Steven Thaw

sjthaw@me.com

Abstract

A Java program which constructs a graph from userselected file and returns the parenthesized list, hierarchy, unreachable vertices, and all vertices of the graph.

CMSC 350 Data structures and analysis

Project 4

Lessons Learned

This assignment’s lessons learned has less to do with programming and more to do with time management. Due to work schedules, wife deployment, being a single parent of 3 now that wife is deployed, and other life issues, I didn’t even get to start the reading until Thursday. I felt very rushed, which lead to a lot of simple mistakes which could have been avoided if I had devoted an hour during the week to doing the readings, so I would have been able to start coding sooner. When I am rushed I find I make more stupid mistakes than when I have a lot of time. On the programming side of the lessons learned, I learned how to better use StringBuilder, which I never really had a firm grasp on, as well as how to use JTextAreas vice JTextFields. I still need a lot of StringBuilder and GUI learning, but was able to manage figuring out what I needed for this assignment thanks to the Java Docs and Geeksforgeeks.com. The final lesson learned, which honestly I think was the most important, is organization. I had a really hard time getting started, so I decided to begin with a pen and paper and wrote out the function definitions of what I needed to do before I actually wrote any code. It really helped having the function names mapped out. Once that I was done I wrote the functions as code, and then filled the functions as I went along. It was very helpful having an outline map to follow when it came to actually writing code.

UML Class Plan



Test Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Input | Expected Output | Pass/Fail |
| 1 | ClassA ClassC ClassE ClassJ  ClassB ClassD ClassG  ClassC ClassA  ClassE ClassB ClassF ClassH  ClassJ ClassB  ClassI ClassC | Parenthesized: ClassA ( ClassC \* ClassE ( ClassB ( ClassD ClassG ) ClassF ClassH ) ClassJ ( ClassB ( ClassD ClassG ) ) )  Hierarchy: ClassA  ClassC \*  ClassE  ClassB  ClassD  ClassG  ClassF  ClassH  ClassJ  ClassB  ClassD  ClassG  Unreachable: ClassI is unreachable  All Verticies: All Vertices:  ClassA: ClassC ClassE ClassJ  ClassC: ClassA  ClassE: ClassB ClassF ClassH  ClassJ: ClassB  ClassB: ClassD ClassG  ClassI: ClassC | Pass |
| 2 | Class1 Class2 Class3  Class4 Class5 | Parenthesized List: Class1 ( Class2 Class3 )  Hierarchy:  Class1  Class2  Class3  Unreachable: Class4 is unreachable  All Verticies:  All Vertices:  Class1: Class2 Class3  Class4: Class5 | Pass |
| 3 | Class1 Class2 Class3 Class4 Class5 Class6  Class2 Class3 Class4 Class5 Class6 | Parenthesized List: Class1 ( Class2 ( Class3 Class4 Class5 Class6 ) Class3 Class4 Class5 Class6 )  Hierarchy:  Class1  Class2  Class3  Class4  Class5  Class6  Class3  Class4  Class5  Class6  Unreachable:  All Verticies: All Vertices:  Class1: Class2 Class3 Class4 Class5 Class6  Class2: Class3 Class4 Class5 Class6 | Pass |
| 4 | Empty File | Display JOptionPane dialog “Please select a valid file” | Pass |

A screenshot of a cell phone

Description automatically generated

Figure Test Case 1

A screenshot of a cell phone

Description automatically generated

Figure Test Case 2

A screenshot of a cell phone

Description automatically generated

Figure Test Case 3

A screenshot of a cell phone

Description automatically generated

Figure Test Case 4