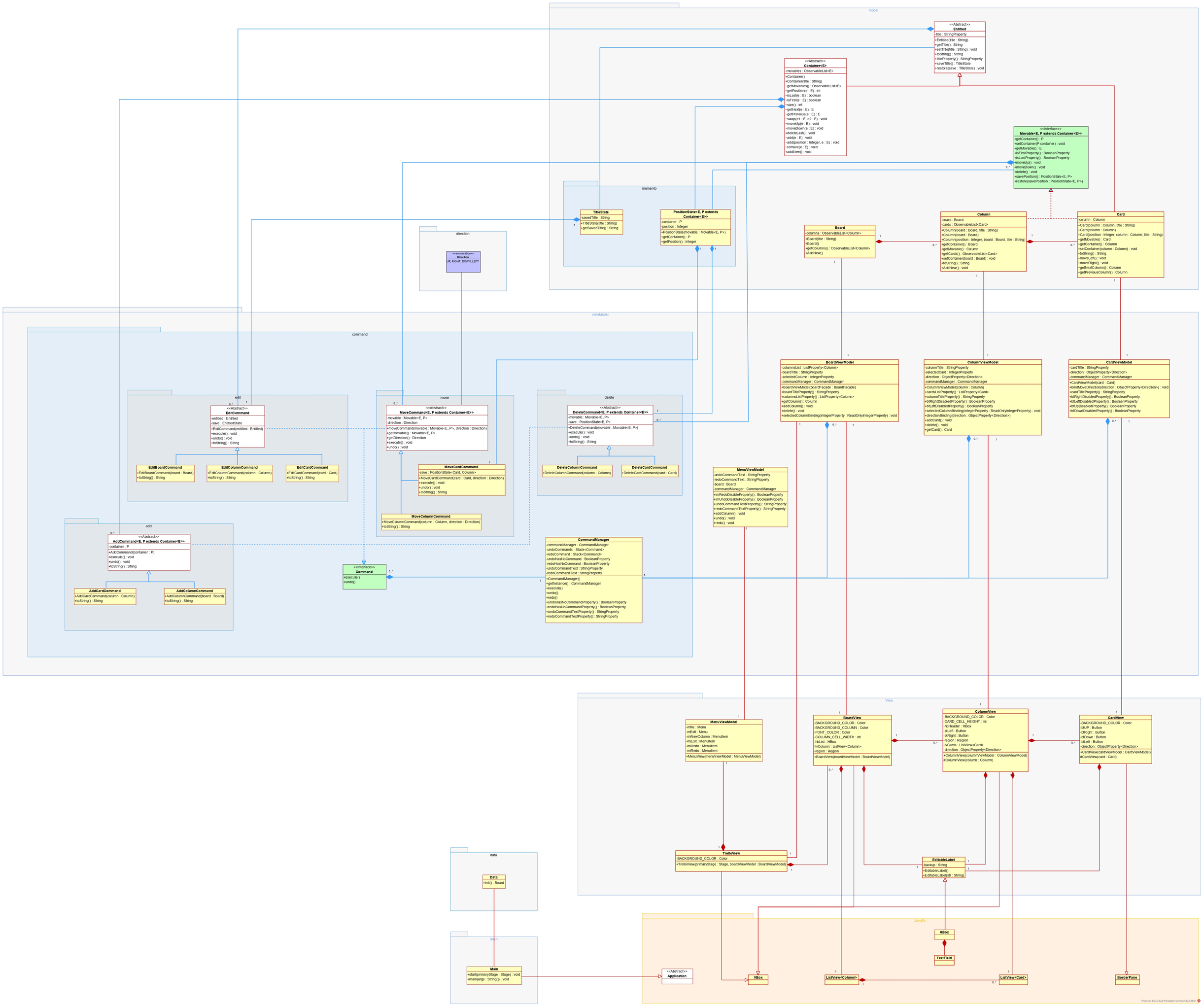
model
<<Abstract>>
Entitled
-title : StringProperty
+Entitled(title : String)
+getTitle() : String
+setTitle(title : String) : void
+toString() : String
+titleProperty() : StringProperty
+saveTitle() : TitleState
+restore(save : TitleState) : void
<<Abstract>>
Container<E>
-movables : ObservableList<E>
+Container()
+Container(title : String)
~getMovables() : ObservableList<E>
~getPosition(e : E) : int
~isLast(e : E) : boolean
~isFirst(e : E) : boolean
~size() : int
~getNext(e : E) : E
~getPrevious(e : E) : E
~swap(e1 : E, e2 : E) : void
~moveUp(e : E) : void
~moveDown(e : E) : void
+deleteLast() : void
~add(e : E) : void
~add(position : Integer, e : E) : void
~remove(e : E) : void
+addNew() : void
<<Interface>>
Movable<E, P extends Container<E>>
+getContainer() : P
+setContainer(P container) : void
+getMovable() : E
+isFirstProperty() : BooleanProperty
+isLastProperty() : BooleanProperty
+moveUp() : void
+moveDown() : void
+delete() : void
+savePosition() : PositionState<E, P>
+restore(savePosition : PositionState<E, P>)
memento
TitleState
-savedTitle : String
+TitleState(title : String)
+getSavedTitle() : String
PositionState<E, P extends Container<E>>
-container : P
-position : Integer
+PositionState(movable : Movable<E, P>)
+getContainer() : P
+getPosition() : Integer
direction
<<enumeration>>
Direction
UP, RIGHT, DOWN, LEFT
Board
-columns : ObservableList<Column>
+Board(title : String)
+Board()
+getColumns() : ObservableList<Column>
+AddNew()
Column
-board : Board
-cards : ObservableList<Card>
+Column(board : Board, title : String)
+Column(board : Board)
+Column(position : Integer, board : Board, title : String)
+getContainer() : Board
+getMovable() : Column
+getCards() : ObservableList<Card>
+setContainer(board : Board) : void
+toString() : String
+AddNew() : void
Card
-column : Column
+Card(column : Column, title : String)
+Card(column : Column)
+Card(position : Integer, column : Column, title : String)
+getMovable() : Card
+getContainer() : Column
+setContainer(column : Column) : void
+toString() : String
+moveLeft() : void
+moveRight() : void
+getNextColumn() : Column
+getPreviousColumn() : Column
viewModel
command
edit
<<Abstract>>
EditCommand
-entitled : Entitled
-save : EntitledState
+EditCommand(entitled : Entitled)
+execute() : void
+undo() : void
+toString() : String
EditBoardCommand
+EditBoardCommand(board : Board)
+toString() : String
EditColumnCommand
+EditColumnCommand(column : Column)
+toString() : String
EditCardCommand
+EditCardCommand(card : Card)
+toString() : String
move
<<Abstract>>
MoveCommand<E, P extends Container<E>>
-movable : Movable<E, P>
-direction : Direction
+moveCommand(movable : Movable<E, P>, direction : Direction)
+getMovable() : Movable<E, P>
+getDirection() : Direction
+execute() : void
+undo() : void
MoveCardCommand
-save : PositionState<Card, Column>
+MoveCardCommand(card : Card, direction : Direction)
+execute() : void
+undo() : void
+toString() : String
MoveColumnCommand
+MoveColumnCommand(column : Column, direction : Direction)
+toString() : String
delete
<<Abstract>>
DeleteCommand<E, P extends Container<E>>
-movable : Movable<E, P>
-save : PositionState<E, P>
+DeleteCommand(movable : Movable<E, P>)
+execute() : void
+undo() : void
+toString() : String
DeleteColumnCommand
+DeleteColumnCommand(column : Column)
DeleteCardCommand
+DeleteCardCommand(card : Card)
add
<<Abstract>>
AddCommand<E, P extends Container<E>>
-container : P
+AddCommand(container : P)
+execute() : void
+undo() : void
+toString() : String
AddCardCommand
+AddCardCommand(column : Column)
+toString() : String
AddColumnCommand
+AddColumnCommand(board : Board)
+toString() : String
<<Interface>>
Command
+execute()
+undo()
CommandManager
-commandManager : CommandManager
-undoCommands : Stack<Command>
-redoCommand : Stack<Command>
-undoHasNoCommand : BooleanProperty
-redoHasNoCommand : BooleanProperty
-undoCommandText : StringProperty
-redoCommandText : StringProperty
+CommandManager()
+getInstance() : CommandManager
+execute()
+undo()
+redo()
+undoHasNoCommandProperty() : BooleanProperty
+redoHasNoCommandProperty() : BooleanProperty
+undoCommandTextProperty() : StringProperty
+redoCommandTextProperty() : StringProperty
BoardViewModel
-columnsList : ListProperty<Column>
-boardTitle : StringProperty
-selectedColumn : IntegerProperty
-commandManager : CommandManager
+BoardViewModel(boardFacade : BoardFacade)
+boardTitleProperty() : StringProperty
+columnsListProperty() : ListProperty<Column>
+getColumn() : Column
+addColumn() : void
+delete() : void
+selectedColumnBinding(integerProperty : ReadOnlyIntegerProperty) : void
ColumnViewModel
-columnTitle : StringProperty
-selectedCard : IntegerProperty
-direction : ObjectProperty<Direction>
-commandManager : CommandManager
+ColumnViewModel(column : Column)
+cardsListProperty() : ListProperty<Card>
+columnTitleProperty() : StringProperty
+btRightDisabledProperty() : BooleanProperty
+btLeftDisabledProperty() : BooleanProperty
+selectedColumnBinding(integerProperty : ReadOnlyIntegerProperty) : void
+directionBinding(direction : ObjectProperty<Direction>)
+addCard() : void
+delete() : void
+getCard() : Card
CardViewModel
-cardTitle : StringProperty
-direction : ObjectProperty<Direction>
-commandManager : CommandManager
+CardViewModel(card : Card)
+bindMoveDirection(direction : ObjectProperty<Direction>) : void
+cardTitleProperty() : StringProperty
+btRightDisabledProperty() : BooleanProperty
+btLeftDisabledProperty() : BooleanProperty
+btUpDisabledProperty() : BooleanProperty
+btDownDisabledProperty() : BooleanProperty
MenuViewModel
-undoCommandText : StringProperty
-redoCommandText : StringProperty
-board : Board
-commandManager : CommandManager
+mRedoDisableProperty() : BooleanProperty
+mUndoDisableProperty() : BooleanProperty
+undoCommandTextProperty() : StringProperty
+redoCommandTextProperty() : StringProperty
+addColumn() : void
+undo() : void
+redo() : void
View
MenuViewModel
-mfile : Menu
-mEdit : Menu
-miNewColumn : MenuItem
-miExit : MenuItem
-miUndo : MenuItem
-miRedo : MenuItem
+MenuView(menuViewModel : MenuViewModel)
BoardView
-BACKGROUND_COLOR : Color
-BACKGROUND_COLUMN : Color
-FONT_COLOR : Color
-COLUMN_CELL_WIDTH : int
-hbList : HBox
-lvColumn : ListView<Column>
-region : Region
+BoardView(boardViewModel : BoardViewModel)
ColumnView
-BACKGROUND_COLOR : Color
-CARD_CELL_HEIGHT : int
-hbHeader : HBox
-btLeft : Button
-btRight : Button
-region : Region
-lvCards : ListView<Card>
-direction : ObjectProperty<Direction>
+ColumnView(columnViewModel : ColumnViewModel)
#ColumnView(column : Column)
CardView
-BACKGROUND_COLOR : Color
-btUP : Button
-btRight : Button
-btDown : Button
-btLeft : Button
-direction : ObjectProperty<Direction>
+CardView(cardViewModel : CardViewModel)
#CardView(card : Card)
TrelloView
-BACKGROUND_COLOR : Color
+TrelloView(primaryStage : Stage, boardViewModel : BoardViewModel)
EditableLabel
-backup : String
+EditableLabel()
+EditableLabel(str : String)
data
Data
+init() : Board
main
Main
+start(primaryStage : Stage) : void
+main(args : String[]) : void
JavaFx
HBox
TextField
<<Abstract>>
Application
VBox
ListView<Column>
ListView<Card>
BorderPane
Powered By Visual Paradigm Community Edition