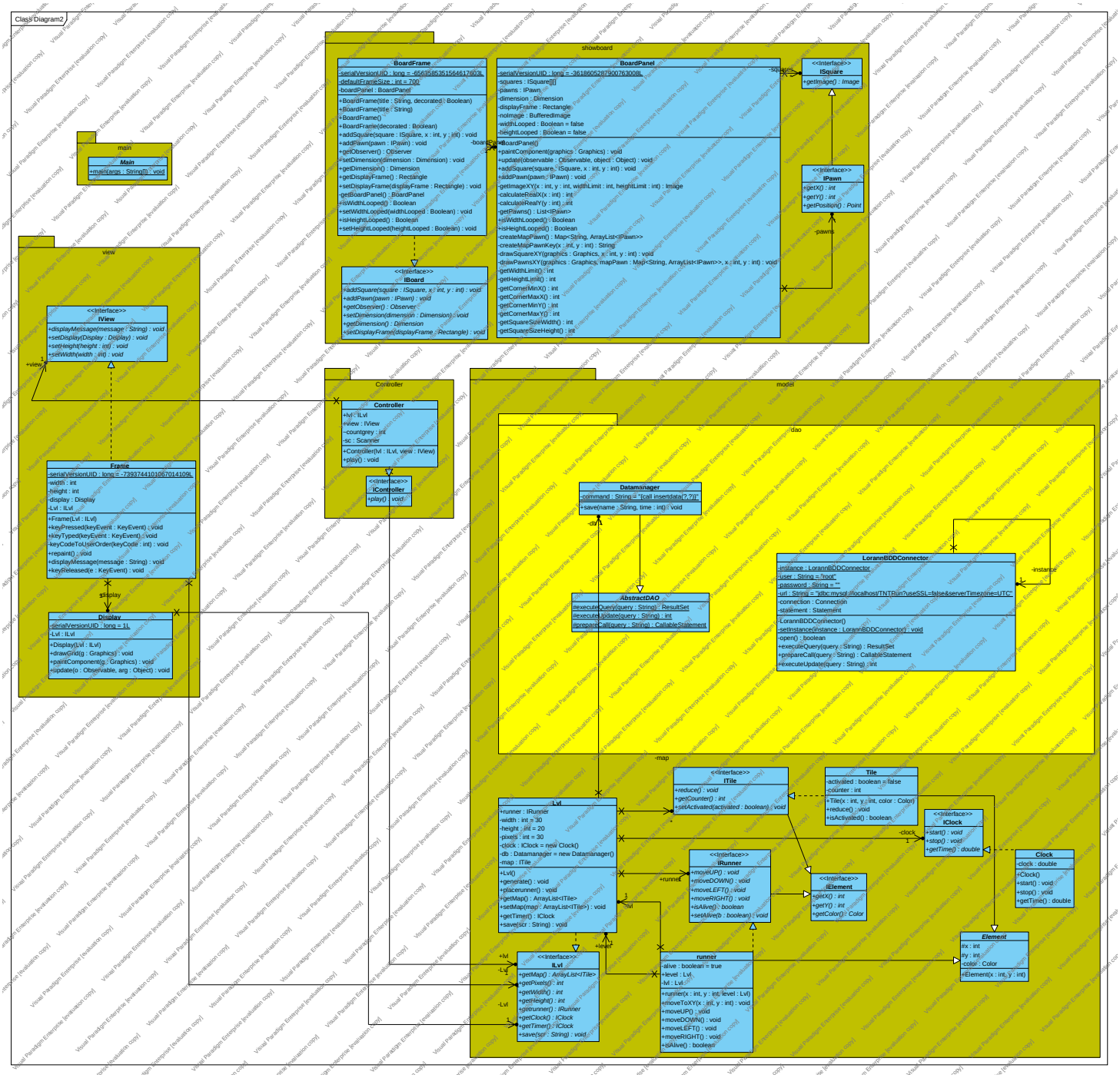
Class Diagram2
showboard
BoardFrame
-serialVersionUID : long = -6563585351564617603L
-defaultFrameSize : int = 700
-boardPanel : BoardPanel
+BoardFrame(title : String, decorated : Boolean)
+BoardFrame(title : String)
+BoardFrame()
+BoardFrame(decorated : Boolean)
+addSquare(square : ISquare, x : int, y : int) : void
+addPawn(pawn : IPawn) : void
+getObserver() : Observer
+setDimension(dimension : Dimension) : void
+getDimension() : Dimension
+getDisplayFrame() : Rectangle
+setDisplayFrame(displayFrame : Rectangle) : void
-getBoardPanel() : BoardPanel
+isWidthLooped() : Boolean
+setWidthLooped(widthLooped : Boolean) : void
+isHeightLooped() : Boolean
+setHeightLooped(heightLooped : Boolean) : void
-boardPanel
BoardPanel
-serialVersionUID : long = -3618605287900763008L
-squares : ISquare[][]
-pawns : IPawn
-dimension : Dimension
-displayFrame : Rectangle
-noImage : BufferedImage
-widthLooped : Boolean = false
-heightLooped : Boolean = false
+BoardPanel()
+paintComponent(graphics : Graphics) : void
+update(observable : Observable, object : Object) : void
+addSquare(square : ISquare, x : int, y : int) : void
+addPawn(pawn : IPawn) : void
-getImageXY(x : int, y : int, widthLimit : int, heightLimit : int) : Image
-calculateRealX(x : int) : int
-calculateRealY(y : int) : int
-getPawns() : List<IPawn>
+isWidthLooped() : Boolean
+isHeightLooped() : Boolean
-createMapPawn() : Map<String, ArrayList<IPawn>>
-createMapPawnKey(x : int, y : int) : String
-drawSquareXY(graphics : Graphics, x : int, y : int) : void
-drawPawnsXY(graphics : Graphics, mapPawn : Map<String, ArrayList<IPawn>>, x : int, y : int) : void
-getWidthLimit() : int
-getHeightLimit() : int
-getCornerMinX() : int
-getCornerMaxX() : int
-getCornerMinY() : int
-getCornerMaxY() : int
-getSquareSizeWidth() : int
-getSquareSizeHeight() : int
-squares
<<Interface>>
ISquare
+getImage() : Image
<<Interface>>
IPawn
+getX() : int
+getY() : int
+getPosition() : Point
-pawns
<<Interface>>
IBoard
+addSquare(square : ISquare, x : int, y : int) : void
+addPawn(pawn : IPawn) : void
+getObserver() : Observer
+setDimension(dimension : Dimension) : void
+getDimension() : Dimension
+setDisplayFrame(displayFrame : Rectangle) : void
main
Main
+main(args : String[]) : void
view
<<Interface>>
IView
+displayMessage(message : String) : void
+setDisplay(Display : Display) : void
+setHeight(height : int) : void
+setWidth(width : int) : void
+view
Frame
-serialVersionUID : long = -7393744101067014109L
-width : int
-height : int
-display : Display
-LvI : ILvI
+Frame(LvI : ILvI)
+keyPressed(keyEvent : KeyEvent) : void
+keyTyped(keyEvent : KeyEvent) : void
-keyCodeToUserOrder(keyCode : int) : void
+repaint() : void
+displayMessage(message : String) : void
+keyReleased(e : KeyEvent) : void
+display
Display
-serialVersionUID : long = 1L
-LvI : ILvI
+Display(LvI : ILvI)
+drawGrid(g : Graphics) : void
+paintComponent(g : Graphics) : void
+update(o : Observable, arg : Object) : void
Controller
Controller
+lvl : ILvI
+view : IView
-countgrey : int
-sc : Scanner
+Controller(lvl : ILvI, view : IView)
+play() : void
<<Interface>>
IController
+play() : void
model
dao
Datamanager
-command : String = "{call insertdata(?,?)}"
+save(name : String, time : int) : void
-dm
AbstractDAO
#executeQuery(query : String) : ResultSet
#executeUpdate(query : String) : int
#prepareCall(query : String) : CallableStatement
LorannBDDConnector
-instance : LorannBDDConnector
-user : String = "root"
-password : String = ""
-url : String = "jdbc:mysql://localhost/TNTRun?useSSL=false&serverTimezone=UTC"
-connection : Connection
-statement : Statement
-LorannBDDConnector()
-setInstance(instance : LorannBDDConnector) : void
-open() : boolean
+executeQuery(query : String) : ResultSet
+prepareCall(query : String) : CallableStatement
+executeUpdate(query : String) : int
-instance
map
LvI
+runner : IRunner
-width : int = 30
-height : int = 20
-pixels : int = 30
-clock : IClock = new Clock()
-db : Datamanager = new Datamanager()
-map : ITile
+LvI()
+generate() : void
+placerunner() : void
+getMap() : ArrayList<ITile>
+setMap(map : ArrayList<ITile>) : void
+getTime() : IClock
+save(scr : String) : void
<<Interface>>
ITile
+reduce() : void
+getCounter() : int
+setActivated(activated : boolean) : void
Tile
-activated : boolean = false
-counter : int
+Tile(x : int, y : int, color : Color)
+reduce() : void
+isActivated() : boolean
<<Interface>>
IClock
+start() : void
+stop() : void
+getTime() : double
-clock
Clock
-clock : double
+Clock()
+start() : void
+stop() : void
+getTime() : double
<<Interface>>
IRunner
+moveUP() : void
+moveDOWN() : void
+moveLEFT() : void
+moveRIGHT() : void
+isAlive() : boolean
+setAlive(b : boolean) : void
+runner
<<Interface>>
IElement
+getX() : int
+getY() : int
+getColor() : Color
+lvl
+level
<<Interface>>
ILvI
+getMap() : ArrayList<ITile>
+getPixels() : int
+getWidth() : int
+getHeight() : int
+getrunner() : IRunner
+getClock() : IClock
+getTime() : IClock
+save(scr : String) : void
+lvl
-LvI
-LvI
runner
-alive : boolean = true
-level : LvI
-lvl : LvI
+runner(x : int, y : int, level : LvI)
+moveToXY(x : int, y : int) : void
+moveUP() : void
+moveDOWN() : void
+moveLEFT() : void
+moveRIGHT() : void
+isAlive() : boolean
Element
#x : int
#y : int
-color : Color
+Element(x : int, y : int)