Andrea Siejna

CS232

Dr. Jump

December 12th, 2014

Project 3: Repeatedly Counting 1 to 9

**public class Sudoku implements Iterable<Integer>**

\* Value that represents an empty square on the Sudoku board.

**private static final int ILLEGAL = 0;**

\* An empty board of Sudoku as a 2D array.

**Integer[][] sudokuArray = new Integer[9][9];**

\* Fills in the Sudoku board with pre-set values.

**public void filledBoard(Integer[][] newSudoku){}**

\* Fills in the Sudoku board with illegal values

**public void defaultBoard(){}**

\* Prints the Sudoku board.

**public void printBoard() {}**

\* Sets a square with the given value.

**public void setSquare(int row, int col, int value) throws NoSuchElementException {}**

\* Gets the cell at the given row and column.

**public int getSquare(int row, int col) throws NoSuchElementException {}**

\* If every cell is a legal move, then the puzzle itself is legal.

**public boolean isLegalPuzzle() {}**

\* Checks to see if the cell is a legal move in the row.

**public boolean isLegalRowMove(int row, int col) {}**

\* Checks to see if the cell is a legal move in the col.

**public boolean isLegalColMove(int row, int col) {}**

\* Checks to see if the cell is a legal move in the block.

**public boolean isLegalBlockMove(int row, int col) {}**

\* Checks to see if the cell is a legal move.

**public boolean isLegalMove(int row, int col) {}**

\* Find a solution starting from the given row and column working in

\* row-major order.

**public boolean solveSudoku(int r, int c) {}**

\* Returns an iterator over the entire Sudoku board.

**public Iterator<Integer> iterator() {}**

\* Returns an iterator over an entire row of the Sudoku board.

**public Iterator<Integer> rowIterator(int currentRow) {}**

\* Returns an iterator over an entire column of the Sudoku board.

**public Iterator<Integer> colIterator(int currentCol) {}**

\* Returns an iterator over an entire block of the Sudoku board.

**public Iterator<Integer> blockIterator(int givenBlock) {}**

**// INNER CLASS ITERATORS**

**public class Row implements Iterator<Integer> {}**

**public class Column implements Iterator<Integer> {}**

**public class Block implements Iterator<Integer> {}**

**public class Board implements Iterator<Integer> {}**

each will have implemented methods **next(), hasNext(), remove();**