Variables

`iconst_x`	load int const x on stack
`iload_2`	load int variable-slot 2
`istore_2`	store variable 2
`iinc 1 -1`	decrement Int variable 1
`iinc 1 1`	increment Int variable 1

Advanced Variables

`bipush 10`	push byte 10 on stack
`aconst _null`	load `null` constant
`newarray int`	Create new array
`astore_1`	store array in var 1
`iastore` Stack: `[x,y,....]`	save int x in array at position y

Arithmetic

`iadd`	add top 2 stack elements
`imul`	multiply top 2 stack elements
`idiv`	divide top 2 stack elem

Advanced Variables

`bipush 10`	push byte 10 on stack
`aconst _null`	load `null` constant
`newarray int`	Create new array
`astore_1`	store array in var 1
`iastore` Stack: `[x,y,....]`	save int x in array at position y

Control flow

`ifle`	Jumpi if less equal
`if_icmpeq`	jump if equal
`if_icmplt`	jump if less theb
`invoke virtua 2 l`	call function 2
`ireturn`	return int
`goto <x>`	goto jump point x
`:<x>`	jump point x