**Date: 11 July 2020 Time : 2.45 Hrs**

Instructions:

 Must read the problem statement thoroughly before start working.

 Give project name as yourname\_year\_number **(Ex. Rohinth\_2020\_001)**

 Create all java classes in package **“com.itech.theatre”** in src folder of the corresponding java project

 Must follow java coding standards

 Use exactly same class names as mentioned

 Use exactly same method signature (method name, return type, method parameter type, position of each method parameter)

 Define attributes with same name and data type as given in class outline.

 Define constructors and getter setters as given in the class outline.

 Ensure attributes are private and other methods which will be called from main method, getter- setter methods and constructor are public.

 Use main method only for input, output, object creation and method invocation.

 Within the given time, complete & mail it

**Theatre Management System**

**Problem Statement:**

 **TheatreManagementSystem** is an application used to manage theatre complex with multiple screens in it. It will automate the Movie release in the screens and the statistics of the film. Movies can be classified into two category it can be a *Feature Movie* or a *Documentary*.

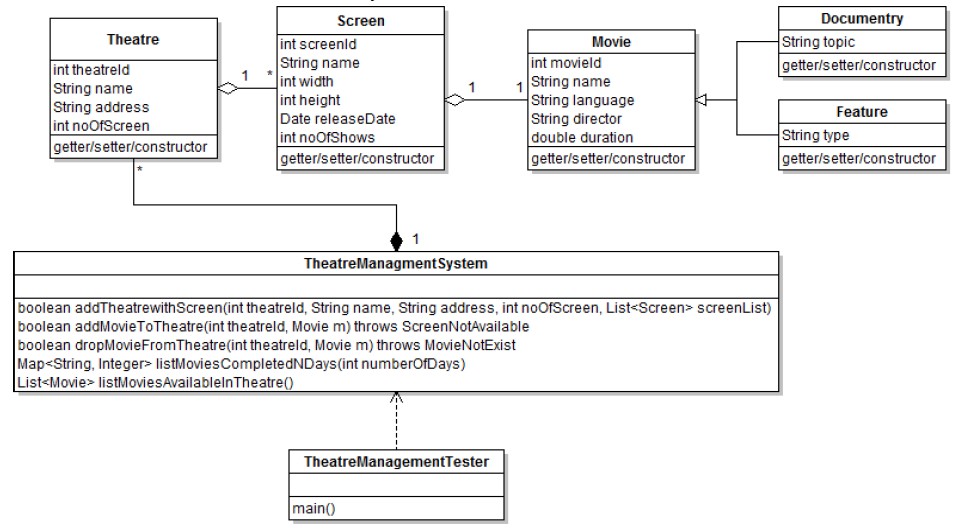
 Develop a Java project which serves this system with following guidelines.

 Class diagram is shared after the guidelines.

 Ensure the same is met 100%, else your solution would be not considered.

**Note : In Screen class, instead of releaseDate attribute declare variable**

**noOfDaysRunning - int**



Methods to be implemented in **TheatreManagementSystem**:

**boolean addTheatrewithScreen(int theatreId, String name, String address, int noOfScreen, List<Screen> screenList):**

This method will add a new theatre with screens to the TheatreManagementSystem. The method contains the parameters for forming the Theatre object and list of screens in it. Before adding the theatre into system, kindly ensure the noofscreen and size of screen list passed should be equal else return false.

[Note: No need to check the existence of theatre inside the TheatreManagementSystem, directly add the Theatre object with ScreenList based on composition relationship given in the diagram.]

**boolean addMovieToTheatre(int theatreId, Movie m) throws ScreenNotAvailable:**

This method will take a Movie object and theatreId as the parameter. It will release the movie in one of the available screen in the theatre. A vacant screen is identified by noOfDaysRunning of the screen should be 0. If there is no screen available for releasing the movie then the method should throw a **ScreenNotAvailable** exception and a message "There is no screen available to release the movie" should be displayed.

**boolean dropMovieFromTheatre(int theatreId, Movie m) throws MovieNotExist:**

This method will drop a movie from the theatre by passing a theatreId and Movie object. Identify the screen in which the movie is released then reset the movie object in the screen(setting the Movie object as null)also reset the noOfDaysRunning to 0.

If the Movie with the given name does not exist then it will throw a **MovieNotExist** Exception and a message "Unable to locate the moviewith specified name" should be displayed.

**Map<String, Integer> listMoviesAvailableInTheatre():**

This method will return a count of Feature movie currently playing in the each theatre. The method will return a Map with theatre name and its respective count of feature movie.

**List<Movie> listMoviesCompletedNDays(int numberOfDays):**

This method will return a List of Movies which completed ***n***days and more after its release. Method accept an int variable as the *number of days,* method need to return all the Movies which belong to

both Feature as well as Documentary which matching this criteria and the list need to sort based on the film name in ascending order.

[Note : Create the TheatreManagementTester class to test all the above functionalities.]

**Tester Hint-**

The first method is the base method for testing all the other functionalities. Kindly invoke the first method

with following test data’s given below.

List<Screen> screenList1 = new ArrayList<Screen>(); Screen s1 = new Screen(1, “Qube-Y”, 1000,500,null,4); Screen s2= new Screen(2, “Qube-X”, 1000,500,null,4);

Screen s3= new Screen(3, “Tracks-T”, 1000,500,null,4);

screenList1.add(s1); screenList1.add(s2); screenList1.add(s3);

obj.addTheatrewithScreen(121,”Natraj”,”TVM”,2, screenList1);

[obj-Object of TheatreManagementSystem class] List<Screen> screenList2= new ArrayList<Screen>(); Screen s4= new Screen(1, “Double-R”, 750,300,null,3); Screen s5= new Screen(2, “Max”, 750,300,null,3); screenList2.add(s4);

screenList2.add(s5); obj.addTheatrewithScreen(122,“Gokul”,”Tanjore”,3, screenList2); Based on the above test data other functionalities can be tested.

===========================================

**Zip Procedure:**

**1. In eclipse right click on your project.**

**2. Select copy, then on desktop paste it.**

**3. Now, on desktop your project folder will be available. Right click that folder and select, send to -> compressed folder.**

**4. Now you can find compressed file (zip) of your project folder.**