Table of Contents

This document contains all my CRC cards for this assignment.

Seat Class

The seat class is the base of my row class, the row class is dependant on the seats. Seats are objects with a seat number and a true/false reserve status. The rows are essentially an ArrayList of chairs

Seats	
Responsibility	Collaborators
Unreserved seat upon creation	Rows
Allocate Seat Number	
Allow you to reserve a seat	
Allow you to unreserve a seat	

Row Class

Row class is essentially an array of chairs. The row class allows you to gather the number of free seats in the row, the first free seat in the row and the row identifier (character) to allow bookings/sessions/cinemas classes to be created

Rows	
Responsibility	Collaborators
Keep track of available seats in the row	Cinema
Allow addition of seats to row	Session
Return the first free seat in the row	bookings
Allocate row name	

Bookings Class

A booking class is a class which contains all booking information for a session. The booking class allows users to create/change/cancel bookings and all created bookings have the important booking information. A session is essentially an ArrayList of bookings with other fields

bookings	
Responsibility	Collaborators
Create a booking	Session
Change a booking	Rows
Cancel a booking	
Returns index of cinema for cinemaNumber in our array of Cinemas	
Returns booking id	
Change a booking	
Cancel a booking	
Has rownumber of the booking	
Has the first seat for the booking	
Contains the rowname for the booking	
Time of booking	
Number of tickets in the booking (number of reserved seats)	

Sessions Class

The session class is supposed to represent an instance of a cinema for a certain showing. Each session contains information regarding, all bookings, Time of Session, Number of rows in the session, cinemaNumber, Movie showing.

Session	
Responsibility	Collaborators
Display all bookings for the session	Cinema
Gets the first available row of the session	Rows
Books seats for a booking	bookings
Free seats for a cancelled/ while changing bookings	
Get the booking index for a certain booking	
Adds bookings to the session	
Contains all information for all cinemas	
All showings of cinemas	
All instances of cinemas	

Cinema Class

The cinema is a blueprint for the entire project. A cinema can be visualised as a building which has sessions, rows and a cinema number, the number of rows in a cinema will dictate how many rows will be in the session. The cinema rows are used as a blueprint for number of rows in the session (also number of seats in rows) the cinema is supposed to allow sessions to be created, each session will have the same number of rows and seats in rows as the original cinema, any added rows to the cinema will also be added to EVERY session. The project will ideally be an ArrayList of cinemas to emulate a modern day theatre system (eg. Event cinemas/reading cinemas).

Cinema	
Responsibility	Collaborators
Sets up all the rows for the cinema and all sessions	Session
Adds any new row to not only itself but all sessions	Rows
Allows location of any booking	
Allows location of any session based on time/booking id/cinema number	
Allows user to add session to the cinema	
Create deep copies of rows for session so the session rows do not effect the core cinema rows	

Operate Class

This is the class that performs all the backend operations. This class takes requests one line at a time. It splits the line from the comments and evaluates the request. This then checks the first keyword of the requests, Cinema, Session, Request, Change, Cancel, Print. This then performs operations that emulate the request. All output is printed to standard output and is stored

Cinema	
Responsibility	Collaborators
Allows creation of cinema to add to ArrayList of all Cinemas	Session
Allows you to add rows to a certain Cinema	Cinema
Allows you to add sessions to a certain Cinema with all details like time/movie/which cinema	bookings
Allows you to create bookings for a session with number of tickets	Rows
Allows you to change bookings for a session	
Allows you to Cancel any booking (deletes the booking from the booking Array in the session)	
Allows you to print all bookings for a session	
Allows you to get the cinemaindex from the cinema arraylist based on cinema number	
Allows you to get the cinema which contains a certain Booking ID (scans all cinemas/all sessions/all bookings)	
Allows you to use any form of requests all handled properly	

CinemaBookingSystem

This is essentially operate Class ^^^ except I didn't want to crowd my main method with too much code, which allows easier debugging (split all functions up across all classes and just made a call to run my tests from operate)

CinemaBookingSystem	
Responsibility	Collaborators
Perform all the operations in operate class (runs all tests)	operate