

## Readme- Ex2:

In this assignment we was asked to create a class of graph and some other part of the graph (node ,edge, Geolocation).

First we create the corner stone of the graph (node ,edge, Geolocation)

For each class we create getters and setters and main constructor .

After we finished programing the basic classes.

We planned the weighted Graph class from HashMap of nodes and HashMap of edges and her function(get node, get edge, add node, connect, node\_iter, edge\_iter...)

We create an iterator for the HashMap of the nodes and the HashMap of the edges so we can go over all the collection.

After that we planned the algorithm class that have shortest path distance function and is connected function. This class have also tsp algorithm and can discover the center of the graph.

We search for material for the algorithm so we can plan the algorithms right and with the best efficiency .

This Is some website that help us to programing the algorithm:

- 1) <https://howtodoinjava.com/java/collections/hashmap/shallow-deep-copy-hashmap/>
- 2) <https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/>
- 3) [https://en.wikipedia.org/wiki/Graph\\_center](https://en.wikipedia.org/wiki/Graph_center)
- 4) [https://en.wikipedia.org/wiki/Travelling\\_salesman\\_problem](https://en.wikipedia.org/wiki/Travelling_salesman_problem)

## **Test for the class:**

### **Node:**

- 1)check the weight of each node that we create -if it is accurate

### **Edge:**

- 1)check the weight of each edge that we create -if it is accurate

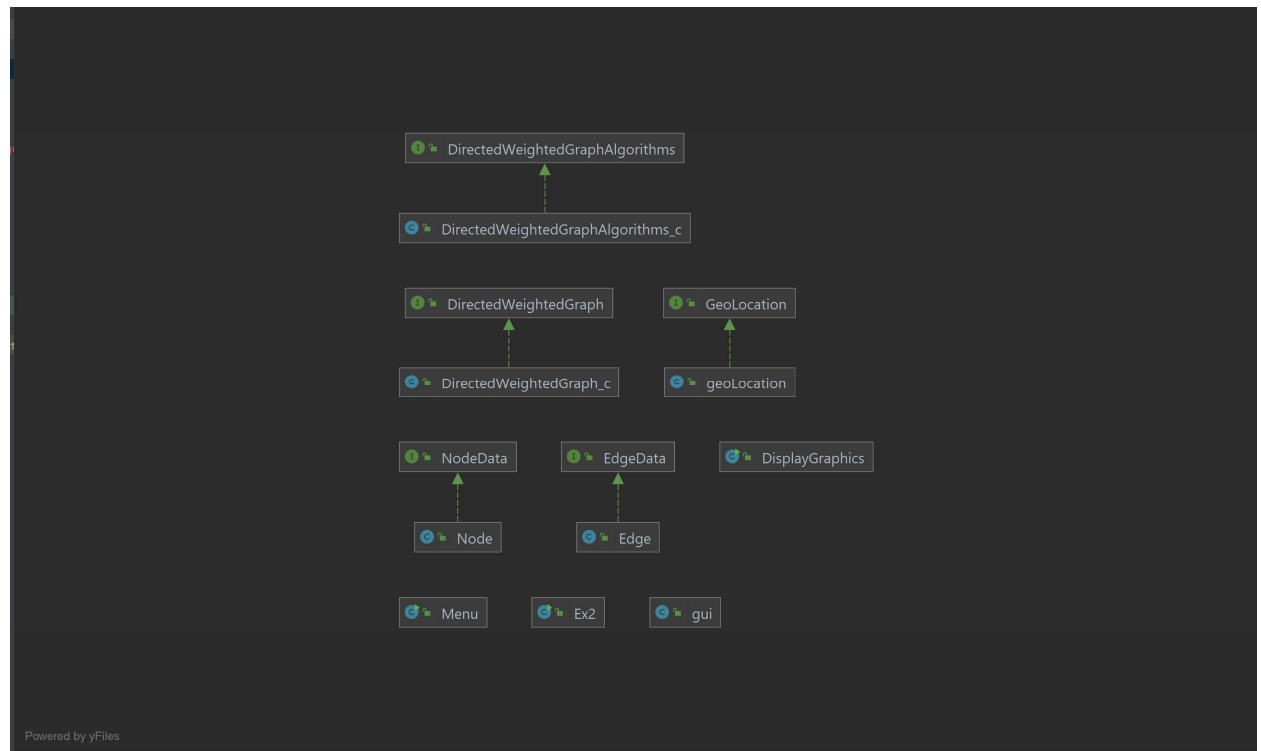
### **DirectedweightedGraph:**

- 1)check if the number of nodes is correct
- 2) check if the number of edges is correct
- 3)check if the graph that we have is really connected and the function is working
- 4)check that connect function create edge as it should be.

**DirectedweightedGraphalgo:**

- 1) check if the graph that we have is really connected and the function is working.
- 2) check if "tsp" and see if it solving the problem
- 3) check the center of each graph we create in the algorithm
- 4)

# UML



### **Explanation about Gui**

In our gui we have 2 menu bars .

The first use for load and save and the second for the function on the graph with the algorithm.

In the first menu bar we can load from json file and save the graph to a file.

In the second bar we have the algorithm function .to find the center of the graph or find the shortest path between two points in the graph.

We used dialog text for receive the argument that we need from the user.(with space between the arguments)

### **Table of time for running algorithm:**

Graph	<b><u>tsp</u></b>	<b><u>shotestpathdist</u></b>	<b><u>center</u></b>	<b><u>IsConnected</u></b>
<b><u>1000</u></b>	337 ms	46 ms	110 ms	62 ms