

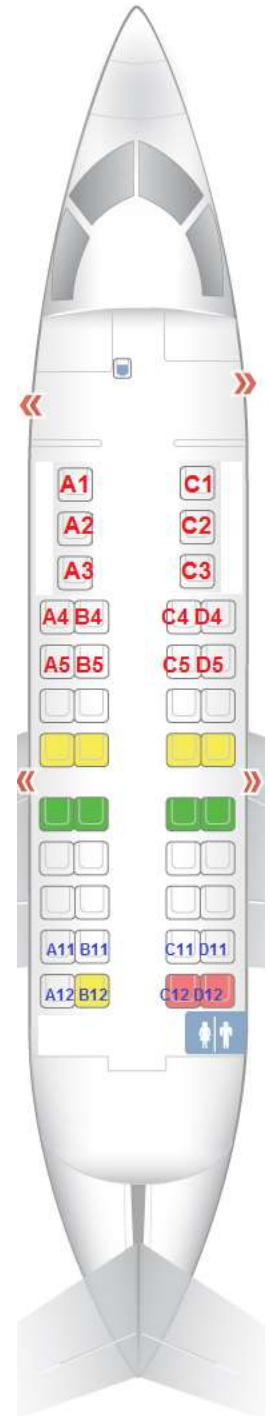
New Objectives:

- To use **String** class.
- To write a **try-catch** block to handle exceptions.
- To read data from a file using the **Scanner** class.
- To write data to a file using the **PrintWriter** class.

Task: Flight Reservation System III

In this phase, you need to consider a new small flight as shown in the figure. Note that in this phase the seat number is a **string** consists of a row number (e.g. 1, 2, ..., 12) and a character (A, B, C, or D) that represents the column number.

- Create a **Seat** class that has:
 - Private String **seatNumber** with getter and setter methods.
 - Private seat **row** number.
 - Private seat **column** number.
 - Private String for **passenger name** with getter and setter methods. Note: if name is null this means seat is empty.
 - A constructor that takes seat number string.
 - A static method **isValid** that takes a seat number string and returns if the seat number is valid or not.
 - A static method **getRow** that takes a seat number string and returns the flight row number if the seat number is valid.
 - A static method **getColumn** that takes a seat number string and returns the flight row number if the seat number is valid.
 - A method **isEmpty** that returns if the seat is empty or not.
 - A **toString** method that returns a string represents seat object's information.



- Create a **Flight** program class that has the following:
 - A private static ragged array of seats reflecting the new mentioned flight seat map.
 - A main method that initialize the seats array and then keep displays a menu containing the following options (similar to phase II):
 1. **Read passengers file.**
 2. **Reserve a new empty seat.**
 3. **Delete a reserved seat.**
 4. **Delete all reserved seats.**
 5. **Update passengers file.**
 6. **Quit.**
 - Regarding **Option 1**: consider the file *passengers.txt* file that contains reserved seats in the following format:

Seat Number: Passenger name

Read the file line-by-line and update the seats array accordingly. Make sure to handle any wrong seat numbers or duplicate reservations for the same seat.
 - Regarding **option 2**: you need to get seat number string and passenger name from the user.
 - Regarding **option 5**: you need to write back all the reserved seats information back to the passengers file.