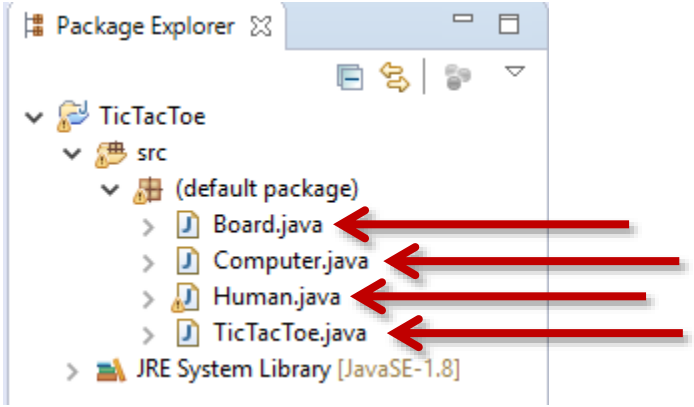
Solution 2: Tic Tac Toe

Step 1: Create a Project

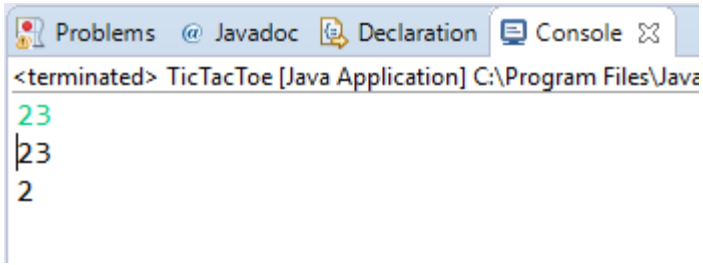



```
public class TicTacToe {

    public static void main(String[] args) {

        System.out.println("Here");

    }

}
```

Making Moves

```
public class TicTacToe {

    public static void main(String[] args) {

        Board board = new Board();
        Human human = new Human();
        Computer computer = new Computer();

        int move = human.getMove();
        board.makeMove(move);

        move = computer.getMove();
        board.makeMove(move);

    }

}
```

```
public class Board {

    public void makeMove(int move) {
        System.out.println(move);
    }
}
```



The Game Board

```
public class Board {

    public void makeMove(int move) {
        System.out.println(move);
    }

    public char getCell(int c) {
        return ' ';
    }

    public void printBoard() {
        System.out.println("X| | ");
        System.out.println("-+-+-");
        System.out.println("O|O| ");
        System.out.println("-+-+-");
        System.out.println("X| | ");
    }

}
```

```
Board board = new Board();
Human human = new Human();
Computer computer = new Computer();

int move = human.getMove();
board.makeMove(move);
board.printBoard();

move = computer.getMove();
board.makeMove(move);
board.printBoard();
```

```
<terminated> TicTacToe [Jav
22
22
X| |
-+-+-
O|O|
-+-+-
X| |
2
```

The Game Board

```
public char getCell(int c) {
    return ' ';
}

public void printBoard() {
    System.out.println(getCell(0)+"|"+getCell(1)+"|"+getCell(2));
    System.out.println("-+-+-");
    System.out.println(getCell(3)+"|"+getCell(4)+"|"+getCell(5));
    System.out.println("-+-+-");
    System.out.println(getCell(6)+"|"+getCell(7)+"|"+getCell(8));
}
```

The Game Board

```
public class Board {

    private int cell0;
    private int cell1;
    private int cell2;

    private char cell3;
    private char cell4;

    private int[] cells; // 0=empty, 1=X, 2=O

    public Board() {
        cells = new int[9];
    }

...

}
```

The Game Board

```
public class Board {

    private int[] cells; // 0=empty, 1=X, 2=O

    public Board() {
        cells = new int[9];
    }

    public void makeMove(int move, char player) {
        if(player=='X') cells[move]=1;
        else if(player=='O') cells[move]=2;
    }

    public char getCell(int c) {
        if(cells[c]==1) return 'X';
        if(cells[c]==2) return 'O';
        return ' ';
    }
```

Better Game Loop

```
System.out.println("");
board.printBoard();
System.out.println("Player X's turn.");
int move = human.getMove();
board.makeMove(move,'X');

System.out.println();
board.printBoard();
System.out.println("Player O's turn.");
move = computer.getMove();
board.makeMove(move,'O');
```

```
<terminated> TicTacToe [Java Appl

 | |
-+-+-
 | |
-+-+-
 | |
Player X's turn.
2

 | |X
-+-+-
 | |
-+-+-
 | |
Player O's turn.
```

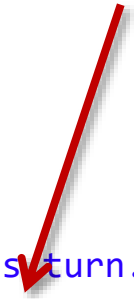
Two Humans

```
Board board = new Board();
Human player1 = new Human();
//Computer player2 = new Computer();
Human player2 = new Human();

while(true) {
    System.out.println("");
    board.printBoard();
    System.out.println("Player X's turn.");
    int move = player1.getMove();
    board.makeMove(move,'X');

    System.out.println("");
    board.printBoard();
    System.out.println("Player O's turn.");
    move = player2.getMove();
    board.makeMove(move,'O');
}
```

Finish the Human

```
public int getMove(Board board) {

    while(true) {
        System.out.println("Pick a cell 0 - 8.");
        int move = getInteger();
        if(move>=0 && move<=8 && board.getCell(move)==' ') {
            return move;
        }
        System.out.println("Invalid cell. Try again.");
    }

}
```

```
System.out.println("");
board.printBoard();
System.out.println("Player X's turn.");
int move = player1.getMove(board);
board.makeMove(move,'X');
```



Computer Player

```
public class Computer {

    private char token;

    public Computer(char tok) {
        token = tok;
    }

    public static int getRandom() {
        Random rand = new Random();
        return rand.nextInt(9);
    }

    public int getMove(Board board) {
        while(true) {
            int move = getRandom();
            if(board.getCell(move)==' ') {
                return move;
            }
        }
    }
}
```

```
Board board = new Board();
Human player1 = new Human();
Computer player2 = new Computer('O');
//Human player2 = new Human();
```

Who Won?

```
    public char getStatus() {

        if(cells[0]==1 && cells[1]==1 && cells[2]==1) return 'X';
        if(cells[3]==1 && cells[4]==1 && cells[5]==1) return 'X';
...
        if(cells[0]==2 && cells[4]==2 && cells[8]==2) return 'O';
        if(cells[2]==2 && cells[4]==2 && cells[6]==2) return 'O';

        if(cells[0]>0 && cells[1]>0 && cells[2]>0 &&
                cells[3]>0 && cells[4]>0 && cells[5]>0 &&
                cells[6]>0 && cells[7]>0 && cells[8]>0)
        {
                 return 'C'; // CAT (tie)
        }

        return ' '; // BLANK - in progress
    }
```

Who Won?

```
public char makeMove(int move, char player) {

    if(player=='X') cells[move]=1;
    else if(player=='O') cells[move]=2;

    return getStatus();

}
```

Final Game Loop

```
char endResult = ' ';
while(true) {
    System.out.println("");
    board.printBoard();
    System.out.println("Player X's turn.");
    int move = player1.getMove(board);
    endResult = board.makeMove(move,'X');
    if(endResult!=' ') break;

    System.out.println("");
    board.printBoard();
    System.out.println("Player O's turn.");
    move = player2.getMove(board);
    endResult = board.makeMove(move,'O');
    if(endResult!=' ') break;
}

System.out.println("The winner is: "+endResult);
board.printBoard();
```

Picking Players

```
Board board = new Board();

//Human player1 = new Human();
Computer player1 = new Computer('X');

//Human player2 = new Human();
Computer player2 = new Computer('O');
```