عنوان: **آشنایی با الگوی طراحی Builder**
نویسنده: وحید فرهمندیان
تاریخ: ۱۳۹۳/۱۲/۰۲ ۲۰:۵
آدرس: www.dotnettips.info
گروه‌ها: C#, OOP, DesignPatterns

سناریوی زیر را در نظر بگیرید:
از شما خواسته شده است تا نحوه‌ی ساخت تلفن همراه را پیاده سازی نمایید. شما در گام اول 2 نوع تلفن همراه را شناسایی نموده‌اید (Android و Windows Phone). پس از شناسایی، احتمالا هر کدام از این انواع را یک کلاس در نظر می‌گیرید و به کمک یک واسط یا کلاس انتزاعی، شروع به ساخت کلاس می‌نمایید، تا در آینده اگر تلفن همراه جدیدی شناسایی شد، راحت‌تر بتوان آن را در پیاده سازی دخیل نمود.
اگر چنین فکر کرده اید باید گفت که 90% با الگوی طراحی Builder آشنا هستید و از آن نیز استفاده می‌کنید؛ بدون اینکه متوجه باشید از این الگو استفاده کرده‌اید. در کدهای زیر این الگو را قدم به قدم بررسی خواهیم نمود. **قدم 1:** تلفن همراه چه بخش هایی می‌تواند داشته باشد؟ (برای مثال یک OS دارند، یک Name دارند و یک Screen) همچنین برای اینکه تلفن همراهی بتواند ساخته شود ابتدا بایستی نام آن‌را بدانیم. کدهای زیر همین رویه را تصدیق می‌نمایند:

```
public class Product
{
        public Product(string name)
        {
            Name = name;
        }
        public string Name { get; set; }
        public string Screen { get; set; }
        public string OS { get; set; }
        public override string ToString()
        {
            return string.Format(Screen + "/" + OS + "/" + Name);
        }
}
```

یک کلاس ساخته‌ایم و نام آن را Product گذاشتیم. بخش‌های مختلفی را نیز در آن تعریف نموده‌ایم. تابع ToString را برای استفاده‌های بعدی override کرده‌ایم (فعلا نیازی بدان نداریم). **قدم 2:** برای ساخت تلفن همراه چه کارهایی باید انجام شود؟ (برای مثال بایستی OS روی آن نصب شود، Screen آن مشخص شود. همچنین بایستی به طریقی بتوانم تلفن همراه ساخته شده‌ی خود را نیز پیدا کنم). کدهای زیر همین رویه را تصدیق می‌نمایند:

```
public interface IBuilder
    {
        void BuildScreen();
        void BuildOS();
        Product Product { get; }
    }
```

یک واسط تعریف کرده‌ایم تا به کمک آن هر تلفن همراهی را که خواستیم بسازیم. **قدم 3:** از آنجا که فقط دو نوع تلفن همراه را فعلا شناسایی کرده‌ایم (Android و Windows Phone) نیاز داریم تا این دو تا را بسازیم.
ابتدا تلفن همراه Android را می‌سازیم:

```
public class ConcreteBuilder1 : IBuilder
    {
        public Product p;
        public ConcreteBuilder1()
        {
            p = new Product("Android Cell Phone");
        }
        public void BuildScreen()
        {
            p.Screen = "Touch Screen 16 Inch!";
        }

        public void BuildOS()
        {
            p.OS = "Android 4.4";
        }
```

```
        public Product Product
        {
            get { return p; }
        }
    }
```

سپس تلفن همراه Windows Phone را می‌سازیم:

```
public class ConcreteBuilder2 : IBuilder
    {
        public Product p;

        public ConcreteBuilder2()
        {
            p = new Product("Windows Phone");
        }
        public void BuildScreen()
        {
            p.Screen = "Touch Screen 32 Inch!";
        }

        public void BuildOS()
        {
            p.OS = "Windows Phone 2014";
        }
        public Product Product
        {
            get { return p; }
        }
    }
```

قدم 4: اول باید OS نصب شود یا Screen مشخص شود؟ برای اینکه توالی کار را مشخص سازم نیاز به یک کلاس دیگر دارم تا اینکار را انجام دهد:

```
public class Director
    {
        public void Construct(IBuilder builder)
        {
            builder.BuildScreen();
            builder.BuildOS();
        }
    }
```

این کلاس در متد Construct خود یک ورودی از نوع IBuilder می‌گیرد و براساس توالی مورد نظر، شروع به ساخت آن می‌کند.
قدم 5: نهایتا میخواهم به برنامه‌ی خود بگویم که تلفن همراه Android را بسازد:

```
Director d = new Director();
ConcreteBuilder1 cb1 = new ConcreteBuilder1();
d.Construct(cb1);
Console.WriteLine(cb1.p.ToString());
```

و به این صورت تلفن همراه من آماده است!
متد ToString در اینجا، همان ToString ابتدای بحث است که آن را Override کردیم.
به این نکته توجه کنید که اگر یک تلفن همراه جدید شناسایی شود، چه مقدار تغییری در کدها نیاز دارید؟ برای مثال تلفن همراه BlackBerry شناسایی شده‌است. تنها کاری که لازم است این است که یک کلاس بصورت زیر ساخته شود:

```
public class BlackBerry: IBuilder
    {
        public Product p;

        public BlackBerry ()
        {
            p = new Product("BlackBerry");
        }
        public void BuildScreen()
        {
            p.Screen = "Touch Screen 8 Inch!";
        }
```

```
        public void BuildOS()
        {
            p.OS = "BlackBerry XXX";
        }
        public Product Product
        {
            get { return p; }
        }
    }
```