

CSC 413 Project Documentation

Summer 2020

Jarvis Yu

920584779

CSC413-02

<https://github.com/csc413-02-SU2020/csc413->

p2-yujarvis08

Table of Contents

| | |
|-----|-------------------------------------|
| 1 | Introduction. 3 |
| 1.1 | Project Overview.. 3 |
| 1.2 | Technical Overview.. 3 |
| 1.3 | Summary of Work Completed. 3 |
| 2 | Development Environment. 3 |
| 3 | How to Build/Import your Project. 3 |
| 4 | How to Run your Project. 3 |
| 5 | Assumption Made. 3 |
| 6 | Implementation Discussion. 3 |
| 6.1 | Class Diagram.. 3 |
| 7 | Project Reflection. 3 |
| 8 | Project Conclusion/Results. 3 |

1 Introduction

1.1 Project Overview

This program reads files and uses functions to calculate things. Depending on the file, the program will compute the factorial of a number, or a Fibonacci number. The numbers and functions are stored onto a stack and the program works its way one by one on the stack to compute these numbers.

1.2 Technical Overview

This program reads a file that is written in a mock language. The file contains bytecodes and arguments for the program to use. The program runs the bytecodes through the Virtual Machine. It reads the bytecodes and determines how they will function. The

bytecodes are then pushed and popped from the stack and frame pointer to calculate the result of whatever integer is inputted.

1.3 Summary of Work Completed

I created the classes for the various bytecodes such as ArgsCode, CallCode, RetrunCode, etc. I then implemented various functions to use in the Virtual Machine and RunTimeStack classes to use for the bytecodes. I then filled in the function resolveAddress and executeProgram in the Program and Virtual Machine classes.

2 Development Environment

Java SE Development Kit 12.0.2

IntelliJ IDEA 2020.1.1 Ultimate Edition

3 How to Build/Import your Project

Open IntelliJ

Then click Open or import

Find and open the root folder of the project

Build the project by clicking on the nav bar on the top

4 How to Run your Project

In IntelliJ, run Interpreter

Then edit the configurations on the top right dropdown table

In the program arguments line, type factorial.x.cod for calculating factorial

Or type fib.x.cod for fibonacci

Then click OK and run the program

5 Assumption Made

Variables in classes must be private

Functions can be public

Virtual Machine is used to run almost everything for bytecodes

6 Implementation Discussion

The bytecodes are stored in a separate location from the virtual machine, bytecode loader, and program. All of the bytecode functions is ran through the virtual machine by using the runVM function. The addresses are resolved through the resolveAddress function in the program file.

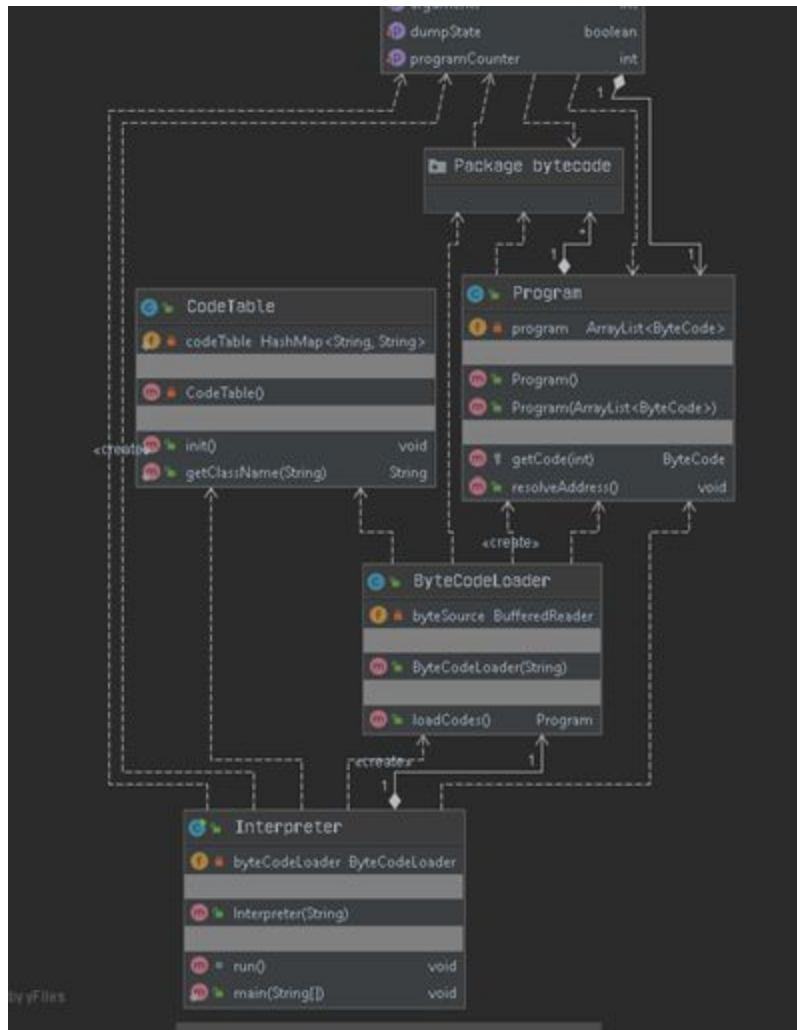
6.1 Class Diagram

| RunTimeStack | |
|-----------------|--------------------|
| runTimeStack | ArrayList<Integer> |
| framePointer | Stack<Integer> |
| RunTimeStack() | |
| dump() | void |
| peek() | int |
| pop() | int |
| newFrameAt(int) | void |
| popFrame() | void |
| store(int) | int |
| load(int) | int |
| push(int) | int |
| arguments | int |

1

<create>

| VirtualMachine | |
|-------------------------|----------------|
| runTimeStack | RunTimeStack |
| returnAddress | Stack<Integer> |
| program | Program |
| isRunning | boolean |
| VirtualMachine(Program) | |
| halt() | void |
| peekRTS() | int |
| pushRTS(int) | void |
| pushProgramCounter(int) | void |
| popRTS() | int |
| popFrame() | void |
| storeRTS(int) | void |
| storeProgramCounter() | void |
| loadRTS(int) | void |
| newFrameAt(int) | void |
| returnProgramCounter() | void |
| executeProgram() | void |



Package bytecode contains all the bytecode classes, I didn't know how to show those in the diagram in IntelliJ

7 Project Reflection

This project was definitely way harder than the first one. The amount of work needed in this assignment was more than the first assignment. I was very overwhelmed by the number of things that I needed to implement. When I implemented these things, there

were many problems when running the code. The errors that I was getting felt impossible to fix because it didn't really tell me what was wrong, it just gave general exceptions. Rereading the instruction pdf many times still got me nowhere. This project really frustrated me. It got to a point where I decided getting a 50% would be better than trying to fix the exceptions.

8 Project Conclusion/Results

After completing this program, I think I have not learned anything to be honest. I still don't know how this program really functions. With the big number of functions needed to be implemented and the number of bytecode classes needed to be created, it makes the whole project very confusing.