

Assignment 02 Documentation

Name : Gio Jung

Student ID : 917160535

Class : Summer 2023, CSC - 413.02

Repositories Link : <https://github.com/csc413-SFSU-Souza/csc413-p2-GioJung97>

Introduction

Project Overview

This is a documentation regarding the project called “Interpreter”. This program is a special calculator which can calculate factorial and also find Fibonacci sequence with the given bytecodes. The project requires fundamental skills and background of Java in order to run the program properly. The basic frame of the code of the project were given and I needed to complete the code by filling the blank methods or creating new classes and interface. After completing the code, the program gives all the process of calculating factorial or Fibonacci sequence depend on the number of user input. If a user inputs other than integer, the program will ask again until the user inputs integer.

Technical Overview

This is a documentation regarding the project called “Interpreter”. The project requires the Data structure skill in order to make the code efficient and prevent the time complexity. My job is to implement an interpreter for the mock language X which is basically writing JVM for java. Regarding the project, there are three big parts; Bytecode, Virtual Machine, and Run Time Stack. Basically, whenever bytecodes requires something from Run Time Stack, the program should request Virtual Machine to get information. Run Time Stack should not be touched and leaked. However, there are 15 bytecodes that I had to create and each bytecodes have different

roles on this program. Skeleton codes are given, so I had a lot of freedom to complete the code. Factorial and Fibonacci bytecodes are also given, so the completed program will show all the process of calculating Fibonacci sequence and Factorial numbers.

Summary of Work Completed

I have completed the work to run the program properly. It was huge project to work on for me, so I started earlier to understand what the program does. I started filling out RunTimeStack class first and created 15 bytecodes classes. Some byte codes are not to be dumped and some bytecodes are able to jump to the stored address (line), I created two clone interface which are dumpable and jumpable, so I could tie certain bytecode classes to write high level detail of codes in Virtual Machine class. I did same thing in resolveAddress method from Program class. Additionally, for BOP bytecode, since I've created operator class from the "Calculator" project, I copied and pasted the operator class and added extra Binary Operation classes.

Development Environment

Version of Java Used : OpenJDK 20.0.1

IDE used : IntelliJ

How to build or import my game in the IDE I used

IntelliJ is one of the IDE (Integrated Development Environment) program for Java and other programming languages. I personally think that it is very convenient for new users to get

used to coding. However, when you try to build a new project, you can simply make a new project. Then it will automatically create several files such as “src” files or Java Libraries (accessible to use). In the “src”, you can create new file such as “class”, “Interface”, or “enum” which most of the coding are taken a place. On the other hand, when you try to import my game in the IDE, you can use git command. You can open Git Bash or terminal and type “git clone <https://github.com/csc413-SFSU-Souza/csc413-p2-GioJung97>”. Then the project file will automatically imported to your computer. Now, you can simply open the project by IntelliJ.

How to run my project

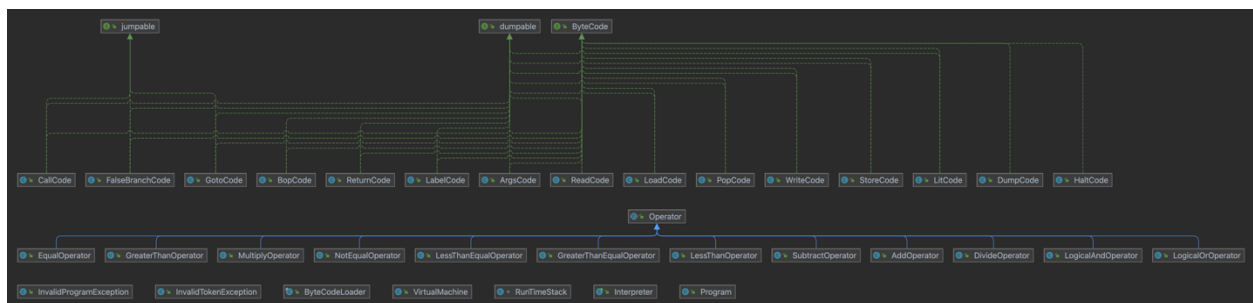
To run my project, you need to carefully choose which class you would like to compile. The only class that can be compiled is Interpreter class. However, since this program requires byte codes files to be read, you need to go through “Edit Configuration”. Once you find “Program Argument”, you need to type the name of the file that is going to be read. For example, “factorial.x.cod”, or “fib.x.cod”. Before you run the program, if you want the program shows all the process of the calculation, you need to make sure if the first line of the files is “DUMP ON” otherwise the program will only show the answer. Once you hit the run button, or compile button, the program will ask an integer which you want it to be calculated and it will show all the process of work.

Assumptions Made when designing and implementing my project

While I was designing and implementing my project, I made several assumptions. A user can only type an integer with this program, so I assumed that I needed to make a loop until the user types an integer. One important thing is that bytecodes cannot request directly to the data structure which is Run Time Stack in this project. So I assumed that I needed to make some methods from Virtual Machine in order to bring data from Run Time Stack, so whenever bytecodes are executing, it can request to VM for the data. For BOP bytecode, since it is about binary operation, I assumed that I could bring the operator codes from the first project and make it applicable to this project, so I could save time creating each binary operation classes.

Implementation Discussion

I was told from the professor that I needed to create 15 byte code classes. So I created bytecodes interface to tie all the byte code classes. Also, since some bytecodes are not to be dumped or to be jumped, I created dumpable and jumpable cloned interface which could avoid low level detail (abstraction) in VM class and Program class. Lastly, all the Binary Operators are created under Operator class which is being used from BOP bytecode.



Project reflection

Since it's big project for me, I was overwhelmed again as I felt for the first project. To be honest, I tried to understand how the complete program works first so I could understand what I was doing, but I didn't fully understand until I tested the draft code. In other words, I was not sure what I was doing while I was creating each byte code. However, once I was able to test the code, I could clearly see how the program should be worked and I could solve the problems that I've made and got more confident on what I had to work on. The hardest part was storing data from RTS to VM and bringing the data in Bytecode because I was concerned if I was leaking any information of RTS which is illegal. Turned out, it was something that I've already known how to do, so I was able to manage storing data. Lastly, using cloneable interface was my first time using while implementing, so I could clearly understand how it works which was a good experience. If I had more experience and knowledge of implementing Java code, I could optimize the code and make more high level abstractions which is one of the purposes of this project. However, I feel like I learned a lot from this project.

Project Conclusion and Results

Overall, this project was great experience for me to gain more knowledge of diverse way of implementing. I have faced many problems while implementing though, I ended up solving all the problems I've had. The program works properly which was one of the main goal for me, however, I am sure that there are better way to save more on time complexity and organize codes with more high level details. This will be my next challenge once I get more knowledge using Java.