gui.canvas.AutomataCanvas + LINE_WIDTH:float = 2.0f + EDGE_WIDTH:float = 8.0f + ACCEPTING_DIST:float = 10.0f - scale_x:double = 100.0f - scale_y:double = 100.0f - state_size:double = 50.0f - colourBlindMode:boolean = false $\hbox{-} G: Graphics Context \\$ - isEmpty:boolean = true isLoading:boolean = true - graph:VisualRepresentation + draw(String text):void + draw(double width, double height, int nodeCount):void drawAutomata():void - setupScalling(doble width, double height, int nodeCount):void - resetCanvas():void gui.canvas.VisualRepresentation - edges:ArrayList<VisualEdge> - states:ArrayList<VIsualState> - title:String - text:Stirng - progressText:String[4] - labelSetList:ArrayList<String> + VisualRepresentation(Automaton a, String text, String title) + VisualRepresentation(ArrayList<VisualState> states, ArrayList<VisualEdge> edges, String text, String title) + copy():VisualRepresentation + copy(String newText):VisualRepresentation 0..* 1..* gui.canvas.VisualEdge gui.canvas.VisualState - accepting:boolean source:int - highlight:StateHighlights destination:int - id:int - input:String - x:int - highlight: Edge Highlights - y:int - label:String - highlight:StateHighlights + VisualEdge(int source, int dest, int input) + VisualEdge(int source, int dest, int input, EdgeHighlights hl) + VisualState(int id, boolean isAcc, int x, int y, String label) + draw(AutomataCanvas canvas, int x1, int y1, int x2, int y2, $+ \ Visual State (int \ id, \ boolean \ is Acc, \ int \ x, \ int \ y, \ String \ label, \ State Highlights \ hl)$ VisualEdgeType type):void + draw(AutomataCanvas canvas):void - drawArrow(AutomataCanvas canvas, int x1, int y1, int x2, int y2):void - drawCircle(AutomataCanvas canvas):void - drawText(AutomataCanvas canvas, int x, int y):void - drawDiamond(AutomataCanvas canvas):void stateHighlightFill():Color -stateHighlightStroke():Color <<enumeration>> <<enumeration>> <<enumeration>> gui.canvas.VisualEdgeType gui.canvas.EdgeHighlights gui.canvas.StateHighlights **NORMAL** Straight NORMAL RED **RED** Recursive CurvedUpForward YELLOW YELLOW CurvedDownBack **GREEN GREEN**