Lab 5: Bytecode executor

January 15, 2016

The goal of this lab is to develop a bytecode executor for a stack based virtual machine.

1 Files

Copy the directory */home/TDDA69/Labs/espy/VirtualMachine/* and all of its files to your espy directory. Also copy the file */home/TDDA69/Labs/espy/esvm.py* to your espy directory.

2 Instruction set

The instruction set for the bytecode executor is defined in the file *VirtualMachine/OpCode.py*. It is composed of 46 instructions, divided in 8 groups:

- *stack manipulation*
- *Environment and objects manipulation*
- *Control*
- *Exceptions*
- *Array and Objects creation*
- *Binary arithmetic operations*
- *Binary bolean operations*
- *Unary operations*

Carefully study the file side-by-side with the tests (in */home/TDDA69/Labs/Lab5/Tests/Executor.py*)

3 Bytecode executor

In this lab you will be developing the bytecode executor. Before starting, you should have a look at some of the existing class in the VirtualMachine directory.

- *VirtualMachine.Code* it contains a list of instruction defining a code that can be executed, be it a program or a function
- *VirtualMachine.Instruction* it defines a single instruction for a *Code* object, it contains the instruction number and possible arguments

- *VirtualMachine.OpCode* as mentioned in the previous section, it defines the instruction set
- *VirtualMachine.Stack* it defines the stack that is use by the virtual machine

The class *VirtualMachine.Executor* is the class that you should implement in this lab.

4 Run the test

You can run the test suite for the executor with the following command:

```
tdda69_lab5_tests dir_to_espy
```