```
import java.util.Scanner;


public class Sudoku {
   private static final int SIZE = 9; // Size of the Sudoku grid
   private static final int[][] board = {
         {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}
   }; // Sample Sudoku board


   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);


      while (!isBoardFull()) {
         printBoard();
         System.out.print("Enter row (1-9), column (1-9), and number (1-9)
separated by spaces: ");
         int row = scanner.nextInt() - 1; // Convert to zero-indexed
```

```
        int col = scanner.nextInt() - 1; // Convert to zero-indexed
        int num = scanner.nextInt();

        if (isValidMove(row, col, num)) {
            board[row][col] = num;
        } else {
            System.out.println("Invalid move! Try again.");
        }
    }

    printBoard();
    System.out.println("Congratulations! You have completed the Sudoku puzzle.");
    scanner.close();
  }

  // Method to print the Sudoku board
  private static void printBoard() {
    for (int i = 0; i < SIZE; i++) {
        if (i % 3 == 0 && i != 0) {
            System.out.println("---------------------");
        }
        for (int j = 0; j < SIZE; j++) {
            if (j % 3 == 0 && j != 0) {
                System.out.print("| ");
```

```
            }
            System.out.print(board[i][j] == 0 ? ". " : board[i][j] + " ");
        }
        System.out.println();
    }
}

// Method to check if a move is valid
private static boolean isValidMove(int row, int col, int num) {
    return board[row][col] == 0 && !isInRow(row, num) && !isInCol(col, num) &&
!isInBox(row, col, num);
}

// Method to check if the number is already in the row
private static boolean isInRow(int row, int num) {
    for (int col = 0; col < SIZE; col++) {
        if (board[row][col] == num) {
            return true;
        }
    }
    return false;
}

// Method to check if the number is already in the column
private static boolean isInCol(int col, int num) {
```

```
    for (int row = 0; row < SIZE; row++) {
        if (board[row][col] == num) {
            return true;
        }
    }
    return false;
}

// Method to check if the number is already in the 3x3 box
private static boolean isInBox(int row, int col, int num) {
    int boxRowStart = row - row % 3;
    int boxColStart = col - col % 3;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (board[boxRowStart + i][boxColStart + j] == num) {
                return true;
            }
        }
    }
    return false;
}

// Method to check if the board is full
private static boolean isBoardFull() {
```

```
        for (int i = 0; i < SIZE; i++) {
            for (int j = 0; j < SIZE; j++) {
                if (board[i][j] == 0) {
                    return false;
                }
            }
        }
        return true;
    }
}
```