Assignment4

Generated by Doxygen 1.8.10

Wed Dec 16 2015 18:16:10

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 AssignmentSysDev4 Namespace Reference

Classes

- class Account
- class AirbusA319100
- class AirbusA330300
- class Airline
- class Airport
- class Boeing767300ER
- class Boeing777200LR
- class Boeing777300ER

  This class defines a plane which extends the IPlane interface; other plane classes in this file have the same basic code, the different model numbers are specified for each one. This makes it possible to add more individual properties unique to a model of plane.
- class Boeing7878
- class Boeing7879
- class BombardierCRJ200
- class BombardierCRJ705
- class BombardierQ400
- class Brand
- class BusinessClassCancel

  A 'ConcreteStrategy' class This strategy class allows the purchase price of Business class tickets to be refunded if the ticket is cancelled within 96 hours, i.e. four days of puchase
- class CancelStrategy
- class CompositeElement

  Flights can also be used as Composite elements because one flight might be 'fixed', i.e. the customer has a narrow window of possible departure times at some point in their journey, but the primitive elements added to the list of the Composite element will be flights that are possible connections. The Composite pattern is used here because it allows for this kind of things, i.e. having a list of items with associated branches or 'leaf' items, and having the data grouped this way for flights will allow the system to show alternative flights to complete a journey.
- class Customer
- class EconomyCancel

  A 'ConcreteStrategy' class This stragegy class allows an Economy ticket's purchase price to be refunded if it is cancelled within 24 hours of purchase.
- class EmbraerE175
- class Employee
- class Flight
- class FlightClass

- interface IPlane

  Here we use the Factory design pattern to create planes. Each plane has a model, we allow the use of a simple integer model number. In a full system, planes would likely be picked from a list based on a database table.
- class Plane
- class PlaneCreator

  This Creator class uses a FactoryMethod to determine what type of plane object to set up.
- class PremiumEconomyCancel

  A 'ConcreteStrategy' class This strategy class allows a Premium Economy ticket's price to be refunded if the ticket is cancelled within 48 hours of purchase.
- class PrimitiveElement

  Flights can be primitive elements within the use of the composite pattern. This is because we wish to make lists of flights that offer alternative connections based on times, stopover times, etc. so customers have choices.
- class Program
- class Ticket

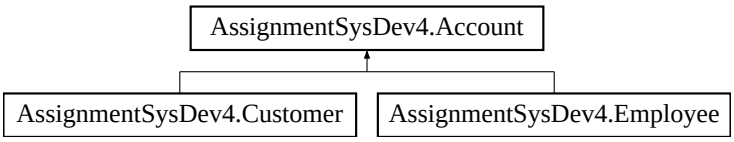
  The ticket class employs different cancellation strategies, set up in individual classes The use of the stragegy pattern for this would, potentially, allow other characteristics of the different ticket classes to be set up.

Chapter 5

Class Documentation

5.1 AssignmentSysDev4.Account Class Reference

Inheritance diagram for AssignmentSysDev4.Account:



Classes

- class **CustomerViewAccount**
- class **EmployeeViewAccount**
- class **Facade**

  A Facade is used here to allow Employees to see Detailed information about accounts, while customers have their own view of their account. This will allow the system to do things like allowing employees to simulate a customer's view and see data only to be shared with employees.

Public Member Functions

- void **ViewAccountInfo** (bool employeeInd, bool delinquentStatus)
- void **UpdateAccount** ()

Properties

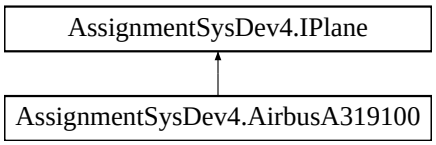
- Ticket **Ticket1** `[get, set]`
- bool **Delinquent** `[get, set]`
- bool **EmployeeInd** `[get, set]`

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Account.cs

5.2 AssignmentSysDev4.AirbusA319100 Class Reference

Inheritance diagram for AssignmentSysDev4.AirbusA319100:



Public Member Functions

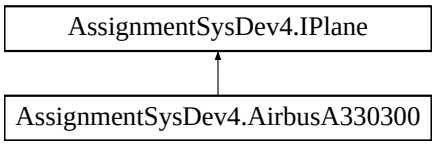
- string **GetModel** ()

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.3 AssignmentSysDev4.AirbusA330300 Class Reference

Inheritance diagram for AssignmentSysDev4.AirbusA330300:



Public Member Functions

- string **GetModel** ()

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.4 AssignmentSysDev4.Airline Class Reference

Public Member Functions

- void **AddAirline** ()
- void **ViewAirline** ()
- void **UpdateAirline** ()
- void **DeleteAirline** ()
- void **AddBrand** (Brand brandName)
- void **ViewBrand** ()
- void **UpdateBrand** ()
- void **DeleteBrand** ()

Static Public Member Functions

- static Airline **AirlineInstatnce** ()

Properties

- Brand **Brand** `[get, set]`

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Airline.cs

5.5 AssignmentSysDev4.Airport Class Reference

Public Member Functions

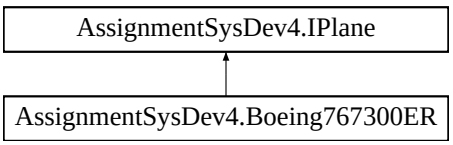
- void **AddNewTerminal** ()
- void **ViewAirport** ()
- void **DeleteAirport** ()
- void **UpdateAirport** ()

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Airport.cs

5.6 AssignmentSysDev4.Boeing767300ER Class Reference

Inheritance diagram for AssignmentSysDev4.Boeing767300ER:



Public Member Functions

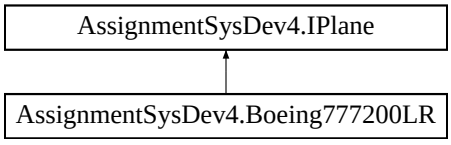
- string **GetModel** ()

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.7 AssignmentSysDev4.Boeing777200LR Class Reference

Inheritance diagram for AssignmentSysDev4.Boeing777200LR:



Public Member Functions

- string **GetModel** ()

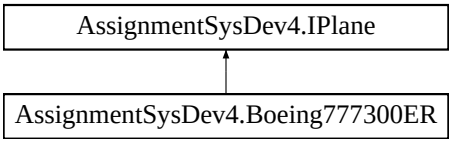
The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.8 AssignmentSysDev4.Boeing777300ER Class Reference

This class defines a plane which extends the IPlane interface; other plane classes in this file have the same basic code, the different model numbers are specified for each one. This makes it possible to add more individual properties unique to a model of plane.

Inheritance diagram for AssignmentSysDev4.Boeing777300ER:



Public Member Functions

- string **GetModel** ()

5.8.1 Detailed Description

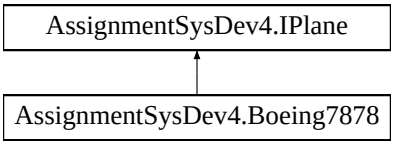
This class defines a plane which extends the IPlane interface; other plane classes in this file have the same basic code, the different model numbers are specified for each one. This makes it possible to add more individual properties unique to a model of plane.

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.9 AssignmentSysDev4.Boeing7878 Class Reference

Inheritance diagram for AssignmentSysDev4.Boeing7878:



Public Member Functions

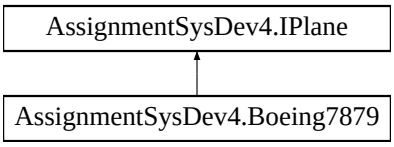
- string **GetModel** ()

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.10 AssignmentSysDev4.Boeing7879 Class Reference

Inheritance diagram for AssignmentSysDev4.Boeing7879:



Public Member Functions

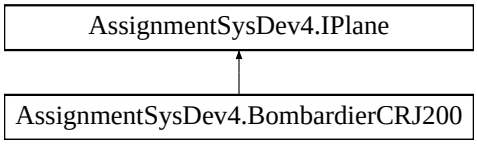
- string **GetModel** ()

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.11 AssignmentSysDev4.BombardierCRJ200 Class Reference

Inheritance diagram for AssignmentSysDev4.BombardierCRJ200:



Public Member Functions

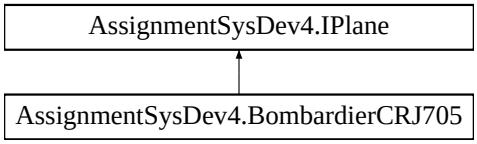
- string **GetModel** ()

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.12 AssignmentSysDev4.BombardierCRJ705 Class Reference

Inheritance diagram for AssignmentSysDev4.BombardierCRJ705:



Public Member Functions

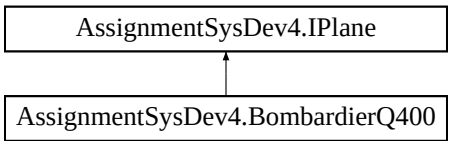
- string **GetModel** ()

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.13 AssignmentSysDev4.BombardierQ400 Class Reference

Inheritance diagram for AssignmentSysDev4.BombardierQ400:



Public Member Functions

- string **GetModel** ()

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.14 AssignmentSysDev4.Brand Class Reference

Public Member Functions

- void **AddBrand** ()
- void **UpdateBrand** ()
- void **DeleteBrand** ()
- void **ViewBrand** ()
- void **AddPlane** ()
- void AddPlanes (int modelNumber, int numberToAdd)

  Here we will take a modelNumber and a number of planes to add to a Brand. Then we can use methods defined in the PlaneCreator class to add those planes to the brand. PlaneCreator will use the Factory method.
- void **ViewPlane** ()
- void **UpdatePlane** ()
- void **DeletePlane** ()

Properties

- string **Name** `[get, set]`

5.14.1 Member Function Documentation

5.14.1.1 void AssignmentSysDev4.Brand.AddPlanes (int *modelNumber,* int *numberToAdd*)

Here we will take a modelNumber and a number of planes to add to a Brand. Then we can use methods defined in the PlaneCreator class to add those planes to the brand. PlaneCreator will use the Factory method.

Parameters

modelNumber	
numberToAdd	

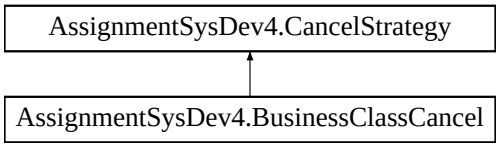
The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Brand.cs

5.15 AssignmentSysDev4.BusinessClassCancel Class Reference

A 'ConcreteStrategy' class This strategy class allows the purchase price of Business class tickets to be refunded if the ticket is cancelled within 96 hours, i.e. four days of puchase

Inheritance diagram for AssignmentSysDev4.BusinessClassCancel:



Public Member Functions

- override bool **Cancel** (Ticket ticketToCancel)

5.15.1 Detailed Description

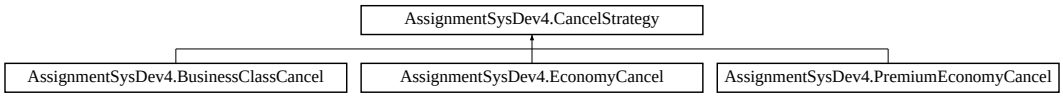
A 'ConcreteStrategy' class This strategy class allows the purchase price of Business class tickets to be refunded if the ticket is cancelled within 96 hours, i.e. four days of puchase

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Ticket.cs

5.16 AssignmentSysDev4.CancelStrategy Class Reference

Inheritance diagram for AssignmentSysDev4.CancelStrategy:



Public Member Functions

- abstract bool **Cancel** (Ticket ticketToCancel)

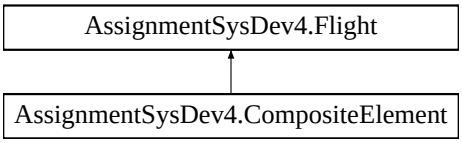
The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Ticket.cs

5.17 AssignmentSysDev4.CompositeElement Class Reference

Flights can also be used as Composite elements because one flight might be 'fixed', i.e. the customer has a narrow window of possible departure times at some point in their journey, but the primitive elements added to the list of the Composite element will be flights that are possible connections. The Composite pattern is used here because it allows for this kind of things, i.e. having a list of items with associated branches or 'leaf' items, and having the data grouped this way for flights will allow the system to show alternative flights to complete a journey.

Inheritance diagram for AssignmentSysDev4.CompositeElement:



Public Member Functions

- **CompositeElement** (Flight baseFlight)
- override void **Add** (Flight newConnection)
- override void **Remove** (Flight oldConnection)
- override void **Display** (int indent)

Additional Inherited Members

5.17.1 Detailed Description

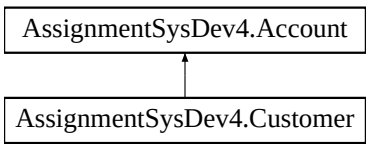
Flights can also be used as Composite elements because one flight might be 'fixed', i.e. the customer has a narrow window of possible departure times at some point in their journey, but the primitive elements added to the list of the Composite element will be flights that are possible connections. The Composite pattern is used here because it allows for this kind of things, i.e. having a list of items with associated branches or 'leaf' items, and having the data grouped this way for flights will allow the system to show alternative flights to complete a journey.

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Flight.cs

5.18 AssignmentSysDev4.Customer Class Reference

Inheritance diagram for AssignmentSysDev4.Customer:



Public Member Functions

- void **QueryFlightSchedual** ()
- void **ReserveTicket** ()
- void **PurchaseTicket** ()

- void **CancelTicket** ()
- void **UpgradeTicket** ()
- void **ChangeSeatLocation** ()
- void **ViewFlightHistory** ()
- void **CreateAccount** ()
- void **ViewFlight** ()
- void **CreateCustomerAccount** ()
- void **CreatePotentionalCustomerAccount** ()

Additional Inherited Members

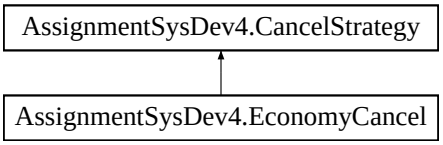
The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Customer.cs

5.19 AssignmentSysDev4.EconomyCancel Class Reference

A 'ConcreteStrategy' class This stragegy class allows an Economy ticket's purchase price to be refunded if it is cancelled within 24 hours of purchase.

Inheritance diagram for AssignmentSysDev4.EconomyCancel:



Public Member Functions

- override bool **Cancel** (Ticket ticketToCancel)

5.19.1 Detailed Description

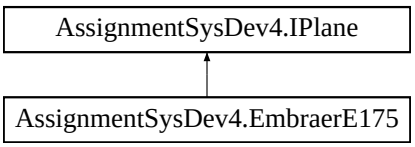
A 'ConcreteStrategy' class This stragegy class allows an Economy ticket's purchase price to be refunded if it is cancelled within 24 hours of purchase.

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Ticket.cs

5.20 AssignmentSysDev4.EmbraerE175 Class Reference

Inheritance diagram for AssignmentSysDev4.EmbraerE175:



Public Member Functions

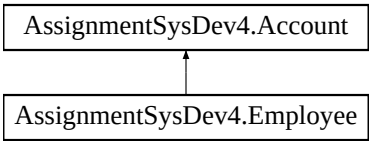
- string **GetModel** ()

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.21 AssignmentSysDev4.Employee Class Reference

Inheritance diagram for AssignmentSysDev4.Employee:



Public Member Functions

- void **AddPlane** ()
- void **AddClassFlight** ()
- void **SchedulaFlight** ()
- void **AddAirCraft** ()
- void **AddFlight** ()
- void **CreateEmployeeAccount** ()

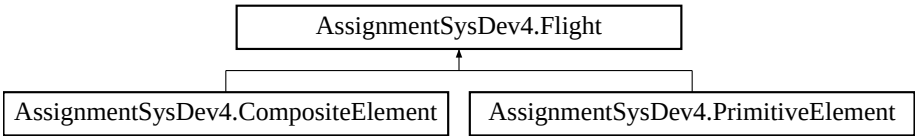
Additional Inherited Members

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Employee.cs

5.22 AssignmentSysDev4.Flight Class Reference

Inheritance diagram for AssignmentSysDev4.Flight:



Public Member Functions

- virtual void **Add** (Flight newFlight)
- virtual void **Remove** (Flight newFlight)
- virtual void **Display** (int indent)
- void **ViewFlight** ()
- void **DeleteFlight** ()

- void **UpdateFlight** ()
- void **AddFlight** ()
- void **AddFlightClass** ()
- void **UpdateFlightClass** ()
- void **DeleteFlightClass** ()
- void **UpdateAirport** ()

Public Attributes

- string **name**

Properties

- FlightClass **FlightClass** `[get, set]`
- string **FlightNumber** `[get, set]`
- Airport **Airport** `[get, set]`

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Flight.cs

5.23 AssignmentSysDev4.FlightClass Class Reference

Public Member Functions

- void **AddNewFlightClass** ()
- void **UpdateFlightClass** ()
- void **ViewFlightClass** ()
- void **DeleteFlightClass** ()

Properties

- Ticket **Ticket** `[get, set]`
- Ticket **Ticket1** `[get, set]`

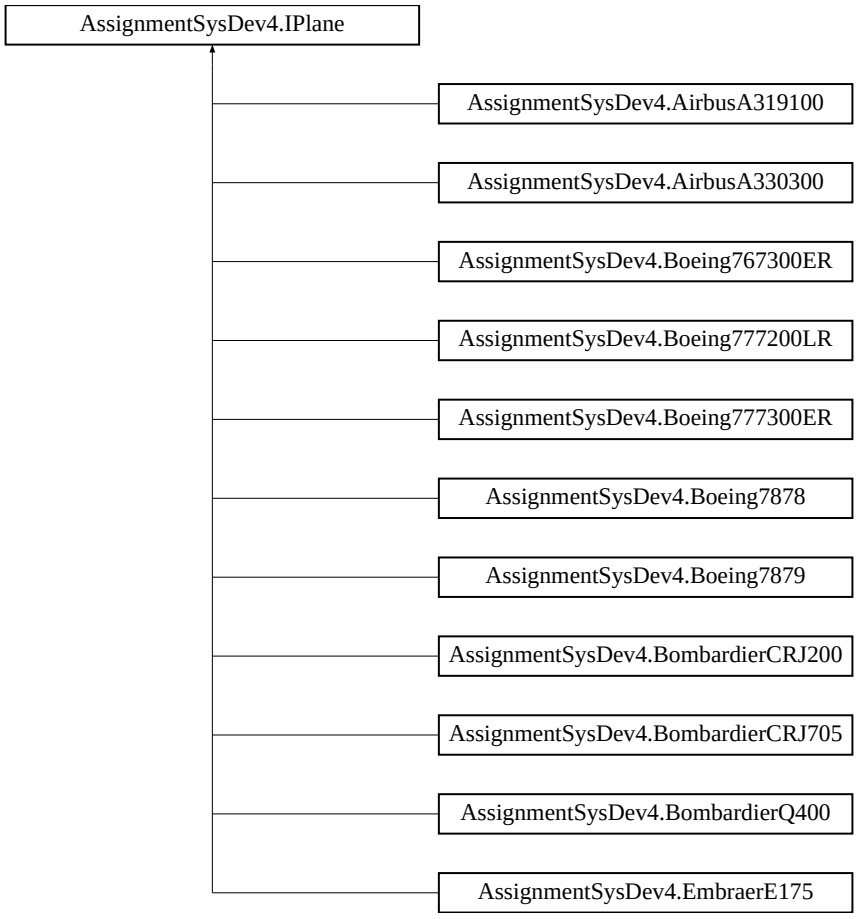
The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/FlightClass.cs

5.24 AssignmentSysDev4.IPlane Interface Reference

Here we use the Factory design pattern to create planes. Each plane has a model, we allow the use of a simple integer model number. In a full system, planes would likely be picked from a list based on a database table.

Inheritance diagram for AssignmentSysDev4.IPlane:



Public Member Functions

- string **GetModel** ()

5.24.1 Detailed Description

Here we use the Factory design pattern to create planes. Each plane has a model, we allow the use of a simple integer model number. In a full system, planes would likely be picked from a list based on a database table.

The documentation for this interface was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.25 AssignmentSysDev4.Plane.IPlane Interface Reference

Public Member Functions

- string **GetModel** ()

The documentation for this interface was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Plane.cs

5.26 AssignmentSysDev4.Plane Class Reference

Classes

- interface IPlane

Public Member Functions

- void **ViewPlane** ()
- void **UpdatePlane** ()
- void **DeletePlane** ()
- void **AddPlane** (int modelNumber)

Properties

- Flight **Flight** `[get, set]`

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Plane.cs

5.27 AssignmentSysDev4.PlaneCreator Class Reference

This Creator class uses a FactoryMethod to determine what type of plane object to set up.

Public Member Functions

- IPlane **FactoryMethod** (int modelNumber)

5.27.1 Detailed Description

This Creator class uses a FactoryMethod to determine what type of plane object to set up.

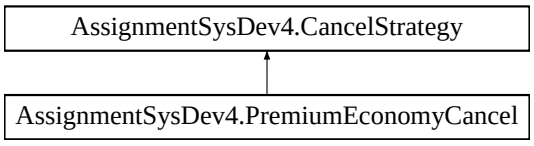
The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/PlaneCreator.cs

5.28 AssignmentSysDev4.PremiumEconomyCancel Class Reference

A 'ConcreteStrategy' class This strategy class allows a Premium Economy ticket's price to be refunded if the ticket is cancelled within 48 hours of purchase.

Inheritance diagram for AssignmentSysDev4.PremiumEconomyCancel:



Public Member Functions

- override bool **Cancel** (Ticket ticketToCancel)

5.28.1 Detailed Description

A 'ConcreteStrategy' class This strategy class allows a Premium Economy ticket's price to be refunded if the ticket is cancelled within 48 hours of purchase.

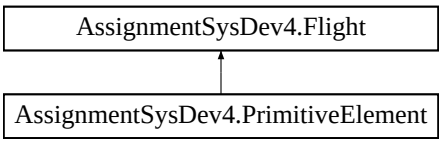
The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Ticket.cs

5.29 AssignmentSysDev4.PrimitiveElement Class Reference

Flights can be primitive elements within the use of the composite pattern. This is because we wish to make lists of flights that offer alternative connections based on times, stopover times, etc. so customers have choices.

Inheritance diagram for AssignmentSysDev4.PrimitiveElement:



Public Member Functions

- **PrimitiveElement** (Flight newFlight)
- override void **Add** (Flight newFlight)
- override void **Remove** (Flight wrongFlight)
- override void **Display** (int indent)

Additional Inherited Members

5.29.1 Detailed Description

Flights can be primitive elements within the use of the composite pattern. This is because we wish to make lists of flights that offer alternative connections based on times, stopover times, etc. so customers have choices.

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Flight.cs

5.30 AssignmentSysDev4.Program Class Reference

5.30.1 Detailed Description

An executive employee may want to create a batch of employee Accounts all at once Sometimes companies will hire people for lower level jobs in groups so they can go through orientation and training as a group.

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Program.cs

5.31 AssignmentSysDev4.Ticket Class Reference

The ticket class employs different cancellation strategies, set up in individual classes The use of the stragegy pattern for this would, potentially, allow other characteristics of the different ticket classes to be set up.

Public Member Functions

- void **SetCancelStrategy** (CancelStrategy cancelstrategy)
- void **Cancel** ()

Properties

- DateTime **PurchaseDate** `[get, set]`

5.31.1 Detailed Description

The ticket class employs different cancellation strategies, set up in individual classes The use of the stragegy pattern for this would, potentially, allow other characteristics of the different ticket classes to be set up.

The documentation for this class was generated from the following file:

- G:/INFO8240 - Systems Design/Assignment4/AssignmentSysDev4/AssignmentSysDev4/Ticket.cs

Index