

Clean Code Challenge

Overview

A small airline wants you to improve their flight booking service.

Users can:

- Enter passenger details
- Print a report containing the flight's financials and other flight details.

You've been asked to:

- Refactor the code using clean coding principles to make the code maintainable
- Add new features
- Write tests to prevent functionality from breaking

We recommend you spend between 2-3 hours on your solution.

Submission Guidelines

The starting point for your solution will be the source code enclosed in the zip file. The primary code is located in the ScheduledFlight class in the FlightBooking.Core project.

Solutions can be written using any recent version of Visual Studio.

When submitting your solution, compress the source code and supporting files into a single archive before sending. Please remove all build output files before compressing (in particular check there are no .exe files or \obj\ folders) as our firewall blocks these.

Please note:

It is very important you don't break any existing functionality. The same passenger data should generate the same summary report (along with any additional information).

The FlightRoute and Plane classes cannot change as other systems depend on them.

There is a sample console application which demonstrates how a user might interact with the flight booking system. Feel free to update this. A separate UI is not required as the airline has hired a web design agency team to produce this.

Good luck!

System description

There are three types of passengers currently catered for:

- General – Normal fare paying passengers
- Loyalty Members – Repeat customers who get benefits for choosing to fly with the airline
- Airline Employee – Employees of the airline who fly for free as a perk

The airline charges a base price for a route, which loyalty members can choose to pay with their loyalty points. Airline employees always fly free.

Loyalty points are given to loyalty members for certain routes. The amount depends on the route. Baggage allowance is allocated on a basis of 1 bag allowed for everyone and 1 extra bag allowed for a loyalty member.

This input:

```
add general Steve 30
add general Mark 12
add general James 36
add general Jane 32
add loyalty John 29 1000 true
add loyalty Sarah 45 1250 false
add loyalty Jack 60 50 false
add airline Trevor 47
add general Alan 34
add general Suzy 21
print summary
```

Generates this summary report:

```
Flight summary for London to Paris
Total passengers: 10
General sales: 6
Loyalty member sales: 3
Airline employee comps: 1
Total expected baggage: 13
Total revenue from flight: 800
Total costs from flight: 500
Flight generating profit of: 300
Total loyalty points given away: 10
Total loyalty points redeemed: 100
THIS FLIGHT MAY PROCEED
```

New features

1. The airline would like to add a new type of passenger – Discounted.

This passenger type:

- Doesn't accrue loyalty points
- Isn't allowed a bag
- Is only charged half the normal base price for the route.

They would like this type of passenger reflected on the summary report in the same way the other passenger types are reported.

2. The system has some business rules for deciding whether the flight can proceed.

These are:

- The revenue generated from the flight must exceed the cost of the flight
- The number of passengers cannot exceed the amount of seats on the plane
- The aircraft must have a minimum percentage of passengers booked for that route

The business would like to keep these default rules but also have an option to choose a different set of 'Relaxed rules'.

In the 'Relaxed rules' if the number of airline employees aboard is greater than the minimum percentage of passengers required, then the revenue generated doesn't need to exceed cost. They want the option to switch between these rules sets. They have indicated they might want more rule sets in the future.

3. Often the airline finds that they overbook flights and are then stuck. As they have a range of aircraft they would like the summary report to list any aircraft which would be able to fly instead.

For example:

```
THIS FLIGHT MAY NOT PROCEED.  
Other more suitable aircraft are:  
Bombardier Q400 could handle this flight.  
ATR 640 could handle this flight.
```

If there are no other planes that could handle the flight then you don't need to list anything extra. You can decide how you make this information available to the summary report.