

Chess layer

ChessException

+ ChessException(msg: String) : void

<<enum>>
Color

- BLACK : int
- WHITE : int

ChessPosition

- column : char
- row : int

toPosition() : Position
fromPosition(position : Position): ChessPosition

King

Bishop

Knight

Pawn

Queen

Rook

1

- chessMatch

1

- chessMatch

ChessMatch

- turn : int
- currentPlayer : Color
- check : boolean
- checkMate : boolean
- enPassantVulnerable : ChessPiece
- promoted : ChessPiece

+ getPieces() : ChessPiece[][]
+ possibleMoves(sourcePosition : ChessPosition) : boolean[][]
+ performChessMove(sourcePosition : ChessPosition, targetPosition: ChessPosition): ChessPiece
+ replacePromotedPiece(type : String) : ChessPiece

ChessPiece

- color: Color
- moveCount: int

+ getChessPosition(): ChessPosition
isThereOpponentPiece(positon : Position) : boolean
increaseMoveCount() : void
decreaseMoveCount() : void

1

- board

Board

- rows : int
- columns : int

+ pieces(row: int, columns : int) Piece
+ piece(position: Position) : Piece
+ placePiece(piece: Piece, position : Position) : void
+ removePiece(position : Position) : Piece
+ positionExists(position: Position) : boolean
+ thereIsAPiece(position: Position) : boolean

1

- board

Piece

position: Position

+ possibleMoves() : boolean [][]
+ possibleMove(position : Position) : boolean
+ isthereAnyPossibleMove() : boolean

- pieces

BoardException

+ BoardException(msg: String) : void

Position

- row : int
- column : int

+ setValues(row : int, column : int) : void