**FLIGHT CLASS**

class Flight

{

private int flightNumber;

private string origin;

private string destination;

private int maxSeats;

private int numPassengers;

private Customer[] passengers;

public Flight(int fn,string or,string dest,int mSeats)

{

maxSeats = mSeats;

flightNumber = fn;

origin = or;

destination = dest;

numPassengers = 0;

passengers = new Customer[maxSeats];

}

public int getFlightNumber() { return flightNumber; }

public string getOrigin() { return origin; }

public string getDestination() { return destination; }

public int getMaxSeats() { return maxSeats; }

public int getNumPassengers() { return numPassengers; }

public bool addPassenger(Customer c)

{

if (numPassengers >= maxSeats) { return false; }

passengers[numPassengers] = c;

numPassengers++;

return true;

}

public int findPassenger(int custId)

{

for (int x = 0; x < maxSeats; x++)

{

if (passengers[x].getId() == custId)

return x;

}

return -1;

}

public bool removePassenger(int custId)

{

int loc = findPassenger(custId);

if (loc == -1) return false;

passengers[loc] = passengers[numPassengers - 1];

numPassengers--;

return true;

}

public string getPassengerList()

{

string s = "\nPassengers on flight " + flightNumber + ":";

for (int x = 0; x < numPassengers; x++)

{

s = s + "\n" + passengers[x].getFirstName() + " " + passengers[x].getLastName();

}

return s;

}

public string toString()

{

string s = "Flight Number: " + flightNumber;

s = s + "\nOrigin: " + origin;

s = s + "\nDestination:" + destination;

s = s + "\nNumber of Passengers:" + numPassengers;

s = s + "\nAvailable seats:" + (maxSeats - numPassengers);

s = s + getPassengerList();

return s;

}

}

**FLIGHT MANAGER CLASS**

class FlightManager

{

private int maxFlights;

private int numFlights;

private Flight[] flightList;

public FlightManager(int max)

{

maxFlights = max;

numFlights = 0;

flightList = new Flight[maxFlights];

}

public bool addFlight(int fn, string origin, string destination,int maxSeats)

{

if (numFlights >= maxFlights) { return false; }

Flight f = new Flight(fn, origin,destination,maxSeats);

flightList[numFlights] = f;

numFlights++;

return true;

}

public int findFlight(int fid)

{

for (int x = 0; x < numFlights; x++)

{

if (flightList[x].getFlightNumber() == fid)

return x;

}

return -1;

}

public bool flightExists(int fid)

{

int loc = findFlight(fid);

if (loc == -1) { return false; }

return true;

}

public Flight getFlight(int fid)

{

int loc = findFlight(fid);

if (loc == -1) { return null; }

return flightList[loc];

}

public bool deleteFlight(int fid)

{

int loc = findFlight(fid);

if (loc == -1) { return false; }

flightList[loc] = flightList[numFlights-1];

numFlights--;

return true;

}

public string getFlightList()

{

string s = "Flight List:";

for (int x = 0; x < numFlights; x++)

{

s = s + "\n" + flightList[x].getFlightNumber() + " from " + flightList[x].getOrigin() + " to " + flightList[x].getDestination();

}

return s;

}

}

**CUSTOMER CLASS**

class Customer

{

private int customerId;

private string firstName;

private string lastName;

private string phone;

private int bookings;

public Customer(int cId,string fname,string lname, string ph)

{

bookings = 0;

customerId = cId;

firstName = fname;

lastName = lname;

phone = ph;

}

public int getId() { return customerId; }

public string getFirstName() { return firstName; }

public string getLastName() { return lastName; }

public string getPhone() { return phone; }

public int getNumBookings() { return bookings; }

public string toString()

{

string s = "Customer "+customerId;

s = s + "\nName: "+firstName+" "+lastName;

s = s + "\nPhone: "+phone;

s = s + "\n Bookings: " + bookings;

return s;

}

}

**AIRLINE COORDINATOR CLASS**

class AirlineCoordinator

{

FlightManager flManager;

CustomerManager custManager;

public AirlineCoordinator(int custIdSeed, int maxCust,int maxFlights)

{

flManager = new FlightManager(maxFlights);

}

public bool addFlight(int flightNo,string origin,string destination, int maxSeats)

{

return flManager.addFlight(flightNo, origin, destination, maxSeats);

}

public string flightList()

{

return flManager.getFlightList();

}

public bool deleteFlight(int fid)

{

return flManager.deleteFlight(fid);

}

}

**MAIN PROGRAM**

class Program

{

static AirlineCoordinator aCoord;

public static void deleteFlight()

{

int id;

Console.Clear();

Console.WriteLine(aCoord.flightList());

Console.Write("Please enter a flight id to delete:");

id = Convert.ToInt32(Console.ReadLine());

if (aCoord.deleteFlight(id))

{

Console.WriteLine("Flight with id {0} deleted..", id);

}

else

{

Console.WriteLine("Flight with id {0} was not found..", id);

}

Console.WriteLine("\nPress any key to continue return to the main menu.");

Console.ReadKey();

}

public static void viewFlights()

{

Console.Clear();

Console.WriteLine(aCoord.flightList());

Console.WriteLine("\nPress any key to continue return to the main menu.");

Console.ReadKey();

}

public static void addFlight(){

int flightNo,maxSeats;

string origin, destination;

Console.Clear();

Console.WriteLine("-----------Add Flight----------");

Console.Write("Please enter the flight number:");

flightNo = Convert.ToInt32(Console.ReadLine());

Console.Write("Please the maximum number of seats:");

maxSeats = Convert.ToInt32(Console.ReadLine());

Console.Write("Please enter the port of Origin:");

origin = Console.ReadLine();

Console.Write("Please enter the destination port:");

destination = Console.ReadLine();

if(aCoord.addFlight(flightNo, origin, destination, maxSeats))

{

Console.WriteLine("Flight successfully added..");

}

else

{

Console.WriteLine("Flight was not added..");

}

Console.WriteLine("\nPress any key to continue return to the main menu.");

Console.ReadKey();

}

public static void showMainMenu()

{

Console.Clear();

Console.WriteLine("XYZ AirLines Limited.\nPlease select a choice from the menu below:\n");

Console.WriteLine("1: Add Flight\n2 :Add Customer\n3: View Flights\n4: View Customers");

Console.WriteLine("5: Delete Customer\n6: Delete Flight");

Console.WriteLine("7:Exit");

}

public static int getValidChoice()

{ int choice;

showMainMenu();

while (!int.TryParse(Console.ReadLine(), out choice) || (choice <1 || choice > 7))

{

showMainMenu();

Console.WriteLine("Please enter a valid choice:");

}

return choice;

}

public static void runProgram()

{

int choice= getValidChoice();

while (choice != 7)

{

if (choice == 1) { addFlight(); }

if (choice == 2) { }

if (choice == 3) { viewFlights(); }

if (choice == 4) { }

if (choice == 5) { }

if (choice == 6) { deleteFlight(); }

choice = getValidChoice();

}

}

static void Main(string[] args)

{

aCoord = new AirlineCoordinator(100, 2, 30);

runProgram();

Console.WriteLine("Thank you for using XYZ Airlines System. Press any key to exit.");

Console.ReadKey();

}

}