

## Assignment: Implement a Flight Booking System

### Objective:

You will develop parts of a simple Flight Booking System based on the Java skeleton created in the previous assignment. This system will involve persisting data using object serialization, and implementing search functionality that works against persisted data. It also involves creating and using a custom exception type.

### Tasks:

#### 1. Add flights:

- Implement a method to add new flights. Make it possible to enter the arrival and departure times using a time zone.

#### 2. Search Functionality:

- Implement a search method to find flights based on departure and destination. It should be possible to search for only departures, only destinations and a combination of departure and destination. Seats and prices don't need to be displayed at this moment (but feel free to think about how to best present this information).

#### 3. Data Persistence:

- Implement object serialization to save and load the state of all **Flights**. Load all flights when starting the application and save them all before quitting.

#### 4. Exception Handling:

- Handle exceptions that may occur during file operations.
- Implement a custom exception called `FlightBookingException` and throw it whenever a new flight is added using an already existing flight number. This exception should not crash the application.

#### 5. User Interface:

- Create a simple text-based interface that allows users to interact with the system.
- Include options to add flights, passengers, bookings, and to search for flights. (Note that you only have to implement the functionality for add flights and search for flights).
- Create a `Main` class and use it for all user interaction (input and output). That is, there should be no print statements but in the `Main` class.

### Example Output:

```
1. Add Flight
2. Add Passenger
```

```
3. Book a Flight
4. Search Flights
5. View Bookings
6. Save and Exit
Choose an option: 1
Enter Flight Number: S0123
Enter Departure City: Stockholm
Enter Destination City: Malmö
Enter Departure Time (yyyy-MM-dd HH:mm +/-HH:mm): 2023-12-12 08:00 +02:00
Enter Arrival Time (yyyy-MM-dd HH:mm +/-HH:mm): 2023-12-12 10:00 +02:00
Flight added successfully.
```

```
Choose an option: 4
Enter Departure City (leave blank for any): Stockholm
Enter Destination City (leave blank for any):
Found Flights:
Flight Number: S0123
Departure City: Stockholm
Destination City: Malmö
Departure Time: 2023-12-12 08:00 +02:00
Arrival Time: 2023-12-12 10:00 +02:00
Flight Number: S0145
Departure City: Stockholm
Destination City: Copenhagen
Departure Time: 2023-12-12 09:00 +02:00
Arrival Time: 2023-12-12 11:30 +02:00
```

**Hints:**

- Use a `DateTimeFormatter` to help parse the date, time and time zone.
- Consider using the class `FlightService` to handle the list of all flights (including saving, loading and searching).