

工厂方法模式

1.

仔细体会上面的示例，对比它们的实现，尤其是从思想层面上，会发现工厂方法模式和 IoC/DI 的思想是相似的，都是“主动变被动”，进行了“主从换位”，从而获得了更灵活的程序结构。

2. 工厂方法模式是通过生成不同操作类对象从而控制哪种类的生成. 即通过子类的new或者传入不同类型的参数等方式来自动选择后面操作.

3.

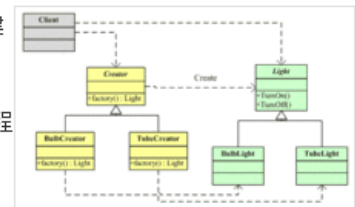
1. 工厂方法模式的定义

定义一个用于创建对象的接口，让子类决定实例化哪一个类，Factory Method 使一个类的实例化延迟到其子类。

2. 主要包括以下几个结构角色

抽象工厂(Creator)角色：是工厂方法模式的核心，与应用程序无关。任何在模式中创建的对象工厂类必须实现这个接口。

具体工厂(Concrete Creator)角色：这是实现抽象工厂接口的具体工厂类，包含与应用程序密切相关的逻辑，并且受到应用程序调用以创建产品对象。在上图中有两个这样的角色：BulbCreator与TubeCreator。



抽象产品(Product)角色：工厂方法模式所创建的对象超类型，也就是产品对象的共同父类或共同拥有的接口。在上图中，这个角色是Light。

具体产品(Concrete Product)角色：这个角色实现了抽象产品角色所定义的接口。某具体产品有专门的具体工厂创建，它们之间往往一一对应。

核心:封装变化, 由工厂类型决定产生的子类对象类型, 由子类对象类型决定方法类型

工厂方法模式与IoC/DI在思想上是类似的

延伸

小结：其实 IoC/DI 对编程带来的最大改变不是在代码上，而是在思想上，发生了“主从换位”的变化。应用程序原本是老大，要获取什么资源都是主动出击，但是在 IoC/DI 思想中，应用程序就变成被动的了，被动地等待 IoC/DI 容器来创建并注入它所需要的资源了。

3. 平行的类层次结构

1. 平行的类层次结构的含义

简单点说，假如有两个类层次结构，其中一个类层次中的每个类在另一个类层次中都有一个对应的类的结构，就被称为平行的类层次结构。

明显工厂方法模式就是这种结构

4. 参数化工厂方法

将用工厂类替代接口, 将变化封装在创建产品的方法参数上