**Tic Tac Toe game in java.**

**Solution:**

**import** java.util.\*;

**public** **class** tictactoe {

**static** HashSet<Integer> *user\_set*=**new** HashSet<Integer>();

**static** HashSet<Integer> *computer\_set*=**new** HashSet<Integer>();

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**char** [][] board= {

{' ','|',' ','|',' '},

{'-','|', '-','|','-'},

{' ','|',' ','|',' '},

{'-','|', '-','|','-'},

{' ','|',' ','|',' '}

};

*display*(board);

Scanner s1=**new** Scanner(System.***in***);

**while**(**true**)

{

System.***out***.println("enter values from 1-9:");

**int** user\_pos=s1.nextInt();

**if**(*user\_set*.contains(user\_pos)||*computer\_set*.contains(user\_pos))

{

System.***out***.print("Retry, enter values from 1-9:");

user\_pos=s1.nextInt();

}

*holder*(board,user\_pos,"neha");

String res=*check\_winner*();

**if**(res.length()>0)

{

System.***out***.println(res);

**break**;

}

**int** computer\_pos= *generate\_random*();

**if**(*user\_set*.contains(computer\_pos)||*computer\_set*.contains(computer\_pos))

{

computer\_pos=*generate\_random*();

}

*holder*(board,computer\_pos,"computer");

res=*check\_winner*();

**if**(res.length()>0)

{

System.***out***.println(res);

**break**;

}

}

}

**static** String check\_winner()

{

HashSet<Integer>r1=**new** HashSet<Integer>();

r1.add(1);

r1.add(2);

r1.add(3);

HashSet<Integer>r2=**new** HashSet<Integer>();

r2.add(4);

r2.add(5);

r2.add(6);

HashSet<Integer>r3=**new** HashSet<Integer>();

r3.add(7);

r3.add(8);

r3.add(9);

HashSet<Integer>c1=**new** HashSet<Integer>();

c1.add(1);

c1.add(4);

c1.add(7);

HashSet<Integer>c2=**new** HashSet<Integer>();

c2.add(2);

c2.add(5);

c2.add(8);

HashSet<Integer>c3=**new** HashSet<Integer>();

c3.add(3);

c3.add(6);

c3.add(9);

HashSet<Integer>d1=**new** HashSet<Integer>();

d1.add(1);

d1.add(5);

d1.add(9);

HashSet<Integer>d2=**new** HashSet<Integer>();

d2.add(3);

d2.add(5);

d2.add(7);

HashSet<HashSet> set=**new** HashSet<HashSet>();

set.add(r1);

set.add(r2);

set.add(r3);

set.add(c1);

set.add(c2);

set.add(c3);

set.add(d1);

set.add(d2);

**for**(HashSet X:set)

{

**if**(*user\_set*.containsAll(X))

**return** "neha won";

**else** **if**(*computer\_set*.containsAll(X))

**return** "neha lose";

}

**if**(*user\_set*.size()+*computer\_set*.size()==9)

**return** "Draw";

**return** "";

}

**static** **int** generate\_random()

{

**int** max=9;

**int** min=1;

**int** range=max-min+1;

**int** result=(**int**)(Math.*random*()\*range)+min;

**return** result;

}

**static** **void** display(**char**[][] board)

{

**for**(**int** r=0;r<board.length;r++)

{

**for**(**int** c=0;c<board[0].length;c++)

{

System.***out***.print(board[r][c]);

}

System.***out***.println();

}

}

**static** **void** holder(**char**[][]board,**int** pos,String user)

{

**char** s='X';

**if**(user.equals("neha"))

{

s='X';

*user\_set*.add(pos);

}

**else** **if**(user.equals("computer"))

{

s='0';

*computer\_set*.add(pos);

}

**switch**(pos)

{

**case** 1: board[0][0]= s;

**break**;

**case** 2: board[0][2]= s;

**break**;

**case** 3: board[0][4]= s;

**break**;

**case** 4: board[2][0]= s;

**break**;

**case** 5: board[2][2]= s;

**break**;

**case** 6: board[2][4]= s;

**break**;

**case** 7: board[4][0]= s;

**break**;

**case** 8: board[4][2]= s;

**break**;

**case** 9: board[4][4]= s;

**break**;

**default**: System.***out***.println("invalid input");

}

System.***out***.println();

*display*(board);

}

}