JavaScript is disabled on your browser.

[Skip navigation links](#1fob9te)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/Sudoku.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/sudoku/GraphicalUI.html)
* [Next Class](http://docs.google.com/sudoku/Utilities.html)
* [Frames](http://docs.google.com/index.html?sudoku/Sudoku.html)
* [No Frames](http://docs.google.com/Sudoku.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#3dy6vkm) |
* [Method](#2s8eyo1)

sudoku

## Class Sudoku

* java.lang.Object
  + sudoku.Sudoku
* public class Sudoku  
  extends java.lang.Object  
  This is the class for representing the model for Sudoku. Rows and columns are numbered 0-8. The value of blank square is zero, the value of a set square is 1-9.

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [Sudoku](http://docs.google.com/sudoku/Sudoku.html#Sudoku--)() Initializes a blank Sudoku |
| * + [Sudoku](http://docs.google.com/sudoku/Sudoku.html#Sudoku-sudoku.Sudoku-)([Sudoku](http://docs.google.com/sudoku/Sudoku.html) s) Initializes a Sudoku to be a copy (deep-copy) of an existing Sudoku |

### Method SummaryAll Methods Instance Methods Concrete Methods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + java.util.ArrayList<java.lang.Integer> | * + [getAllGoodMoves](http://docs.google.com/sudoku/Sudoku.html#getAllGoodMoves-int-int-)(int row, int column) Gets a collection of all (legal) GOOD moves for this square. |
| * + java.util.ArrayList<java.lang.Integer> | * + [getAllLegalMoves](http://docs.google.com/sudoku/Sudoku.html#getAllLegalMoves-int-int-)(int row, int column) Gets a collection of all legal moves for this square |
| * + int | * + [getAnyLegalMove](http://docs.google.com/sudoku/Sudoku.html#getAnyLegalMove-int-int-)(int row, int column) Returns any legal move for a square. |
| * + int | * + [getNumberLegalMoves](http://docs.google.com/sudoku/Sudoku.html#getNumberLegalMoves-int-int-)(int row, int column) Gets the number of legal moves for this square. |
| * + int | * + [getValue](http://docs.google.com/sudoku/Sudoku.html#getValue-int-int-)(int row, int column) Gets the value of an entry in a Sudoku |
| * + boolean | * + [isLegalMove](http://docs.google.com/sudoku/Sudoku.html#isLegalMove-int-int-int-)(int row, int column, int value) Determines if a particular move is legal, looking at other squares in the same row, column and 3x3 square. |
| * + void | * + [setValue](http://docs.google.com/sudoku/Sudoku.html#setValue-int-int-int-)(int row, int column, int value) Sets a square of the Sudoku to a specific value. |
| * + java.lang.String | * + [toString](http://docs.google.com/sudoku/Sudoku.html#toString--)() Returns a string representation of the Sudoku board. |

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, wait, wait, wait

### Constructor Detail

#### Sudoku public Sudoku() Initializes a blank Sudoku

#### Sudoku public Sudoku([Sudoku](http://docs.google.com/sudoku/Sudoku.html) s) Initializes a Sudoku to be a copy (deep-copy) of an existing SudokuParameters: s - Sudoku to copy

### Method Detail

#### getValue public int getValue(int row, int column) Gets the value of an entry in a SudokuParameters: row - row to retrieve (0-8) column - column to retrieve (0-8) Returns: 0 if entry is empty, 1-9 otherwise.

#### setValue public void setValue(int row, int column, int value) Sets a square of the Sudoku to a specific value.Parameters: row - row to set (0-8) column - column to set (0-8) value - (1-9) Throws: java.lang.IllegalArgumentException - is the square is already filled in or if it immediately conflicts with a value in same row, column or square.

#### getNumberLegalMoves public int getNumberLegalMoves(int row, int column) Gets the number of legal moves for this square. This method will only be invoked on blank squares.Parameters: row - row of entry column - column of entry Returns: the number of legal moves (0-9)

#### getAllLegalMoves public java.util.ArrayList<java.lang.Integer> getAllLegalMoves(int row, int column) Gets a collection of all legal moves for this squareParameters: row - row of entry column - column of entry Returns: a collection of all of the legal moves

#### getAllGoodMoves public java.util.ArrayList<java.lang.Integer> getAllGoodMoves(int row, int column) Gets a collection of all (legal) GOOD moves for this square. Applies only to the honors section. Students in the honor section will provide code that finds better moves than simply all legal moves. Students in the other sections can just return all the legal moves.Parameters: row - row of entry column - column of entry Returns: a collection of all of the good moves

#### isLegalMove public boolean isLegalMove(int row, int column, int value) Determines if a particular move is legal, looking at other squares in the same row, column and 3x3 square.Parameters: row - row of entry column - column of entry value - value of move being considered Returns: true if the square is blank and move is legal

#### getAnyLegalMove public int getAnyLegalMove(int row, int column) Returns any legal move for a square. Return 0 if no legal moves existParameters: row - row for entry column - column for entry Returns: an arbitrary legal move for the square, or 0 if no legal moves exist

#### toString public java.lang.String toString() Returns a string representation of the Sudoku board. This should consist of 9 lines of 9 characters each, with a . for a blank square or a digit for an occupied square.Overrides: toString in class java.lang.Object

[Skip navigation links](#1y810tw)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/Sudoku.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/sudoku/GraphicalUI.html)
* [Next Class](http://docs.google.com/sudoku/Utilities.html)
* [Frames](http://docs.google.com/index.html?sudoku/Sudoku.html)
* [No Frames](http://docs.google.com/Sudoku.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#3dy6vkm) |
* [Method](#2s8eyo1)