phases

LayerPhase
+apply(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)

CycleBreakingLayerPhase
+apply(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)
+basic(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)
+advanced(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)
+hasCycle(layoutGraph : ElkNode)
+reverse(e : ElkEdge)

LayerAssignmentLayerPhase
+apply(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)
+basic(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)
+topoSort(layoutGraph : ElkNode, src : ElkNode)
+addDummyNodes(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)

CrossingMinimizationLayerPhase
+apply(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)
+basic(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)

NodePlacementLayerPhase
+apply(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)

EdgeRoutingLayerPhase
+apply(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)

LayeredLayoutingLayoutProvider
-phases : LayerPhase[]
+layout(layoutGraph : ElkNode, monitor : IElkPrograssMonitor)

helper

Graph

Help

Edge

Node

LongEdge

EdgeRoutingProvider

Vector2

properties

GraphProperty
-layers : List<List<ElkNode>>
-longEdges : List<LongEdge>

NodeProperty
-isDummy : boolean
-layer : int
-visiting : boolean
-visited : boolean
-barycenterVal : double

EdgeProperty
-isReversed : boolean
-isDummy : boolean
-visited : boolean
-parent : LongEdge

Firstly I added a LayerPhase Interface as suggested by the Task. You can see it at the top of the diagram. It's five subclasses represent the five Layer Phases. Those are loaded into the phases Array in the LayeredLayoutingLayoutProvider which is designed to handle any n phases of a layout algorithm, so I can reuse it if we need to implement a layered algorithm again.
The reason to separate the phases into classes is to divide them and their sub methods into different text files to make navigating through the code easier in Eclipse.

I also included the helper package which contains the Graph classes which are used in the cycle breaking algorithm, the EdgeRoutingProvider which implements simple straight line edge routing, the Vector2 class from the last assignment, a LongEdge Class to group together dummy edges and nodes to put the pack together in the Edge Routing Phase and the Help class which allows me to access and lazily load the Property instances of ElkNodes and ElkEdges.

The property classes contain useful attributes for the Elk-Graph, Nodes and Edges.