**Title: Cinema Ticket Reservation System in Java**

*Description: The project involves developing a cinema ticket reservation system in Java. The system should allow the user to search for available movies, showtimes, and cinema halls, reserve tickets, and pay for them.*

For this project, you will use the following **data structures**:

**Linked list:** Java provides the LinkedList class in the java.util package to store information about the available movies. Each element of the list will be an instance of a custom class representing a specific movie.

**Array:** You will use an array to store information about movie showtimes. Each element of the array will be an instance of a custom class containing information about the showtimes of a specific movie, including the start time, end time, and cinema hall.

**Queue:** Java provides the LinkedList class in the java.util package, which implements the Queue interface. You will use this to store ticket reservation requests. Requests will be added to the queue in the order they are received.

**HashMap:** Java provides the HashMap class in the java.util package. You will use a HashMap to store information about cinema halls, where the key will be the hall number and the value will be an instance of a custom class representing details about the hall.

And more…

The program will be written in **Java using object-oriented programming concepts**. You will create classes for the different types of data structures used in the program. The program can have a graphical user interface or can simply be a console application.

Additional features to consider for the system include:

**Rating movies:** You can use a HashMap to store ratings and comments where the key is the movie and the value is a list of ratings/comments.

**Graph for movie relationships**: You can represent movie relationships using custom classes and collections like HashMap or ArrayList.

**Stack for reservations**: You can use the Stack class provided in the java.util package to store current reservations, making it easy to cancel or modify them.

**Recommendation system:** You can implement a recommendation system using custom logic and the data structures available in Java. (Hash Map, Graph, Priority Queue,..)

**Age-based recommendation:** You can use conditional logic to recommend appropriate movies based on the user's age.

**Customization feature for seat selection:** You can represent seats in the cinema hall using a matrix, which can be implemented using a 2D array.

**Loyalty point system:** You can use a HashMap to store information about users and their points.

**Employee management:** You can use various data structures to manage employee details and track sales, such as HashMap, Array Lists, or custom classes.

**Attendance monitoring:** You can use a HashMap to store information about cinema attendance and ticket prices.

**Price reduction feature:** You can implement price reduction logic using custom code and data structures available in Java.

**Implementation:** The program will be written in Java using object-oriented programming concepts. You will create classes for the different types of data structures used in the program. The program can have a graphical user interface or can simply be a console application.

Movie

* Name
* Length
* Release date
* Directors
* Distributors
* About
* List of genres
* price

AvailableMovies

Linked List of movies

MovieShowtime

* Movie
* StartTime
* EndTime
* CinemaHall

AvailableShowtimes

Array of showtimes

CinemaHall

* HallNumber
* Array of seats

CinemaHalls

Hashmap of cinema Halls

Reserve seat/ ticket

* + Name
  + Contact
  + cinema hall number
  + List of seats
  + Date
  + Time
  + Price
  + MovieName

Seat

* rowNumber
* columnNumber

Popcorn: Small, Medium, Large Qty, price, name

Beverages: Coke, Sprite, Rootbeer

Payment Mode: Cash, Debit, Credit

Employee Management System

* Total number of sales

Only employees can change the price of goods.

A client can reserve many seats

In real life there is one cinema hall. The seats of that cinema hall are reserved everyday. If I have like three movies on the same day and I schedule one seat in one movie, that seat must not necessarily be schedule in another movie.

Two movies cannot be scheduled at the same time.

For the seat problem, a showtime will receive the array of seats(number of seats). Or implement prototype for the cinemaHall and send a clone to the showtime.

Updated the Ticket class with all its properties and methods. cinemaHall variable and setCinemaHall methods were added to the MovieShowtime class. The getMovie method in MovieShowtime returns a movie. Added the getShowtime function in the AvailableShowtime class. The reserveSeat methods were added to the Ticket, MovieShowtime and CinemaHall classes.