Project Title: Theater Seat Reservation System

Description:

The goal of this project is to create a Java application that manages the seat reservation process in a theater. The application will involve file manipulations, Java OOP principles, collections, generics, and exception handling.

Features:

Seat Class: Create a Seat class that represents an individual seat in the theater. It should have attributes such as seat number, row, and availability status.

Theater Class: Develop a Theater class that manages the collection of seats. It should support operations like displaying the seating arrangement, reserving a seat, canceling a reservation, and checking seat availability. Theater class should have a contractor that gets theater name, number of rows and column , … , and fills the seats collection base on this row and column.

File Import/Export: Implement methods to import and export the theater's seating arrangement from/to a file. The file should store the seat details in a structured format (e.g., CSV or JSON).

Seat Availability: Create a class that checks the availability of a specific seat and displays its current status.

Exception Handling: Implement appropriate exception handling to handle scenarios like invalid file formats, file not found, or errors during file operations.

User Interaction: Design a simple command-line interface that allows users to interact with the theater seat reservation system. Provide options to view the seating arrangement, reserve a seat, cancel a reservation, and check seat availability.

Guidelines:

Implement the Seat and Theater classes using proper Java OOP principles, such as encapsulation and method design.

Use appropriate data structures, like arrays or lists, to manage the collection of seats in the theater.

Apply generics to make your code more flexible and reusable.

Handle exceptions gracefully by providing meaningful error messages to the user.

Use file reading and writing techniques to import and export the theater's seating arrangement from/to files.

Design a user-friendly command-line interface that guides users through various operations.

Timeframe:

This project can be completed in approximately 3-4 days.

Remember to break down the project into smaller tasks and plan accordingly. Good luck with your project!