```
public class project3 {
  private static final int SIZE = 9;

  public static void main(String[] args) {

    int[][] sudoku = {
        { 5, 3, 0, 0, 7, 0, 0, 0, 0 },
        { 6, 0, 0, 1, 9, 5, 0, 0, 0 },
        { 0, 9, 8, 0, 0, 0, 0, 6, 0 },
        { 8, 0, 0, 0, 6, 0, 0, 0, 3 },
        { 4, 0, 0, 8, 0, 3, 0, 0, 1 },
        { 7, 0, 0, 0, 2, 0, 0, 0, 6 },
        { 0, 6, 0, 0, 0, 0, 2, 8, 0 },
        { 0, 0, 0, 4, 1, 9, 0, 0, 5 },
        { 0, 0, 0, 0, 8, 0, 0, 7, 9 }
    };

    if (solveSudoku(sudoku)) {
      printSudoku(sudoku);
    } else {
      System.out.println("No solution exists.");
    }

  }

  private static boolean solveSudoku(int[][] sudoku) {
    for (int row = 0; row < SIZE; row++) {
      for (int col = 0; col < SIZE; col++) {
        if (sudoku[row][col] == 0) {
          for (int num = 1; num <= SIZE; num++) {
            if (isSafe(sudoku, row, col, num)) {
              sudoku[row][col] = num;
              if (solveSudoku(sudoku)) {
                return true;
              }
              sudoku[row][col] = 0; // Backtrack if placing the number doesn't lead to a solution
            }
          }
          return false; // If no number can be placed, backtrack
        }
      }
    }
    return true; // Puzzle solved
  }

  private static boolean isSafe(int[][] sudoku, int row, int col, int num) {
    return !usedInRow(sudoku, row, num) &&
        !usedInCol(sudoku, col, num) &&
        !usedInBox(sudoku, row - row % 3, col - col % 3, num);
  }

  private static boolean usedInRow(int[][] sudoku, int row, int num) {
    for (int col = 0; col < SIZE; col++) {
      if (sudoku[row][col] == num) {
        return true;
```

```
      }
    }
    return false;
  }

  private static boolean usedInCol(int[][] sudoku, int col, int num) {
    for (int row = 0; row < SIZE; row++) {
      if (sudoku[row][col] == num) {
        return true;
      }
    }
    return false;
  }

  private static boolean usedInBox(int[][] sudoku, int startRow, int startCol, int num) {
    for (int i = 0; i < 3; i++) {
      for (int j = 0; j < 3; j++) {
        if (sudoku[startRow + i][startCol + j] == num) {
          return true;
        }
      }
    }
    return false;
  }

  private static void printSudoku(int[][] sudoku) {
    for (int i = 0; i < SIZE; i++) {
      for (int j = 0; j < SIZE; j++) {
        System.out.print(sudoku[i][j] + " ");
      }
      System.out.println();
    }
  }
}
```