*Road*

compareTo – Compares 2 objects to see if they are equal

contains – Returns true only if the edge contains the given town

equals - Returns true if each of the ends of the road r is the same as the ends of this road.

getDestination - Returns the second town on the road

getName – Returns the road name

getSource – Returns the first town on the road

getWeight – Returns the distance of the road

toString – Returns String representation on object

*Graph*

getEdge - Returns an edge connecting source vertex to target vertex if such vertices and such edge exist in this graph. Otherwise returns null.

addEdge - Creates a new edge in this graph, going from the source vertex to the target vertex, and returns the created edge.

addVertex - Adds the specified vertex to this graph if not already present. More formally, adds the specified vertex, v, to this graph if this graph contains no vertex u such that u.equals(v).

containsEdge - Returns true if and only if this graph contains an edge going from the source vertex to the target vertex.

containsVertex - Returns true if this graph contains the specified vertex.

edgeSet - Returns a set of the edges contained in this graph. The set is backed by the graph, so changes to the graph are reflected in the set.

edgesOf - Returns a set of all edges touching the specified vertex (also referred to as adjacent vertices). If no edges are touching the specified vertex returns an empty set.

removeEdge - Removes an edge going from source vertex to target vertex if such vertices and such edge exist in this graph.

removeVertex - Removes the specified vertex from this graph including all its touching edges if present. More formally, if the graph contains a vertex u such that u.equals(v), the call removes all edges that touch u and then removes u itself.

vertexSet - Returns a set of the vertices contained in this graph. The set is backed by the graph, so changes to the graph are reflected in the set.

shortestPath - Find the shortest path from the sourceVertex to the destinationVertex call the dijkstraShortestPath with the sourceVertex

dijkstraShortestPath - Dijkstra's Shortest Path Method.