Gabriel Alfaro

ID: 913142053

CSC 413.02 Summer 2018

https://github.com/csc413-02-su18/csc413-p2-GabrielAlfaro

2. Introduction

a. Project Review

The java project involves implementing an interpreter class that can read in and translate a language X. The whole program reads in the source code from a file and converts the language into strings that the interpreter can find and execute the needed functions. Along with implementing a Virtual Machine which will talk with the Program class and be the middle man for instructions being passed and executed. We create the bytecodes and as the strings are read and pushed into a list they are read with the variables that were also read in after the function name. Then the ByteCode class acts as the abstract class to give the byte codes the information and allow the byte code classes to perform their action.

b. Technical Review

in this java project we were given an Interpreter that communicated with the Virtual Machine and other classes to be able to understand and run the code from the mock language given to us. The program class stores the bytecodes and their indexes, having the addresses converted for the virtual machine to read. The ByteCodeLoader takes the Source file line by line and separates the strings into respective tokens. Eventually loading all the codes into ByteCodeLoader and creating instances of each. The RunTimeStack contains the runTimeStack array list and frame Pointer stack, essentially keeping track of what byte code has been run and where on the runTimeStack the frame Pointer is or needs to go. The Virtual Machine, acting as the middle man then uses a loop to get the codes at each line of the program and executing them based on the byte code they matched. Virtual Machine is also running smaller function to compute the needed action. Which Interpreter calls one by one, so the Virtual Machine is talking with the Interpreter to evaluate and translate the incoming strings from the Source file. Then inheritance is used for ByteCode to allow the other byte code subclasses to have access to their members and variables. This allows for when a byte code is read to create the bytecode object and being able to call Bytecode, using polymorphism to take the form of a needed byte code class and call that class with the parameters read from the source file.

c. Summary of Work Completed

My program doesn’t work. I couldn’t fully implement all the classes or create a functional abstract Byte Code class. The code isn’t much and will not run without errors. I’m annoyed but also even more motivated to not just understand the next project but to go back and fully complete the other two projects.

3. Development Environment

   a. Version of Java used

      -JDK 1.8 (Default)

   b. IDE used

      -NetBeans IDE 8.2

4. How to build or import your game in the IDE used

   a. first you clone the java files from the link to the repositories in the title page.

   b. Unzip the cloned java file.

   c. Open a NetBeans IDE with at least version 8.2.

   d. Click on File -> new Project and click on Java with existing sources.

   e. Click next and find the version of Java cloned.

   f. Once the main package “interpreter” is found, select it and click Finish.

   g. Go to your command Prompt and type: java Interpreter (filename to be read).

5. How to run your project

To run the project, pass the file through to Interpreter and the program should run. If the dump is on, then the Bytecodes will be printed out.

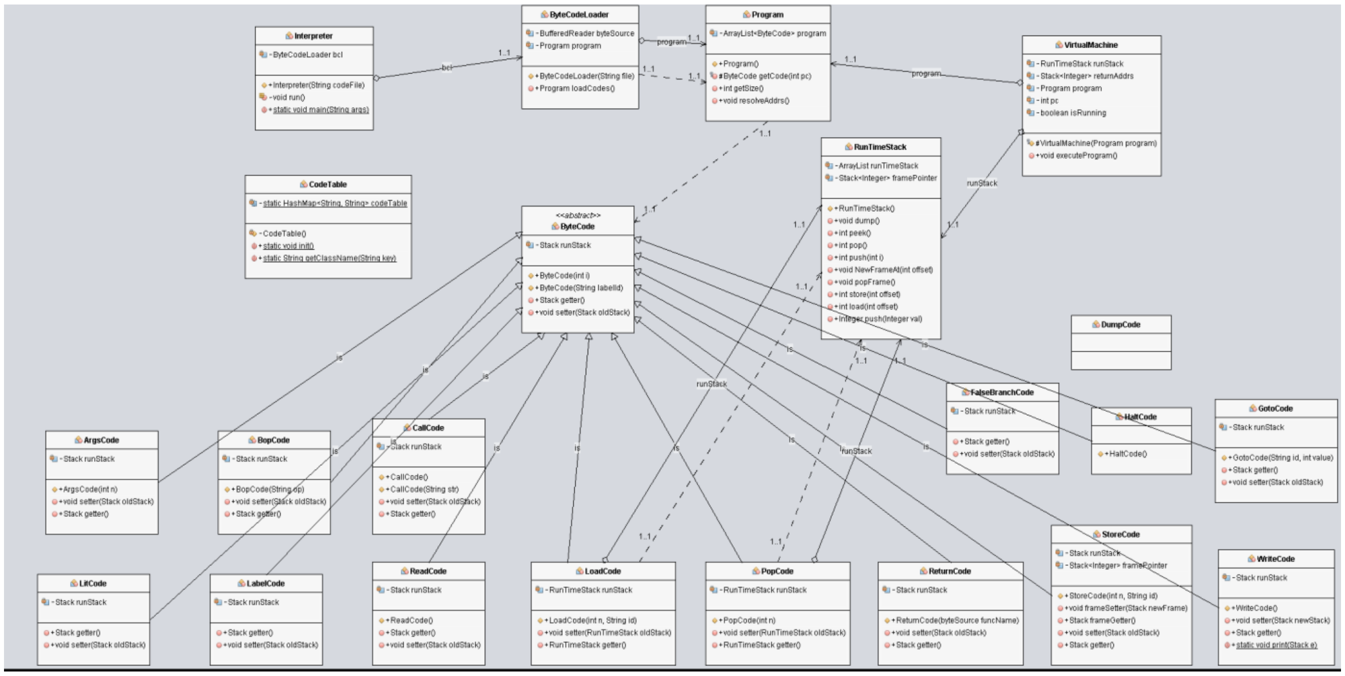
6. Assumptions Made when Designing and Implementing your Project

- Assuming the file read has valid strings to be read.

- Assuming the values being handled are integers.

- Assume no errors in the given bytecode source programs.

7. Implementation Documentation



8. Projection Reflection

The project was very interesting and something I have never seen before in terms of understanding all the moving objects. This also led to a bit of confusion with the variable names and having to adjust to the name of a variable but also the type. Suggestion wise, I would say use more visuals to help describe the relationships like Virtual Machine using Program for the byte codes. I like this project more than the calculator because it was something I haven't really done and helps me get more comfortable with having to keep track of several variables and functions.

9. Project Conclusion and Results

The project overall did not work, I was not able to successfully implement the dump code along with others. I became confusing while trying to understand the implementation of Virtual Machine and the example given on pop code.