CSCD 212
Design Patterns
Final Project

Problem Description and Constraints

The problem statement may be incomplete. Unlike assignments you may have had previously, in which the problem is fully described (possibly with bits of code provided for you), the problem statement given below is not necessarily complete or consistent. Do not make assumptions about what is needed!
If the information is not provided, ask questions. Your program must be robust. You can presume that only valid inputs are provided.

You are required to complete a Java program that implements a basic Cruise System (CS). The CS allows a client to create cruise ships, cruise ports, cruise rooms.

Each cruise ship is associated with a main cruise port. A cruise ship has a starting port and a destination port. The originating and destination ports cannot be the same.

Each cruise ship is associated with a cruise room (inside cabin, cabin with a port, cabin with a balcony, cabin suite, all these rooms have associated numbers which consist of a letter and a room number. The letter represents the deck name) and a cruise package (drinks package, wi-fi package, dinner with the captain, etc). A cruise package and cruise cabin consist of a CruiseManager that provides a single point of access to the functions provided (Façade Pattern).

NOTES:

- A cruise can be more than two ports.
- Cruise ships have a maximum capacity.
- The ports will be loaded from a file named ports.txt. The file format is location country code, location name.
- Cruise ships information is read from a file. Parent company, ship name, ship name, ship name, …
- There are multiple types of ships (adults only, family friendly, etc)
- Clients can subscribe to watch for cruise deals.
- Other information can be read from the keyboard, or a file.

In this assignment, you will be required to implement the following functionality:

PART 1

1. Create a cruise port. A cruise port must have a name consisting of location name, location country code. No two cruise ports can have the same location name and location country. (https://www.whatsinport.com/)
2. Create a cruise ship. A cruise ship has a ship parent company and a ship name. No two cruise ships can have the same ship name.
3. Create a cruise package and a cruise room. This is specific for a cruise ship.
4. Create a cruise. A cruise consists of starting port, the destination port, a departure date, the number of days in the destination port, the number of days on the water to the destination port.
5. Allow the client to find available cruises, based on the starting and destination ports, and the departure date.
6. Allow the client to book a cruise, including room, and package based on the available cruises.
7. Print all the details.

CSCD 212
Design Patterns
Final Project

Required classes (Design Constraints)

CruiseManager: This class provides the interface (Façade) to the system. That is, clients interact with the system by calling operations in the CruiseManager. The CruiseManager is linked to all the ports and cruise ship objects in the system. When it is created, the CrusieManager has no links. To create cruise ports and cruise ships, the methods named createCruisePort and createCruiseShip operations defined in this CruiseManager class must be invoked. A printout of information on everything that is happening is obtained by invoking the method displayCruiseSystemDetails.

The classes you create and the methods you create are up to you.

It is expected that you will use the following patterns:

- Strategy
- Factory
- Builder
- Observer
- Prototype
- Façade
- Adapter?
- Other patterns not mentioned here?

You will need to write your own tests.

This is an individual final project.

You will submit your code pushed to your repo. As part of your submission you must write a document that explains how you used the patterns. I don't want code here I want in your own words how the patterns were used.

Example: The strategy pattern is used with the cruise ship interface and the cruise ship classes named X, Y, Z.

This will be a detailed explanation of how you are using the pattern. I don't want the pattern definition. This document with your code will show me you understand the basics of design patterns.