavily policed. Each entry inside it will have a type from a set of types that have been registered with the virtual machine and these types will contain information about what is contained within each element on the heap.