

09 Mar 16 7:53

Row.java

Sidan 1/3

```

import java.util.Scanner;
import java.io.*;

/**
 * Represents a row of seats.
 * The seats are numbered 0, 1, 2, ...
 */
public class Row {
    private String[] seats;

    /**
     * Creates a row with without aisle indication
     * @param size the number of seats in the row
     */
    public Row(int size) {

        // Uppgift A2

    }

    /**
     * Gets the name of the passenger at a specified seat
     * @param seat seat number
     * @return the passenger
     */
    // Uppgift A3

    /**
     * Print the names of the passengers in the row
     */
    public void printPassengers() {
        for (int i=0; i<seats.length; i++) {

            // Uppgift A4

        }
    }

    /**
     * Searches the row for a passenger name.
     * @param name Name of the person to be searched
     * @return The seat number or -1 if the person is not found
     */
    public int find(String name) {

        // Uppgift A5

    }

    /**
     * Returns a string indicating booked and free seats:
     * Example:
     * [. B B . . B .]
     * B indicates a reserved and . indicates a free seat.
     */
    public String toString() {
        String result = "";

        // Uppgift A6

        return "[" + result + "];"
    }
}

```

09 Mar 16 7:53

Row.java

Sidan 2/3

```

/**
 * Books a seat.
 * @param seat seat number (i.e. 1, 2, 3, ... )
 * @param passenger passenger name
 * @return true if the booking was successful, else false
 */
// Uppgift A7
public boolean book(int seat, String passenger) {

    if ( ... ) {
        System.out.println("*** Illegal seat number:" + seat);
        return false;
    } else if ( ... ) {
        System.out.println("*** Seat already booked");
        return false;
    } else {
        ...
        return true;
    }
}

/**
 * Get the number of seast in the row
 * return number of seats
 */
public int rowLength() {
    return seats.length;
}

/**
 * Save the complete row.
 * Format:
 * First line contains the number of seats (int)
 * Then one line for each seat containing the passenger name.
 * If the seat is not booked the string "none" is written.
 * @param pw an open printwriter
 */
public void save(PrintWriter pw) {
    pw.println(seats.length);
    for (String name : seats) {
        if (name==null) {
            pw.println("none");
        } else {
            pw.println(name);
        }
    }
}

/**
 * Creates and returns a Row object using information
 * read by a scanner. See the save method for details
 */
public static Row load(Scanner fscan) {

    // Uppgift B1

}

```

09 Mar 16 7:53

Row.java

Sidan 3/3

```

/**
 * Demonstration program for the Row class
 */
public static void main(String[] args) {
    Row row = new Row(6);
    row.book(3, "Jeltz");
    row.book(2, "Ford Prefect");
    row.book(1, "Trillian");
    System.out.println("Row: " + row.toString());
    System.out.println("Find Jeltz: " + row.find("Jeltz"));
    System.out.println("Find Zaphod: " + row.find("Zaphod"));

    System.out.println("\nBookings on this row: " + row.toString());

    System.out.println("\nTry to book an already booked seat ");
    row.book(2, "Arthur Dent");

    System.out.println("\nPassengers on this row:");
    row.printPassengers();
}

}

/* Output:

Initial row      : [ . . . . . ]
Row with bookings: [ . B B B . . ]
Find Jeltz : 3
Find Zaphod: -1

Try to book an already booked seat
*** Seat already booked

Passengers on this row:
Trillian
Ford Prefect
Jeltz

*/

```

07 Mar 16 19:55

Cabin.java

Sidan 1/1

```

import java.util.ArrayList;

public class Cabin {
    private Row[] theRows;
    private String flightId;

    public Cabin(String flightId, int numberOfRows, int rowLength) {
        ... // Uppgift B2
    }

    /**
     * Creates a passenger list i.e. an arraylist with passenger objects
     */
    public ArrayList<Passenger> passengerList() {
        ... // Uppgift B3
    }

    /**
     * Creates a sorted arraylist of passenger objects
     * @param list An arraylist with passenger objects
     * @return An arraylist with sorted passenger objects
     */
    public static ArrayList<Passenger> sort(ArrayList<Passenger> list) {
        ... // Uppgift B4
    }

    /**
     * Prints a name sorted passenger list
     */
    public void printPassengers() {
        ... // Uppgift B5
    }

    // Other methods ...
}

```