

```

public class Movie {
    private String name;
    private int[] times;
    private Seats[] seats; // correction point

    public Movie(String name, int[] time, Seats[] s) {
        super();
        this.name = name;
        this.times = time;
        seats = s;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int[] getTimes() {
        return times;
    }

    public void setTimes(int[] time) {
        this.times = time;
    }

    public Seats[] getSeats() {
        return seats;
    }

    public void setSeats(Seats[] seats) {
        this.seats = seats;
    }

    public void printAllTimes() { //print all show time of a single movie, assuming it's in one Cinema.
        System.out.println("Timetable for " + getName() + ":");

        for (int i = 0; i < times.length - 1; i++) {
            System.out.print(times[i] + ", ");
        }
        System.out.println(times[times.length - 1] + ". ");
        System.out.println(" ");
    }

    public int checkTime(int time) { //ตรวจว่ามีเวลาอบฉายตามพารามิเตอร์จริงหรือเปล่า ถ้าจริงก็หาค่า index ตำแหน่งที่เวลานั้นอยู่ในอาร์เรย์ times ถ้าไม่
        //จริงให้หาค่า -1
        for (int i = 0; i < times.length; i++) {
            if (times[i] == time) {
                return i;
            }
        }

        return -1;
    }

    public void clearSeats() {
        for (int i = 0; i < seats.length; i++) {
            seats[i].clearSeats();
        }
    }
}

```

ผมย้ายมาไว้ที่นี่ จะได้แมทซ์เข้ากับตารางเวลาได้ตรงๆ index จะได้ตรงกัน

Auto gen ได้ อย่าลืม

```
}
```

```
package lab00;
```

```
public class Seats {  
    private int[][] chairs; //เป็นอาร์เรย์สองมิติ  
    private int row, column;  
  
    public Seats(int row, int column) {  
        if (row <= 0 || column <= 0) { //ตรวจจำนวน row , column แล้วให้ค่าเริ่มต้นที่ถูกต้องแบบนี้ น่าจะดีกว่าปล่อยให้เฉยๆโดยไม่เช็ค  
            System.out  
                .println("Invalid row or column number, default value of 1s are given.");  
            row = 1;  
            column =1;  
        }  
  
        chairs = new int[row][column];  
        for (int i = 0; i < row; i++) {  
            for (int j = 0; j < column; j++) {  
                chairs[i][j] = 0; //ตอนแรกต้องไม่มีคนนั่งเลย  
            }  
        }  
  
        this.row = row;  
        this.column = column;  
    }  
  
    public void displaySeats() {  
        for (int i = 0; i < row; i++) {  
            for (int j = 0; j < column; j++) {  
                if (chairs[i][j] == 0)  
                    System.out.print("0");  
                else  
                    System.out.print("X");  
            }  
            System.out.println("");  
        }  
        System.out.println("");  
        System.out.println("");  
        System.out.println("");  
    }  
  
    public void takeSeat(int rowIndex, int columnIndex) {  
        if (rowIndex <0 || columnIndex <0 || rowIndex > chairs.length-1 || columnIndex >  
chairs[0].length-1){  
            System.out  
                .println("Invalid row or column number, seat taking cancelled.");  
            return;  
        }  
  
        if (chairs[rowIndex][columnIndex] == 0)  
            chairs[rowIndex][columnIndex] = 1;  
    }  
  
    public void giveUpSeat(int rowIndex, int columnIndex) {  
        if (rowIndex <0 || columnIndex <0 || rowIndex > chairs.length-1 || columnIndex >  
chairs[0].length-1){  
            System.out  
                .println("Invalid row or column number, seat returning cancelled.");  
            return;  
        }  
    }  
}
```

```

        if (chairs[rowIndex][columnIndex] == 1)
            chairs[rowIndex][columnIndex] = 0;
    }

    public void clearSeats() {
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < column; j++) {
                chairs[i][j] = 0;
            }
        }
    }

    public static void main(String[] args) { // main ตรงนี้ผมเอาไว้ทดสอบเล่นๆเฉยๆ นะ
        Seats s = new Seats(5, 10);
        s.displaySeats();
        s.takeSeat(3, 4);
        s.takeSeat(2, 7);
        s.displaySeats();
        s.giveUpSeat(3, 7);
        s.displaySeats();
        s.giveUpSeat(3, 4);
        s.displaySeats();
        s.clearSeats();
        s.displaySeats();
    }
}

```

```
package lab00;
```

```

public class Cinema {
    private Movie[] movies; // assume no duplicate movie name
    private int earning;

    public Cinema(Movie[] movies) {
        super();
        this.movies = movies;
        earning = 0;
    }

    public Movie[] getMovies() {
        return movies;
    }

    public void setMovies(Movie[] movies) {
        this.movies = movies;
    }

    public int getEarning() {
        return earning;
    }

    public void setEarning(int earning) {
        this.earning = earning;
    }

    public Movie findMovie(String name) { // ตรงนี้เป็นส่วนของโค้ดที่เขียนในหลายๆเมธอด เลยจับรวมกันเขียนเป็นเมธอดขึ้นมา
        for (int i = 0; i < movies.length; i++) {
            if (movies[i].getName().equals(name)) {
                return movies[i];
            }
        }
        System.out.println("There is no movie of the given name.");
        return null;
    }
}

```

```

public void printMovieTime(String movieName) {
    Movie m = findMovie(movieName);
    if (m == null) {
        return;
    }
    m.printAllTimes();
}

public void displaySeats(String movieName, int time) {
    Movie m = findMovie(movieName);
    if (m == null) {
        return;
    }

    int timeSlot = m.checkTime(time);

    if (timeSlot != -1) { // There are movie and show time, so we
        // display seats in that movie's timeslot.
        Seats s = m.getSeats()[timeSlot];
        s.displaySeats();
    } else {
        System.out
            .println("The given time does not match the actual showing time.");
    }
}

public void takeorGiveSeat(String movieName, int time, int rowIndex,
    int columnIndex, boolean take) { //เมธอดร่วม จะได้เขียนซ้ำน้อยหน่อยตอน take หรือ give up
    Movie m = findMovie(movieName);
    if (m == null) {
        return;
    }

    int timeSlot = m.checkTime(time);

    if (timeSlot != -1) { // There are movie and show time, so we
        // book seat or give up seat.
        Seats s = m.getSeats()[timeSlot];
        if (take == true) {

            s.takeSeat(rowIndex, columnIndex);
            earning += 100;
        } else {
            s.giveUpSeat(rowIndex, columnIndex);
            earning -= 100;
        }
    } else {
        System.out
            .println("The given time does not match the actual showing time.");
    }
}

public void takeSeat(String movieName, int time, int rowIndex,
    int columnIndex) {
    takeorGiveSeat(movieName, time, rowIndex, columnIndex, true);
}

public void giveUpSeat(String movieName, int time, int rowIndex,
    int columnIndex) {
    takeorGiveSeat(movieName, time, rowIndex, columnIndex, false);
}

public void clearSeats(String movieName, int time) {

```

```

    Movie m = findMovie(movieName);
    if (m == null) {
        return;
    }

    int timeSlot = m.checkTime(time);

    if (timeSlot != -1) { // There are movie and show time, so we
        // clear seats for that particular show.
        Seats s = m.getSeats()[timeSlot];
        s.clearSeats();
    } else {
        System.out
            .println("The given time does not match the actual showing time.");
    }
}

public void clearSeats(String movieName){// clear seats at every show time of a given movie.
    Movie m = findMovie(movieName); //เมธอดนี้ไม่ได้สั่ง แต่อยากเขียน
    if (m == null) {
        return;
    }

    m.clearSeats();
}

public void clearAll(){
    for (int i = 0; i < movies.length; i++) {
        movies[i].clearSeats();
    }
}
}

```

```
package lab00;
```

```

public class Lido {
    private Cinema firstCinema, secondCinema;

    public Lido(Cinema firstCinema, Cinema secondCinema) {
        super();
        this.firstCinema = firstCinema;
        this.secondCinema = secondCinema;
    }

    public Cinema getFirstCinema() {
        return firstCinema;
    }

    public void setFirstCinema(Cinema firstCinema) {
        this.firstCinema = firstCinema;
    }

    public Cinema getSecondCinema() {
        return secondCinema;
    }
}

```

โจทย์ดูเหมือนจะพยายามให้ fix จำนวนที่นั่งแต่ละโรงไปเลย แต่ผม
ว่ามันจำกั๊ดเกินไป ผมไปกำหนดจำนวนที่นั่งตอนทำ main แทน

```

public void setSecondCinema(Cinema secondCinema) {
    this.secondCinema = secondCinema;
}

public void showAll(String movieName) {
    System.out.println("First cinema show time for " + movieName + ".");
    firstCinema.printMovieTime(movieName);
    System.out.println("Second cinema show time for " + movieName + ".");
    secondCinema.printMovieTime(movieName);
    System.out.println("Done");
}

public static void main(String[] args) {
    Movie[] moviesFor1stCinema = new Movie[2];

    int[] spiderManTimein1stCinema = { 1300, 1530, 1800 };
    Seats[] seatsForSpiderManin1stCinema = new Seats[3];
    for (int i = 0; i < seatsForSpiderManin1stCinema.length; i++) {
        seatsForSpiderManin1stCinema[i] = new Seats(10, 20);
    }

    int[] hobbitTimein1stCinema = { 1830, 2200 };
    Seats[] seatsForHobbitin1stCinema = new Seats[2];
    for (int i = 0; i < seatsForHobbitin1stCinema.length; i++) {
        seatsForHobbitin1stCinema[i] = new Seats(10, 20);
    }

    moviesFor1stCinema[0] = new Movie("Amazing Spider-Man",
        spiderManTimein1stCinema, seatsForSpiderManin1stCinema);
    moviesFor1stCinema[1] = new Movie("The Hobbit: There and Back Again",
        hobbitTimein1stCinema, seatsForHobbitin1stCinema);

    Cinema a = new Cinema(moviesFor1stCinema);

    Movie[] moviesFor2ndCinema = new Movie[2];

    int[] spiderManTimein2ndCinema = { 1330, 1600, 1830 };
    Seats[] seatsForSpiderManin2ndCinema = new Seats[3];
    for (int i = 0; i < seatsForSpiderManin2ndCinema.length; i++) {
        seatsForSpiderManin2ndCinema[i] = new Seats(10, 30);
    }

    int[] hobbitTimein2ndCinema = { 1900, 2230 };
    Seats[] seatsForHobbitin2ndCinema = new Seats[2];
    for (int i = 0; i < seatsForHobbitin2ndCinema.length; i++) {
        seatsForHobbitin2ndCinema[i] = new Seats(10, 30);
    }

    moviesFor2ndCinema[0] = new Movie("Amazing Spider-Man",
        spiderManTimein2ndCinema, seatsForSpiderManin2ndCinema);
    moviesFor2ndCinema[1] = new Movie("The Hobbit: There and Back Again",
        hobbitTimein2ndCinema, seatsForHobbitin2ndCinema);

    Cinema b = new Cinema(moviesFor2ndCinema);

    Lido c = new Lido(a,b);
    c.showAll("Amazing Spider-Man");

    //
    c.getFirstCinema().takeSeat("Amazing Spider-Man", 1300, 2, 10);
    c.getFirstCinema().takeSeat("Amazing Spider-Man", 1300, 5, 13);
    c.getFirstCinema().takeSeat("Amazing Spider-Man", 1300, 5, 13);

    System.out.println("First cinema after booking Spider-Man at 1300:");
    c.getFirstCinema().displaySeats("Amazing Spider-Man", 1300);

    c.getFirstCinema().takeSeat("Amazing Spider-Man", 1800, 5, 13);
}

```

เตรียมใจ 1

เตรียมใจ 2

จากนี้ทำตามคำสั่ง test ในโจทย์

```

c.getFirstCinema().takeSeat("Amazing Spider-Man", 1330, 5, 13);

System.out.println("First cinema after booking Spider-Man at 1300:");
c.getFirstCinema().displaySeats("Amazing Spider-Man", 1300);

System.out.println("First cinema after booking Spider-Man at 1800:");
c.getFirstCinema().displaySeats("Amazing Spider-Man", 1800);

System.out.println("Now start booking the second cinema:....");
c.getSecondCinema().takeSeat("Amazing Spider-Man", 1330, 5, 13);
c.getSecondCinema().takeSeat("The Hobbit: There and Back Again", 2230, 7, 14);

System.out.println("Second cinema after booking Hobbit at 2230:");
c.getSecondCinema().displaySeats("The Hobbit: There and Back Again", 2230);

c.getSecondCinema().giveUpSeat("The Hobbit: There and Back Again", 2230, 7, 14);
System.out.println("Second cinema after cancelling Hobbit at 2230:");
c.getSecondCinema().displaySeats("The Hobbit: There and Back Again", 2230);

c.getFirstCinema().clearSeats("Amazing Spider-Man", 1300);
System.out.println("First cinema after clearing all seats for Spider-Man at 1300:");
c.getFirstCinema().displaySeats("Amazing Spider-Man", 1300);

c.getFirstCinema().clearAll();
c.getSecondCinema().clearAll();

System.out.println("First cinema after clearing all seats for Spider-Man-> at 1800:");
c.getFirstCinema().displaySeats("Amazing Spider-Man", 1800);

System.out.println("Second cinema after clearing Spiderman -> at 1330");
c.getSecondCinema().displaySeats("Amazing Spider-Man", 1330);

}
}

```

โจทย์ผิด ต้องแก้เวลาเป็นอย่าง
นี้ ถึงจะได้คำตอบตามที่โจทย์
คาดหวังไว้

```

First cinema show time for Amazing Spider-Man.
Timetable for Amazing Spider-Man:
1300, 1530, 1800.

Second cinema show time for Amazing Spider-Man.
Timetable for Amazing Spider-Man:
1330, 1600, 1830.

Done
First cinema after booking Spider-Man at 1300:
00000000000000000000
00000000000000000000
0000000000X000000000
00000000000000000000
00000000000000000000
000000000000X0000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000

The given time does not match the actual showing time.
First cinema after booking Spider-Man at 1300:
00000000000000000000
00000000000000000000
0000000000X000000000
00000000000000000000
00000000000000000000

```

นี่คือสิ่งที่พิมพ์ออกมาตอนรัน

000000000000X00000
000000000000000000
000000000000000000
000000000000000000
000000000000000000

First cinema after booking Spider-Man at 1800:

000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000X00000
000000000000000000
000000000000000000
000000000000000000
000000000000000000

Now start booking the second cinema:....

Second cinema after booking Hobbit at 2230:

00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
000000000000X000000000
00000000000000000000
00000000000000000000

Second cinema after cancelling Hobbit at 2230:

00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000

First cinema after clearing all seats for Spider-Man at 1300:

000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000

First cinema after clearing all seats for Spider-Man-> at 1800:

000000000000000000
000000000000000000
000000000000000000

