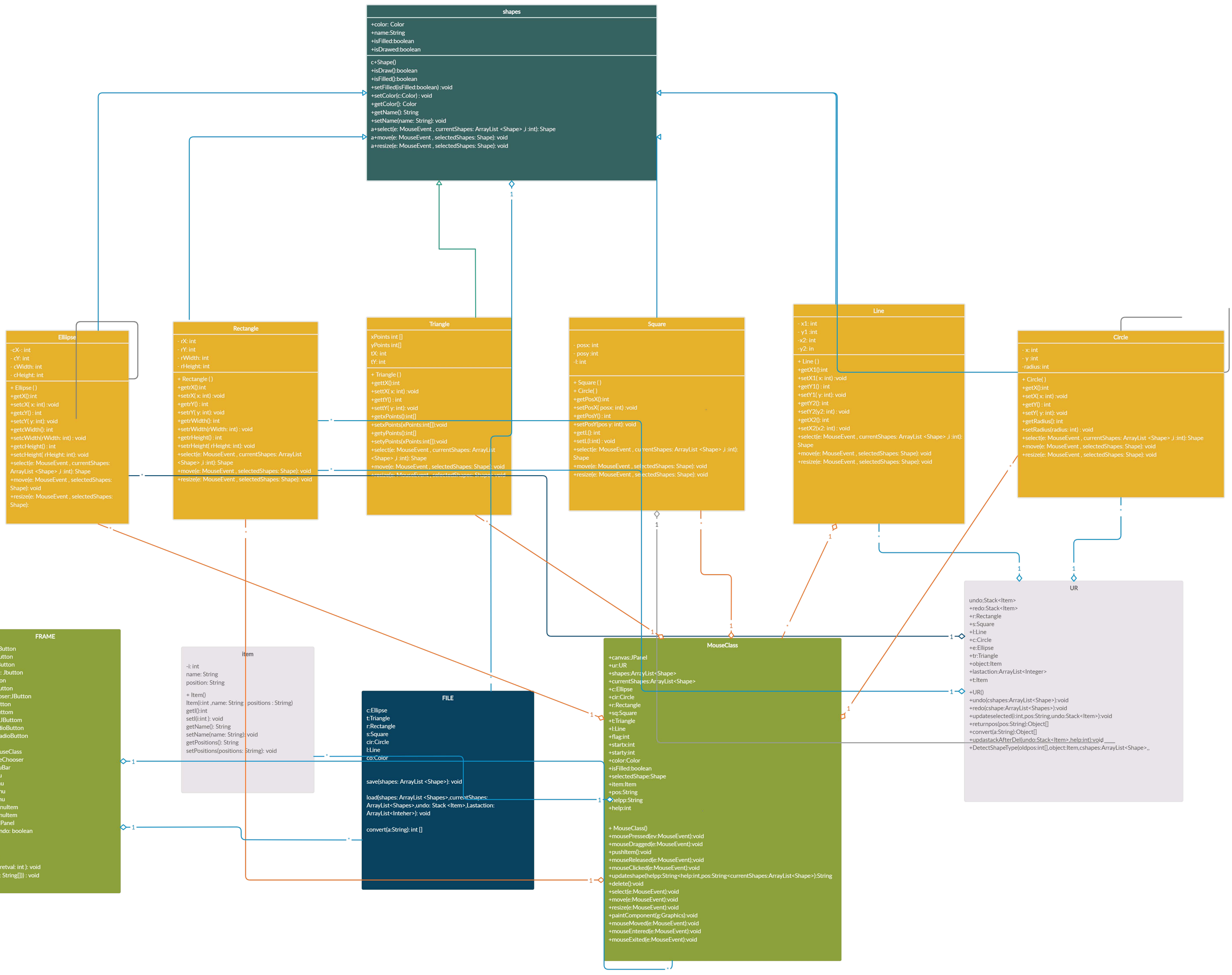
shapes
+color: Color
+name:String
+isFilled:boolean
+isDrawed:boolean
c+Shape()
+isDraw():boolean
+isFilled():boolean
+setFilled(isFilled:boolean) :void
+setColor(c:Color) : void
+getColor(): Color
+getName(): String
+setName(name: String): void
a+select(e: MouseEvent , currentShapes: ArrayList <Shape> ,i :int): Shape
a+move(e: MouseEvent , selectedShapes: Shape): void
a+resize(e: MouseEvent , selectedShapes: Shape): void
Ellipse
-cX-: int
- cY: int
- cWidth: int
- cHeight: int
+ Ellipse ( )
+getX():int
+setcX(x: int) :void
+getcY() : int
+setcY(y: int): void
+getcWidth(): int
+setcWidth(rWidth: int) : void
+getcHeight() : int
+setcHeight(rHeight: int): void
+select(e: MouseEvent , currentShapes: ArrayList <Shape> ,i :int): Shape
+move(e: MouseEvent , selectedShapes: Shape): void
+resize(e: MouseEvent , selectedShapes: Shape):
Rectangle
- rX: int
- rY: int
- rWidth: int
- rHeight: int
+ Rectangle ( )
+getrX():int
+setrX(x: int) :void
+getrY() : int
+setrY(y: int): void
+getrWidth(): int
+setrWidth(rWidth: int) : void
+getrHeight() : int
+setrHeight(rHeight: int): void
+select(e: MouseEvent , currentShapes: ArrayList <Shape> ,i :int): Shape
+move(e: MouseEvent , selectedShapes: Shape): void
+resize(e: MouseEvent , selectedShapes: Shape): void
Triangle
xPoints int []
yPoints int[]
tX: int
tY: int
+ Triangle ( )
+gettX():int
+settX(x: int) :void
+gettY() : int
+settY(y: int): void
+getxPoints():int[]
+setxPoints(xPoints:int[]):void
+getyPoints():int[]
+setyPoints(xPoints:int[]):void
+select(e: MouseEvent , currentShapes: ArrayList <Shape> ,i :int): Shape
+move(e: MouseEvent , selectedShapes: Shape): void
Square
- posx: int
- posy :int
-l: int
+ Square ( )
+ Circle( )
+getPosX():int
+setPosX(posx: int) :void
+getPosY() : int
+setPosY(pos y: int): void
+getL(): int
+setL(l:int) : void
+select(e: MouseEvent , currentShapes: ArrayList <Shape> ,i :int): Shape
+move(e: MouseEvent , selectedShapes: Shape): void
+resize(e: MouseEvent , selectedShapes: Shape): void
Line
- x1: int
- y1 :int
-x2: int
-y2: in
+ Line ( )
+getX1():int
+setX1(x: int) :void
+getY1() : int
+setY1(y: int): void
+getY2(): int
+setY2(y2: int) : void
+getX2(): int
+setX2(x2: int) : void
+select(e: MouseEvent , currentShapes: ArrayList <Shape> ,i :int): Shape
+move(e: MouseEvent , selectedShapes: Shape): void
+resize(e: MouseEvent , selectedShapes: Shape): void
Circle
- x: int
- y :int
-radius: int
+ Circle( )
+getX():int
+setX(x: int) :void
+getY() : int
+setY(y: int): void
+getRadius(): int
+setRadius(radius: int) : void
+select(e: MouseEvent , currentShapes: ArrayList <Shape> ,i :int): Shape
+move(e: MouseEvent , selectedShapes: Shape): void
+resize(e: MouseEvent , selectedShapes: Shape): void
FRAME
Square: JButton
Circle: JButton
Ellipse: JButton
Rectangle: Jbutton
line:JButton
delete:JButton
colorChooser:JButton
move:JButton
resize:JButtom
changeC: JButtom
filled:JRadioButton
border: JradioButton
url:String
paint: MouseClass
SLfile:JFileChooser
bar:JmenuBar
file:JMenu
edit:JMenu
save:JMenu
load:JMenu
undo:Jmenultem
redo:JMenultem
Bshapes:JPanel
PressedUndo: boolean
+ frame()
OpenFile(retval: int): void
main(args: String[]) : void
item
-i: int
name: String
position: String
+ Item()
Item(i:int ,name: String positions : Strimg)
getI():int
setI(i:int): void
getName(): String
setName(name: String): void
getPositions(): String
setPositions(positions: Strimg): void
FILE
c:Ellipse
t:Triangle
r:Rectangle
s:Square
cir:Circle
l:Line
co:Color
save(shapes: ArrayList <Shape>): void
load(shapes: ArrayList <Shapes>,currentShapes: ArrayList<Shapes>,undo: Stack <Item>,Lastaction: ArrayList<Inteher>): void
convert(a:String): int []
MouseClass
+canvas:JPanel
+ur:UR
+shapes:ArrayList<Shape>
+currentShapes:ArrayList<Shape>
+c:Ellipse
+cir:Circle
+r:Rectangle
+sq:Square
+t:Triangle
+l:Line
+flag:int
+startx:int
+starty:int
+color:Color
+isFilled:boolean
+selectedShape:Shape
+item:Item
+pos:String
+helpp:String
+help:int
+ MouseClass()
+mousePressed(ev:MouseEvent):void
+mouseDragged(e:MouseEvent):void
+pushItem():void
+mouseReleased(e:MouseEvent):void
+mouseClicked(e:MouseEvent):void
+updateshape(helpp:String<help:int,pos:String<currentShapes:ArrayList<Shape>):String
+delete():void
+select(e:MouseEvent):void
+move(e:MouseEvent):void
+resize(e:MouseEvent):void
+paintComponent(g:Graphics):void
+mouseMoved(e:MouseEvent):void
+mouseEntered(e:MouseEvent):void
+mouseExited(e:MouseEvent):void
UR
undo:Stack<Item>
+redo:Stack<Item>
+r:Rectangle
+s:Square
+l:Line
+c:Circle
+e:Ellipse
+tr:Triangle
+object:Item
+lastaction:ArrayList<Integer>
+t:Item
+UR()
+undo(cshapes:ArrayList<Shape>):void
+redo(cshape:ArrayList<Shapes>):void
+updateselected(i:int,pos:String,undo:Stack<Item>):void
+returnpos(pos:String):Object[]
+convert(a:String):Object[]
+updastackAfterDel(undo:Stack<Item>,help:int):void
+DetectShapeType(oldpos:int[],object:Item,cshapes:ArrayList<Shape>,,