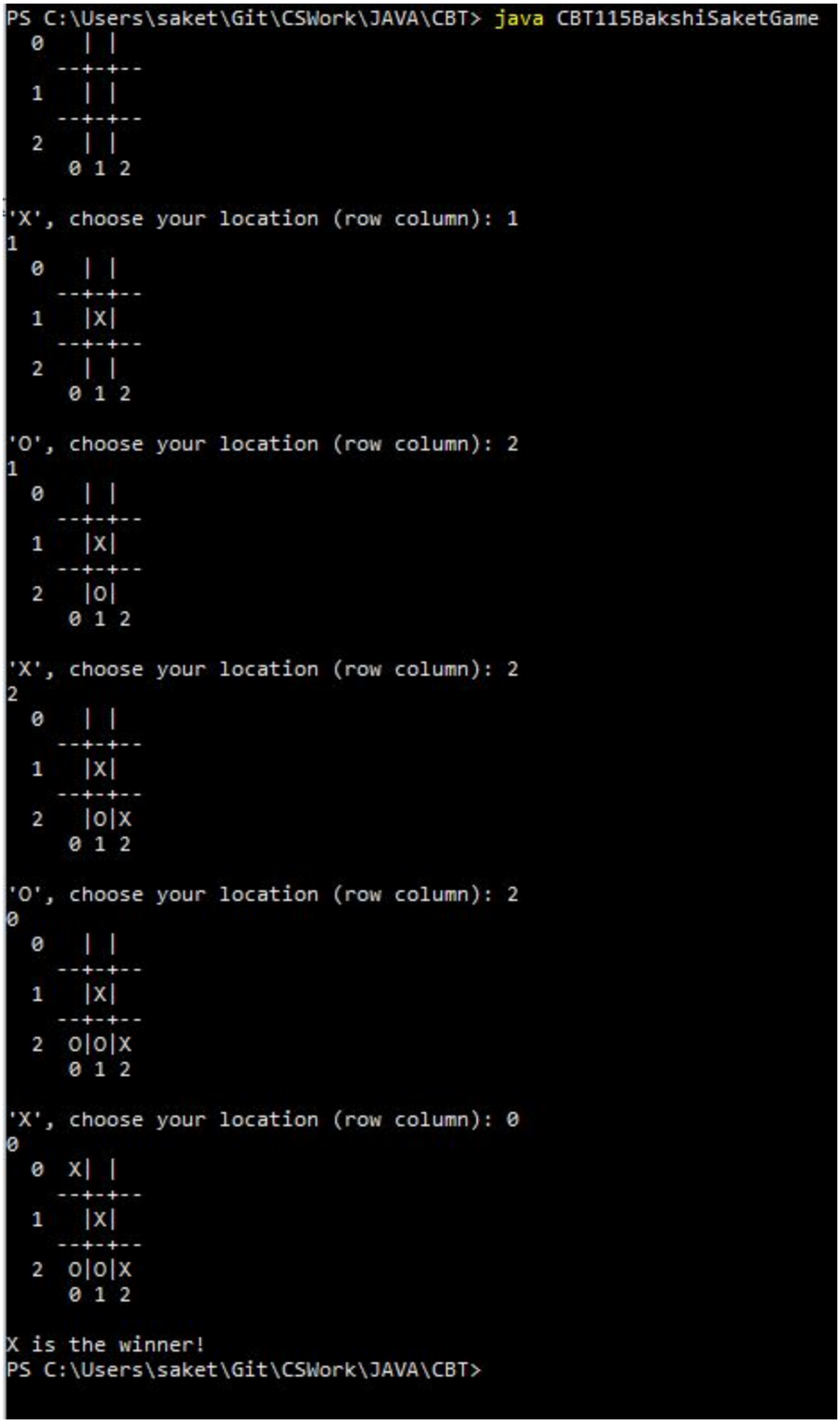


```
public class CBT115BakshiSaket
{
        private String[][] board;
```

```
	private int turns;

	public CBT115BakshiSaket()
	{
		this.board = new String[3][3];
		this.turns = 0;

		for(int a = 0; a < 3; a++)
			for(int b = 0; b < 3; b++)
				this.board[a][b] = " ";
	}

	public boolean isWinner( String p) {
   // top row
   if ( winCheck(p, 0,0, 0,1, 0,2) ) return true;
   // middle row
   if ( winCheck(p, 1,0, 1,1, 1,2) ) return true;
   // bottom row
   if ( winCheck(p, 2,0, 2,1, 2,2) ) return true;
   // left column
   if ( winCheck(p, 0,0, 1,0, 2,0) ) return true;
   // middle column
   if ( winCheck(p, 0,1, 1,1, 2,1) ) return true;
   // right column
   if ( winCheck(p, 0,2, 1,2, 2,2) ) return true;
   // diagonal top-left to bottom-right
   if ( winCheck(p, 0,0, 1,1, 2,2) ) return true;
   // diagonal bottom-left to top-right
   if ( winCheck(p, 2,0, 1,1, 0,2) ) return true;

   return false;
	}

	private boolean winCheck(String p, int a, int b, int c, int d, int e, int f) {
	return board[a][b].equals(board[c][d]) && board[a][b].equals(board[e][f]) &&
board[a][b].equals(p);
   }

	public boolean isFull()
	{
	if ( turns == 9 )
		return true;
		else
```

```
                    return false;
            // return turns == 9;
}

public boolean isCat() {
return isFull() && !isWinner("X") && !isWinner("O");
}

public boolean isValid( int r, int c) {
if( 0 <= r && r <= 2 && 0 <= c && c <= 2)
            return true;
else
            return false;
}

public int numTurns() {
return turns;
}

public String playerAt( int r, int c) {
if ( isValid(r,c) )
            return board[r][c];
else
            return "@";
}

public boolean isTaken(int r, int c) {
String p = playerAt(r,c);
if ( p.equals(" ") )
            return false;
else
            return true;
}

public String toString() {
String out = "";
out += "  0  " + board[0][0] +"|"+ board[0][1] +"|"+ board[0][2] +"\n";
out += "    --+-+--" + "\n";
out += "  1  " + board[1][0] +"|"+ board[1][1] +"|"+ board[1][2] +"\n";
out += "    --+-+--" + "\n";
out += "  2  " + board[2][0] +"|"+ board[2][1] +"|"+ board[2][2] +"\n";
out += "     0 1 2 " + "\n";
return out;
```

```
        }

  // Modifier / Mutator Method
        public void playMove( String p, int r, int c ) {
        board[r][c] = p;
        turns++;
        }
}


import java.util.Scanner;

public class CBT115BakshiSaketGame
{
        public static void main(String[] args) {
                Scanner keyboard = new Scanner(System.in);

                String p = "X";
                CBT115BakshiSaket ttt = new CBT115BakshiSaket();
                int r, c;

        while ( ! (ttt.isWinner("X") || ttt.isWinner("O") || ttt.isFull() ) ) {
                System.out.println(ttt);
                System.out.print("'" + p + "', choose your location (row column): ");
                r = keyboard.nextInt();
                c = keyboard.nextInt();

                while ( ! ttt.isValid(r,c) || ttt.isTaken(r,c) ) {
                if ( ttt.isValid(r,c) == false )
                        System.out.println("Not a valid location. Try again.");
                else if ( ttt.isTaken(r,c) )
                        System.out.println("Location already full. Try again.");

                System.out.print( "Choose your location (row column): ");
        r = keyboard.nextInt();
        c = keyboard.nextInt();
        }
        ttt.playMove( p, r, c );

        if ( p.equals("X") )
        p = "O";
        else
        p = "X";
```

```
    }

    System.out.println(ttt);

    if ( ttt.isWinner("X") )
                    System.out.println("X is the winner!");
    else if ( ttt.isWinner("O") )
            System.out.println("O is the winner!");
    else if ( ttt.isCat() )
            System.out.println("The game is a tie.");
            }
}
```