Windows Based Car Selling System Using Concept of Late Binding in Reflection

TABLE OF CONTENTS

ABSTRACT

We will designing the windows application using the concept of late binding and reflection for Car Shop. In this application the user will be able to select vendors models of the cars and its colour and the count of cars the user wish to buy and according the

user can make order of its choice. All the ordered cars will come in a sold cars option that will be available in the interface of the application and we will provide the option to sell with the help of sell button .This application will be having the features of reflection, late binding and properties. We will also make a refresh button option to refresh the updates of the cars which are available. Using the windows form console we will design the interface and the required buttons and text field and listbox.

Reflection concept in C# is the ability to inspect metadata of assembly at runtime meaning assembly content is described by looking at the assembly metadata at run time namespace.

Its advantages includes- It allows view attribute information at runtime, it allows examining various types in an assembly
and instantiate these types, it allows late binding to methods and properties, it allows creating new types at runtime and then
performs some tasks using those types.

Implementation

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using CarShop.Common;
using System.IO;
using System.Reflection;

namespace CarShop
{
    public partial class Shop : Form
    {
        private IList<CommonInterfacesClasses.IVentor> vendors;
        private IList<CommonInterfacesClasses.ICarModel> curentModels;
        private List<Color> currentColors;
        private List<CommonInterfacesClasses.OrderedCar> orderList;
        private List<CommonInterfacesClasses.OrderedCar> finishList;
        public Shop()
        {
            InitializeComponent();

            this.finishList = new List<CommonInterfacesClasses.Ordered-
Car>();
            this.vendors = new List<CommonInterfacesClasses.IVentor>();
            this.curentModels = new List<CommonInterfacesClasses.ICar-
Model>();
            this.orderList = new List<CommonInterfacesClasses.Ordered-
Car>();
            this.listBox_ventors.DisplayMember = "VendorName";
            FillVentors();
        }

        private void FillVentors()
        {
            string pathFolder = System.AppDomain.CurrentDomain.BaseDi-
rectory;
            string[] dllFiles = Directory.GetFiles(pathFolder, "*.dll");

            foreach (string item in dllFiles)
            {
                try
                {
                    Assembly assembly = Assembly.LoadFile(item);
                    foreach (Type type in assembly.GetTypes())
                    {
```

```
                        Type iface = type.GetInterface("IVentor");
                        if (iface != null && !type.IsAbstract)
                        {

                            CommonInterfacesClasses.IVentor ventor =
(CommonInterfacesClasses.IVentor)
                                Activator.CreateInstance(type);
                            this.vendors.Add(ventor);
                            this.listBox_ventors.Items.Add(ventor);
                        }
                    }
                }
                catch (ReflectionTypeLoadException e)
                {
                    MessageBox.Show("Error loading plugin \n" + e.Message);
                }
            }
        }

        private void listBox_ventors_SelectedIndexChanged(object sender,
EventArgs e)
        {
            this.listBox_Models.Items.Clear();
            this.listBox_Colors.Items.Clear();

            CommonInterfacesClasses.IVentor temp = (CommonInterfaces-
Classes.IVentor)((ListBox)sender).SelectedItem;
            curentModels = temp.GetProductionList();

            if (this.curentModels.Count != 0)
                foreach (CommonInterfacesClasses.ICarModel item in
curentModels)
                    this.listBox_Models.Items.Add(item.Name);
        }

        private void listBox_Models_SelectedIndexChanged(object sender,
EventArgs e)
        {
            this.listBox_Colors.Items.Clear();
            foreach (CommonInterfacesClasses.ICarModel item in curent-
Models)
                if (this.listBox_Models.SelectedItem != null && item.-
Name == this.listBox_Models.SelectedItem.ToString())
                {
                    this.currentColors = item.Colors;
                    if (this.currentColors != null)
                        foreach (Color color in currentColors)
                            listBox_Colors.Items.Add(color.Name);
                }
        }

        private void button_refresh_Click(object sender, EventArgs e)
        {
            if (this.vendors != null && this.vendors.Count != 0)
                this.vendors.Clear();
            if (this.currentColors != null && this.currentColors.Count !
= 0)
                this.currentColors.Clear();
            if (this.curentModels != null && this.curentModels.Count !=
0)
                this.curentModels.Clear();
            if (this.orderList != null && this.orderList.Count != 0)
                this.orderList.Clear();
```

```
            if (this.finishList != null && this.finishList.Count != 0)
                this.finishList.Clear();

            this.listBox_Colors.Items.Clear();
            this.listBox_Models.Items.Clear();
            this.listBox_ventors.Items.Clear();
            this.listBox_Ordered.Items.Clear();
            this.listBox_allCars.Items.Clear();
            this.finishList.Clear();

            this.numericUpDown1.Value = 0;
            this.FillVentors();
        }

        private void button_ADD_Click(object sender, EventArgs e)
        {
            if (!(this.listBox_ventors.SelectedIndex > -1 &&
                this.listBox_Colors.SelectedIndex > -1 &&
                this.listBox_Models.SelectedIndex > -1 &&
                this.numericUpDown1.Value > 0))
            {
                MessageBox.Show("Select all options ");
                return;
            }
            CommonInterfacesClasses.OrderedCar orderCar =
                new CommonInterfacesClasses.OrderedCar(

this.vendors[this.listBox_ventors.SelectedIndex].VendorName,

this.curentModels[this.listBox_Models.SelectedIndex].Name,
                    this.currentColors[this.listBox_Colors.SelectedIn-
dex],
                    (int)this.numericUpDown1.Value);

            orderList.Add(orderCar);

            this.listBox_Ordered.Items.Add(orderCar.VendorName + " " +
orderCar.ModelName +
                " " + orderCar.ModelColor.Name + " " + orderCar.Count);
        }

        private void button_makeOrder_Click(object sender, EventArgs e)
        {
            if (this.listBox_Ordered.SelectedIndex < 0)
            {
                MessageBox.Show("Nothing Selected");
                return;
            }

            for (int i = 0; i < this.orderList[this.listBox_Ordered.Se-
lectedIndex].Count; i++)
            {
                this.listBox_allCars.Items.Add(this.orderList[this.list-
Box_Ordered.SelectedIndex].VendorName + " " +

this.orderList[this.listBox_Ordered.SelectedIndex].ModelName + " " +

this.orderList[this.listBox_Ordered.SelectedIndex].ModelColor.Name);

                this.finishList.Add(new CommonInterfacesClasses.Ordered-
Car(

this.orderList[this.listBox_Ordered.SelectedIndex].VendorName,
```

```
this.orderList[this.listBox_Ordered.SelectedIndex].ModelName,

this.orderList[this.listBox_Ordered.SelectedIndex].ModelColor, -1));
            }
            this.orderList.RemoveAt(this.listBox_Ordered.SelectedIndex);
            this.listBox_Ordered.Items.RemoveAt(this.listBox_Ordered.Se-
lectedIndex);

        }

        private void button_Buy_Click(object sender, EventArgs e)
        {
            if (this.listBox_allCars.Items.Count == 0)
            {
                MessageBox.Show("Consignment note is not filled");
                return;
            }
            string message = String.Empty;
            for (int i = 0; i < this.listBox_allCars.Items.Count; i++)
                message += String.Format("{0}, {1}, {2}\n", this.finish-
List[i].VendorName,
                    this.finishList[i].ModelName,
this.finishList[i].ModelColor.Name);

            this.finishList.Clear();
            this.listBox_allCars.Items.Clear();

            MessageBox.Show(message);
        }

        private void label_Ventors_Click(object sender, EventArgs e)
        {

        }

        private void Shop_Load(object sender, EventArgs e)
        {

        }

        private void listBox_Ordered_SelectedIndexChanged(object sender,
EventArgs e)
        {

        }
    }
}
```

Common interfaces

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;
using System.IO;
using System.Reflection;
```

```
namespace CarShop.Common
{
    public class CommonInterfacesClasses
    {
        public interface IVentor
        {
            string VendorName { get; }
            IList<ICarModel> GetProductionList();
        }
        public abstract class BaseVentor : IVentor
        {
            public string VendorName
            {
                get;
                set;
            }

            public IList<ICarModel> GetProductionList()
            {
                IList<ICarModel> models = new List<ICarModel>();
                string directoryPath = AppDomain.CurrentDomain.BaseDi-
rectory;
                string[] files = Directory.GetFiles(directoryPath,
"*.dll");

                foreach (string item in files)
                {
                    Assembly assembly = null;
                    try { assembly = Assembly.LoadFile(item); }
                    catch (ReflectionTypeLoadException ) { }

                    foreach (Type type in assembly.GetTypes())
                    {
                        Type temp = type.GetInterface("ICarModel");
                        if (temp != null)
                        {
                            CarInfoAttribute atr =
(CarInfoAttribute)type.GetCustomAttribute(typeof(CarInfoAttribute));
                            if (atr != null && atr.VendorName ==
this.VendorName)
                            {
                                ICarModel model = Activator.CreateIn-
stance(type) as ICarModel;
                                models.Add(model);
                            }
                        }
                    }
                }
                return models;
            }
        }
        public interface ICarModel
        {
            string Name { get; }
            List<Color> Colors { get; }
        }
        public sealed class CarInfoAttribute : Attribute
        {
            public string VendorName { get; set; }
        }
        public class OrderedCar
        {
            public string VendorName { get; set; }
```

```
            public string ModelName { get; set; }
            public Color ModelColor { get; set; }
            public int Count { get; set; }

            public OrderedCar(string vendorName, string modelName, Color
color, int count)
            {
                this.VendorName = vendorName;
                this.ModelName = modelName;
                this.ModelColor = color;
                this.Count = count;
            }
        }
    }
}
```

Ventor

Opel

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using CarShop.Common;

namespace CarShop.Ventor.Opel
{
    public class Opel: CommonInterfacesClasses.BaseVentor
    {
        public Opel()
        {
            this.VendorName = "Opel";
        }

    }
}
```

Ventor Mercedes

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using CarShop.Common;
namespace CarShop.Ventor.Mercedes
{
    public class Mercedes: CommonInterfacesClasses.BaseVentor
    {
        public Mercedes()
        {
            this.VendorName = "Mercedes";
        }
    }
}
```

Car Model

Opel

Cadet

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using CarShop.Common;

namespace CarShop.Model.Cadet
{
    [CarShop.Common.CommonInterfacesClasses.CarInfo(VendorName ="Opel")]
    public class Cadet : CommonInterfacesClasses.ICarModel
    {
        public List<System.Drawing.Color> Colors
        {
            get;
            private set;
        }

        public string Name
        {
            get;
            private set;
        }

        public Cadet()
        {
            this.Name = "Cadet";
            this.Colors = new List<System.Drawing.Color>();
            this.Colors.Add(System.Drawing.Color.Azure);
            this.Colors.Add(System.Drawing.Color.Beige);
            this.Colors.Add(System.Drawing.Color.Black);
            this.Colors.Add(System.Drawing.Color.Indigo);
        }
    }
}
```

Car Model

omega

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CarShop.Model.Omega
{
    [Common.CommonInterfacesClasses.CarInfo(VendorName = "Opel")]
    public class Omega:Common.CommonInterfacesClasses.ICarModel
    {
        public List<System.Drawing.Color> Colors
        {
            get;
            private set;
        }

        public string Name
        {
            get;
            private set;
        }
```

```
        public Omega()
        {
            this.Name = "Omega";
            this.Colors = new List<System.Drawing.Color>();
            Colors.Add(System.Drawing.Color.Green);
            Colors.Add(System.Drawing.Color.Maroon);
            Colors.Add(System.Drawing.Color.Lime);
        }

    }
}
```

Car model

Benz

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CarShop.Model.Benz
{
    [Common.CommonInterfacesClasses.CarInfo(VendorName = "Mercedes")]
    public class Benz : Common.CommonInterfacesClasses.ICarModel
    {
        public List<System.Drawing.Color> Colors
        {
            get;
            private set;
        }

        public string Name
        {
            get;
            private set;
        }
        public Benz()
        {
            this.Name = "Benz";
            this.Colors = new List<System.Drawing.Color>();
            Colors.Add(System.Drawing.Color.Beige);
            Colors.Add(System.Drawing.Color.Orange);
            Colors.Add(System.Drawing.Color.Pink);
        }
    }
}
```

Screenshots