**Stack-Based Virtual Machine for Expression Evaluation**

Check these instructions in [JVM Spec](http://lara.epfl.ch/w/compilation:jvm_spec): iadd, imul, iload, istore, bipush

[abstract](http://scala-lang.org) [class](http://scala-lang.org) Instruction

[case](http://scala-lang.org) [class](http://scala-lang.org) Bipush(c : Int) [extends](http://scala-lang.org) Instruction

[case](http://scala-lang.org) [class](http://scala-lang.org) Iadd [extends](http://scala-lang.org) Instruction

[case](http://scala-lang.org) [class](http://scala-lang.org) Imul [extends](http://scala-lang.org) Instruction

[case](http://scala-lang.org) [class](http://scala-lang.org) Iload(slot : Int) [extends](http://scala-lang.org) Instruction

[case](http://scala-lang.org) [class](http://scala-lang.org) Istore(slot : Int) [extends](http://scala-lang.org) Instruction

[class](http://scala-lang.org) VM([var](http://scala-lang.org) code : Array[Instruction],

[var](http://scala-lang.org) pc : Int, // program counter (current instruction)

[var](http://scala-lang.org) local : Array[Int], // local variables

[var](http://scala-lang.org) operand : Array[Int], // operand stack

[var](http://scala-lang.org) top : Int // top of operand stack

)

{

[def](http://scala-lang.org) step = {

code(pc) [match](http://scala-lang.org) {

[case](http://scala-lang.org) Bipush(c) =>

operand(top + 1) = c

top = top + 1

[case](http://scala-lang.org) Iadd() =>

operand(top - 1) = operand(top - 1) + operand(top)

top = top - 1

[case](http://scala-lang.org) Imul() =>

operand(top - 1) = operand(top - 1) \* operand(top)

top = top - 1

[case](http://scala-lang.org) Iload(n) =>

operand(top + 1) = local(n)

top = top + 1

[case](http://scala-lang.org) Istore(n) =>

local(n) = operand(top)

top = top - 1

}

pc = pc + 1

}

[def](http://scala-lang.org) run = {

[while](http://scala-lang.org) (pc < code.length)

step

}

}

[object](http://scala-lang.org) Test {

[def](http://scala-lang.org) main(args : Array[String]) = {

[val](http://scala-lang.org) code : Array[Instruction] = List(

Iload(0),

Iload(1),

Imul(),

Iload(1),

Iload(2),

Imul(),

Iadd(),

Iload(0),

Iload(2),

Imul(),

Iadd(),

Bipush(2),

Imul()).toArray

[val](http://scala-lang.org) local : Array[Int] = List(3,4,5).toArray

[val](http://scala-lang.org) operand = [new](http://scala-lang.org) Array[Int](10)

[val](http://scala-lang.org) vm = [new](http://scala-lang.org) VM(code,0,local,operand,0);

vm.run

println(vm.operand(vm.top))

}

}