***Лабараторна робота №5***

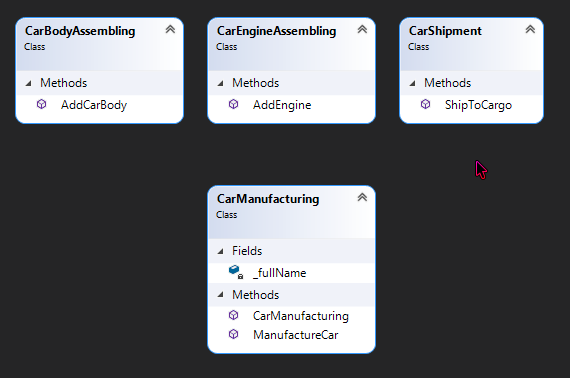
***Варіант №14***

**Завдання №1**

**14.** В якості  предметної області використовується процес виробництва машин. У системі виробництва машин існує 3 підсистеми: Підсистема зборки кузова з методом AddCarBody() метод, в якому описуються операції вибору типу кузова, кольору, шасі. Підсистема зборки двигуна з методом AddEngіne(), у якому задається параметри двигуна. Підсистема відвантаження готового автомобіля з методом ShіpToCargo() де розраховується дата відвантаження, а саме: поточна дата+ 10 днів. У конструкторі класу Фасад приймається значення ПІБ покупця. Користувачу видається інформація про те, коли він одержить машину.

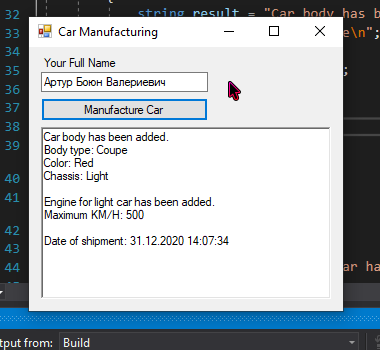
***Результати виконання завдання 1:***

*Діаграма класів додатку (рис.1):*



**Рис. 1 Діаграма класів додатку**

*Результати роботи додатку (рис.2)*



**Рис. 2 Результат виконаня додатку**

*Лістинг програмного коду:*

*\*\* MainForm.cs \*\**

using System.Windows.Forms;

namespace Lab\_5\_1

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

private void button1\_Click(object sender, System.EventArgs e)

{

if (textBox1.Text.Length <= 10)

MessageBox.Show("Full name must contain more than 10 characters", "Error");

else

{

CarManufacturing carManufacturing = new CarManufacturing(textBox1.Text);

richTextBox1.Text = carManufacturing.ManufactureCar();

}

}

}

}

*\*\* CarManufacturing.cs \*\**

using System;

namespace Lab\_5\_1

{

class CarManufacturing // Facade class

{

private readonly string \_fullName;

public CarManufacturing(string fullName)

{

\_fullName = fullName;

}

public string ManufactureCar()

{

string result = "";

CarBodyAssembling carBodyAssembling = new CarBodyAssembling();

CarEngineAssembling carEngineAssembling = new CarEngineAssembling();

CarShipment carShipment = new CarShipment();

result += carBodyAssembling.AddCarBody();

result += "\n\n";

result += carEngineAssembling.AddEngіne();

result += "\n\n";

result += carShipment.ShіpToCargo();

return result;

}

}

class CarBodyAssembling

{

public string AddCarBody()

{

string result = "Car body has been added.\n";

result += "Body type: Coupe\n";

result += "Color: Red\n";

result += "Chassis: Light";

return result;

}

}

class CarEngineAssembling

{

public string AddEngіne()

{

return "Engine for light car has been added.\nMaximum KM/H: 500";

}

}

class CarShipment

{

public string ShіpToCargo()

{

DateTime dateOfShipment = DateTime.Now.AddDays(10);

return $"Date of shipment: {dateOfShipment}";

}

}

}

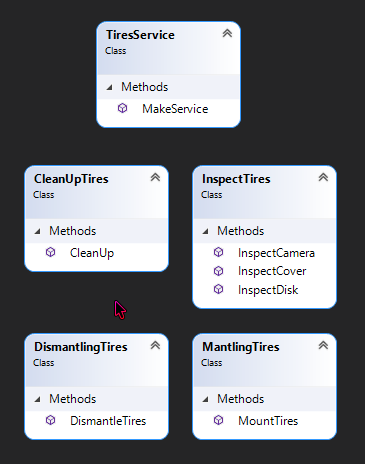
**Завдання №2**

Застосувати патерн Фасад для предметної області за варіантами.

(14. Шиномонтаж)

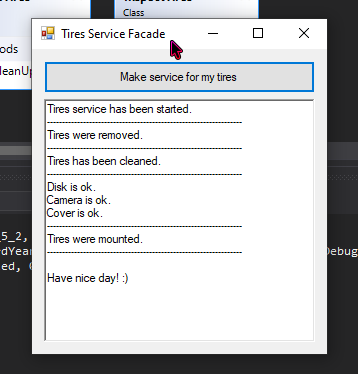
***Результати виконання завдання 2:***

*Діаграма класів додатку (рис.3):*



**Рис. 3 Діаграма класів додатку**

*Результати роботи додатку (рис.4)*



**Рис. 4 Результат виконаня додатку**

*Лістинг програмного коду:*

*\*\* MainForm.cs \*\**

using System;

using System.Windows.Forms;

namespace Lab\_5\_2

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

TiresService tiresService = new TiresService();

var result = tiresService.MakeService();

result += "\nHave nice day! :)";

richTextBox1.Text = result;

}

}

}

*\*\* TiresService.cs \*\**

namespace Lab\_5\_2

{

public class TiresService

{

public string MakeService()

{

string divider = "\n-----------------------------------------------------------------\n";

string result = "Tires service has been started.";

result += divider;

DismantlingTires dismantlingTires = new DismantlingTires();

MantlingTires mantlingTires = new MantlingTires();

CleanUpTires cleanUpTires = new CleanUpTires();

InspectTires inspectTires = new InspectTires();

result += dismantlingTires.DismantleTires();

result += divider;

result += cleanUpTires.CleanUp();

result += divider;

result += inspectTires.InspectDisk();

result += "\n";

result += inspectTires.InspectCamera();

result += "\n";

result += inspectTires.InspectCover();

result += divider;

result += mantlingTires.MountTires();

result += divider;

return result;

}

}

public class DismantlingTires

{

public string DismantleTires()

{

return "Tires were removed.";

}

}

public class MantlingTires

{

public string MountTires()

{

return "Tires were mounted.";

}

}

public class CleanUpTires

{

public string CleanUp()

{

return "Tires has been cleaned.";

}

}

public class InspectTires

{

public string InspectDisk()

{

return "Disk is ok.";

}

public string InspectCover()

{

return "Cover is ok.";

}

public string InspectCamera()

{

return "Camera is ok.";

}

}

}