Комманды, поддерживаемые компилятором и виртуальной машиной.			
Байткод	Стек	Мнемоника	Описание
0x00	eval_stack	nop	Просто занимает место в памяти. Ничего не делает. Мусор.
0x01	eval_stack	push	Кладет в стека константу. Единственная команда которая принимает операнд от программиста. Синтаксис: push константа Синтаксис констант: 'char' – кладет в стек код символа (ASCII). (64 разрядное) 0xN – кладет в стек 64 разрадное знаковое целое число [0-9]+ - кладет в стек 64 разрядное знаковое целое число [0-9]+.[0-9]+ - кладет в стек 64 разрядное число с плавающей точкой label – кладет в стек 64 разрядный адрес метки.
0x02 - 0x09	eval_stack	iadd dadd isub dsub idiv ddiv imul dmul	Арифметические команды. i – принимают операнды как знаковое целое d – принимают операнды как числа с плавающей запятой Стек до использования: b - scndOperand a - firstOperand ... Стек после: res – result …
0x0A	eval_stack	mod	Возвращвет остаток от деления. Принимает операнды как знаковое целое. Стек до использования: b - divider a – dividend … Стек после использования: rem – remainder ...
0x0B	eval_stack	branch	Безусловный переход по адресу. Переход осуществляется внутри функции. Стек до использования: addr

			… Стек после: …
0x0C	eval_stack	branchif	Если за вершиной стека лежит ненулевое значение – переход по адресу на вершине стека. Стек до: a – addr b – compare result … Стек после: ...
0x0D	eval_stack ctx_stack	invoke	Вызывает функцию по id на вершине стека. Стек до: b – function_id … Стек после: ... Если функция принимает N аргументов, то с eval_stack снимается N + 1 значений (N аргументов и 1 function_id). При этом function_id должен находиться на вершине.
0x0E	ctx_stack	retvoid	Возврат из функции без возвращаемого значения.
0x0F	eval_stack ctx_stack	ret	Возврат из функции с возвращаемым значением. Кладет на вершину eval_stack вызвавшей функции значение с вершины eval_stack вызываемой функции.
0x10	eval_stack	dup	Дублирует значение в стеке
0x11	eval_stack	swap	Меняет местами два последних значения в стеке.
0x12	eval-stack	iprint	Выводит значение с вершины стека как знаковое целое. Не меняет стек.
0x13	eval-stack	dprint	Выводит значение с вершины стека как число с плавающей запятой. Не меняет стек.
0x14	eval_stack	sprint	Принимает значение с вершины стека как индекс в константном поле строк.
0x15	-	halt	Останов.
0x16 - 0x21	eval_stack	[id]cmpeq [id]cmpne	Сравнивают два числа и кладут на стек результат. (1 – если истина. 0 – если ложь)

		[id]cmpg [id]cmpge [id]cmpl [id]cmple	i – сравнивать как знаковое целое d – сравнивать как числа с плавющей точкой
0x22	eval_stack	neg	Принимает с верштны стека число как беззнаковое целое. Если оно равно 0, то кладет на стек 1. Иначе 0.
0x23	eval_stack	i2d	Принимает с вершины стека число как знаковое целое. Переводит его в формат чисел с плавающей запятой. Кладет на вершину стека.
0x24	eval_stack	d2i	Принимает с вершины стека число как число с плавающей запятой. Переводит его в знаковое целое. Кладет на вершину стека.
0x25	data_stack eval_stack	load	Выгружает из data_stack 8 байт в eval_stack.
0x26	data_stack eval_stack	store	Выгружает из eval_stack 8 байт в data_stack.
0x27-0x2A	eval_stack	[id]inc [id]dec	Инкремент и декремент числа на вершине стека. I – принимает число как знаковое целое d – принимает число как число с плавающей запятой
0x2B	data_stack	clrloc	Очищает data_stack.
0x2C	eval_stack	clreval	Очищает eval_stack.
0x3D	eval_stack	pop	Очищает вершину стека.
0x3E	data_stack eval_stack	loadcp	Копирует значение вершины data_stack и кладет его на вершину eval_stack.
0x3F	data_stack	storecp	Копирует значение вершины eval_stack и кладет его на вершину data_stack.
0x40	data_stack	cprint	Выводит значение с вершины стека как символ. Не меняет стек.