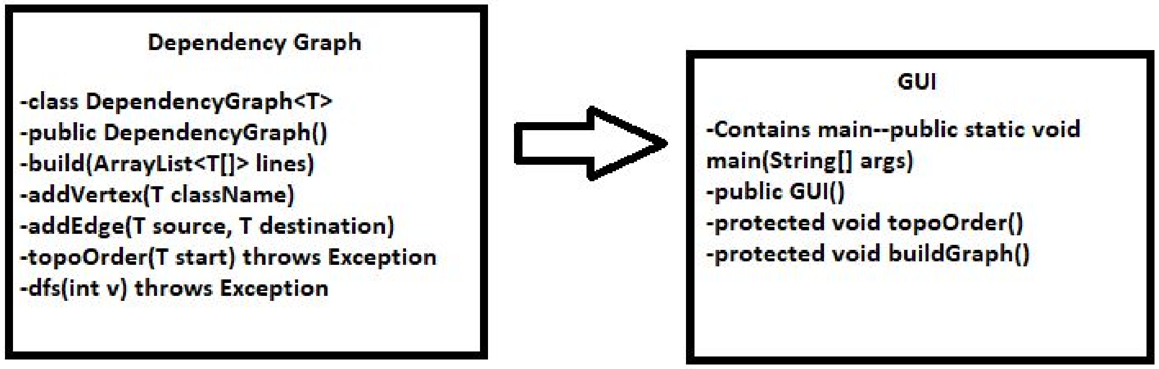
CMSC 350 7983 Data Structures and Analysis

Project 4

Name: Mark Tasker

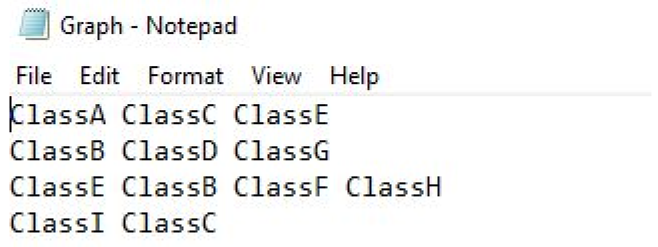
Date: December 15th, 2019

UML Diagram:


Dependency Graph
-class DependencyGraph<T>
-public DependencyGraph()
-build(ArrayList<T[]> lines)
-addVertex(T className)
-addEdge(T source, T destination)
-topoOrder(T start) throws Exception
-dfs(int v) throws Exception
GUI
-Contains main--public static void main(String[] args)
-public GUI()
-protected void topoOrder()
-protected void buildGraph()

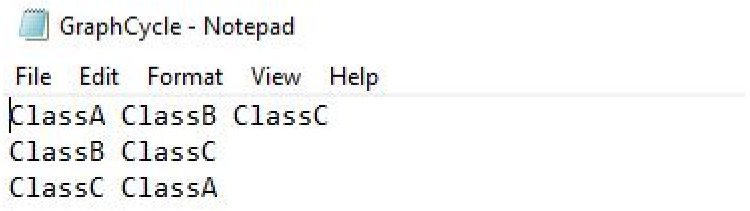
Text Files:

Graph.txt



```
ClassA ClassC ClassE
ClassB ClassD ClassG
ClassE ClassB ClassF ClassH
ClassI ClassC
```

GraphCycle.txt

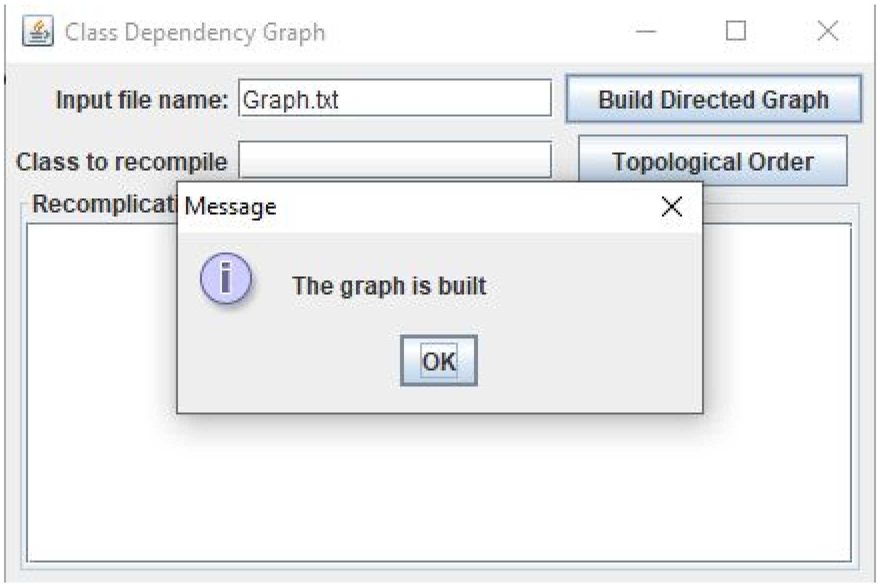


```
ClassA ClassB ClassC
ClassB ClassC
ClassC ClassA
```

Test Plan

Test Case 1a: Testing that program sorts classA dependencies. No Cycle

1) Enter Graph.txt into input file name text field
2) Click Build Directed Graph button
3) Confirm that graph was built
4) Enter ClassA into the Class to recompile text field
5) Click Topological Order button
6) Correct dependencies ClassA ClassC ClassE ClassB ClassD ClassG ClassF ClassH listed

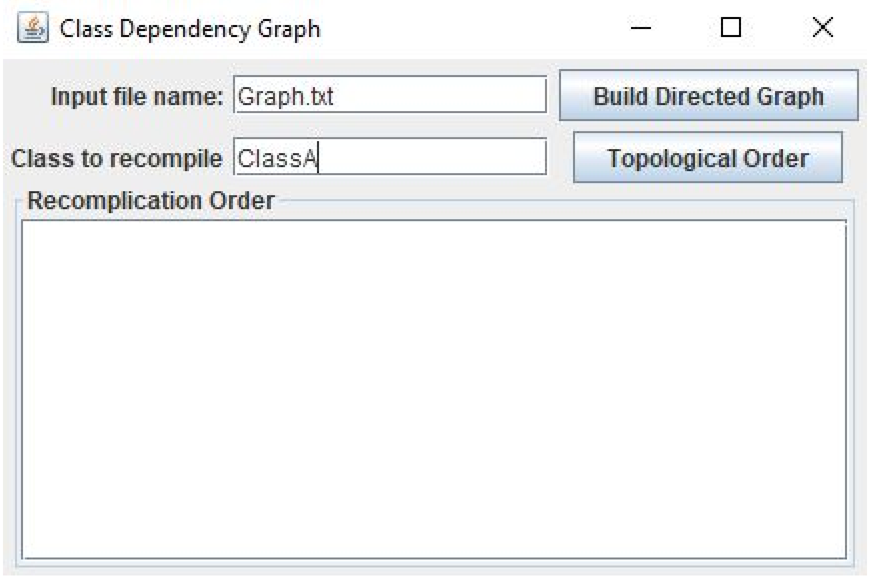

Class Dependency Graph

Input file name: Graph.txt

Class to recompile

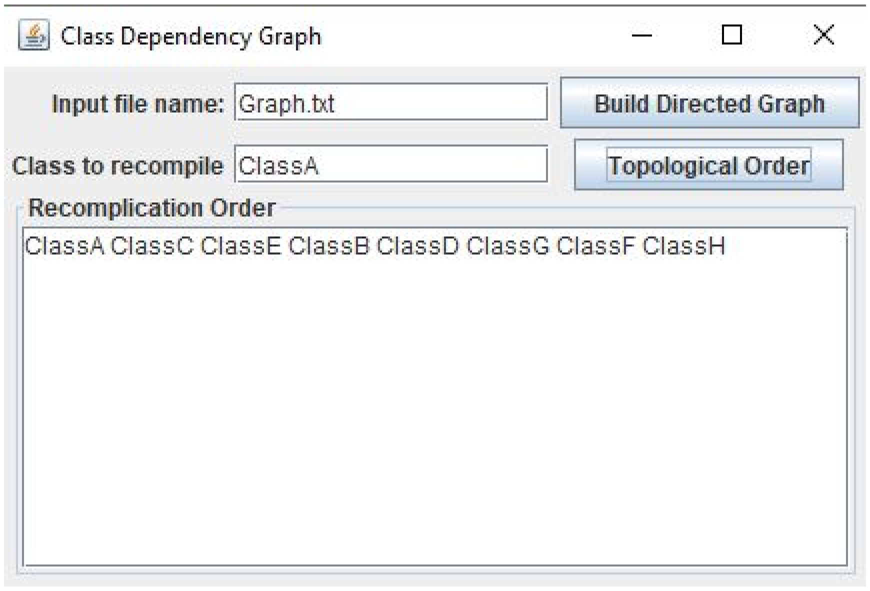
Message

The graph is built


Class Dependency Graph

Input file name: Graph.txt

Class to recompile ClassA


Class Dependency Graph

Input file name: Graph.txt

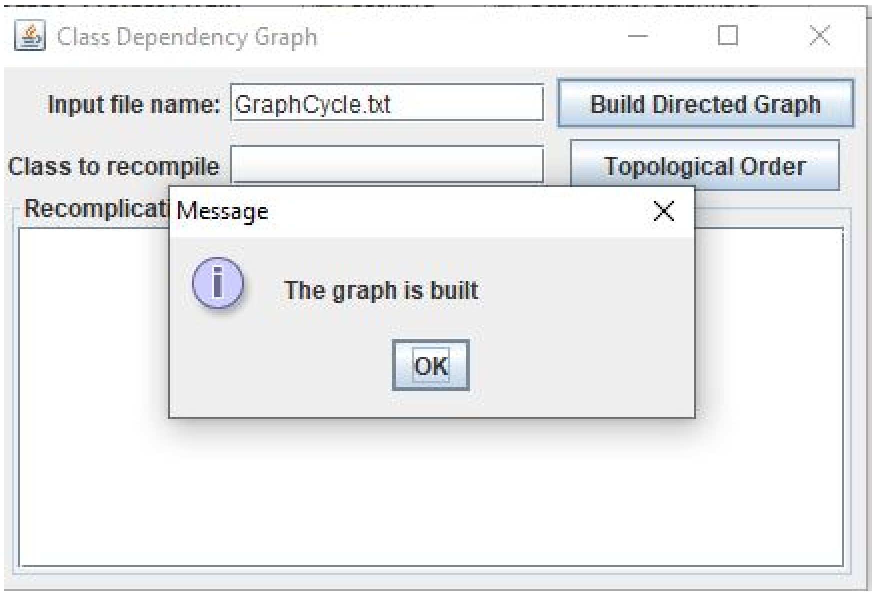
Class to recompile ClassA

Recomplication Order

```
ClassA ClassC ClassE ClassB ClassD ClassG ClassF ClassH
```

Test Case 1b: Testing how program interacts with cycle graph

1) Enter GraphCycle.txt into input file name text field
2) Click Build Directed Graph button
3) Confirm that graph was built
4) Enter ClassA into the Class to recompile text field
5) Click Topological Order button
6) Confirm error dialogue popup


Class Dependency Graph

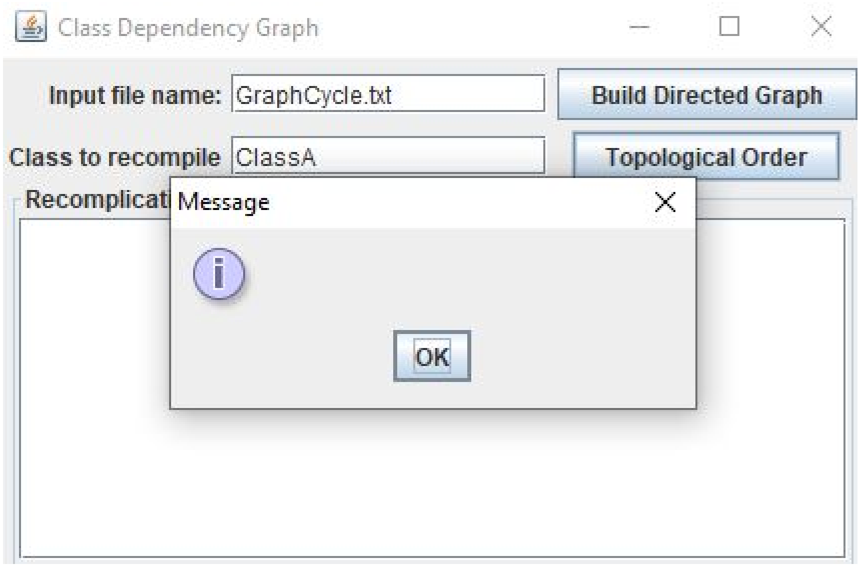
Input file name: GraphCycle.txt

Class to recompile

Recomplicati

Message

The graph is built

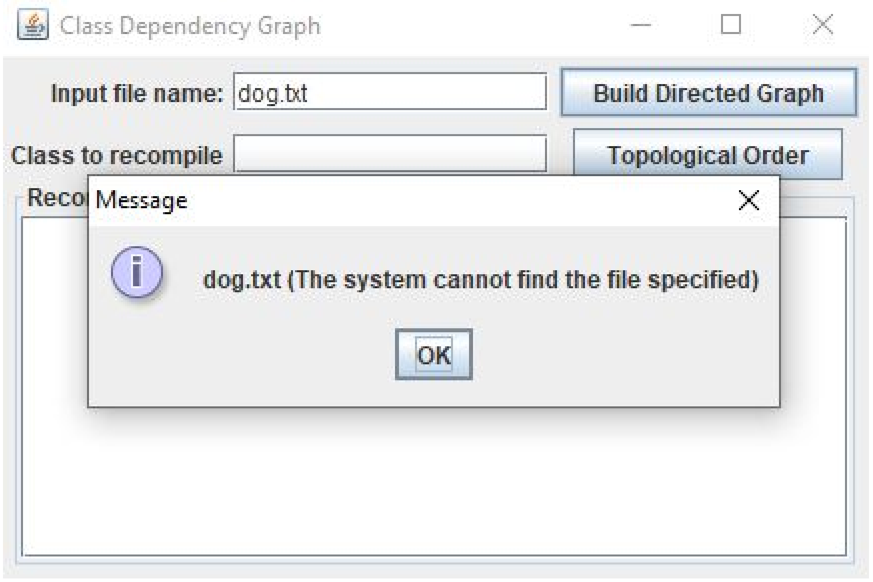


Input file name: GraphCycle.txt

Class to recompile ClassA

<u>Test Case 2:</u> Test how program interacts with incorrect file name for input file

1) Enter dog.txt
2) Click Build Directed Graph button
3) Confirm error prompt indicating file not found.

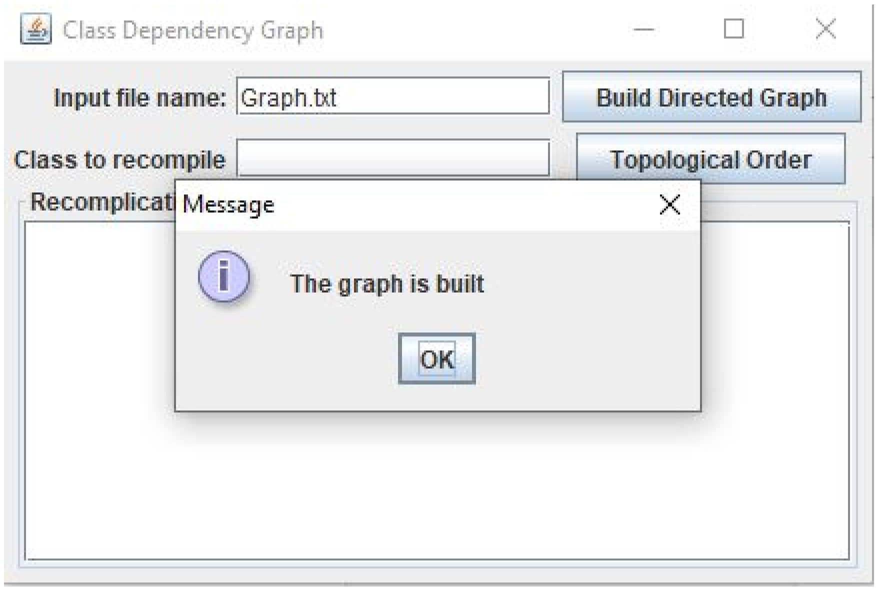


Input file name: dog.txt

dog.txt (The system cannot find the file specified)

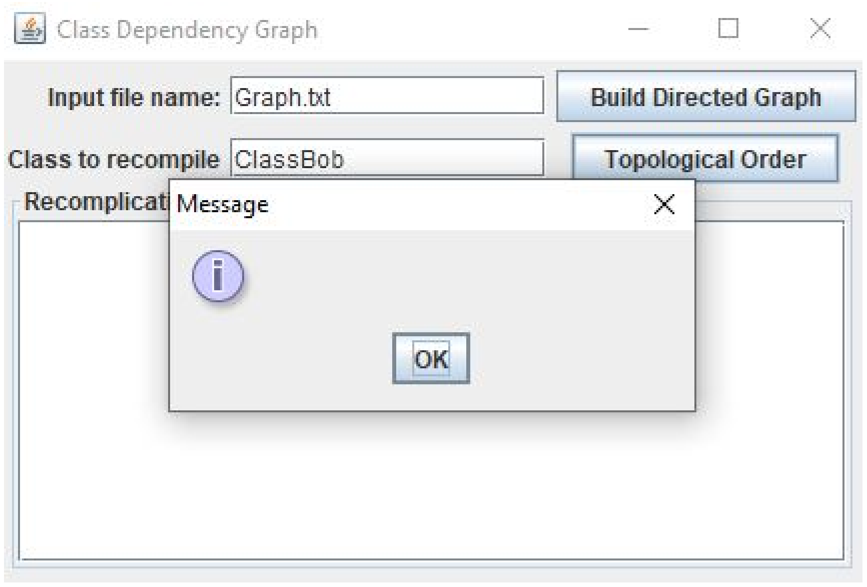
<u>Test Case 3:</u> Test how program responds to an invalid class name after graph file loads successfully.

1) Enter Graph.txt into input file name text field
2) Click Build Directed Graph button
3) Confirm that graph was built
4) Enter ClassBob into Class to recompile text field
5) Confirm error prompt



Graph.txt

The graph is built



Graph.txt

ClassBob

Lesson Learned Discussion

During Project 4 I learned to write a program that behaves like a Java command line compiler. The program recompiled particular classes and provided their respective dependencies. As part of the project code, I had to learn a lot more about how topological ordering works so that I could make it function for the program. In addition, I’ve learned a lot about handwriting the GUIs, but I haven’t really figured out how to make a GUI that isn’t ugly. I’ve interacted with Java-based GUIs at my job, and they always seem to look so dated and old. There’s something very 1980’s about Java, and I hate it. That being said, I know the primary purpose of these projects has been to learn the concepts of the code, but it really gets to me how ugly everything looks.