WRAV102/MSEV102: Practical 7

*Submit in Moodle before Monday 10 October 14:00*

**Add your name, surname and student number in the space provided below:**

Name and Surname: Ayabonga Kemese

Student number:s225175614

**Combine all your tasks into one file to submit in Moodle:**

1. Add your personal details in the space provided above.
2. Copy your code from the .cs file in Visual Studio to the provided space under the relevant heading. *Copy all your code – from line 1 up to the last curly bracket (remember that your will have more than one file containing code for the task).*
3. Use the SAVE AS functionality to save your document with your student number in the filename – P7-*studentnumber.*docx (e.g. P7-*224390000*, *where you have your student number after the -*)

**Submit your file in Moodle:**

1. Return to the WRAV102/MSEV102 Moodle site – upload your Word document to the Prac 7 link in the *Week 10* section.

**Task**

**Solution Code:**

**Car.cs file**

*<<Copy and paste your code from the Car.cs file here>>*

class carClass

{

string name, available;

int registorNo;

double rent;

public carClass(int registorNo, string name, string available, double rent)

{

this.registorNo = registorNo;

this.name = name;

this.available = available;

this.rent = rent;

}

}

**CarPool.cs file**

*<<Copy and paste your code from the CarPool.cs file here>>*

class carPool

{

static int sortstate;

List<carClass> carinfo = new List<carClass>();

public static object sortedstate { get; internal set; }

public carPool()

{

List<carClass> carinfo = new List<carClass>();

sortstate = -1;

}

public static void streamread(List<carClass> carInfo)

{

string[] fields;

const Char DELIM = ',';

string inputline, name, initial;

int registorNo;

double balance;

StreamReader SR = new StreamReader("patientinfo.dat");

inputline = SR.ReadLine();

while (inputline != null)

{

fields = inputline.Split(separator: DELIM);

registorNo = Convert.ToInt32(fields[0]);

name = fields[1];

initial = fields[2];

balance = Convert.ToDouble(fields[3]);

carClass patient = new carClass(registorNo, name, initial, balance);

carInfo.Add(patient);

inputline = SR.ReadLine();

}

SR.Close();

}

public static int getRegNo()

{

Console.Write("Enter cars registor number : ");

int carNo = Convert.ToInt32(Console.ReadLine());

return carNo;

}

public static void carAdd(List<carClass> carInfo)

{

carInfo.Count();

string name, available;

int carNo;

double balance;

carNo = getRegNo();

Console.Write("Enter cars name : ");

name = Console.ReadLine();

name.ToUpper();

Console.Write("Is car still available (Y/N) : ");

available = Console.ReadLine();

available.ToUpper();

Availability(available, carInfo.Count(), name, carInfo);

Console.Write("Enter car price: ");

balance = Convert.ToDouble(Console.ReadLine());

carClass patient = new carClass(carNo, name, available, balance);

carInfo.Add(patient);

}

public static void carsell(List<carClass> carInfo)

{

string name, available;

int carNo;

double balance;

Console.Write("Enter cars registor number : ");

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

Console.Write("Enter cars name : ");

name = Console.ReadLine();

name.ToUpper();

Console.Write("Is car still available (Y/N) : ");

available = Console.ReadLine();

available.ToUpper();

Console.Write("Enter car price: ");

balance = Convert.ToDouble(Console.ReadLine());

carClass patient = new carClass(carNo, name, available, balance);

carInfo.Add(patient);

}

public static void Availability(string availablity, int pos, string name, List<carClass> carInfo)

{

List<string> avail = new List<string>();

if (availablity == "Y")

avail[pos]=name;

Sell(avail,pos,carInfo);

}

public static void Sell(List<string> availcars, int pos, List<carClass> carInfo)

{

Console.WriteLine("Choose car to sell :");

foreach (string car in availcars)

{

pos = availcars.IndexOf;

Console.WriteLine(pos+" "+ car);

}

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

pos--;

Console.WriteLine(carInfo[pos]);

Console.WriteLine("sell car Y/N");

string sell = Console.ReadLine();

sell.ToUpper();

if (sell == "Y")

{ carInfo.RemoveAt(pos);}

}

public static void BubbleSortAsc(List<carClass> carInfo)

{

for(int pass = 0; pass < carInfo.Count; pass++)

{

for(int comp = 1; comp <=carInfo.Count-pass; comp++)

{ carClass car1 = (carClass)carInfo[comp-1];

carClass car2 = (carClass)carInfo[comp];

if (car1.getRegNo().CompareTo(car2.getRegNo()) > 0)

swap(comp - 1, comp);

}

}

sortstate = 1;

Console.WriteLine(sortstate);

}

public static void SelectionSortAsc(List<carClass> carInfo)

{

int minPos = 0;

for (int pass = 0; pass < carInfo.Count; pass++)

{

for (int comp = 1; comp < carInfo.Count-pass; comp++)

{

carClass car1 = (carClass)carInfo[comp];

carClass car2 = (carClass)carInfo[minPos];

if (car1.CompareTo(car2) < 0)

minPos = comp;

}

swap(pass-1,minPos);

minPos = pass;

sortstate = 2;

}

}

public static void insertSort(List<carClass> carInfo)

{

for (int pass = 1; pass < carInfo.Count; pass++)

{

carClass newOne = (carClass)carInfo[pass];

int curPos = pass--;

carClass car2 = (carClass)carInfo[curPos];

while ((curPos != -1) && (newOne.getRegNo()<car2.getRegNo()))

{

curPos--;

if (curPos != -1)

car2 = (carClass)carInfo[curPos];

}

carInfo.RemoveAt(pass);

carInfo.Insert(curPos++, newOne);

}

}

private static void swap(int swap, int swap2)

{

carClass temp = (carClass)carinfo[swap2];

carinfo[swap2] = carinfo[swap];

carinfo[swap] = temp;

}

}

**Program.cs file**

*<<Copy and paste your code from the Program.cs file here>>*

internal class Program

{

static void Main(string[] args)

{ List< carClass> garage = new carClass();

int menu = 0;

while (menu != 10)

{

Mainmenu();

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

switch (menu)

{

case 1:

carPool.carAdd(garage);

break;

case 2:

Console.WriteLine(carPool.sortedstate);

break;

case 3:

carPool.streamread(garage);

break;

case 4:

carPool.BubbleSortAsc(garage);

break;

case 5:

carPool.insertSort(garage);

break;

case 6:

carPool.SelectionSortAsc(garage);

break;

case 7:

garage.Remove();

break;

case 8:

carPool.Availability(string a, int pos, string name, garage);

break;

case 9:

garage.Find();

}

}

}

static void Mainmenu()

{

Console.WriteLine("Choose menu Item :");

Console.WriteLine("1. Add car");

Console.WriteLine("2. view sorted method");

Console.WriteLine("3. update list");

Console.WriteLine("4. sort by register number");

Console.WriteLine("5. sort name");

Console.WriteLine("6. sort by daily price");

Console.WriteLine("7. sell car");

Console.WriteLine("8. display available car");

Console.WriteLine("9. find a car");

Console.WriteLine("10 to quit app");

}

}