```scala
package main

import bc.ByteCode
import factory.VirtualMachineFactory
import vm.VirtualMachine

import scala.collection.immutable.Stack


/**
  * Created by Eric's laptop on 16/04/2017.
  */
class VirtualMachineImpl (stack: List[Byte]) extends VirtualMachine {

  var virtualMachine: VirtualMachine = new VirtualMachineImpl (stack)
  //val bcp = VirtualMachineFactory.byteCodeFactory

  override def execute(bc: Vector[ByteCode]): VirtualMachine = {

        //var virtualMachine: VirtualMachine = new VirtualMachineImpl (List())
        executeOne(bc)
        virtualMachine = executeOne(bc)._2
        if (bc.head.bytecode == null) {
          return virtualMachine
      } else {
        execute(executeOne(bc)._1)
      }
  }

  override def executeOne (bc: Vector[ByteCode]): (Vector[ByteCode], VirtualMachine)= {

    var yy = bc.head.execute(virtualMachine)
    (bc.tail, yy )
    }




  override def push(value: Int): VirtualMachine = {
    val stack1 = value.toByte :: stack
    val virtualMachine = new VirtualMachineImpl(stack1)
    virtualMachine
  }

  override def pop(): (Int, VirtualMachine)= {

    val poppedInt = stack.head
    val stack1 = stack.tail
    val virtualMachine = new VirtualMachineImpl(stack1)
    (poppedInt, virtualMachine)
  }
  override def state: Vector[Int] = {

    val vectorStack: Vector[Int] = stack.map(_.toInt).to[Vector
]
    vectorStack
  }
}
```