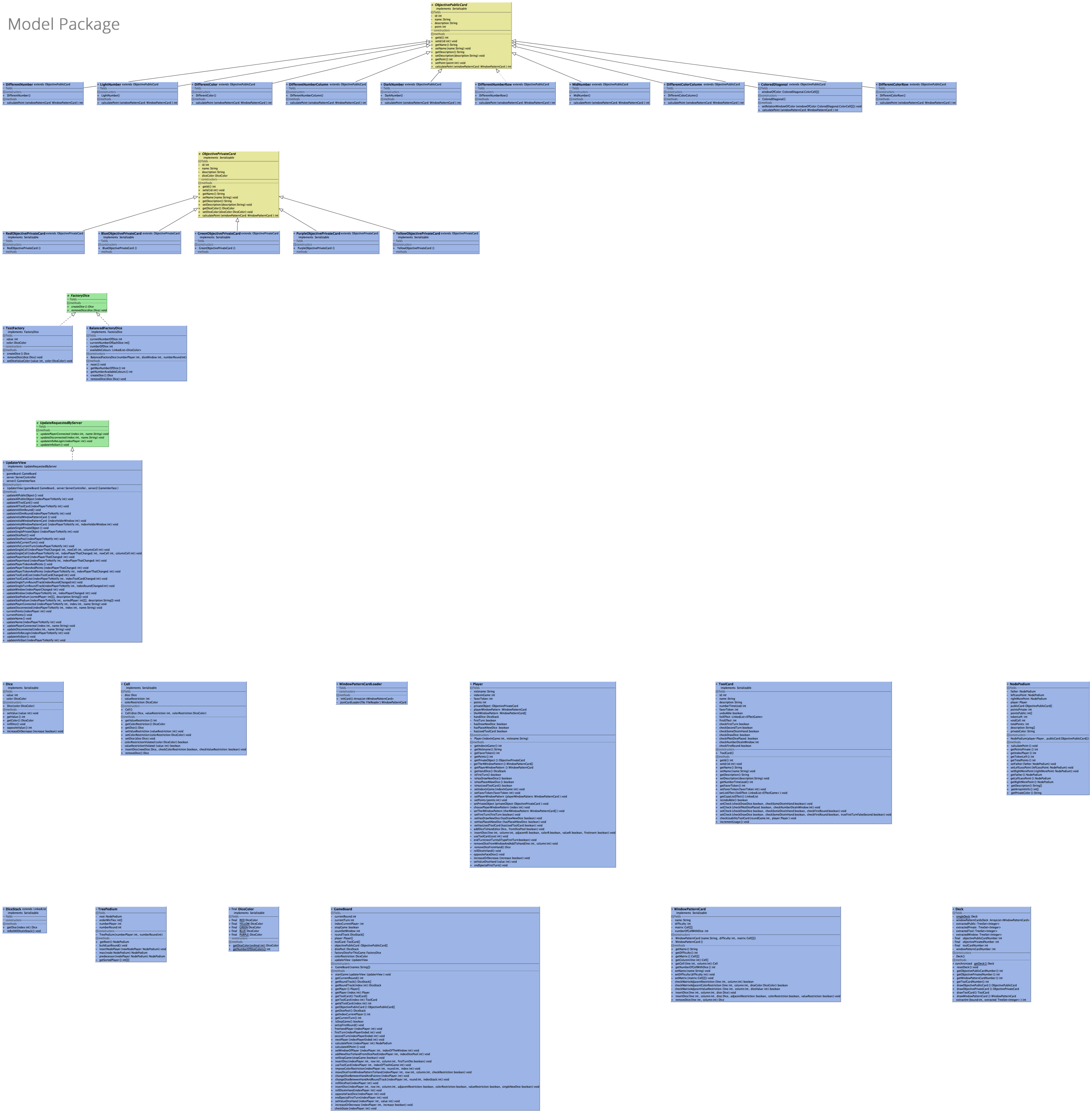
Model Package


+ ObjectivePublicCard
implements Serializable
fields
- id:int
- name:String
- description:String
- point:int
constructors
methods
+ getId():int
+ setId(id:int):void
+ getName():String
+ setName(name:String):void
+ getDescription():String
+ setDescription(description:String):void
+ getPoint():int
+ setPoint(point:int):void
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ DifferentNumber extends ObjectivePublicCard
fields
constructors
+ DifferentNumber()
methods
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ LightNumber extends ObjectivePublicCard
fields
constructors
+ LightNumber()
methods
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ DifferentColor extends ObjectivePublicCard
fields
constructors
+ DifferentColor()
methods
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ DifferentNumberColumn extends ObjectivePublicCard
fields
constructors
+ DifferentNumberColumn()
methods
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ DarkNumber extends ObjectivePublicCard
fields
constructors
+ DarkNumber()
methods
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ DifferentNumberRow extends ObjectivePublicCard
fields
constructors
+ DifferentNumberRow()
methods
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ MidNumber extends ObjectivePublicCard
fields
constructors
+ MidNumber()
methods
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ DifferentColorColumn extends ObjectivePublicCard
fields
constructors
+ DifferentColorColumn()
methods
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ ColoredDiagonal extends ObjectivePublicCard
fields
- windowOfColor:ColoredDiagonal.ColorCell[][]
constructors
+ ColoredDiagonal()
methods
- setRelationWindowOfColor (windowOfColor:ColoredDiagonal.ColorCell[][]):void
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ DifferentColorRow extends ObjectivePublicCard
fields
constructors
+ DifferentColorRow()
methods
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ ObjectivePrivateCard
implements Serializable
fields
- id:int
- name:String
- description:String
- diceColor:DiceColor
constructors
methods
+ getId():int
+ setId(id:int):void
+ getName():String
+ setName(name:String):void
+ getDescription():String
+ setDescription(description:String):void
+ getDiceColor():DiceColor
+ setDiceColor(diceColor:DiceColor):void
+ calculatePoint (windowPatternCard: WindowPatternCard):int
+ RedObjectivePrivateCard extends ObjectivePrivateCard
implements Serializable
fields
constructors
+ RedObjectivePrivateCard ()
methods
+ BlueObjectivePrivateCard extends ObjectivePrivateCard
implements Serializable
fields
constructors
+ BlueObjectivePrivateCard ()
methods
+ GreenObjectivePrivateCard extends ObjectivePrivateCard
implements Serializable
fields
constructors
+ GreenObjectivePrivateCard ()
methods
+ PurpleObjectivePrivateCard extends ObjectivePrivateCard
implements Serializable
fields
constructors
+ PurpleObjectivePrivateCard ()
methods
+ YellowObjectivePrivateCard extends ObjectivePrivateCard
implements Serializable
fields
constructors
+ YellowObjectivePrivateCard ()
methods
+ FactoryDice
fields
methods
+ createDice ():Dice
+ removeDice(dice:Dice):void
+ TestFactory
implements FactoryDice
fields
- value:int
- color:DiceColor
constructors
methods
+ createDice ():Dice
+ removeDice(dice:Dice):void
+ setDiceValueColor (value:int, color:DiceColor):void
+ BalancedFactoryDice
implements FactoryDice
fields
- currentNumberOfDice:int
- currentNumberOfEachDice:int[]
- numberOfDice:int
- availableColours :LinkedList<DiceColor>
constructors
+ BalancedFactoryDice (numberPlayer:int, diceWindow:int, numberRound:int)
methods
+ reset ():void
+ getMaxNumberOfDice ():int
+ getNumberAvailableColours ():int
+ createDice ():Dice
+ removeDice(dice:Dice):void
+ UpdateRequestedByServer
fields
methods
+ updatePlayerConnected (index:int, name:String):void
+ updateDisconnected(index:int, name:String):void
+ updateInfoReLogin(indexPlayer: int):void
+ updateInfoStart ():void
+ UpdaterView
implements UpdateRequestedByServer
fields
- gameBoard:GameBoard
- server:ServerController
- server2:GameInterface
constructors
+ UpdaterView (gameBoard:GameBoard , server:ServerController, server2:GameInterface)
methods
- updateAllPublicObject ():void
- updateAllPublicObject (indexPlayerToNotify: int):void
- updateAllToolCard ():void
- updateAllToolCard (indexPlayerToNotify: int):void
- updateInitDimRound():void
- updateInitDimRound(indexPlayerToNotify: int):void
- updateInitialWindowPatternCard ():void
- updateInitialWindowPatternCard (indexHolderWindow:int):void
- updateInitialWindowPatternCard (indexPlayerToNotify: int, indexHolderWindow:int):void
- updateSinglePrivateObject ():void
- updateSinglePrivateObject (indexPlayerToNotify: int):void
~ updateDicePool ():void
- updateDicePool (indexPlayerToNotify: int):void
~ updateInfoCurrentTurn():void
- updateInfoCurrentTurn (indexPlayerToNotify: int):void
~ updateSingleCell (indexPlayerThatChanged: int, rowCell: int, columnCell:int):void
- updateSingleCell (indexPlayerToNotify: int, indexPlayerThatChanged: int, rowCell: int, columnCell:int):void
~ updatePlayerHand (indexPlayerThatChanged: int):void
- updatePlayerHand (indexPlayerToNotify: int, indexPlayerThatChanged: int):void
~ updatePlayerTokenAndPoints ():void
~ updatePlayerTokenAndPoints (indexPlayerThatChanged: int):void
- updatePlayerTokenAndPoints (indexPlayerToNotify: int, indexPlayerThatChanged: int):void
~ updateToolCardCost (indexToolCardChanged:int):void
- updateToolCardCost (indexPlayerToNotify: int, indexToolCardChanged:int):void
~ updateSingleTurnRoundTrack(indexRoundChanged:int):void
- updateSingleTurnRoundTrack(indexPlayerToNotify: int, indexRoundChanged:int):void
~ updateWindow (indexPlayerChanged: int):void
- updateWindow (indexPlayerToNotify: int, indexPlayerChanged: int):void
~ updateStatPodium (sortedPlayer: int[][], description:String[]):void
- updateStatPodium (indexPlayerToNotify: int, sortedPlayer: int[][], description:String[]):void
- updatePlayerConnected (indexPlayerToNotify: int, index:int, name:String):void
- updateDisconnected (indexPlayerToNotify: int, index:int, name:String):void
- currentPoints (indexPlayer: int):void
~ currentPoints ():void
- updateName ():void
- updateName (indexPlayerToNotify: int):void
+ updatePlayerConnected (index:int, name:String):void
+ updateDisconnected (index:int, name:String):void
+ updateInfoReLogin (indexPlayerToNotify: int):void
+ updateInfoStart ():void
+ updateInfoStart (indexPlayerToNotify: int):void
+ Dice
implements Serializable
fields
- value:int
- color:DiceColor
constructors
+ Dice(color:DiceColor)
methods
+ setValue (value: int):void
+ getValue ():int
+ getColor():DiceColor
+ rollDice():void
+ oppositeValue ():int
+ increaseOrDecrease (increase: boolean):void
+ Cell
implements Serializable
fields
- dice:Dice
- valueRestriction :int
- colorRestriction:DiceColor
constructors
+ Cell ()
+ Cell (dice:Dice, valueRestriction:int, colorRestriction:DiceColor)
methods
+ getValueRestriction ():int
+ getColorRestriction ():DiceColor
+ getDice ():Dice
+ setValueRestriction (valueRestriction: int):void
+ setColorRestriction (colorRestriction:DiceColor):void
+ setDice (dice:Dice):void
- colorRestrictionViolated (color:DiceColor):boolean
- valueRestrictionViolated (value: int):boolean
+ insertDice (newDice:Dice, checkColorRestriction:boolean, checkValueRestriction: boolean):void
+ removeDice():Dice
+ WindowPatternCardLoader
fields
constructors
methods
+ initCard ():ArrayList<WindowPatternCard>
- jsonCardLoader(file: FileReader):WindowPatternCard
+ Player
fields
- nickname:String
- indexInGame:int
- favorToken:int
- points:int
- privateObject :ObjectivePrivateCard
- playerWindowPattern :WindowPatternCard
- the4WindowPattern :WindowPatternCard[]
- handDice:DiceStack
- firstTurn:boolean
- hasDrawNewDice :boolean
- hasPlaceANewDice :boolean
- hasUsedToolCard:boolean
constructors
+ Player (indexInGame:int, nickname:String)
methods
+ getIndexInGame ():int
+ getNickname ():String
+ getFavorToken():int
+ getPoints ():int
+ getPrivateObject ():ObjectivePrivateCard
~ getThe4WindowPattern ():WindowPatternCard[]
+ getPlayerWindowPattern ():WindowPatternCard
+ getHandDice():DiceStack
+ isFirstTurn():boolean
+ isHasDrawNewDice():boolean
+ isHasPlaceANewDice ():boolean
+ isHasUsedToolCard():boolean
+ setIndexInGame (indexInGame:int):void
+ setFavorToken(favorToken:int):void
+ setPlayerWindowPattern (playerWindowPattern: WindowPatternCard):void
+ setPoints (points:int):void
~ setPrivateObject (privateObject: ObjectivePrivateCard):void
~ choosePlayerWindowPattern (index:int):void
~ setThe4WindowPattern (the4WindowPattern: WindowPatternCard[]):void
+ setFirstTurn(firstTurn:boolean):void
+ setHasDrawNewDice (hasDrawNewDice: boolean):void
+ setHasPlaceANewDice (hasPlaceANewDice: boolean):void
+ setHasUsedToolCard (hasUsedToolCard:boolean):void
~ addDiceToHand(dice:Dice, fromDicePool:boolean):void
+ insertDice (line:int, column:int, adjacentR:boolean, colorR:boolean, valueR:boolean, firstInsert: boolean):void
~ useToolCard(cost:int):void
~ endTurn(nextTurnIsATypeFirstTurn:boolean):void
~ removeDiceFromWindowAndAddToHand(line:int, column:int):void
+ removeDiceFromHand():Dice
~ rollDiceInHand():void
~ oppositeFaceDice():void
~ increaseOrDecrease (increase: boolean):void
~ setValueDiceHand (value: int):void
+ endSpecialFirstTurn():void
+ ToolCard
implements Serializable
fields
- id:int
- name:String
- description:String
- numberTimeUsed:int
- favorToken:int
- undoAble:boolean
- listEffect :LinkedList<EffectGame>
- firstEffect :int
- checkFirstTurn:boolean
- checkSecondTurn:boolean
- checkSomeDiceInHand:boolean
- checkDrawDice:boolean
- checkIfNotDicePlaced :boolean
- checkNumberDiceInWindow:int
- checkFirstRound:boolean
constructors
+ ToolCard()
methods
+ getId():int
+ setId (id:int):void
+ getName ():String
+ setName (name:String):void
+ getDescription():String
+ setDescription (description:String):void
+ getNumberTimeUsed():int
+ getFavorToken():int
+ setFavorToken (favorToken:int):void
~ setListEffect (listEffect: LinkedList<EffectGame>):void
+ getCopyListEffect ():LinkedList
+ isUndoAble():boolean
+ setCheck (checkDrawDice:boolean, checkSomeDiceInHand:boolean):void
+ setCheck (checkIfNotDicePlaced: boolean, checkNumberDiceInWindow:int):void
+ setCheck (checkDrawDice:boolean, checkSomeDiceInHand:boolean, checkFirstRound:boolean):void
+ setCheck (checkDrawDice:boolean, checkSomeDiceInHand:boolean, checkFirstRound:boolean, trueFirstTurnFalseSecond:boolean):void
+ checkUsabilityToolCard (roundGame:int, player:Player):void
+ incrementUsage ():void
+ NodePodium
fields
- father:NodePodium
- leftLessPoint :NodePodium
- rightMorePoint :NodePodium
- player:Player
- publicCard:ObjectivePublicCard[]
- pointsPrivate :int
- pointsPublic:int[]
- tokenLeft :int
- voidCell:int
- totalPoints :int
- description:String[]
- privateColor :String
constructors
~ NodePodium(player:Player , publicCard:ObjectivePublicCard[])
methods
+ calculatePoint ():void
~ getPointsPrivate ():int
~ getIndexPlayer ():int
~ getTokenLeft ():int
~ getTotalPoints ():int
~ setFather (father: NodePodium):void
~ setLeftLessPoint (leftLessPoint: NodePodium):void
~ setRightMorePoint (rightMorePoint: NodePodium):void
~ getFather ():NodePodium
~ getLeftLessPoint ():NodePodium
~ getRightMorePoint ():NodePodium
+ getDescription():String[]
+ getArrayIntInfo ():int[]
+ getPrivateColor ():String
+ DiceStack extends LinkedList
implements Serializable
fields
constructors
methods
+ getDice (index:int):Dice
+ reRollAllDiceInStack ():void
~ TreePodium
fields
- root:NodePodium
- orderWinTies:int[]
- numberPlayer:int
- numberRound:int
constructors
~ TreePodium(numberPlayer:int, numberRound:int)
methods
+ getRoot():NodePodium
- buildLastRound():void
~ insertNodePlayer (newNodePlayer: NodePodium):void
- max(node:NodePodium):NodePodium
- predecessor(nodePlayer: NodePodium):NodePodium
~ getSortedPlayer ():int[][]
+ final DiceColor
implements Serializable
fields
+ final RED:DiceColor
+ final YELLOW:DiceColor
+ final GREEN:DiceColor
+ final BLUE:DiceColor
+ final PURPLE:DiceColor
constructors
methods
+ getDiceColor(ordinal:int):DiceColor
+ getNumberOfDiceColors():int
+ GameBoard
fields
- currentRound:int
- currentTurn:int
- indexCurrentPlayer :int
- stopGame:boolean
- countSetWindow:int
- roundTrack:DiceStack[]
- player:Player[]
- toolCard:ToolCard[]
- objectivePublicCard :ObjectivePublicCard[]
- dicePool:DiceStack
- factoryDiceForThisGame:FactoryDice
- colorRestriction:DiceColor
- updaterView :UpdaterView
constructors
+ GameBoard (names:String[])
methods
+ startGame (updaterView: UpdaterView):void
+ getCurrentRound():int
+ getRoundTrack():DiceStack[]
+ getRoundTrack(index:int):DiceStack
+ getPlayer ():Player[]
+ getPlayer (index:int):Player
+ getToolCard():ToolCard[]
+ getToolCard(index:int):ToolCard
+ getIdToolCard(index:int):int
+ getObjectivePublicCard ():ObjectivePublicCard[]
+ getDicePool ():DiceStack
+ getIndexCurrentPlayer ():int
+ getCurrentTurn():int
+ isStopGame():boolean
- setUpFirstRound():void
- freeHandPlayer (indexPlayer: int):void
- firstTurn(indexPlayerEnded:int):void
- secondTurn(indexPlayerEnded:int):void
+ nextPlayer (indexPlayerEnded:int):void
+ calculatePoint (indexPlayer: int):NodePodium
+ calculateAllPoint ():void
+ setWindowOfPlayer (indexPlayer: int, indexOfTheWindow: int):void
+ addNewDiceToHandFromDicePool(indexPlayer: int, indexDicePool:int):void
+ setStopGame (stopGame:boolean):void
+ insertDice (indexPlayer: int, row:int, column:int, firstTurnDie:boolean):void
+ useToolCard (indexPlayer: int, indexOfToolInGame:int):void
+ imposeColorRestriction (indexPlayer: int, round:int, index:int):void
+ moveDiceFromWindowPatternToHand(indexPlayer: int, row:int, column:int, checkRestriction:boolean):void
+ changeDiceBetweenHandAndFactory (indexPlayer: int):void
+ changeDiceBetweenHandAndRoundTrack(indexPlayer: int, round:int, indexStack:int):void
+ rollDicePool (indexPlayer: int):void
+ insertDice (indexPlayer: int, row:int, column:int, adjacentRestriction: boolean, colorRestriction:boolean, valueRestriction: boolean, singleNewDice:boolean):void
+ rollDiceInHand(indexPlayer: int):void
+ oppositeFaceDice(indexPlayer: int):void
+ endSpecialFirstTurn(indexPlayer: int):void
+ setValueDiceHand (indexPlayer: int, value: int):void
+ increaseOrDecrease (indexPlayer: int, increase: boolean):void
- checkState (indexPlayer: int):void
+ WindowPatternCard
implements Serializable
fields
- name:String
- difficulty:int
- matrix:Cell[][]
- numberOfCellWithDice :int
constructors
+ WindowPatternCard (name:String, difficulty:int, matrix:Cell[][])
+ WindowPatternCard ()
methods
+ getName ():String
+ getDifficulty ():int
+ getMatrix ():Cell[][]
+ getColumn (line:int):Cell[]
+ getCell (line:int, column:int):Cell
+ getNumberOfCellWithDice ():int
~ setName (name:String):void
~ setDifficulty (difficulty:int):void
~ setMatrix (matrix:Cell[][]):void
- checkMatrixAdjacentRestriction (line:int, column:int):boolean
- checkMatrixAdjacentColorRestriction (line:int, column:int, diceColor:DiceColor):boolean
- checkMatrixAdjacentValueRestriction (line:int, column:int, diceValue: int):boolean
+ insertDice(line:int, column:int, dice:Dice):void
+ insertDice (line:int, column:int, dice:Dice, adjacentRestriction: boolean, colorRestriction:boolean, valueRestriction: boolean):void
+ removeDice(line:int, column:int):Dice
+ Deck
fields
- singleDeck:Deck
- windowPatternCardsDeck :ArrayList<WindowPatternCard>
- extractedPublic :TreeSet<Integer>
- extractedPrivate :TreeSet<Integer>
- extractedTool :TreeSet<Integer>
- extractedWindow :TreeSet<Integer>
- final objectivePublicCardNumber:int
- final objectivePrivatedNumber :int
- final toolCardNumber:int
- windowPatternCardNumber :int
constructors
- Deck()
methods
+ synchronized getDeck():Deck
+ resetDeck ():void
+ getObjectivePublicCardNumber():int
+ getObjectivePrivatedNumber ():int
+ getWindowPatternCardNumber ():int
+ getToolCardNumber():int
+ drawObjectivePublicCard ():ObjectivePublicCard
+ drawObjectivePrivateCard ():ObjectivePrivateCard
+ drawToolCard():ToolCard
+ drawWindowPatternCard ():WindowPatternCard
- extractInt (bound:int, extracted: TreeSet<Integer>):int