RESOLUÇÃO (DIAGRAMA DE CLASSES):

public class App Dijkstra dijkstra = new Dijkstra(); Vertex v1 = new Vertex("A"); v1.addNeighbour(new Edge(2, v1, v2)); vertexList.add(new Vertex("A")); edgeList.add(new Edge(1, vertexList.get(1), vertexList.get(2))); BellmanFord algorithm = new BellmanFord(edgeList, vertexList); public class Dijkstra public class BellmanFord computePath(Vertex sourceVertex) getShortestPathTo(Vertex private List<Vertex> vertexList; targetVerte) private List<Edge> edgeList; BellmanFord(List<Edge> edgeList, *List<Vertex>* vertexList) shotestPath(Vertex sourceVertex, public class Edge Vertex targetVertex) hasCycle(Edge edge) private double weight; private Vertex startVertex; private Vertex targetVertex; public class Vertex public Edge(double weight, Vertex startVertex, Vertex targetVertex) private String name; private boolean visited; private List<Edge> edges; private double minDistance = Double.MAX VALUE; private Vertex previousVertex; addNeighbour(Edge edge)