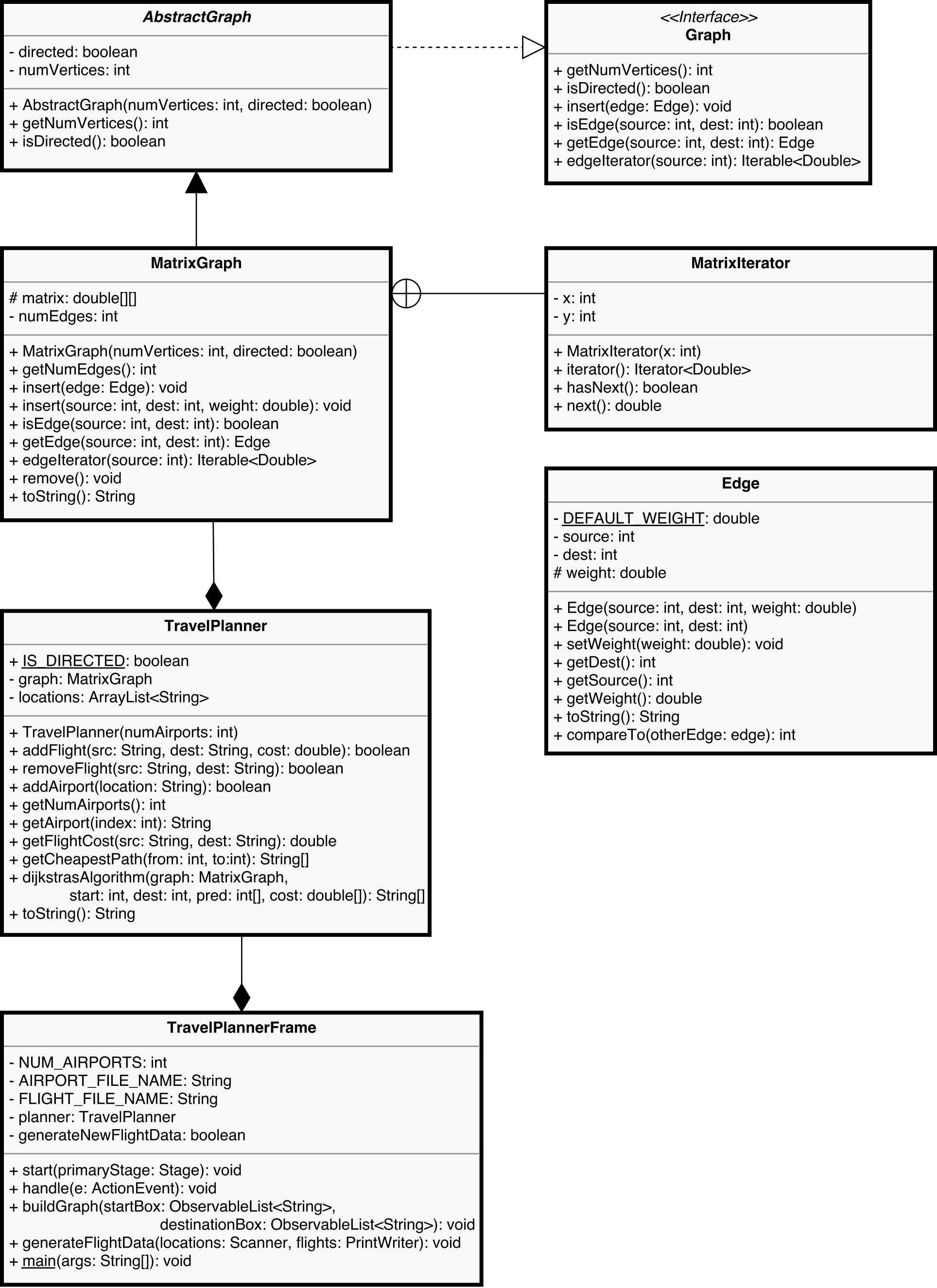
AbstractGraph
- directed: boolean
- numVertices: int
+ AbstractGraph(numVertices: int, directed: boolean)
+ getNumVertices(): int
+ isDirected(): boolean
<<Interface>>
Graph
+ getNumVertices(): int
+ isDirected(): boolean
+ insert(edge: Edge): void
+ isEdge(source: int, dest: int): boolean
+ getEdge(source: int, dest: int): Edge
+ edgeIterator(source: int): Iterable<Double>
MatrixGraph
matrix: double[][]
- numEdges: int
+ MatrixGraph(numVertices: int, directed: boolean)
+ getNumEdges(): int
+ insert(edge: Edge): void
+ insert(source: int, dest: int, weight: double): void
+ isEdge(source: int, dest: int): boolean
+ getEdge(source: int, dest: int): Edge
+ edgeIterator(source: int): Iterable<Double>
+ remove(): void
+ toString(): String
MatrixIterator
- x: int
- y: int
+ MatrixIterator(x: int)
+ iterator(): Iterator<Double>
+ hasNext(): boolean
+ next(): double
Edge
- DEFAULT_WEIGHT: double
- source: int
- dest: int
weight: double
+ Edge(source: int, dest: int, weight: double)
+ Edge(source: int, dest: int)
+ setWeight(weight: double): void
+ getDest(): int
+ getSource(): int
+ getWeight(): double
+ toString(): String
+ compareTo(otherEdge: edge): int
TravelPlanner
+ IS_DIRECTED: boolean
- graph: MatrixGraph
- locations: ArrayList<String>
+ TravelPlanner(numAirports: int)
+ addFlight(src: String, dest: String, cost: double): boolean
+ removeFlight(src: String, dest: String): boolean
+ addAirport(location: String): boolean
+ getNumAirports(): int
+ getAirport(index: int): String
+ getFlightCost(src: String, dest: String): double
+ getCheapestPath(from: int, to:int): String[]
+ dijkstrasAlgorithm(graph: MatrixGraph,
start: int, dest: int, pred: int[], cost: double[]): String[]
+ toString(): String
TravelPlannerFrame
- NUM_AIRPORTS: int
- AIRPORT_FILE_NAME: String
- FLIGHT_FILE_NAME: String
- planner: TravelPlanner
- generateNewFlightData: boolean
+ start(primaryStage: Stage): void
+ handle(e: ActionEvent): void
+ buildGraph(startBox: ObservableList<String>,
destinationBox: ObservableList<String>): void
+ generateFlightData(locations: Scanner, flights: PrintWriter): void
+ main(args: String[]): void