package tictactoe;

import java.util.Scanner;

import java.util.Random;

import java.util.List;

import java.util.ArrayList;

import java.util.Arrays;

public class TicTacToe {

//All the methods for operating the game smoothly are defined by the methods which are in this class:

static ArrayList<Integer> playerPositions= new ArrayList<Integer>(); //Creating an ArrayList for storing winning positions of the board for user inputs

static ArrayList<Integer> cpuPositions= new ArrayList<Integer>(); //Creating an ArrayList for storing winning positions of the board for CPU random entries

//Creating a function named PGameBoard()which prints the empty game board using nested for loop

public static void PGameBoard(char[][] gameBoard)

{

for(char[] row:gameBoard){

for(char c:row)

{

System.out.print(c);

}

System.out.println();}

}

//Creating a function named Placement() which places the symbol(O or X) on the desired position of game board and also switch between user and CPU turn

public static void Placement(char [][] gameBoard,int pos,String user)

{

//switching between user and CPU turns and also adding the occupied places of board to the ArrayList to help identify the winner

char symbol=' ';

if(user.equals("Player")){

symbol='X';

playerPositions.add(pos);}

else if(user.equals("CPU")){

symbol='O';

cpuPositions.add(pos);}

//putting the placement on gameBoard by using switch case:

switch(pos){

case 1:

gameBoard[0][0]=symbol;

break;

case 2:

gameBoard[0][2]=symbol;

break;

case 3:

gameBoard[0][4]=symbol;

break;

case 4:

gameBoard[2][0]=symbol;

break;

case 5:

gameBoard[2][2]=symbol;

break;

case 6:

gameBoard[2][4]=symbol;

break;

case 7:

gameBoard[4][0]=symbol;

break;

case 8:

gameBoard[4][2]=symbol;

break;

case 9:

gameBoard[4][4]=symbol;

break;

default:

break;}

}

// Creating a function which will check if there is a winner or a tie using lists,List of lists and Array lists

// this function is copied from a youtube video https://www.youtube.com/watch?v=gQb3dE-y1S4

public static String CheckWin()

{

//Each of the following list shows the winning coonditions of board filling

List topR=Arrays.asList(1,2,3);

List midR=Arrays.asList(4,5,6);

List endR=Arrays.asList(7,8,9);

List topC=Arrays.asList(1,4,7);

List midC=Arrays.asList(2,5,8);

List endC=Arrays.asList(3,6,9);

List cross1=Arrays.asList(1,5,9);

List cross2=Arrays.asList(7,5,3);

//adding all those lists to a single list of list so that the operation will be easier

List<List> winning=new ArrayList<List>();

winning.add(topR);

winning.add(midR);

winning.add(endR);

winning.add(topC);

winning.add(midC);

winning.add(endC);

winning.add(cross1);

winning.add(cross2);

//checking the winner or tie

for(List l: winning){

if(playerPositions.containsAll(l)){

return "Congratulations!! You Won...";}

else if(cpuPositions.containsAll(l)){

return "Sorry!! CPU Won...";}

else if(playerPositions.size()+cpuPositions.size()==9){

return "A tie...";}}

return "";

}

//Creating main function which calls all the ablove functions,creates an empty game board, takes user input and print the results

public static void main(String[] args)

{

//initializing a 2D array for creating an empty game board

char[][] gameBoard= {{' ','|',' ','|',' '},

{'-','+','-','+','-'},

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

{'-','+','-','+','-'},

{' ','|',' ','|',' '}}; //creating a game board from 2D array is copied from a youtubr video https://www.youtube.com/watch?v=gQb3dE-y1S4

PGameBoard(gameBoard);//calling the function to print(create) the game board

//taking input of user placement and also allowing CPU to randomly put its symbol 'O':

while(true)

{

Scanner scan=new Scanner(System.in);

System.out.println("enter your placement(1-9): ");

int playerPos=scan.nextInt(); //user input for placement

while(playerPositions.contains(playerPos)||cpuPositions.contains(playerPos)) //contains method functionality is copied from a youtube video https://www.youtube.com/watch?v=gQb3dE-y1S4

{

System.out.println("Place occupied!! choose the correct position.");

playerPos=scan.nextInt();

}

Placement(gameBoard,playerPos,"Player"); //calling the function to put the symbol of player on game board according to the input position

String result=CheckWin();

if(result.length()>0){

System.out.println("--------------------------\n"+result+"\n--------------------------");

break;}

Random rand=new Random(); //random CPU input for placement

int CPUpos=rand.nextInt(9)+1;

while(playerPositions.contains(CPUpos)||cpuPositions.contains(CPUpos))

{

CPUpos=rand.nextInt(9)+1;

}

Placement(gameBoard,CPUpos,"CPU");//calling the function to put the symbol of CPU on game board according to the random function position

PGameBoard(gameBoard);

result=CheckWin();

if(result.length()>0){ //checking whether there is a winner

System.out.println("--------------------------\n"+result+"\n--------------------------");

break;}

}

}

}