

# Project 4

William Crutchfield

IntelliJ Ultimate 2017.1.3

Windows 10

## Contents

Assumptions.....	3
UML Diagram .....	4
Test Cases.....	5
Final Product .....	6
Lessons Learned .....	10

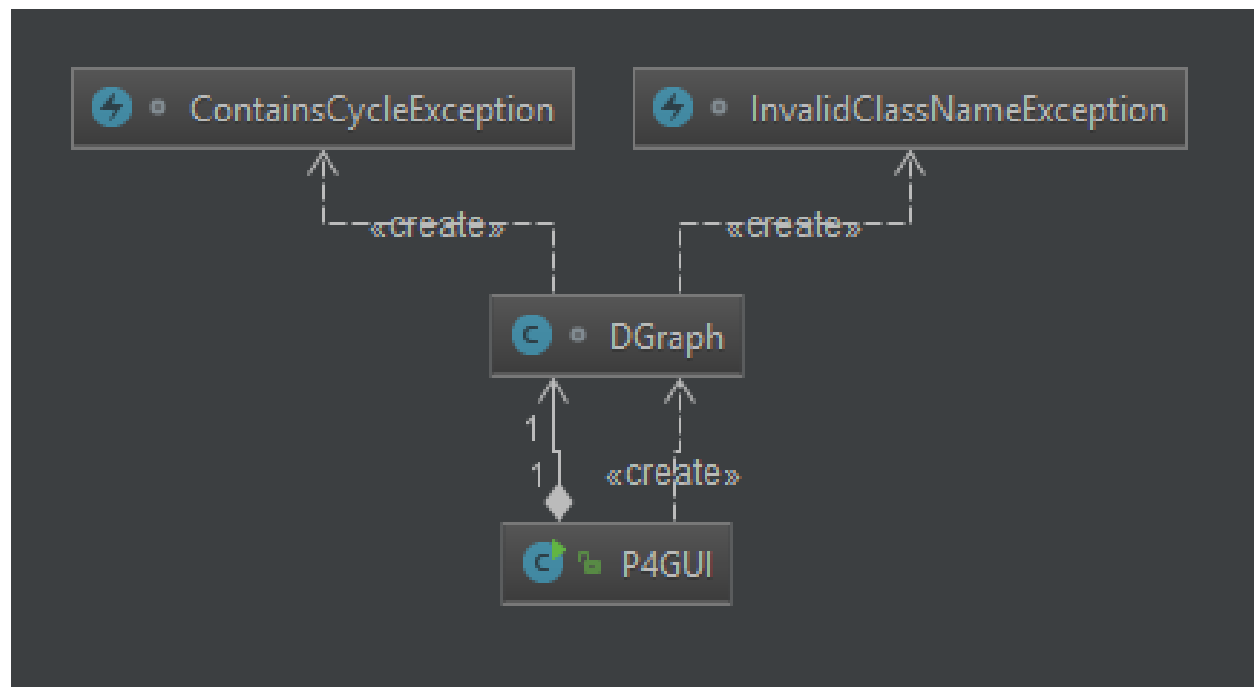
# Assumptions

---

In this project, the only assumption I made was where the program reads the text files from. I made a “res” directory in my project where the text files must be located, to be read by the program. I figured it was the best/most convenient way to go about it.

# UML Diagram

---



# Test Cases

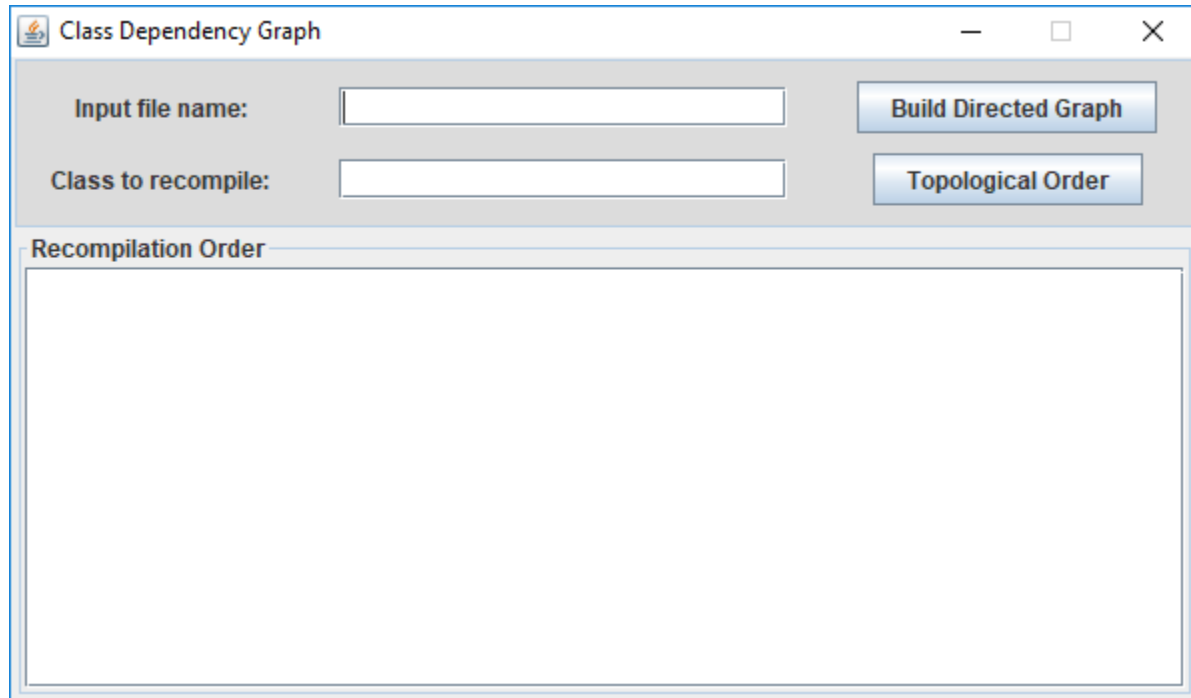
---

Aspect Tested	Input	Expected Output	Actual Output	Test Outcome
Original File - graph.txt (Graph without Cycles)	"graph.txt", "ClassA"	Topologically sorted graph starting from ClassA	ClassA ClassC ClassE ClassB ClassD ClassG ClassF ClassH	Passed
Graph with Cycles	"graph2.txt", "ClassE"	Error - Cycle Detected	Error - Cycle Detected	Passed
Invalid File Name	"graph"	Error - Invalid File	Error - Invalid File	Passed
Invalid Class Name	"Graph.txt", ClassZ	Error - Invalid Class	Error - Invalid Class	Passed

# Final Product

---

Upon starting the program, a window will open that looks like this.



**Class Dependency Graph**

Input file name:

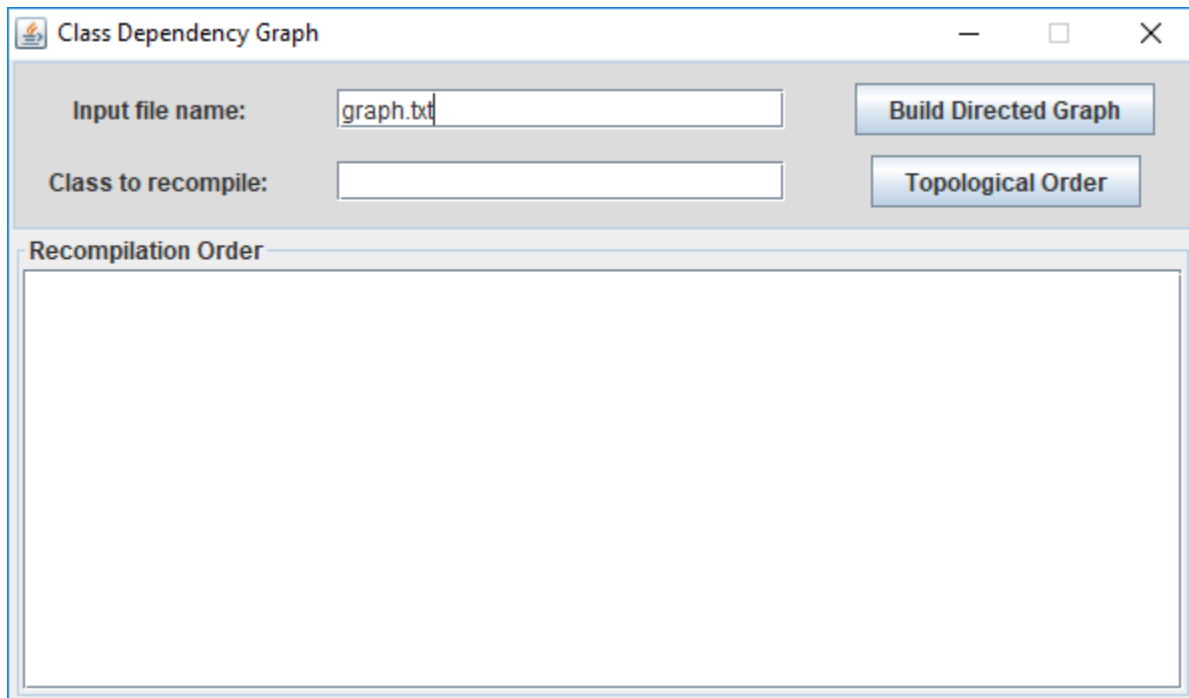
Class to recompile:

**Build Directed Graph**

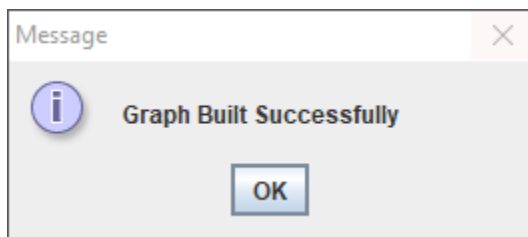
**Topological Order**

**Recompilation Order**

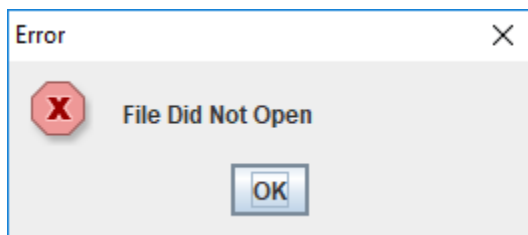
To start off, we can enter a file name into the appropriate textbox.



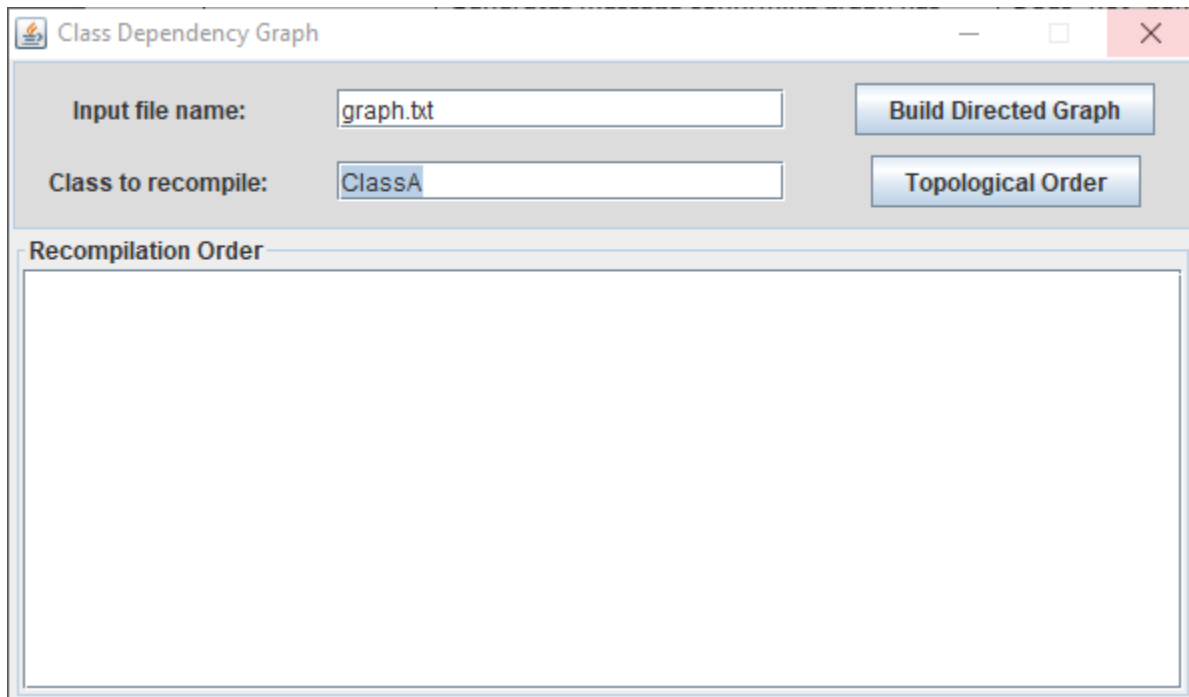
Next, we can click on the “Build Directed Graph” button. If it is successful, we will see this message box pop up!



However, if we input an invalid file name we will see this message.

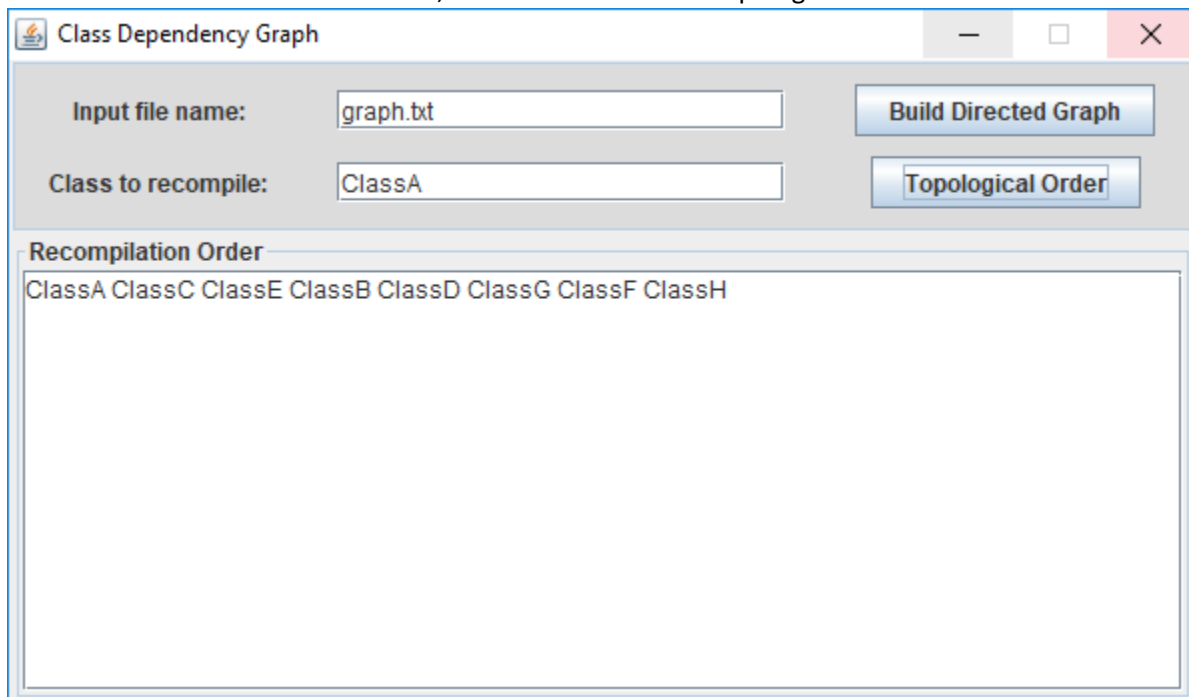


Next we can input a Class to recompile, we will use ClassA.



The image shows a window titled "Class Dependency Graph" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there are two input fields and two buttons. The first input field is labeled "Input file name:" and contains the text "graph.txt". The second input field is labeled "Class to recompile:" and contains the text "ClassA". To the right of the first input field is a button labeled "Build Directed Graph". To the right of the second input field is a button labeled "Topological Order". Below these fields and buttons is a large rectangular area labeled "Recompilation Order" which is currently empty.

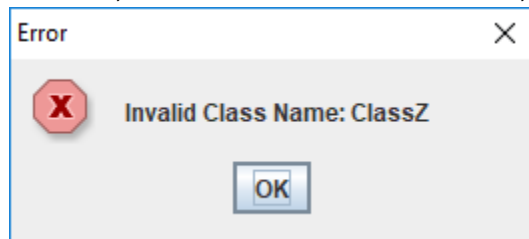
Once we have entered a Class name, we can then click the Topological Order button.



The image shows the same "Class Dependency Graph" window as before, but now the "Topological Order" button has been clicked. The "Recompilation Order" area now displays the text "ClassA ClassC ClassE ClassB ClassD ClassG ClassF ClassH" in a single line.



However, if we enter an invalid Class name, for example "ClassZ" we will get this message.



# Lessons Learned

---

Overall, this was a fun project that I felt challenged my abilities. The biggest thing I took out of this project was how to use multiple Array Lists to effectively build a Directed Graph. Once, I figured out how to build all the appropriate variables, I had some trouble at first understanding how to topologically order the graph. Once I did some more research on depth-first search algorithms and re-read some of the readings from week seven, I could then implement the dfs algorithm correctly. I also felt a lot more comfortable when it came to generics on this project due to using them throughout the previous project.

On a side note, my output for graph.txt is not the exact same as yours, however, after double and triple checking the readings, as well as my outputs, it looks as if it is still topologically ordered. So, my question is, how come this is the case? It is due to my implementation? As both the example result, and mine, look to be correct. I would love to know the answer/your thoughts on this!

Lastly, I would just like to say that I thoroughly enjoyed this course, and the feedback you gave me throughout it. (On this project, I tried to improve the cohesiveness compared to last project, let me know if I took the right approach.) I can honestly say that I feel this has been the most beneficial CS course in my academic career so far! Thanks!