using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CarDealer

{

/// This class manages all of the car data and

/// exposes all the functions needed for the program.

public class CarManagment

{

//create array

private readonly int maxCars = 21;

Car[] Cars;

public CarManagment()

{

// add value to array

Cars = new Car[maxCars];

Cars [1] = new Car(1,"FGR45G", "German", "Mercedes", 2008, 11000, "fgjh");

Cars[13] = new Car(13, "FGR45G", "German", "Mercedes", 2008, 15000, "fgjh");

Cars[9] = new Car(9, "FGR45G", "German", "Mercedes", 2008, 12000, "fgjh");

Cars[10] = new Car(10, "FGR45G", "German", "Mercedes", 2008, 35000, "fgjh");

}

//

public Car[] GetCars()

{

//return car in array

return Cars;

}

public void SetCar(Car[] newValue)

{

//set a new car

Cars = newValue;

}

public Car GetCar(int bay)

{

//return car using position

return Cars[bay];

}

public Car GetCarPrice(int price)

{

//return car using position

return Cars[price];

}

public bool AddCar(Car carToAdd)

{

//validate my car

//This spot is NOT free to add an car return

if (Cars[carToAdd.bay] != null)

{

return false;

}

Cars[carToAdd.bay] = carToAdd;

return true;

}

public void RemoveCar(Car carToRemove)

{

//remove from array

Cars[carToRemove.bay] = null;

}

public int FindCar(Car [] Cars, int x)

//

{

for (int i =0;i <Cars.Length; i++)

{

if(Cars[i] != null)

if (Cars[i].GetPrice() == x)

{

return i;

}

}

return -1;

}

public int BinarySearch(int[] ArrayForSearch, int l, int r, int x)

{

if (r >= l)

{

int mid = l + (r - l) / 2;

/\* if the element is present

at the middle itself \*/

if (ArrayForSearch[mid] == x)

return mid;

/\* If element is smaller than mid, then

it can only be present in left subarray \*/

if (ArrayForSearch[mid] > x)

return BinarySearch(ArrayForSearch, l, mid - 1, x);

return BinarySearch(ArrayForSearch, mid + 1, r, x);

}

return -1;

}

}

}