

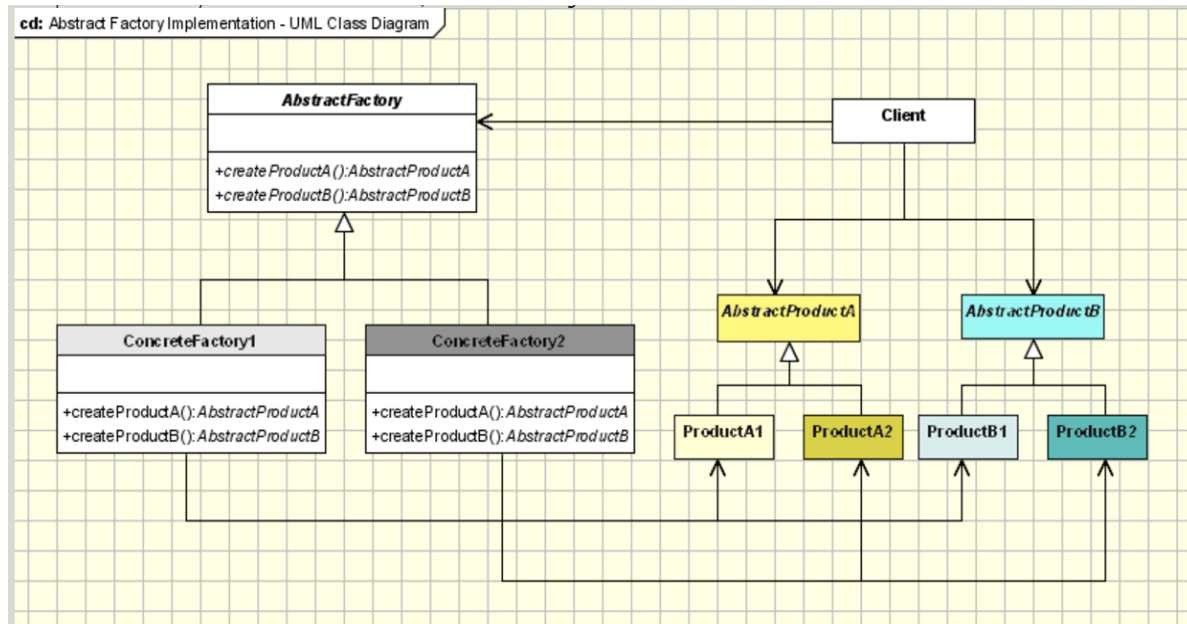
Introduction

For this assignment, we were tasked with making a program that displays the Abstract Factory pattern. For my program, I chose to make a car factory. The two types of factories I made were Chevy and Ford. These factories were able to make a vehicle of three different types; sports, sedan, and truck.

UML Diagram

To the right you will see the UML diagram for the abstract factory pattern.

The ChevyFactory class and the FordFactory class are used to fulfill the role of the ConcreteFactory. The



`abstractSportsCar`, `abstractTruck`, and the `abstractSedan` were all used to demonstrate the `abstractProduct` classes. Finally, the `ChevyCamaro`, `ChevyMalibu`, `ChevySilverado`, `FordMustang`, `FordFusion`, and the `FordF150` classes were all used to demonstrate the `concreteProduct` classes.

Below I will show the code used to make the program fulfill the requirements of the abstract factory pattern. First will be the code for `Form1` and `Form2`.

```
public partial class Form1 : Form
{
    ChevyFactory factory1;
    FordFactory factory2;
    ChevyCamaro camaro;
    ChevyMalibu malibu;
    ChevySilverado silverado;
    FordMustang mustang;
    FordFusion fusion;
    FordF150 f150;

    public Form1()
    {
        InitializeComponent();
        factory1 = new ChevyFactory();
    }
}
```

```
        factory2 = new FordFactory();
        camaro = new ChevyCamaro();
        malibu = new ChevyMalibu();
        silverado = new ChevySilverado();
        mustang = new FordMustang();
        fusion = new FordFusion();
        f150 = new FordF150();
    }

    private void m_btnCamaro_Click(object sender, EventArgs e)
    {
        factory1.m_BodyPart = ChevyFactory.BodyType.Sports;
        factory1.m_EnginePart = ChevyFactory.Engine.V8;
        factory1.m_TirePart = ChevyFactory.Tires.Sports;
        factory1.m_BrandType = ChevyFactory.Brand.Chevy;
        factory1.createCamaro(factory1.m_BodyPart, factory1.m_EnginePart,
factory1.m_TirePart, factory1.m_BrandType);
    }

    private void m_btnMalibu_Click(object sender, EventArgs e)
    {
        factory1.m_BodyPart = ChevyFactory.BodyType.Sedan;
        factory1.m_EnginePart = ChevyFactory.Engine.FourCylinder;
        factory1.m_TirePart = ChevyFactory.Tires.Regular;
        factory1.m_BrandType = ChevyFactory.Brand.Chevy;
        factory1.createMalibu(factory1.m_BodyPart, factory1.m_EnginePart,
factory1.m_TirePart, factory1.m_BrandType);
    }

    private void m_btnSilverado_Click(object sender, EventArgs e)
    {
        factory1.m_BodyPart = ChevyFactory.BodyType.Truck;
        factory1.m_EnginePart = ChevyFactory.Engine.V6;
        factory1.m_TirePart = ChevyFactory.Tires.HeavyDuty;
        factory1.m_BrandType = ChevyFactory.Brand.Chevy;
        factory1.createSilverado(factory1.m_BodyPart, factory1.m_EnginePart,
factory1.m_TirePart, factory1.m_BrandType);
    }

    private void m_btnMustang_Click(object sender, EventArgs e)
    {
        factory2.m_BodyPart = FordFactory.BodyType.Sports;
        factory2.m_EnginePart = FordFactory.Engine.V8;
        factory2.m_TirePart = FordFactory.Tires.Sports;
        factory2.m_BrandType = FordFactory.Brand.Ford;
        factory2.createMustang(factory2.m_BodyPart, factory2.m_EnginePart,
factory2.m_TirePart, factory2.m_BrandType);
    }

    private void m_btnFusion_Click(object sender, EventArgs e)
    {
        factory2.m_BodyPart = FordFactory.BodyType.Sedan;
        factory2.m_EnginePart = FordFactory.Engine.FourCylinder;
        factory2.m_TirePart = FordFactory.Tires.Regular;
        factory2.m_BrandType = FordFactory.Brand.Ford;
        factory2.createFusion(factory2.m_BodyPart, factory2.m_EnginePart,
factory2.m_TirePart, factory2.m_BrandType);
    }
}
```

```
private void m_btnF150_Click(object sender, EventArgs e)
{
    factory2.m_BodyPart = FordFactory.BodyType.Truck;
    factory2.m_EnginePart = FordFactory.Engine.V6;
    factory2.m_TirePart = FordFactory.Tires.HeavyDuty;
    factory2.m_BrandType = FordFactory.Brand.Ford;
    factory2.createF150(factory2.m_BodyPart, factory2.m_EnginePart,
factory2.m_TirePart, factory2.m_BrandType);
}

public partial class Form2 : Form
{
    public Form2()
    {

    }

    public Form2(ChevyFactory.BodyType m_BodyPart, ChevyFactory.Engine
m_EnginePart, ChevyFactory.Tires m_TirePart, ChevyFactory.Brand m_BrandType)
    {
        InitializeComponent();
        m_tbBody.Text = m_BodyPart.ToString();
        m_tbEngine.Text = m_EnginePart.ToString();
        m_tbTires.Text = m_TirePart.ToString();
        m_tbLogo.Text = m_BrandType.ToString();
    }

    public Form2(FordFactory.BodyType m_BodyPart, FordFactory.Engine
m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand m_BrandType)
    {
        InitializeComponent();
        m_tbBody.Text = m_BodyPart.ToString();
        m_tbEngine.Text = m_EnginePart.ToString();
        m_tbTires.Text = m_TirePart.ToString();
        m_tbLogo.Text = m_BrandType.ToString();
    }
}
```

Next, I will move on to the AbstractCarFactory, ChevyFactory, and the FordFactory classes.

```
public abstract class AbstractCarFactory // this is the AbstractFactory class
{
    public abstract ChevyCamaro createCamaro(ChevyFactory.BodyType m_BodyPart,
ChevyFactory.Engine m_EnginePart, ChevyFactory.Tires m_TirePart,
ChevyFactory.Brand m_BrandType);
    public abstract ChevyMalibu createMalibu(ChevyFactory.BodyType m_BodyPart,
ChevyFactory.Engine m_EnginePart, ChevyFactory.Tires m_TirePart,
ChevyFactory.Brand m_BrandType);
    public abstract ChevySilverado createSilverado(ChevyFactory.BodyType
m_BodyPart, ChevyFactory.Engine m_EnginePart, ChevyFactory.Tires m_TirePart,
ChevyFactory.Brand m_BrandType);
    public abstract FordMustang createMustang(FordFactory.BodyType m_BodyPart,
FordFactory.Engine m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand
m_BrandType);
    public abstract FordFusion createFusion(FordFactory.BodyType m_BodyPart,
FordFactory.Engine m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand
m_BrandType);
}
```

```
        public abstract FordF150 createF150(FordFactory.BodyType m_BodyPart,
FordFactory.Engine m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand
m_BrandType);
    }

public class ChevyFactory : AbstractCarFactory // this is the ConcreteFactory1
class
{
    public enum BodyType { Sports, Sedan, Truck }
    public enum Engine { FourCylinder, V6, V8 }
    public enum Tires { Sports, Regular, HeavyDuty }
    public enum Brand { Chevy, Ford }
    private BodyType _BodyPart;
    private Engine _EnginePart;
    private Tires _TirePart;
    private Brand _BrandType;

    public BodyType m_BodyPart
    {
        get { return _BodyPart; }
        set
        {
            _BodyPart = value;
        }
    }

    public Engine m_EnginePart
    {
        get { return _EnginePart; }
        set
        {
            _EnginePart = value;
        }
    }

    public Tires m_TirePart
    {
        get { return _TirePart; }
        set
        {
            _TirePart = value;
        }
    }

    public Brand m_BrandType
    {
        get { return _BrandType; }
        set
        {
            _BrandType = value;
        }
    }

    public override ChevyCamaro createCamaro(ChevyFactory.BodyType m_BodyPart,
ChevyFactory.Engine m_EnginePart, ChevyFactory.Tires m_TirePart,
ChevyFactory.Brand m_BrandType)
    {
        return new ChevyCamaro(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
    }
}
```

```
    }

    public override ChevyMalibu createMalibu(ChevyFactory.BodyType m_BodyPart,
ChevyFactory.Engine m_EnginePart, ChevyFactory.Tires m_TirePart,
ChevyFactory.Brand m_BrandType)
    {
        return new ChevyMalibu(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
    }

    public override ChevySilverado createSilverado(ChevyFactory.BodyType
m_BodyPart, ChevyFactory.Engine m_EnginePart, ChevyFactory.Tires m_TirePart,
ChevyFactory.Brand m_BrandType)
    {
        return new ChevySilverado(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
    }

    public override FordMustang createMustang(FordFactory.BodyType m_BodyPart,
FordFactory.Engine m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand
m_BrandType)
    {
        throw new NotImplementedException();
    }

    public override FordFusion createFusion(FordFactory.BodyType m_BodyPart,
FordFactory.Engine m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand
m_BrandType)
    {
        throw new NotImplementedException();
    }

    public override FordF150 createF150(FordFactory.BodyType m_BodyPart,
FordFactory.Engine m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand
m_BrandType)
    {
        throw new NotImplementedException();
    }
}

public class FordFactory : AbstractCarFactory // this is the ConcreteFactory2
class
{
    public enum BodyType { Sports, Sedan, Truck }
    public enum Engine { FourCylinder, V6, V8 }
    public enum Tires { Sports, Regular, HeavyDuty }
    public enum Brand { Chevy, Ford }
    private BodyType _BodyPart;
    private Engine _EnginePart;
    private Tires _TirePart;
    private Brand _BrandType;

    public BodyType m_BodyPart
    {
        get { return _BodyPart; }
        set
        {
            _BodyPart = value;
        }
    }
}
```

```
    }

    public Engine m_EnginePart
    {
        get { return _EnginePart; }
        set
        {
            _EnginePart = value;
        }
    }

    public Tires m_TirePart
    {
        get { return _TirePart; }
        set
        {
            _TirePart = value;
        }
    }

    public Brand m_BrandType
    {
        get { return _BrandType; }
        set
        {
            _BrandType = value;
        }
    }

    public override FordMustang createMustang(FordFactory.BodyType m_BodyPart,
FordFactory.Engine m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand
m_BrandType)
    {
        return new
FordMustang(m_BodyPart,m_EnginePart,m_TirePart,m_BrandType);
    }

    public override FordFusion createFusion(FordFactory.BodyType m_BodyPart,
FordFactory.Engine m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand
m_BrandType)
    {
        return new FordFusion(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
    }

    public override FordF150 createF150(FordFactory.BodyType m_BodyPart,
FordFactory.Engine m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand
m_BrandType)
    {
        return new FordF150(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
    }

    public override ChevyCamaro createCamaro(ChevyFactory.BodyType m_BodyPart,
ChevyFactory.Engine m_EnginePart, ChevyFactory.Tires m_TirePart,
ChevyFactory.Brand m_BrandType)
    {
        throw new NotImplementedException();
    }
}
```

```
        public override ChevyMalibu createMalibu(ChevyFactory.BodyType m_BodyPart,
ChevyFactory.Engine m_EnginePart, ChevyFactory.Tires m_TirePart,
ChevyFactory.Brand m_BrandType)
        {
            throw new NotImplementedException();
        }

        public override ChevySilverado createSilverado(ChevyFactory.BodyType
m_BodyPart, ChevyFactory.Engine m_EnginePart, ChevyFactory.Tires m_TirePart,
ChevyFactory.Brand m_BrandType)
        {
            throw new NotImplementedException();
        }
    }
```

Finally, we will move on to the concreteProduct classes, which are ChevyCamaro, ChevyMalibu, ChevySilverado, FordMustang, FordFusion, and FordF150.

```
// these are the ConcreteProductA classes
public class ChevyCamaro : AbstractSportsCar
{
    public ChevyCamaro()
    {
    }

    public ChevyCamaro(ChevyFactory.BodyType m_BodyPart, ChevyFactory.Engine
m_EnginePart, ChevyFactory.Tires m_TirePart, ChevyFactory.Brand m_BrandType)
    {
        Form2 f2 = new Form2(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
        f2.Visible = true;
    }
}

public class FordMustang : AbstractSportsCar
{
    public FordMustang()
    {
    }

    public FordMustang(FordFactory.BodyType m_BodyPart, FordFactory.Engine
m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand m_BrandType)
    {
        Form2 f2 = new Form2(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
        f2.Visible = true;
    }
}

// these are the ConcreteProductB classes
public class ChevyMalibu : AbstractSedan
{
    public ChevyMalibu()
    {
    }
}
```

```
        public ChevyMalibu(ChevyFactory.BodyType m_BodyPart, ChevyFactory.Engine
m_EnginePart, ChevyFactory.Tires m_TirePart, ChevyFactory.Brand m_BrandType)
        {
            Form2 f2 = new Form2(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
            f2.Visible = true;
        }
    }

    public class FordFusion : AbstractSedan
    {
        public FordFusion()
        {
        }

        public FordFusion(FordFactory.BodyType m_BodyPart, FordFactory.Engine
m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand m_BrandType)
        {
            Form2 f2 = new Form2(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
            f2.Visible = true;
        }
    }

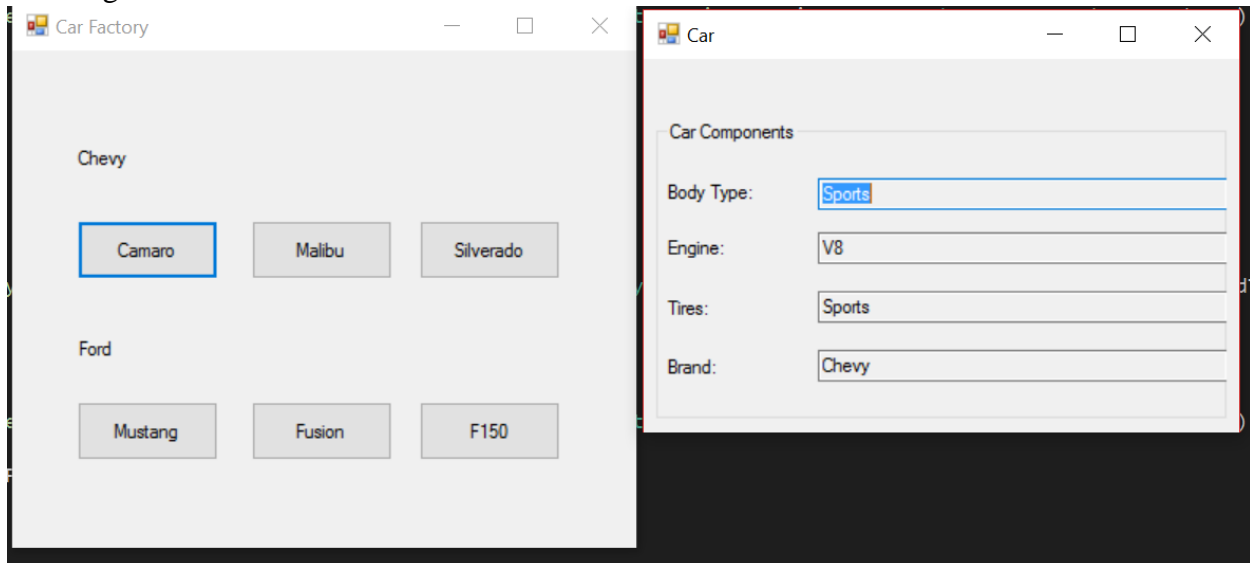
//these are the ConcreteProductC classes
    public class ChevySilverado : AbstractTruck
    {
        public ChevySilverado()
        {
        }

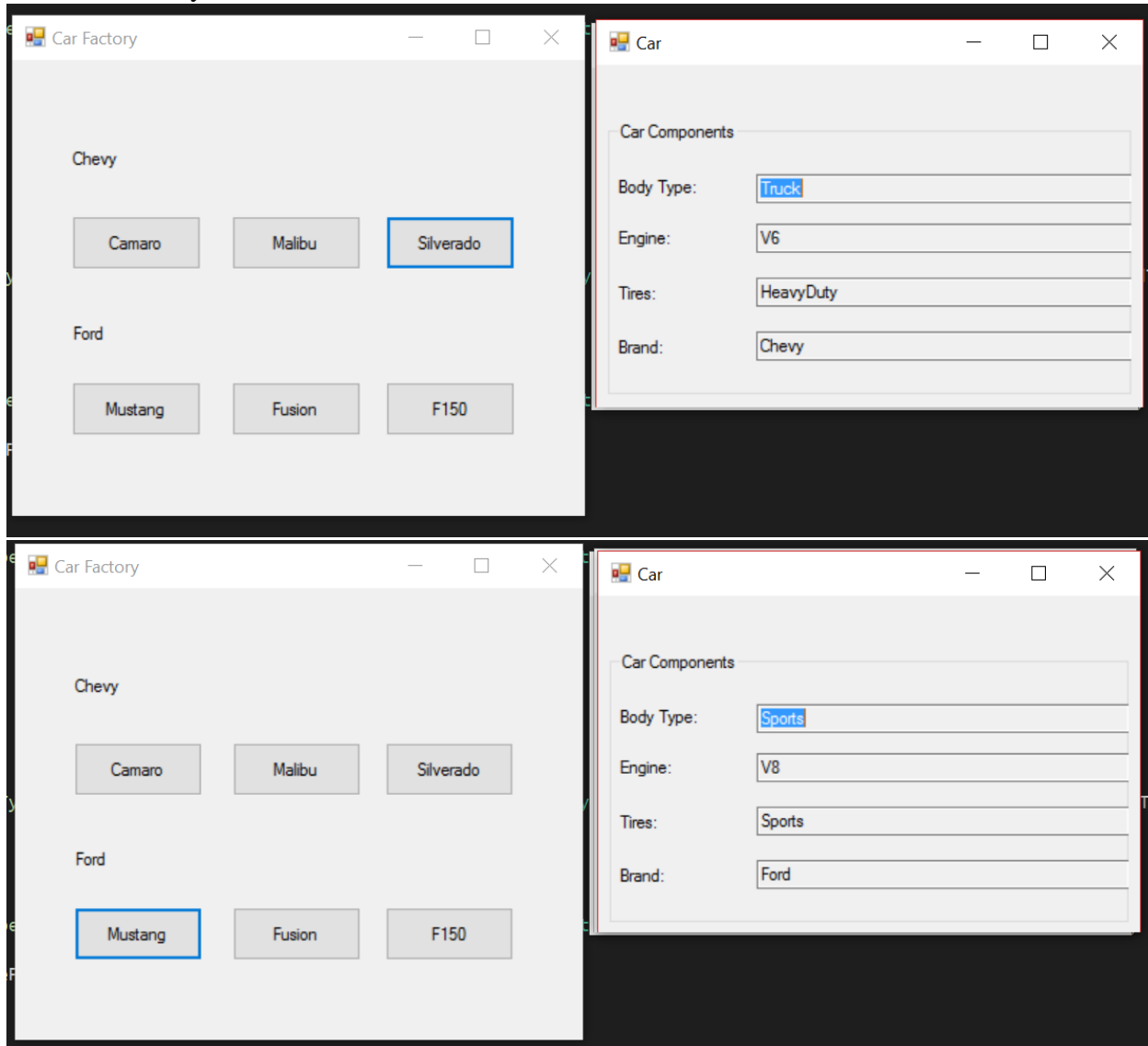
        public ChevySilverado(ChevyFactory.BodyType m_BodyPart,
ChevyFactory.Engine m_EnginePart, ChevyFactory.Tires m_TirePart,
ChevyFactory.Brand m_BrandType)
        {
            Form2 f2 = new Form2(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
            f2.Visible = true;
        }
    }

    public class FordF150 : AbstractTruck
    {
        public FordF150()
        {
        }

        public FordF150(FordFactory.BodyType m_BodyPart, FordFactory.Engine
m_EnginePart, FordFactory.Tires m_TirePart, FordFactory.Brand m_BrandType)
        {
            Form2 f2 = new Form2(m_BodyPart, m_EnginePart, m_TirePart,
m_BrandType);
            f2.Visible = true;
        }
    }
}
```


Here are some screenshots of my code working, the first one shows what happens when you press the Camaro button. The next one shows the Silverado button, and the last one shows the Mustang button.





Conclusion

Overall, I thought this was a pretty easy project to put together after doing the FactoryMethod pattern. The one major problem I ran into dealt with the fact that I originally didn't have the factories creating the product, it was just being accessed by the form. After fixing that problem, the rest of the code was smooth sailing.