

# Factory Method 模式

## 引入。

- 【案例】

承接上一篇简单工厂模式。黑枣克隆公司在初始发展阶段使用简单工厂模式来生产克隆狗，狗的分类方式局限于按颜色分类。由于市场需求变化，客户对同一颜色狗有了体型大小要求，分为大型狗及小型狗两类。

- 【分析OOA】

如果我是黑枣克隆公司老板，我可能要开设分厂。使用大型狗工厂专门生产大型狗，小型狗工厂专门生产小型狗。其实软件工程师也会去这样实现。工厂方法模式使用继承自抽象工厂角色的多个子类来代替简单工厂模式中的“上帝类”。

原文链接：<https://blog.csdn.net/uuleaf/article/details/7572514>

## 定义。

描述1：工厂方法是针对每一种产品提供一个工厂类。通过不同的工厂实例来创建不同的产品实例。

描述2：工厂方法就是为配一个产品提供一个独立的工厂类，经过不一样的工厂实例来建立不一样的产品实例。

## 理解。

为生产不同种类的产品开设不同的工厂。

## 作用。

- 可以将生成实例的框架（framework）和实际负责生成实例的类解耦。
  - 父类决定实例的生成方式，但并不决定所要生成的具体的类，具体的处理全部交给子类处理。

## 关键字。

- 特定的工厂生产特定的产品。

## 启示。

不用new关键字来生成实例，而是通过调用生成实例的专用方法来生成实例，这样就可以防止父类与其他具体类耦合

## 登场角色。

- 产品(Product)。
  - 属于框架，抽象类。

- 定义了Factory Method模式中生成那些实例所持有的接口（API），但具体的处理由子类ConcreteProduct角色实现。
- Product类扮演此角色。
- 创建者(Creator)。
  - 属于框架，抽象类。
  - 负责生成Product角色，但具体的处理由ConcreteCreator角色决定。
  - Factory类扮演此角色。
- 具体的产品(ConcreteProduct)。
  - 属于加工方。
  - 决定了具体的产品。
  - IDCard类扮演此角色。
- 具体的创建则(ConcreteCreator)。
  - 属于加工方。
  - 负责生成具体的产品。
  - IDCardFactory类扮演此角色。

## 类图。

