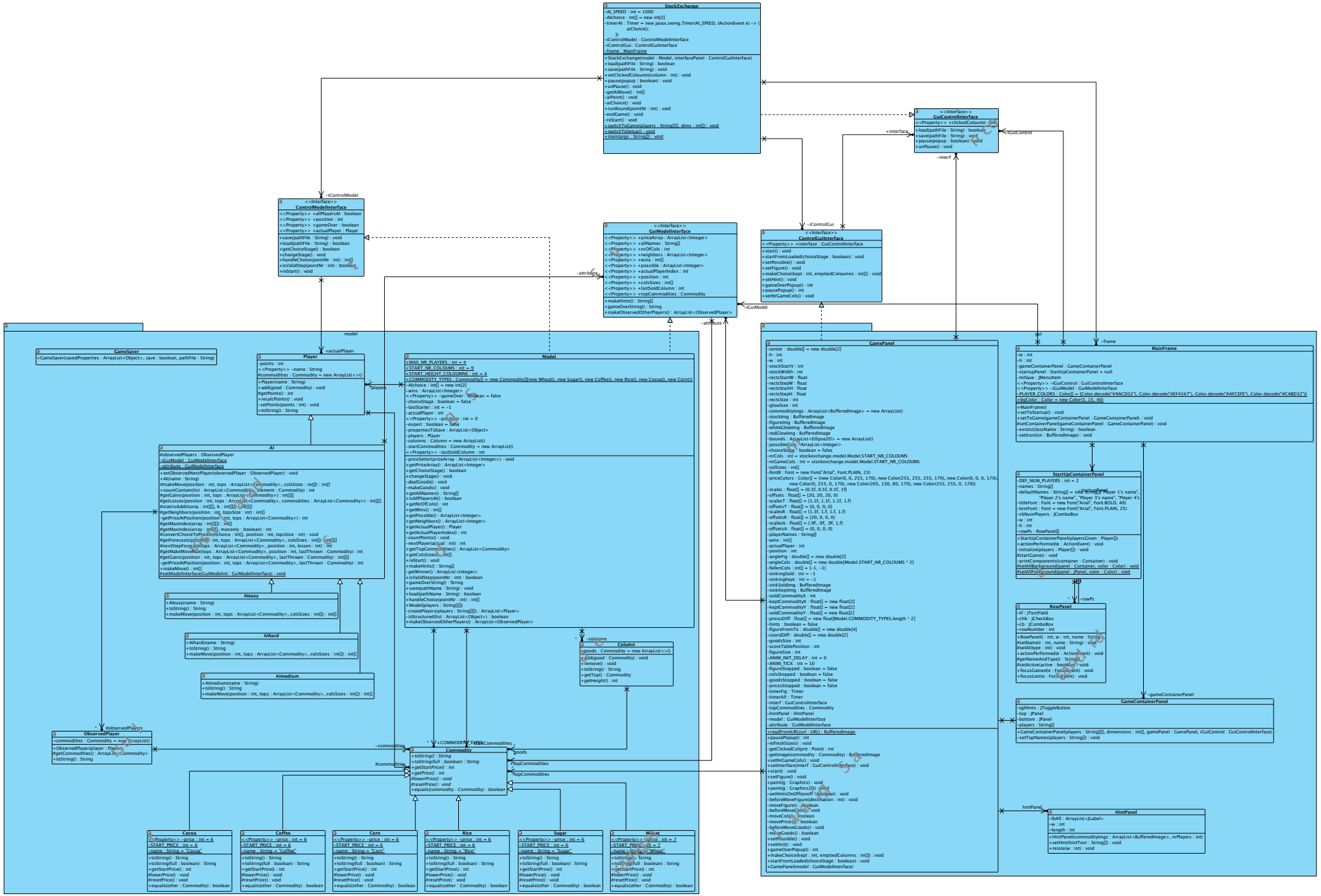
StockExchange
-AI_SPEED : int = 1000
-AIchoice : int[] = new int[2]
-timerAI : Timer = new javax.swing.Timer(AI_SPEED, (ActionEvent e) -> {
aiChoice();
})
-iControlModel : ControlModelInterface
-iControlGui : ControlGuiInterface
-frame : MainFrame
+StockExchange(model : Model, interfacePanel : ControlGuiInterface)
+load(pathFile : String) : boolean
+save(pathFile : String) : void
+setClickedColoumn(column : int) : void
+pause(popup : boolean) : void
+unPause() : void
-getAIMove() : int[]
-aiPoint() : void
-aiChoice() : void
+runRound(pointNr : int) : void
-endGame() : void
-reStart() : void
+switchToGame(players : String[][], dims : int[]) : void
+switchToSetup() : void
+main(args : String[]) : void
<<Interface>>
GuiControlInterface
<<Property>> +clickedColoumn : int[]
+load(pathFile : String) : boolean
+save(pathFile : String) : void
+pause(popup : boolean) : void
+unPause() : void
+interface
-iGuiControl
-interf
-iControlModel
<<Interface>>
ControlModelInterface
<<Property>> +allPlayersAI : boolean
<<Property>> +position : int
<<Property>> +gameOver : boolean
<<Property>> +actualPlayer : Player
+save(pathFile : String) : void
+load(pathFile : String) : boolean
+getChoiceStage() : boolean
+changeStage() : void
+handleChoice(pointNr : int) : int[]
+isValidStep(pointNr : int) : boolean
+reStart() : void
-iControlGui
<<Interface>>
ControlGuiInterface
<<Property>> +interface : GuiControlInterface
+start() : void
+startFromLoaded(choiceStage : boolean) : void
+setPossible() : void
+setFigure() : void
+makeChoice(kept : int, emptiedColoumns : int[]) : void
+setHint() : void
+gameOverPopup() : int
+pausePopup() : int
+setNrGameCols() : void
<<Interface>>
GuiModelInterface
<<Property>> +priceArray : ArrayList<Integer>
<<Property>> +allNames : String[]
<<Property>> +nrOfCols : int
<<Property>> +neighbors : ArrayList<Integer>
<<Property>> +wins : int[]
<<Property>> +possible : ArrayList<Integer>
<<Property>> +actualPlayerIndex : int
<<Property>> +position : int
<<Property>> +colsSizes : int[]
<<Property>> +lastSoldColumn : int
<<Property>> +topCommodities : Commodity
+makeHints() : String[]
+gameOverString() : String
+makeObservedOtherPlayers() : ArrayList<ObservedPlayer>
-attributes
-iGuiModel
-attribute
model
GameSaver
+GameSaver(savedProperties : ArrayList<Object>, save : boolean, pathFile : String)
Player
-points : int
<<Property>> -name : String
#commodities : Commodity = new ArrayList<>()
+Player(name : String)
+add(good : Commodity) : void
#getPoints() : int
+recalcPoints() : void
-setPoints(points : int) : void
+toString() : String
+actualPlayer
-players
Model
+MAX_NR_PLAYERS : int = 4
+START_NR_COLOUMS : int = 9
+START_HEIGHT_COLOUMNS : int = 4
+COMMODITY_TYPES : Commodity[] = new Commodity[]{new Wheat(), new Sugar(), new Coffee(), new Rice(), new Cocoa(), new Corn()}
-AIchoice : int[] = new int[2]
-wins : ArrayList<Integer>
<<Property>> -gameOver : boolean = false
-choiceStage : boolean = false
-lastStarter : int = -1
-actualPlayer : int
<<Property>> -position : int = 0
-expert : boolean = false
-propertiesToSave : ArrayList<Object>
-players : Player
-columns : Column = new ArrayList()
-startCommodities : Commodity = new ArrayList()
<<Property>> -lastSoldColumn : int
-priceSetter(priceArray : ArrayList<Integer>) : void
+getPriceArray() : ArrayList<Integer>
+getChoiceStage() : boolean
+changeStage() : void
-dealGoods() : void
-makeGoods() : void
+getAllNames() : String[]
+isAllPlayersAI() : boolean
+getNrOfCols() : int
+getWins() : int[]
+getPossible() : ArrayList<Integer>
+getNeighbors() : ArrayList<Integer>
+getActualPlayer() : Player
+getActualPlayerIndex() : int
-countPoints() : void
-nextPlayer(actual : int) : int
+getTopCommodities() : ArrayList<Commodity>
+getColsSizes() : int[]
+reStart() : void
+makeHints() : String[]
-getWinner() : ArrayList<Integer>
+isValidStep(pointNr : int) : boolean
+gameOverString() : String
+save(pathName : String) : void
+load(pathName : String) : boolean
+handleChoice(pointNr : int) : int[]
+Model(players : String[][])
-createPlayers(players : String[][]) : ArrayList<Player>
-isStructured(list : ArrayList<Object>) : boolean
+makeObservedOtherPlayers() : ArrayList<ObservedPlayer>
AI
#observedPlayers : ObservedPlayer
-iGuiModel : GuiModelInterface
-attribute : GuiModelInterface
+setObservedNextPlayer(observedPlayer : ObservedPlayer) : void
+AI(name : String)
#makeMove(position : int, tops : ArrayList<Commodity>, colsSizes : int[]) : int[]
+countContain(list : ArrayList<Commodity>, element : Commodity) : int
#getGains(position : int, tops : ArrayList<Commodity>) : int[][]
#getLosses(position : int, tops : ArrayList<Commodity>, commodities : ArrayList<Commodity>) : int[][]
#matrixAddition(a : int[][], b : int[][]) : int[][]
#getNeighbors(position : int, topsSize : int) : int[]
-getPriceAtPosition(position : int, tops : ArrayList<Commodity>) : int
#getMaxIndex(array : int[]) : int[]
#getMaxIndex(array : int[]), maxonly : boolean) : int
#convertChoiceToPoint(choice : int[], position : int, topsSize : int) : void
#getForecasts(popup : int, tops : ArrayList<Commodity>, colsSizes : int[]) : int[][]
#nextStepForecast(tops : ArrayList<Commodity>, position : int, lessen : int) : int
#getMakeMoveMax(tops : ArrayList<Commodity>, position : int, lastThrown : Commodity) : int
#getGains(position : int, tops : ArrayList<Commodity>, lastThrown : Commodity) : int[][]
-getPriceAtPosition(position : int, tops : ArrayList<Commodity>, lastThrown : Commodity) : int
+makeMove() : int[]
#setModelInterface(GuiModelInt : GuiModelInterface) : void
Aleasy
+Aleasy(name : String)
+toString() : String
+makeMove(position : int, tops : ArrayList<Commodity>, colsSizes : int[]) : int[]
Alhard
+Alhard(name : String)
+toString() : String
+makeMove(position : int, tops : ArrayList<Commodity>, colsSizes : int[]) : int[]
Almedium
+Almedium(name : String)
+toString() : String
+makeMove(position : int, tops : ArrayList<Commodity>, colsSizes : int[]) : int[]
Column
-goods : Commodity = new ArrayList<>()
+add(good : Commodity) : void
+remove() : void
+toString() : String
+getTop() : Commodity
+getHeight() : int
-columns
ObservedPlayer
-commodities : Commodity = new ArrayList()
+ObservedPlayer(player : Player)
#getCommodities() : ArrayList<Commodity>
+toString() : String
#observedPlayers
-commodities
Commodity
+toString() : String
+toString(full : boolean) : String
+getStartPrice() : int
+getPrice() : int
#flowerPrice() : void
#resetPrice() : void
#equals(commodity : Commodity) : boolean
#commodities
+COMMODITY_TYPES
-startCommodities
goods
topCommodities
topCommodities
Cocoa
<<Property>> -price : int = 6
-START_PRICE : int = 6
-name : String = "Cocoa"
+toString() : String
+toString(full : boolean) : String
+getStartPrice() : int
#flowerPrice() : void
#resetPrice() : void
+equals(other : Commodity) : boolean
Coffee
<<Property>> -price : int = 6
-START_PRICE : int = 6
-name : String = "Coffee"
+toString() : String
+toString(full : boolean) : String
+getStartPrice() : int
#flowerPrice() : void
#resetPrice() : void
+equals(other : Commodity) : boolean
Corn
<<Property>> -price : int = 6
-START_PRICE : int = 6
-name : String = "Corn"
+toString() : String
+toString(full : boolean) : String
+getStartPrice() : int
#flowerPrice() : void
#resetPrice() : void
+equals(other : Commodity) : boolean
Rice
<<Property>> -price : int = 6
-START_PRICE : int = 6
-name : String = "Rice"
+toString() : String
+toString(full : boolean) : String
+getStartPrice() : int
#flowerPrice() : void
#resetPrice() : void
+equals(other : Commodity) : boolean
Sugar
<<Property>> -price : int = 6
-START_PRICE : int = 6
-name : String = "Sugar"
+toString() : String
+toString(full : boolean) : String
+getStartPrice() : int
#flowerPrice() : void
#resetPrice() : void
+equals(other : Commodity) : boolean
Wheat
<<Property>> -price : int = 7
-START_PRICE : int = 7
-name : String = "Wheat"
+toString() : String
+toString(full : boolean) : String
+getStartPrice() : int
#flowerPrice() : void
#resetPrice() : void
+equals(other : Commodity) : boolean
GamePanel
-center : double[] = new double[2]
-h : int
-w : int
-stockStartV : int
-stockWidth : int
-rectsStartW : float
-rectsStepW : float
-rectsStartH : float
-rectsStepH : float
-rectsSize : int
-glowSize : int
-commodityImgs : ArrayList<BufferedImage> = new ArrayList()
-stocking : BufferedImage
-figureImg : BufferedImage
-whiteGlowImg : BufferedImage
-redGlowImg : BufferedImage
-bounds : ArrayList<Ellipse2D> = new ArrayList()
-possibleCols : ArrayList<Integer>
-choiceStage : boolean = false
-nrCols : int = stockexchange.model.Model.START_NR_COLOUMS
-nrGameCols : int = stockexchange.model.Model.START_NR_COLOUMS
-colSizes : int[]
-fontB : Font = new Font("Arial", Font.PLAIN, 23)
-priceColors : Color[] = {new Color(0, 0, 255, 170), new Color(255, 255, 255, 170), new Color(0, 0, 0, 170),
new Color(0, 255, 0, 170), new Color(205, 130, 80, 170), new Color(255, 255, 0, 170)}
-scales : float[] = {0.5f, 0.5f, 0.5f, 1f}
-offsets : float[] = {20, 20, 20, 0}
-scalesT : float[] = {1.1f, 1.1f, 1.1f, 1.f}
-offsetsT : float[] = {0, 0, 0, 0}
-scalesR : float[] = {1.3f, 1.f, 1.f, 1.f}
-offsetsR : float[] = {20, 0, 0, 0}
-scalesA : float[] = {.9f, .9f, .9f, 1.f}
-offsetsA : float[] = {0, 0, 0, 0}
-playerNames : String[]
-wins : int[]
-actualPlayer : int
-position : int
-angleFig : double[] = new double[2]
-angleCols : double[] = new double[Model.START_NR_COLOUMS * 2]
-fallenCols : int[] = {-1, -1, -1}
-sinkingSold : int = -1
-sinkingKept : int = -1
-sinkSoldImg : BufferedImage
-sinkKeptImg : BufferedImage
-soldCommodityX : int
-keptCommodityX : float[] = new float[2]
-keptCommodityY : float[] = new float[2]
-soldCommodityY : float[] = new float[2]
-pricesDiff : float[] = new float[Model.COMMODITY_TYPES.length * 2]
-hints : boolean = false
-figureFromTo : double[] = new double[4]
-coordDiff : double[] = new double[2]
-goodsSize : int
-scoreTablePosition : int
-figureSize : int
-ANIM_INIT_DELAY : int = 0
-ANIM_TICK : int = 10
-figureStopped : boolean = false
-colsStopped : boolean = false
-goodsStopped : boolean = false
-pricesStopped : boolean = false
-timerFig : Timer
-timerAll : Timer
-interf : GuiControlInterface
-topCommodities : Commodity
-hintPanel : HintPanel
-model : GuiModelInterface
-attribute : GuiModelInterface
+readFromURL(url : URL) : BufferedImage
+pausePopup() : int
+refreshSizes() : void
-getClickedColPnt : Point2D : int
-getImage(commodity : Commodity) : BufferedImage
+setNrGameCols() : void
+setInterface(interf : GuiControlInterface) : void
+start() : void
+setFigure() : void
+paint(g : Graphics) : void
+paint(g : Graphics2D) : void
-setHintsOnOffionoff : boolean) : void
-beforeMoveFigure(destination : int) : void
-moveFigure() : boolean
-beforeMoveCols() : void
-moveCols() : boolean
-movePrice() : boolean
-beforeMoveGoods() : void
-moveGoods() : boolean
+setPossible() : void
+setHint() : void
+gameOverPopup() : int
+makeChoice(kept : int, emptiedColumns : int[]) : void
+startFromLoaded(choiceStage : boolean) : void
+GamePanel(model : GuiModelInterface)
MainFrame
-w : int
-h : int
-gameContainerPanel : GameContainerPanel
-startupPanel : StartUpContainerPanel = null
-miSave : JMenuItem
<<Property>> -iGuiControl : GuiControlInterface
<<Property>> -iGuiModel : GuiModelInterface
-PLAYER_COLORS : Color[] = {Color.decode("#9ACD32"), Color.decode("#EF4347"), Color.decode("#AFCEFE"), Color.decode("#CABD32")}
+bgColor : Color = new Color(15, 15, 40)
+MainFrame()
+setToStartup() : void
+setToGame(gameContainerPanel : GameContainerPanel) : void
#setContainerPanel(gameContainerPanel : GameContainerPanel) : void
-exists(className : String) : boolean
-setIcon(ic : BufferedImage) : void
StartUpContainerPanel
-DEF_NUM_PLAYERS : int = 2
-names : String[]
-defaultNames : String[] = new String[]{"Player 1's name",
"Player 2's name", "Player 3's name", "Player 4's
-titleFont : Font = new Font("Arial", Font.BOLD, 40)
-textFont : Font = new Font("Arial", Font.PLAIN, 25)
-cbNumPlayers : JComboBox
-w : int
-h : int
-rowPs : RowPanel[]
+StartUpContainerPanel(playersGiven : Player[])
+actionPerformed(e : ActionEvent) : void
-initialize(players : Player[]) : void
#startGame() : void
-printComponents(container : Container) : void
#setAllBackground(color : Container, color : Color) : void
#setAllForeground(panel : JPanel, color : Color) : void
-rowPs
RowPanel
-tf : JTextField
-chb : JCheckBox
-cb : JComboBox
-rowNumber : int
+RowPanel(i : int, w : int, name : String)
#setName(i : int, name : String) : void
#setAlType(i : int) : void
+actionPerformed(e : ActionEvent) : void
#getNameAndType() : String[]
#setActive(active : boolean) : void
+focusGained(e : FocusEvent) : void
+focusLost(e : FocusEvent) : void
-frame
-gameContainerPanel
GameContainerPanel
-tglHints : JToggleButton
-top : JPanel
-bottom : JPanel
-players : String[]
+GameContainerPanel(players : String[][], dimensions : int[], gamePanel : GamePanel, iGuiControl : GuiControlInterface)
+setTopNames(players : String[]) : void
hintPanel
HintPanel
-lbAll : ArrayList<JLabel>
-w : int
-length : int
+HintPanel(commodityImgs : ArrayList<BufferedImage>, nrPlayers : int)
+setHint(hintText : String[]) : void
+resize(w : int) : void