TAD - Data structures

TAD - Graph
Graph = List<Vertices> vertices, List<Edge> edges
{inv: vertices != Ø & edges !=Ø}
Principal operations: ● AddVertice (Modifier): Value x List<Edge> -----> Graph ● AddEdge(Modifier): Edge -----> Graph ● DeleteVertice(Modifier): Value -----> Graph ● DeleteEdge(Modifier): Vertice x Vertice -----> Graph ● CreateGraph(Constructor): List<Edge> x List<Node> -----> Graph
AddVertice (Value value, List<Edge> edges) “Add a vertice to the graph” {pre: A value and a list of edges} {post: Modified graph}
AddEdge (Edge edge) “Add an edge to the graph” {pre: The edge might relate existent vertices} {post: Modified edges list}
DeleteVertice(Value) “Delete a vertex from the graph” {pre: the value that will be deleted must belong to the graph} {post: Modified vertices list }
DeleteEdge (Vertice entry, Vertice exit) “Delete an edge from the graph” {pre: Two vertices} {post: Modified tree}
CreateGraph (List<Vertices> vertices, List <Edges> edges) “Create a new graph” {pre: The list of vertices might be not empty} {post: A new graph}