

Class

Town
implements Comparable<Town>

+ name : String
+ towns : ArrayList<Town>

- Town(name : String)
- Town(templateTown : Town)
- getName() : String
- compareTo(o : Town) : int
- toString() : String
- hashCode() : int
- equals(o : Object) : boolean
- getTowns() : ArrayList<Town>
- setTowns(towns : ArrayList<Town>) : void
- addTowns(town : Town) : void

Class

Road implements
Comparable<Road>

+ town : Town
+ town1 : Town
+ weight : int
+ name : String

- Road(source : Town, destination : Town, degrees : int, name : String)
- Road(source : Town, destination : Town, name : String)
- contains(town : Town) : boolean
- toString() : String
- getName() : String
- getDestination : Town
- getSource() : Town
- compareTo(r : Road) : int
- getWeight() : int
- equals(o : Object) : boolean
- setSource(town1 : Town) : void
- setDestination(town2 : Town) : void
- setDistance(distance : int) : void
- setName(name : String) : void

Class

Graph implements

GraphInterface < Town, Road >

- towns : ArrayList < Town >
- edges : ArrayList < LinkedList >
- hash : HashMap < Town, Town >
- hash2 : HashMap < String, < String, Road > >
- arr : ArrayList < String >
- + before : String[]
- + dest : int[]

- getEdge (sourceVertex : Town, destinationVertex : Town) : Road
- addEdge (sourceVertex : Town, destinationVertex : Town, weight : int, description : String) : Road
- addVertex (town : Town) : boolean
- containsEdge (sourceVertex : Town, destinationVertex : Town) : boolean
- containsVertex (town : Town) : boolean
- edgeSet () : Set < Road >
- edgesOf (vertex : Town) : Set < Road >
- removeEdge (sourceVertex : Town, destinationVertex : Town, weight : int, description : String) : Road
- removeVertex (town : Town) : boolean
- vertexSet () : Set < Town >
- shortestPath (sourceVertex : Town, destinationVertex : Town) : ArrayList < String >
- dijkstraShortestPath (sourceVertex : Town) : void

Shahed Pasi
Assignment 6 UML

UML Pg. 2

Class
TownGraphManager
implements TownGraphManagerInterface

- graph : new Graph()
- addRoad(town1 : String, town2 : String, weight : int, roadName : String) : boolean
- getRoad(town1 : String, town2 : String) : String
- addTown(v : String) : boolean
- getTown(name : String) : Town
- containsTown(v : String) : boolean
- containsRoadConnection(town1 : String, town2 : String) : boolean
- allRoads() : ArrayList<String>
- deleteRoadConnection(town1 : String, town2 : String, road : String) : boolean
- deleteTown(v : String) : boolean
- allTowns() : ArrayList<String>
- getPaths(town1 : String, town2 : String)
- populateTownGraph(selectedFile : File) : void
throws FileNotFoundException,
IOException