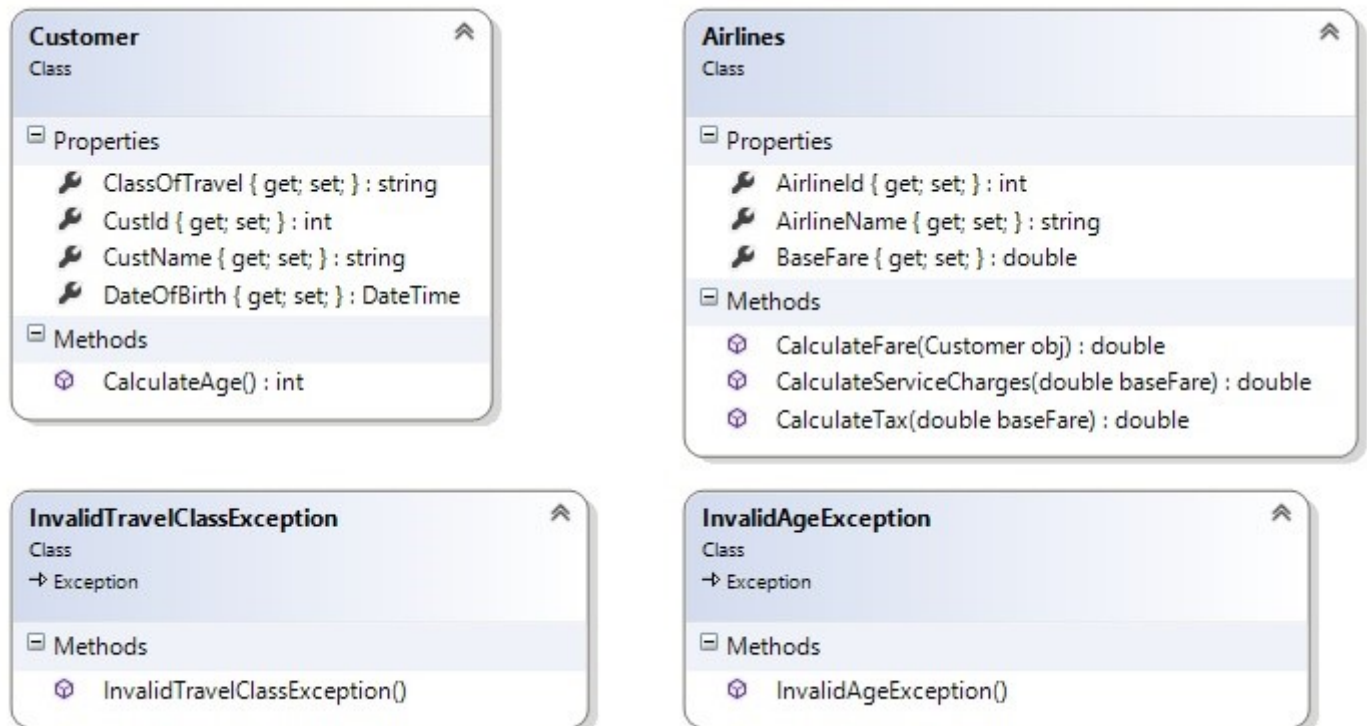


Problem Statement:

Objective: To understand the business requirements and to implement Exception Handling with User defined exceptions.

Problem Description: FastAirlines is a company that helps users to book Airline tickets. It wants to ensure that the users are prompted with proper messages whenever they go wrong during the process of booking.

Class Diagram:



Implementation Details:

Customer Class

`CustId`, `CustName` and `ClassOfTravel` are Auto-Implemented properties.

Airlines Class

`AirlineId`, `AirlineName` and `BaseFare` are Auto-Implemented properties.

CalculateServiceCharges(double baseFare):

The Service charges is calculated as 5% of the base fare and returned.

CalculateTax (double baseFare):

The tax is calculated as 15% of the base fare and returned.

CalculateFare (Customer obj):

- Invoke the CalculateAge method of customer class.
- If the age is greater than 0 and less than 100, then check
 - If the ClassOfTravel chosen by customer is Business, set the BaseFare to 10000.0
 - If the ClassOfTravel chosen by customer is Premium, set the BaseFare to 5000.0
 - If the ClassOfTravel chosen by customer is Economy, set the BaseFare to 1200.0
 - In all other cases InvalidTravelClass Exception has to be raised and handled
- Raise InvalidAgeException otherwise and handle it
- The totalAmount is calculated as BaseFare + ServiceCharges + Tax
- Return TotalAmount

InvalidTravelClassException Class

InvalidTravelClassException():

The constructor should display an appropriate message to choose particular class of travel.

InvalidAgeException Class

InvalidAgeException():

The constructor should display an appropriate message to prompt the user about invalid age.