|  |
| --- |
| chessBoard |
| -cell [] [] : boardCells |
| +chessBoard()  +void: makeBoard()  +void: movePiece(int: rankIn, char: fileIn, int: rankOut, int: fileOut)  +chessPiece: getPiece(int: rankIn, char fileIn)  +Boolean: testSpecialCases()  String: toString |

|  |
| --- |
| cell |
| -int: rank  -char: file  -String cellDisplay  -chessPiece: currentPiece |
| +cell()  +cell(int: rankIn, char: fileIn, String: cellDisp)  +char: getFile()  +int: fetRank()  +void: setRank(int:rankIn)  +void: setFil(char: fileIn)  +void: setDisplay(String: cellDispIn)  +void: setPiece(chessPiece: pieceIn)  +chessPiece: removePiece()  +chessPiece: getPiece()  +boolean: equals(cell: cellIn)  +String: getDisplay()  +String: toString() |

|  |
| --- |
| Chess04 |
|  |
| +main(String[]) : void  +playGame(playArea: chessboard, Keyboard: Scanner, checkColor: colors, counter:long) |

|  |
| --- |
| <<Enum>>  colors |
|  |
| WHITE  BLACK |

|  |
| --- |
| *chessPiece* |
| ~String: displayChars  ~colors: pieceColor |
| *void: move(cell [][]: board, int: rankIn, char: fileIn, cell: end)*  *boolean: checkValidMove (int: rankIn, char: fileIn, cell: end)*  +String: getCharacters()  +colors:getColor()  +void: takePiece( chessPiece: takePiece)  Boolean:take (cell[] []: board, int:rankIn, char:fileIn, cell: end) |

|  |
| --- |
| Queen |
|  |
| +Queen(colors: colorIn)  +cell: take()  +void:move(cell[]:board, int: rankIn, char:fileIn, cell:end)  +boolean: checkValidMove(cell[]:board, int: rankIn, char:fileIn, cell:end) |

|  |
| --- |
| King |
|  |
| +King(colors: colorIn)  +cell: take()  +void:move(cell[]:board, int: rankIn, char:fileIn, cell:end)  +boolean: checkValidMove(cell[]:board, int: rankIn, char:fileIn, cell:end) |

|  |
| --- |
| Knight |
|  |
| +Knight(colors: colorIn)  +cell: take()  +void:move(cell[]:board, int: rankIn, char:fileIn, cell:end)  +boolean: checkValidMove(cell[]:board, int: rankIn, char:fileIn, cell:end) |

|  |
| --- |
| Bishop |
|  |
| +Bishop(colors: colorIn)  +cell: take()  +void:move(cell[]:board, int: rankIn, char:fileIn, cell:end)  +boolean: checkValidMove(cell[]:board, int: rankIn, char:fileIn, cell:end) |

|  |
| --- |
| Rook |
|  |
| +Rook(colors: colorIn)  +cell: take()  +void:move(cell[]:board, int: rankIn, char:fileIn, cell:end)  +boolean: checkValidMove(cell[]:board, int: rankIn, char:fileIn, cell:end) |

|  |
| --- |
| Pawn |
|  |
| +Pawn(colors: colorIn)  +cell: take()  +void:move(cell[]:board, int: rankIn, char:fileIn, cell:end)  +boolean: checkValidMove(cell[]:board, int: rankIn, char:fileIn, cell:end) |