```java
import java.util.Scanner;

public class Main {
    private static char[][] board;
    private static char currentPlayer;
    private static boolean isGameFinished;

    public static void main(String[] args) {
        initializeGame();
        playGame();
    }

    private static void initializeGame() {
        board = new char[3][3];
        currentPlayer = 'X';
        isGameFinished = false;

        // Initialize the board with empty spaces
        for (int row = 0; row < 3; row++) {
            for (int col = 0; col < 3; col++) {
                board[row][col] = ' ';
            }
        }
    }

    private static void playGame() {
        System.out.println("Welcome to Tic-Tac-Toe!");
        System.out.println("Player 1: X | Player 2: O");
        System.out.println("Let's start the game!");

```

```
31          while (!isGameFinished) {
32              printBoard();
33              makeMove();
34              checkWinCondition();
35              checkDrawCondition();
36              switchPlayer();
37          }
38      }
39
40      private static void printBoard() {
41          System.out.println("---------");
42          for (int row = 0; row < 3; row++) {
43              System.out.print("| ");
44              for (int col = 0; col < 3; col
    ++) {
45                  System.out.print(board[row][
    col] + " | ");
46              }
47              System.out.println("\n---------"
    );
48          }
49      }
50
51      private static void makeMove() {
52          Scanner scanner = new Scanner(System.
    in);
53          int row, col;
54
55          do {
56              System.out.print("Player " +
    currentPlayer + ", enter the row (1-3): ");
57              row = scanner.nextInt() - 1;
58              System.out.print("Player " +
    currentPlayer + ", enter the column (1-3): "
    );
```

```
59              col = scanner.nextInt() - 1;
60          } while (!isValidMove(row, col));
61
62          board[row][col] = currentPlayer;
63      }
64
65      private static boolean isValidMove(int
    row, int col) {
66          if (row < 0 || row >= 3 || col < 0
     || col >= 3) {
67              System.out.println("Invalid move
    ! Please enter row and column values between
    1 and 3.");
68              return false;
69          }
70          if (board[row][col] != ' ') {
71              System.out.println("Invalid move
    ! The cell is already occupied.");
72              return false;
73          }
74          return true;
75      }
76
77      private static void checkWinCondition() {
78          if (checkRows() || checkColumns() ||
    checkDiagonals()) {
79              isGameFinished = true;
80              printBoard();
81              System.out.println("Player " +
    currentPlayer + " wins! Congratulations!");
82          }
83      }
84
85      private static boolean checkRows() {
86          for (int row = 0; row < 3; row++) {
```

```
 87              if (board[row][0] ==
    currentPlayer && board[row][1] ==
    currentPlayer && board[row][2] ==
    currentPlayer) {
 88                  return true;
 89              }
 90          }
 91          return false;
 92      }
 93
 94      private static boolean checkColumns() {
 95          for (int col = 0; col < 3; col++) {
 96              if (board[0][col] ==
    currentPlayer && board[1][col] ==
    currentPlayer && board[2][col] ==
    currentPlayer) {
 97                  return true;
 98              }
 99          }
100          return false;
101      }
102
103      private static boolean checkDiagonals
    () {
104          if ((board[0][0] == currentPlayer
     && board[1][1] == currentPlayer && board[2
    ][2] == currentPlayer) ||
105                  (board[0][2] ==
    currentPlayer && board[1][1] ==
    currentPlayer && board[2][0] ==
    currentPlayer)) {
106              return true;
107          }
108          return false;
109      }
```

```
110
111     private static void checkDrawCondition
    () {
112         if (!isGameFinished) {
113             boolean isBoardFull = true;
114             for (int row = 0; row < 3; row
    ++) {
115                 for (int col = 0; col < 3;
    col++) {
116                     if (board[row][col] ==
    ' ') {
117                         isBoardFull = false;
118                         break;
119                     }
120                 }
121             }
122             if (isBoardFull) {
123                 isGameFinished = true;
124                 printBoard();
125                 System.out.println("It's a
    draw! The game is over.");
126             }
127         }
128     }
129
130     private static void switchPlayer() {
131         if (currentPlayer == 'X') {
132             currentPlayer = 'O';
133         } else {
134             currentPlayer = 'X';
135         }
136     }
137 }
138
```