## graph

**<<Interface>>**
**Edge**
+setStartVertex(v : Vertex) : void
+setFinishVertex(v : Vertex) : void
+getStartVertex() : Vertex
+getFinishVertex() : Vertex

**EdgeNotFound**
-serialVersionUID : long = 1L
+EdgeNotFound(id1 : int, id2 : int)

**SameVertex**
-serialVersionUID : long = 1L
+SameVertex()

**VertexNotFound**
-serialVersionUID : long = 1L
+VertexNotFound(id : int)
+VertexNotFound(msg : String)

**Link**
-weight : int
-v1 : Vertex
-v2 : Vertex
+Link(v1 : Vertex, v2 : Vertex, weight : int)
+setStartVertex(v : Vertex) : void
+setFinishVertex(v : Vertex) : void
+getStartVertex() : Vertex
+getFinishVertex() : Vertex
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String

**<<Interface>>**
**Vertex**
+addEdge(v : Vertex, weight : int) : boolean
+removeEdge(v : Vertex) : boolean
+containsEdge(v : Vertex) : boolean
+findEdge(v : Vertex) : Edge
+degreeOf() : int
+degreeOfInward() : int
+degreeOfOutward() : int

**<<Interface>>**
**Graph**
+DEFAULT_EDGE_WEIGHT : int = 1
+findVertex(id : int) : Vertex
+addEdge(id1 : int, id2 : int) : boolean
+addEdge(id1 : int, id2 : int, weight : int) : boolean
+addVertex() : void
+findEdge(id1 : int, id2 : int) : Edge
+degreeOf(id : int) : int
+degreeOfInward(id : int) : int
+degreeOfOutward(id : int) : int
+getAllEdges(id : int) : Edge[]
+getInwardEdges(id : int) : Edge[]
+getOutwardEdges(id : int) : Edge[]
+getEdgeWeight(id1 : int, id2 : int) : int
+removeEdge(id1 : int, id2 : int) : boolean
+removeVertex(id : int) : boolean
+setEdgeWeight(id1 : int, id2 : int, weight : int) : boolean

**Node**
-id : int
-cnt : int = 1
-links : Link = new LinkedList<Link>()
+Node()
+addEdge(v : Vertex, weight : int) : boolean
+removeEdge(v : Vertex) : boolean
+containsEdge(v : Vertex) : boolean
+findEdge(v : Vertex) : Edge
+getAllEdges() : Edge[]
+getInwardEdges() : Edge[]
+getOutwardEdges() : Edge[]
+degreeOf() : int
+degreeOfInward() : int
+degreeOfOutward() : int
+hashCode() : int
+equals(obj : Object) : boolean
+toString() : String

**UndirectedGraph**
#n : int
#nodes : Node[]
+UndirectedGraph(n : int)
+getSize() : int
+findVertex(id : int) : Vertex
+addEdge(id1 : int, id2 : int) : boolean
+addEdge(id1 : int, id2 : int, weight : int) : boolean
+addVertex() : void
+findEdge(id1 : int, id2 : int) : Edge
+degreeOf(id : int) : int
+degreeOfInward(id : int) : int
+degreeOfOutward(id : int) : int
+getAllEdges(id : int) : Edge[]
+getInwardEdges(id : int) : Edge[]
+getOutwardEdges(id : int) : Edge[]
+getEdgeWeight(id1 : int, id2 : int) : int
+removeEdge(id1 : int, id2 : int) : boolean
+removeVertex(id : int) : boolean
+setEdgeWeight(id1 : int, id2 : int, weight : int) : boolean
+getAllVertices() : Vertex[]
+toString() : String

## xml

**UserHandler**
-bWeight : boolean = false
-bMove : boolean = false
+startElement(uri : String, localName : String, qName : String, attributes : Attributes) : void
+characters(ch : char[], start : int, length : int) : void

**InitialValues**
-finalinst : float
-antcolsize : int
-plevel : float
-nbnodes : int
-nestnode : int
-graphInfo : ArrayList<int[]> = new ArrayList<int[]>()
-actualNode : int
-alpha : float
-beta : float
-delta : float
-eta : float
-rho : float
~setFinalinst(finalinst : float) : void
~setAntcolsize(antcolsize : int) : void
~setPlevel(plevel : float) : void
~setNbnodes(nbnodes : int) : void
~setNestnode(nestnode : int) : void
~setAlpha(alpha : float) : void
~setBeta(beta : float) : void
~setDelta(delta : float) : void
~setEta(eta : float) : void
~setRho(rho : float) : void
~addNode(nodeidx : int) : void
~addRelation(targetnode : int) : void
~addWeight(weight : int) : void

**Xml**
+validateFile(filename : String) : boolean
-validate(xmlFile : String) : void
+readValues(filename : String) : boolean

## tsp

**EvaporationEvent**
-eevents : int = 0
-node1 : int
-node2 : int
-maze : TSPGraph
~EvaporationEvent(currTime : double, node1 : int, node2 : int, maze : TSPGraph)
~EvaporationEvent(oldEvent : EvaporationEvent)
+doEvent() : LinkedList<Event>
+getEventNo() : int
+getNodeStart() : int
+getNodeFinish() : int

**TSPGraph**
-rho : double
-eta : double
-pheromones : double[][]
-W : int
-bestCycle : int[]
-bestCycleCost : int = Integer.MAX_VALUE
+TSPGraph(n : int, info : List<int[]>, rho : double, eta : double)
-TSPGraphBuilder(info : List<int[]>) : void
-containsEdge(id1 : int, id2 : int) : boolean
-computeW() : void
-checkEdgeEvaporation(id1 : int, id2 : int) : boolean
+layPheromones(path : int[], amount : double) : void
+getPathCost(path : int[]) : int
+getEdgePheromones(id1 : int, id2 : int) : double
+decrementEdgePheromones(id1 : int, id2 : int) : boolean
+getBestCost() : int
+setBestCycle(path : int[], cost : int) : void
+bestPathString() : String

**ObservationEvent**
-observation : int = 1
-tau : double
~ObservationEvent(tau : double)
+doEvent() : LinkedList<Event>

**Event**
-eventTime : double
-random : Random = new Random()
+doEvent() : LinkedList<Event>
+expRandom(m : double) : double

**Main**
+main(args : String[]) : void

**AntEvent**
-mevents : int = 0
-delta : double
-nextNode : int
+ant : Ant
~AntEvent(currTime : double, delta : double, ant : Ant)
~AntEvent(oldEvent : AntEvent)
+getEventNo() : int
+doEvent() : LinkedList<Event>

**Ant**
-generator : Random = new Random()
-nestNode : int
-alpha : double
-beta : double
-gamma : double
-currNode : int
-nodeCount : int = 1
-visited : int[]
-maze : TSPGraph
~Ant(nest : int, alpha : double, beta : double, gamma : double, maze : TSPGraph)
+isVisited(node : int) : boolean
+getNextNode() : int
+getNextNodeWeight(nextNode : int) : double
+doPath() : int[]
+isHamiltonian() : boolean
-cleanVisited() : void
+move(nextNode : int) : void
+reset() : void
+getNest() : int
+getNode() : int
+toString() : String