```java
/*
CSE017 Fall 2019
@Bratislav Petkovic
Homework # 2
Program: Team
*/
import java.text.DecimalFormat;
import java.util.* ;
import java.util.ArrayList;
import java.lang.String ;
public class Team{

//imported Class for formatting output
static DecimalFormat ft = new DecimalFormat("0.00");
private String locationName ;
private String teamName ;
int[] weeksPlayed = new int[17] ; //the week the team plays 1,0
private String[] dateList = new String [17]; //mm/dd format , if team doesnt play
00/00

/*
Constructor 1
@param : name of the team's city, name of team, list of dates the team plays
*/

public Team( String locationName, String teamName, String[] dateList){
    this.locationName = locationName;
    this.teamName = teamName ;
    this.dateList = dateList;
    weeksPlayed = weekPlayed(dateList) ;
}
/*
Constructor 2
@param : name of the team's city, list of dates the team plays
*/
public Team(String locationName, String[] dateList){
    this.locationName = locationName;
    this.teamName = "N/A" ;
    this.dateList = dateList;
    weeksPlayed = weekPlayed(dateList) ;
}
/*
Constructor 3
@param : list of dates the team plays
*/
public Team( String[] dateList){
    this.locationName =  "N/A";
    this.teamName =  "N/A" ;
    this.dateList = dateList;
    weeksPlayed = weekPlayed(dateList) ;
}

/*
weekPlayed
@param : list of dates the team plays
@return : returns an integer array of 0s and 1s, 0 if not playing, 1 if playing
that week
*/
public int[] weekPlayed(String[] dateList){
    for(int i = 0; i< dateList.length; i++){
        String currentDate = dateList[i];
        if(currentDate == "00/00"){
            weeksPlayed[i]=0;
        }
        else if (currentDate != "00/00"){
            weeksPlayed[i]=1;
        }
    }
    return weeksPlayed ;
}


/*
byWeek
@return: integer of which week is the byweek
*/
public int byWeek(){
    int byWeekCheck = -1;
    for(int i = 0; i < dateList.length;i++){
        String currentWeek = dateList[i] ;
        if(currentWeek == "00/00"){
            byWeekCheck = i + 1;
        }
    }
    return byWeekCheck;
}

/*
estimatedSales
@param : week, team, price of ticket, number of seats total, number of seats sold
@return: double type of the combined sale of tickets
*/
public static double estimateSales(int week,Team team,double ticketPrice,int
numSeats,double seatsSold){
    double estimatedSale ;
    //Team is here so that we can check that its not a byweek
    if(team.weeksPlayed[week-1] == 0){
        estimatedSale = 0;
    }
    else {
        estimatedSale = numSeats * ticketPrice * seatsSold ;
    }
    return estimatedSale ;
}
/*
CheckSchedule
@param : team array of length 2, week number
@return : boolean of if the two teams play on the same day, true if they do
*/
public static boolean checkSchedule(Team[] team, int week){
    boolean statusChecked = false ;
    String team1date = team[0].dateList[week-1];
    String team2date = team[1].dateList[week-1];
    //do play on the same date
    if(team1date == team2date){
        statusChecked = true ;
    }
    //do not play on same date
    else if(team1date != team2date){
        statusChecked = false ;
    }
    return statusChecked ;

}
/*
printSales
@param : team array of size 1-3, price of 1 ticket, number of seats total, week
number
Prints out a table of tickets sold as a function of Percentage of seats sold
*/
public static void printSales(Team[] team, double ticketPrice, int numSeats, int
week ){
    double[] seatsSoldPerc = new double[]{0.70,0.75,0.80,0.85,0.90,0.95,1.0} ;
    double generatedRev ;
    int multiplier = 0;
    //Seats, Percent, Sales.
    //3 team possibility
    if (team.length == 3){
        Team [] matchup1 = new Team[]{team[0],team[1]} ;
        Team [] matchup2 = new Team[]{team[0],team[2]} ;
        Team [] matchup3 = new Team[]{team[1],team[2]} ;
        // should return boolean, true if teams play on same date
        boolean overlappingGame1 = checkSchedule(matchup1, week) ;
        boolean overlappingGame2 = checkSchedule(matchup2, week) ;       // should
return boolean
        boolean overlappingGame3 = checkSchedule(matchup3, week) ;       // should
return boolean
        System.out.println("          Seats       Sold           Revenue") ;
        System.out.println("          -----       ----           -------") ;
        if(overlappingGame1==true & overlappingGame2==true ){
            //they all play on the same date,
            //if 2 combos play on same day all 3 play on same day
            //multiplier is 1
            multiplier = 1;
            for(int i = 0; i< seatsSoldPerc.length;i++){
                generatedRev = estimateSales(week, team[0], ticketPrice, numSeats,
seatsSoldPerc[i]) * multiplier ;
                System.out.println("          "+numSeats+"
"+ft.format(seatsSoldPerc[i])+"           "+generatedRev+ " ");
            }
        }
        else if(overlappingGame1==false & (overlappingGame2==false &
overlappingGame3==false)){
            // all three play on different days
            //multiplier is 3
            multiplier = 3;
            if(team[0].dateList[week-1]=="00/00"){multiplier = 2;}
            else if(team[1].dateList[week-1]=="00/00"){multiplier = 2;}
            else if(team[2].dateList[week-1]=="00/00"){multiplier = 2;}
            else if(team[0].dateList[week-1]=="00/00" & team[1].dateList[week-
1]=="00/00"){multiplier = 1;}
            else if(team[0].dateList[week-1]=="00/00" & team[2].dateList[week-
1]=="00/00"){multiplier = 1;}
            else if(team[1].dateList[week-1]=="00/00" & team[2].dateList[week-
1]=="00/00"){multiplier = 1;}
            for(int i = 0; i< seatsSoldPerc.length;i++){
                generatedRev = estimateSales(week, team[0], ticketPrice, numSeats,
seatsSoldPerc[i]) * multiplier ;
                System.out.println("          "+numSeats+"
"+ft.format(seatsSoldPerc[i])+"            "+generatedRev+ " ");
            }
        }
        else if(overlappingGame1==true & (overlappingGame2==false &
overlappingGame3==false)){
            // 2 teams play on same day, 1 team plays on different day
            //multiplier is 2
            multiplier = 2 ;
            for(int i = 0; i< seatsSoldPerc.length;i++){
                generatedRev = estimateSales(week, team[0], ticketPrice, numSeats,
seatsSoldPerc[i]) * multiplier ;
                System.out.println("          "+numSeats+"
"+ft.format(seatsSoldPerc[i])+"            "+generatedRev+ " ");
            }
        }

        else if(overlappingGame1==false & (overlappingGame2==true &
overlappingGame3==false)){
            // 2 teams play on same day, 1 team plays on different day
            //multiplier is 2
            multiplier = 2 ;
            for(int i = 0; i< seatsSoldPerc.length;i++){
                generatedRev = estimateSales(week, team[0], ticketPrice, numSeats,
seatsSoldPerc[i])* multiplier ;
                System.out.println("          "+numSeats+"
"+ft.format(seatsSoldPerc[i])+"            "+generatedRev+ " ");
            }
        }

        else if(overlappingGame1==false & (overlappingGame2==false &
overlappingGame3==true)){
            // 2 teams play on same day, 1 team plays on different day
            //multiplier is 2
            multiplier = 2 ;
            for(int i = 0; i< seatsSoldPerc.length;i++){
                generatedRev = estimateSales(week, team[0], ticketPrice, numSeats,
seatsSoldPerc[i])* multiplier ;
                System.out.println("          "+numSeats+"
"+ft.format(seatsSoldPerc[i])+"            "+generatedRev+ " ");
        }

    }
}
    else if (team.length == 2) {
        System.out.println("          Seats       Sold           Revenue") ;
        System.out.println("          -----       ----           -------") ;
        boolean Game1 = checkSchedule(team, week) ;   // should return boolean
        if (Game1 == true){
            multiplier=1;
            for(int i = 0; i< seatsSoldPerc.length;i++){
                generatedRev = estimateSales(week, team[0], ticketPrice, numSeats,
seatsSoldPerc[i])* multiplier ;
                System.out.println("          "+numSeats+"
"+ft.format(seatsSoldPerc[i])+"            "+generatedRev+ " ");
        }
        }
        else if (Game1 == false){
            multiplier = 2;
            for(int i = 0; i< seatsSoldPerc.length;i++){
                generatedRev = estimateSales(week, team[0], ticketPrice, numSeats,
seatsSoldPerc[i])* multiplier ;
                System.out.println("          "+numSeats+"
"+ft.format(seatsSoldPerc[i])+"           "+generatedRev+ " ");
        }
        }
    }
    else if (team.length == 1){
        System.out.println("          Seats       Sold           Revenue") ;
        System.out.println("          -----       ----           -------") ;
        //does the team play on that week
        int teamByWeek = team[0].byWeek() ;
        if (teamByWeek != week){
            multiplier=1;
            for(int i = 0; i< seatsSoldPerc.length;i++){
                generatedRev = estimateSales(week, team[0], ticketPrice, numSeats,
seatsSoldPerc[i])* multiplier ;
                System.out.println("          "+numSeats+"
"+ft.format(seatsSoldPerc[i])+"           "+generatedRev+ " ");
        }
        }
        else if (teamByWeek == week){
            multiplier = 0;
            for(int i = 0; i< seatsSoldPerc.length;i++){
                generatedRev = estimateSales(week, team[0], ticketPrice, numSeats,
seatsSoldPerc[i])* multiplier ;
                System.out.println("          "+numSeats+"
"+ft.format(seatsSoldPerc[i])+"           "+generatedRev+ " ");
        }
        }
    }


}
}
```