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>>                         │
│ + iterator(): Iterator<Vertex<V>>                        │
│ + minSpanningTree(v: Vertex<V>): Tree<V, E>              │
│ + totalVertexWeight(): double                            │
│ + totalEdgeWeight(): double                              │
│ + canReach(v1: Vertex<V>, v2: Vertex<V>): boolean        │
│ + hasCycle(): boolean                                    │
└────────────────────────────────────────────────────────┘
```

```
                                ┌──────────────────────┐ ┌─┐
                                │    « interface »     │ │V│
                                │      Iterable        │ └─┘
                                └──────────────────────┘
                                ┌──────────────────────────────┐
                                │  DuplicateVertexException     │
                                └──────────────────────────────┘
                                ┌──────────────────────────────┐
                                │  DuplicateEdgeException        │
                                └──────────────────────────────┘
                                ┌──────────────────────────────┐
                                │  NoSuchVertexException         │
                                └──────────────────────────────┘
                                ┌──────────────────────────────┐
                                │  NoSuchEdgeException           │
                                └──────────────────────────────┘
```

```
                                ┌──────────────────────┐ ┌─┐
                                │    « interface »     │ │V│
                                │      Comparable       │ └─┘
                                └──────────────────────┘
```

```
┌────────────────────────────────────────┐ ┌─┐
│               Vertex                    │ │V│
├────────────────────────────────────────┤ └─┘
│ + Vertex(label: V)                      │
│ + Vertex(label: V, weight: double)      │
│ + getLabel(): V                         │
│ + getWeight(): double                   │
│ + setLabel(label: V)                    │
│ + setWeight(weight: double)             │
│ + compareTo(other: Vertex<V>): int      │
│ + toString(): String                    │
└────────────────────────────────────────┘
```

```
┌─────────────────────────────────────────────┐ ┌─ ─ ─┐
│                   Tree                        │ │ V, E│
├─────────────────────────────────────────────┤ └─ ─ ─┘
│ + Tree()                                      │
│ + getParent(): Tree<V, E>                     │
│ + getVertex(): Vertex<V>                      │
│ + isChild(): boolean                          │
│ + isParent(): boolean                         │
│ + isRoot(): boolean                           │
│ + isLeaf(): boolean                           │
│ + isDegenerate(): boolean                     │
│ + height(): int                               │
│ + depth(): int                                │
│ + isSiblingOf(t: Treev<V, E>): boolean        │
│ + isChildOf(t: Tree<V, E>): boolean           │
│ + isParentOf(t: Tree<V, E>): boolean          │
│ + makeNewRoot(t: Tree<V, E>)                  │
└─────────────────────────────────────────────┘
```

```
┌─────────────────────────────────────────────────┐ ┌─ ─ ─┐
│                      Edge                         │ │ V, E│
├─────────────────────────────────────────────────┤ └─ ─ ─┘
│ Edge(v1: Vertex<V>, v2: Vertex<V>, label: E)      │
│ Edge(v1: Vertex<V>, v2: Vertex<V>, lable: E,      │
│ weight: double)                                   │
│ getVertex1(): Vertex<V>                           │
│ getVertex2(): Vertex<V>                           │
│ getLabel(): E                                     │
│ getWeight(): double                               │
│ setVertex1(v: Vertex<V>)                          │
│ setVertex2(v: Vertex<V>)                          │
│ setLabel(label: E)                                │
│ setWeight(weight: double)                         │
│ equals(other: Edge<V, E>): boolean                │
│ toString(): String                                │
└─────────────────────────────────────────────────┘
```