Graph Implementation

Class Diagram

Graph `[V, E]`

```
+ Graph()
+ clear()
+ clearEdges()
+ vertexCount(): int
+ edgeCount(): int
+ addVertex(v: Vertex<V>)
+ addVertices(vertices: ArrayList<Vertex<V>>)
+ containsVertex(v: Vertex<V>): boolean
+ addEdge(e: Edge<V, E>)
+ addEdges(edges: ArrayList<Edge<V, E>>)
+ containsEdge(e: Edge<V, E>): boolean
+ removeVertex(v: Vertex<V>): ArrayList<Edge<V, E>>
+ removeEdge(e: Edge<V, E>)
+ adjacentCount(v: Vertex<V>): int
+ vertices(): ArrayList<Vertex<V>>
+ adjacent(v: Vertex<V>): ArrayList<Vertex<V>>
+ edges(): ArrayList<Edge<V, E>>
+ minSpanningTree(v: Vertex<V>): Tree<V, E>
+ totalVertexWeight(): int
+ totalEdgeWeight(): int
+ hasCycle(): boolean
+ maxVertexWeight(): int
+ minVertexWeight(): int
```

DuplicateVertexException

DuplicateEdgeException

NoSuchVertexException

NoSuchEdgeException

Vertex `[V]`

```
+ Vertex(label: V)
+ Vertex(label: V, weight: int)
+ getLabel(): V
+ getWeight(): int
+ setLabel(label: V)
+ setWeight(weight: int)
+ equals(other: Vertex<V>): boolean
+ toString(): String
```

Tree `[V, E]`

```
+ BinaryTree()
+ getParent(): Tree<V, E>
+ getVertex(): Vertex<V>
+ isChild(): boolean
+ isParent(): boolean
+ isRoot(): boolean
+ isLeaf(): boolean
+ isDegenerate(): boolean
+ height(): int
+ depth(): int
+ isSuperTreeOf(t: Tree<V, E>) boolean
+ isSubtreeOf(t: Tree<V, E>) boolean
+ isSiblingOf(): boolean
+ isChildOf(): boolean
+ isParentOf(): boolean
+ makeNewRoot(t: Tree<V, E>)
+ iterator(): iterator<V>
```

Edge `[V, E]`

```
Edge(v1: Vertex<V>, v2: Vertex<V>, label: E)
Edge(v1: Vertex<V>, v2: Vertex<V>, lable: E,
weight: int)
getVertex1(): Vertex<V>
getVertex2(): Vertex<V>
getLabel(): E
getWeight(): int
setVertex1(v: Vertex<V>)
setVertex2(v: Vertex<V>)
setLabel(label: E)
setWeight(weight: int)
equals(other: Edge<V, E>): boolean
toString(): String
```

« interface » Iterable `[V]`