

<<Java Class>> GUIController (default package)	
<ul style="list-style-type: none"> GUIController() <u>main(String[]):void</u> getBoard():SudokuBoard getGeneratedBoard():SudokuBoard getSolvedBoardArray():SudokuBoard setNewSudoku():void 	

<<Java Class>> GUI (default package)	
<ul style="list-style-type: none"> frame: JFrame pane: Container frameSize: Dimension gridInsets: Insets left: JPanel right: JPanel 	
<ul style="list-style-type: none"> GUI() initialise():void createLeft9x9():JPanel createRight9x9():JPanel updateRight9x9():JPanel updateLeft9x9():JPanel 	

<<Java Class>> GenerateSudoku (default package)	
<ul style="list-style-type: none"> NUMBER_ZERO: int MAX_NUM_COLUMNS: int MAX_NUM_ROWS: int 	
<ul style="list-style-type: none"> GenerateSudoku() getBoard():SudokuBoard generateRandomBoard():SudokuBoard 	

<<Java Class>> SudokuFileReader (default package)	
<ul style="list-style-type: none"> board: int[][] MAX_NUM_COLUMNS: int MAX_NUM_ROWS: int 	
<ul style="list-style-type: none"> SudokuFileReader() readInFile():SudokuBoard 	

<<Java Class>> SolveSudoku (default package)	
<ul style="list-style-type: none"> SolveSudoku() testSolver():void testNoSolutions():void recursiveBruteForceSolver(SudokuBoard):SudokuBoard noSolutions(SudokuBoard):int findEmptyCell(SudokuBoard):int[] getPossibilities(int,int,SudokuBoard):LinkedList<Integer> isValid(SudokuBoard):boolean <u>isValidSubGrid(SudokuBoard,int,int):boolean</u> <u>isValidColumn(SudokuBoard,int):boolean</u> <u>isValidRow(SudokuBoard,int):boolean</u> <u>isComplete(SudokuBoard):boolean</u> 	

<<Java Class>> SudokuBoard (default package)	
<ul style="list-style-type: none"> sBoard: int[][] MAX_NUM_COLUMNS: int MAX_NUM_ROWS: int 	
<ul style="list-style-type: none"> SudokuBoard() SudokuBoard(int[][]) printBoard():void getBoardArray():int[][] getCellNum(int,int):int setCellNum(int,int,int):void setZero(int,int,int):void copy():SudokuBoard 	

-gui
0..1

-solvedPuzzle 0..1 puzzle 0..1

-board
0..1