# **UML** pathFinder

# «interface» Comparable

### Class Graph

numVertices : intnumEdges : intgraph : Vertex[]

- cities : HashMap<String, Integer>

+ <<constructor>> (size : int)

+ addVertex()

+ countVertices(): int

+ countEdges(): int

+ addEdge()

+ addEdgeHelper()

+ getTotalDistanceInKm() : double

+ areAdjacent(): boolean

+ getAdjacencyList() : Vertex[]

+ heuristicStrategy() : double

+ heuristicCalculation() : double

+ aStarConnection() : ArrayList<Vertex>

+ aStarSearch() : ArrayList<Vertex>

+ findPath(): ArrayList<Vertex>

+ reset()

+ printOptimalPath()

## Class Edge

v1 : Vertexv2 : Vertex# weight : double

+ <<constructor>> (vert1: Vertex, vert2 :

Vertex, weight : double) + getWeight() : double

+ getOther() : Vertex

+ getEither() : Vertex

### Class Vertex

 $\hbox{-} straight Line: double\\$ 

# cities : String

# edges : ArrayList<Edge>

hCost : double# fCost ; double# gCost : double# parent : Vertex

+ <<constructor>> (name : String,

straightline : double

+ getEdge() : ArrayList<Edge>

+ getStraightLineDistance():

double

+ criteriaFunction()

+ setGCost()

+ setDistance() : double

+ shareEdge() : Edge

Class Navigator

+ main()

+ userInteraction()