Homework 5 Design Algorithm

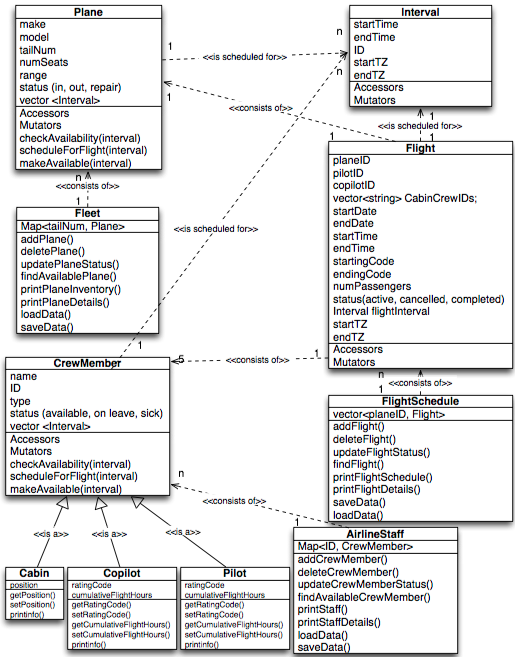
*Riya Danait*

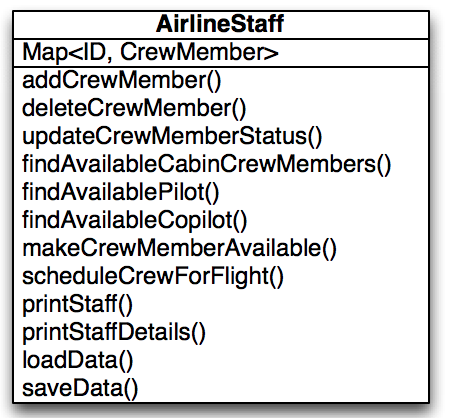
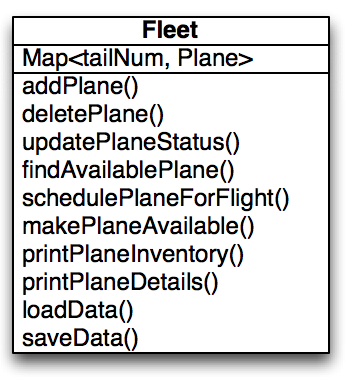
*Professor Keathly*

*CSCE 1040.002*

*4/20/2018*

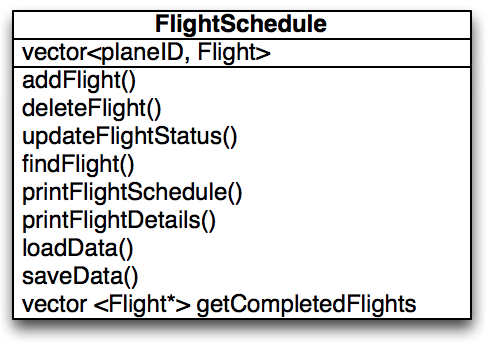
Update Homework 5 Diagram Set



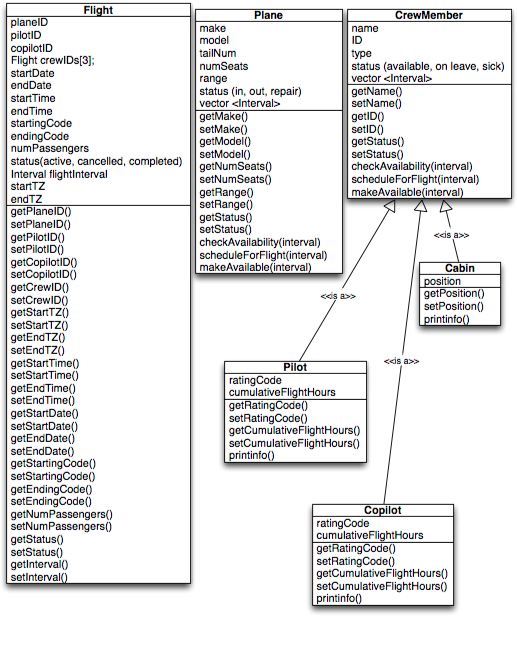
 

class diagram for AirlineStaff

class diagram Fleet



class diagram for FlightSchedule



Updated Homework 5 Design Algorithm

In Homework 5, I am designing a system to schedule flight crews and aircraft for Mean Green Airlines. There will be 3 entities called Flight, Plane, and CrewMember, with collections FlightSchedule, Fleet, and AirlineStaff. In addition, there will be an Interval class to manage the times of the flights. The derived classes for Pilot, CoPilot, and Cabin inherit from the base class CrewMember have an **is-a relationship**.

The following pseudocode describes the use case scenarios and algorithms for 18 functions:

1. Adding a Flight to FlightSchedule

if adding flight:

find an available Plane with sufficient capacity given numPassengers and start/end time and

date from user input along with an **IANA format time zone**.

if no plane is available (all planes are already scheduled for given date/time, or not enough

passenger capacity in free planes):

return saying Flight cannot be scheduled.

else:

Plane is available. Get Pilot, Copilot, 3 crew members from Staff given start/end date time

If no crew members are available (all assigned to other flights at same time or out sick):

Return saying Flight cannot be scheduled

Else:

Plane and Crew are available. Schedule the flight by allocating a Flight object.

Store Plane ID, Crew IDs, start/end date and time, and source and destination

airports in the Flight object.

Add Flight object to the FlightSchedule map.

Mark crew and aircraft as “busy” for the given date and time period

1. Updating a Flight in FlightSchedule

if updating flights:

go through all flights in FlightSchedule to determine which are active based on current date/time

If Flight is ACTIVE:

if current time >= end time:

Mark Flight as COMPLETED

Remove Flight from FlightSchedule

Delete Flight Object.

Mark corresponding Crew IDs and Plane ID as “free” so that they can be

scheduled for the next Flight.

else:

if Flight is SCHEDULED:

if current time >= start time:

Mark Flight as ACTIVE

else:

Process the next Flight in the FlightSchedule

1. Deleting a Flight from FlightSchedule

if deleting Flight:

get Plane ID of the Flight to be deleted, from user input

Find the matching Flight in FlightSchedule map based on the Plane ID

If Flight not found in Flight Schedule

Return no such Flight

Else:

Check Flight status by comparing current time to begin/end time of the flight.

If Flight is ACTIVE:

Return Flight cannot be deleted as it is currently Active.

If Flight is SCHEDULED or COMPLETED:

Remove Flight from FlightSchedule map

Delete the Flight Object.

Mark Plane ID and crew IDs as “free” so that they can be scheduled for a

another Flight.

1. Adding a Crew Member to AirlineStaff

if adding a crew member:

get user input on type of crew member (Pilot, CoPilot, Cabin)

based on type, allocate an object of the corresponding derived class

get user input for crew ID

find crew member with ID

if crew member does not exist:

get user input for Crew Name, Type and Status

else

return saying Crew member already exists

1. Deleting a Crew Member from AirlineStaff

if deleting crew member:

get crew ID from user input

find matching crew member with the ID

check if crew member is scheduled on a flight

if not:

remove crew member from airline staff

else:

return saying crew member is already scheduled for a flight

1. Printing the details of a Crew Member in AirlineStaff

if printing details for a singular crew member:

find ID

if crew member does not exist:

return saying crew member does not exist

else:

print details for crew member with that ID

1. Adding a Plane to Fleet

if adding an aircraft:

get user input for tailNum

find aircraft with tailNum

if aircraft does not exist:

get user input for make, model, numSeats, range, status

else

return saying aircraft already exists

1. Deleting a Plane from Fleet

if deleting aircraft:

get tailNum from user input

find matching plane with tailNum

check to see if aircraft is ACTIVE

if not:

remove plane from aircrafts

else:

return saying the aircraft is currently active

1. Printing the details of a Plane from Fleet

if printing details for a plane:

find tailNum

if tailNum does not exist:

return saying aircraft does not exist

else:

print details for plane with that tailNum

1. Loading and storing data (happens when the program runs)

if program is starting:

prompt user for the name of the input file containing details for all collections and entities

if closing the program:

prompt the user to enter the name of the output file to output the Mean Green system details

1. Updating a CrewMember Status in AirlineStaff

if updating CrewMember:

go through all of AirlineStaff to determine who all are AVAILABLE based on current status

If CrewMember is AVAILABLE:

Mark CrewMember as either ON LEAVE or SICK depending on user input

Mark corresponding CrewMember ID as “busy” or “unavailable”

else:

if CrewMember is ON LEAVE or SICK:

Mark CrewMember as AVAILABLE depending on user input

Mark corresponding CrewMember ID as “free” or “available”

else:

continue through AirlineStaff

1. Updating a Plane Status in Fleet

if updating Plane:

go through all of Fleet to determine which are IN based on current status

If Plane is IN:

Mark Plane as either OUT or REPAIR depending on user input

Mark corresponding plane ID as “busy” or “unavailable”

else:

if Plane is OUT or REPAIR:

Mark Plane as IN depending on user input

Mark corresponding Plane ID as “free” or “available”

else:

continue through Fleet

1. Updating FlightSchedule based on Current Time

Find all completed flights based on the current time

For each completed flight, make the plane, pilot, crew members available

Delete flight from the flight schedule

1. Print a list of Flights based on Status

if Flight is ACTIVE:

print that Flight with that Flight ID is active and plane cannot be scheduled at the currentTime until after the endTime

mark ID as “busy”

if Flight is CANCELLED:

print that Flight with that Flight ID is cancelled and plane/crew members can be used to schedule another flight

mark ID as “free”

if Flight is COMPLETED:

print that Flight with that Flight ID is completely and plane/crew members can be used to schedule another flight

mark ID as “free”

1. Printing the details of a Single Flight in FlightSchedule

if printing the details for a singular flight:

Find Flight from FlightSchedule given Plane ID from user input.

if Flight not found in FlightSchedule:

Return saying Flight for the given Plane ID does not exist

else:

Print details of the Flight for the given Plane ID

1. Printing the entire FlightSchedule

if printing all entries in a flight schedule:

print all

1. Printing the entire AirlineStaff

if printing all entries in airline staff:

print all

1. Printing the entire Fleet

if printing all entries in fleet:

print all