Caretaker

- addMemento(m:Memento) : void
- getMemento(i:int) : Memento
- undoMemento(canvas:Stack<string>) : void
- redoMemento(canvas:Stack<string>) : void

redolist<string>   canvas<Memento>

Stack`1   List`1

history<string>

Memento

- Memento(shapesClass:ShapesClass)
- getDetails() : string

Originator

- createMemento() : Memento
- getDetails() : string

Program

- Main(args:string[]) : void
- printHome() : void
- printHelp() : void
- printCommands() : void
- printCanvas(canv:List<String>) : string
- PrintStack(s:Stack<string>) : string

Shapes

Memento

- Memento(shapesClass:ShapesClass)
- getDetails() : string

shapesClass

ShapesClass

Originator

- createMemento() : Memento
- getDetails() : string

shapesClass

ShapesClass

Program

- Main(args:string[]) : void
  printHome() : void
  printHelp() : void
  printCommands() : void
- printCanvas(canv:List<String>) : string
- PrintStack(s:Stack<string>) : string

C Shape

- circle(x:int, y:int, r:int, stroke:string, strokewidth:int, fill:string) : void
- rectangle(x:int, y:int, w:int, h:int, stroke:string, strokewidth:int, fill:string) : void
- ellipse(x:int, y:int, rx:int, ry:int, stroke:string, strokewidth:int, fill:string) : void
- line(x:int, y:int, x2:int, y2:int, stroke:string, strokewidth:int, fill:string) : void
- polyline(p1:int, p2:int, p3:int, p4:int, p5:int, p6:int, p7:int, p8:int, stroke:string, strokewidth:int, fill:string) : void
- polygon(p1:int, p2:int, p3:int, p4:int, p5:int, p6:int, p7:int, p8:int, stroke:string, strokewidth:int, fill:string) : void
- circlet(x:int, y:int, r:int, stroke:string, strokewidth:int, fill:string, translate:string) : void
- rectanglet(x:int, y:int, w:int, h:int, stroke:string, strokewidth:int, fill:string, translate:string) : void
- ellipset(x:int, y:int, rx:int, ry:int, stroke:string, strokewidth:int, fill:string, translate:string) : void
- linet(x:int, y:int, x2:int, y2:int, stroke:string, strokewidth:int, fill:string, translate:string) : void
- polylinet(p1:int, p2:int, p3:int, p4:int, p5:int, p6:int, p7:int, p8:int, stroke:string, strokewidth:int, fill:string, translate:string) : void
- polygont(p1:int, p2:int, p3:int, p4:int, p5:int, p6:int, p7:int, p8:int, stroke:string, strokewidth:int, fill:string, translate:string) : void
- circles(x:int, y:int, r:int, stroke:string, strokewidth:int, fill:string, scale:string) : void
- rectangles(x:int, y:int, w:int, h:int, stroke:string, strokewidth:int, fill:string, scale:string) : void
- ellipses(x:int, y:int, rx:int, ry:int, stroke:string, strokewidth:int, fill:string, scale:string) : void
- lines(x:int, y:int, x2:int, y2:int, stroke:string, strokewidth:int, fill:string, scale:string) : void
- polylines(p1:int, p2:int, p3:int, p4:int, p5:int, p6:int, p7:int, p8:int, stroke:string, strokewidth:int, fill:string, scale:string) : void
- polygons(p1:int, p2:int, p3:int, p4:int, p5:int, p6:int, p7:int, p8:int, stroke:string, strokewidth:int, fill:string, scale:string) : void
- circler(x:int, y:int, r:int, stroke:string, strokewidth:int, fill:string, rotate:string) : void
- rectangler(x:int, y:int, w:int, h:int, stroke:string, strokewidth:int, fill:string, rotate:string) : void
- ellipser(x:int, y:int, rx:int, ry:int, stroke:string, strokewidth:int, fill:string, rotate:string) : void
- liner(x:int, y:int, x2:int, y2:int, stroke:string, strokewidth:int, fill:string, rotate:string) : void
- polyliner(p1:int, p2:int, p3:int, p4:int, p5:int, p6:int, p7:int, p8:int, stroke:string, strokewidth:int, fill:string, rotate:string) : void
- polygonr(p1:int, p2:int, p3:int, p4:int, p5:int, p6:int, p7:int, p8:int, stroke:string, strokewidth:int, fill:string, rotate:string) : void
- circlesk(x:int, y:int, r:int, stroke:string, strokewidth:int, fill:string, skew:string) : void
- rectanglesk(x:int, y:int, w:int, h:int, stroke:string, strokewidth:int, fill:string, skew:string) : void
- ellipsesk(x:int, y:int, rx:int, ry:int, stroke:string, strokewidth:int, fill:string, skew:string) : void
- linesk(x:int, y:int, x2:int, y2:int, stroke:string, strokewidth:int, fill:string, skew:string) : void
- polylinesk(p1:int, p2:int, p3:int, p4:int, p5:int, p6:int, p7:int, p8:int, stroke:string, strokewidth:int, fill:string, skew:string) : void
- polygonsk(p1:int, p2:int, p3:int, p4:int, p5:int, p6:int, p7:int, p8:int, stroke:string, strokewidth:int, fill:string, skew:string) : void
- path(p1:int, p2:int) : void
- style(stroke:string, strokewidth:int, fill:string) : void
- settranslate(a:string) : void
- setstroke(a:string) : void
- setstrokewidth(a:int) : void
- setfill(a:string) : void
- setr(a:int) : void
- setx(a:int) : void
- sety(a:int) : void
- setw(a:int) : void
- seth(a:int) : void
- setrx(a:int) : void
- setry(a:int) : void
- setx2(a:int) : void
- sety2(a:int) : void
- setp1(a:int) : void
- setp2(a:int) : void
- setp3(a:int) : void
- setp4(a:int) : void
- setp5(a:int) : void
- setp6(a:int) : void
- setp7(a:int) : void
- setp8(a:int) : void
- getstroke() : string
- getstrokewidth() : int
- getfill() : string
- getr() : int
- getx() : int
- gety() : int
- getw() : int
- geth() : int
- getrx() : int
- getry() : int
- getx2() : int
- gety2() : int
- getp1() : int
- getp2() : int
- getp3() : int
- getp4() : int
- getp5() : int
- getp6() : int
- getp7() : int
- getp8() : int
- gettranslate() : string
- getCircle() : string
- getRectangle() : string
- getEllipse() : string
- getLine() : string
- getPolyline() : string
- getPolygon() : string
- getPath() : string
- getCircleT() : string
- getRectangleT() : string
- getEllipseT() : string
- getLineT() : string
- getPolylineT() : string
- getPolygonT() : string
- getPathT() : string
- getCircleS() : string
- getRectangleS() : string
- getEllipseS() : string
- getLineS() : string
- getPolylineS() : string
- getPolygonS() : string
- getPathS() : string
- getCircleR() : string
- getRectangleR() : string
- getEllipseR() : string
- getLineR() : string
- getPolylineR() : string
- getPolygonR() : string
- getCircleSK() : string
- getRectangleSK() : string
- getEllipseSK() : string
- getLineSK() : string
- getPolylineSK() : string
- getPolygonSK() : string
- getStyle() : string

ShapesClass

temp_shape : string «get» «set»

ShapesClass(temp_shape:string)
getDetails() : string