

همان طور که میدانید از الگوی Factory به عنوان روشی برای کاهش وابستگی اجزای یک سیستم استفاده میشود. در این مقاله میخواهیم با استفاده از جنریک‌ها، الگوی Abstract Factory را پیاده سازی کنیم.

(1) ایجاد یک کلاس به نام AbstractFactory و یک متد جنریک به نام CreateObject

```
public class AbstractFactory
{
    public static T CreateObject<T>() where T : class , new()
    {
        return new T();
    }
}
```

(2) ساخت کلاسهای مورد نظر

```
public class Product
{
    public void DisplayInfo()
    {
        Console.WriteLine("Product Class Craeted. ");
    }
}
```

```
public class Category
{
    public void DisplayInfo()
    {
        Console.WriteLine("Category Class Created.");
    }
}
```

(3) حال در یک برنامه‌ی کنسول ویندوز، از کلاس AbstractFactory به شکل زیر استفاده میکنیم

```
static void Main(string[] args)
{
    var p = AbstractFactory.CreateObject< Product>();
    p.DisplayInfo();
    Console.WriteLine("=====");

    var c = AbstractFactory.CreateObject<Category>();
    c.DisplayInfo();
    Console.WriteLine("=====");

    Console.ReadKey();
}
```

خروجی کد بالا

```
file:///c:/users/cloud/documents/  
Product Class Craeted.  
=====  
Category Class Created.  
=====
```

نظرات خوانندگان

نویسنده: سعید

تاریخ: ۲۱:۱۶ ۱۳۹۳/۰۸/۱۲

تو این حالت میشه به صورت مستقیم هم از کلاس Product، آبجکت تعریف کرد. پس در این حالت کاهش وابستگی دقیقاً به چه صورتی هست؟ چه فرقی هست بین AbstractFactory.CreateObject<Product>() و new Product () ؟

نویسنده: محسن خان

تاریخ: ۲۱:۳۵ ۱۳۹۳/۰۸/۱۲

مطلب فوق کمی خلاصه شده هست. یک مثال عملی اون رو برای کاهش وابستگی‌ها، می‌تونید اینجا مطالعه کنید: [استفاده از Factories برای حذف Service locators در برنامه‌های WinForms](#)

نویسنده: احمد نواصری

تاریخ: ۱۰:۲۸ ۱۳۹۳/۰۸/۲۲

درسته . اما در این حالت دیگر کلاینت شما مستقیماً به کار ساخت Object را انجام نمیدهد و کار را به کلاس دیگری واگذار کرده.