# 工厂方法模式



## 问题引入

在现实生活中社会分工越来越细，越来越专业化。各种产品由专门的工厂生产，彻底告别了自给自足的小农经济时代，这大大缩短了产品的生产周期，提高了生产效率，提升了产品质量。同样，在软件开发中能否做到软件对象的生产和使用相分离呢？能否在满足“开闭原则”的前提下，客户随意增删或改变对软件相关对象的使用呢？这就是工厂方法模式要解决的问题。

## 工厂方法模式的定义和特点

工厂方法模式（FactoryMethod Pattren）的定义：定义一个创建产品对象的工厂接口，将产品对象的实际创建工作交由具体工厂类完成。

工厂方法模式是一种创建型模式，具有软件对象的创建和使用相分离的特点。

## 名词解释

产品：要创建的对象被称为“产品“。

工厂：创建产品的对象称为“工厂“。

抽象工厂：提供了创建产品的接口，调用者通过它访问具体工厂的工厂方法来创建产品。

具体工厂：主要是实现抽象工厂中创建产品的工厂方法，完成具体产品的实现，工厂方法模式中，一个具体工厂负责创建一种具体产品。

抽象产品：定义了产品的规范，描述了产品的主要特性和功能。

具体产品：实现了抽象产品角色所定义的接口，由具体工厂来创建，它同具体工厂之间一一对应。

## 简单工厂模式

如果要创建的产品种类不多，可以将产品的创建全部交由一个工厂类完成，这种模式叫“简单工厂模式”，它并不属于GoF的23种经典设计模式，且它违背了“开闭原则“，因为每增加一种新的产品就需要去修改工厂类，添加新的”生产线“。

工厂方法模式是对简单工厂模式的进一步抽象，遵循了开闭原则。

## 工厂方法模式的优点和缺点

工厂方法模式的优点有：

1. 实现了软件对象的创建和使用分离，客户端只需要知道具体工厂的名称就能得到想要的产品，不需要关心产品的具体创建细节。
2. 满足开闭原则，在系统增加新的产品时只需要添加具体产品类和对应的具体工厂类，无须对原工厂进行任何修改。

工厂方法模式的缺点：

1. 每增加一个新的产品就需要添加一个新的具体产品类和一个对应的具体工厂类，增加了系统的复杂度。

## 工厂方法模式的应用场景

工厂方法模式通常适用于以下场景：

1. 客户只知道创建产品的工厂名，而不知道具体的产品名。如 TCL 电视工厂、海信电视工厂等。
2. 创建对象的任务由多个具体子工厂中的某一个完成，而抽象工厂只提供创建产品的接口。
3. 客户不关心创建产品的细节，只关心获得的产品种类是否正确。