

创建型模式

原创

xyzso1z

最后发布于2019-02-28 14:53:02

阅读数 54

☆ 收藏

编辑 展开

创建型模式有五种具体的模式，分别为：

- 1. 单例模式
- 2. 原型模式
- 3. 建造模式 (Builder模式)
- 4. 工厂方法模式
- 5. 抽象工厂模式

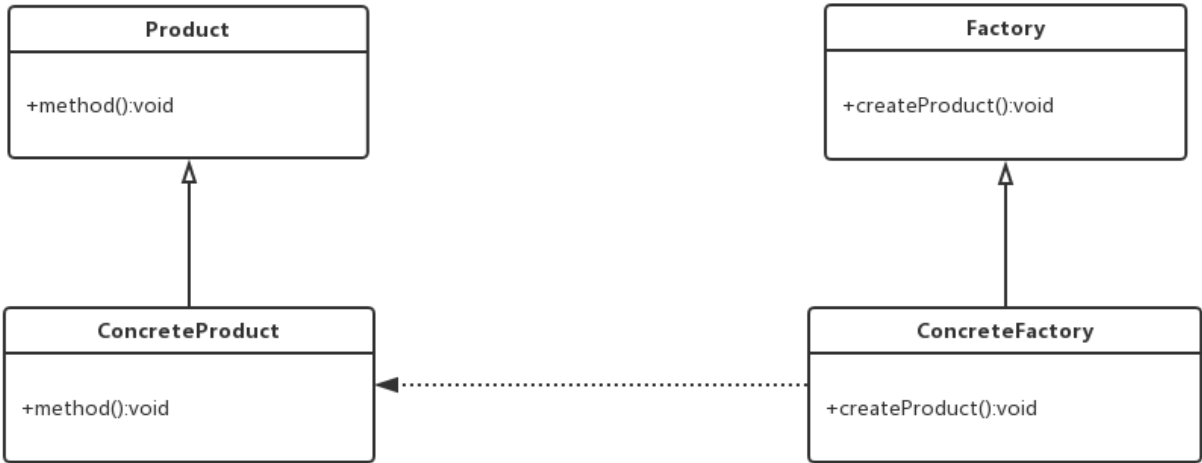
对比:

模式	定义	使用场景
单例模式	确保某一个类只有一个实例，而且自行实例化并向整个系统提供这个实例	确保某各类有且只有一个对象的场景
原型模式	用原型实例指定创建对象的种类，并通过拷贝这些原型创建新的对象	1.类初始化需要消耗非常多的资源；2.通过new 产生一个对象需要非常繁琐的数据准备或对象需要提供给其它对象访问，而且各个调用者可能需要修改其值时。
建造模式	将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示	1.相同的方法，不同的执行顺序，产生不同的时间结果时；2.多个部件或零件，都可以装配到一个产品上，但是产生的运行结果又不相同时；3.产品类非常复杂，或者产品类中的调用顺序不同产生不同的运行结果时；4.初始化一个对象特别复杂时。
工厂方法模式	定义一个用于创建对象的接口，让子类决定实例化哪个类	在任何需要生成复杂对象的地方，都可以使用工厂方法模式。复杂对象适合使用工厂模式完成创建的对象无需使用工厂模式
抽象工厂模式	为创建一组相关或者是相互依赖的对象提供一个接口，而不需要指定它们的具体类	一个对象族有相同的约束时可以使用抽象工厂模式

工厂方法与抽象工厂比较:

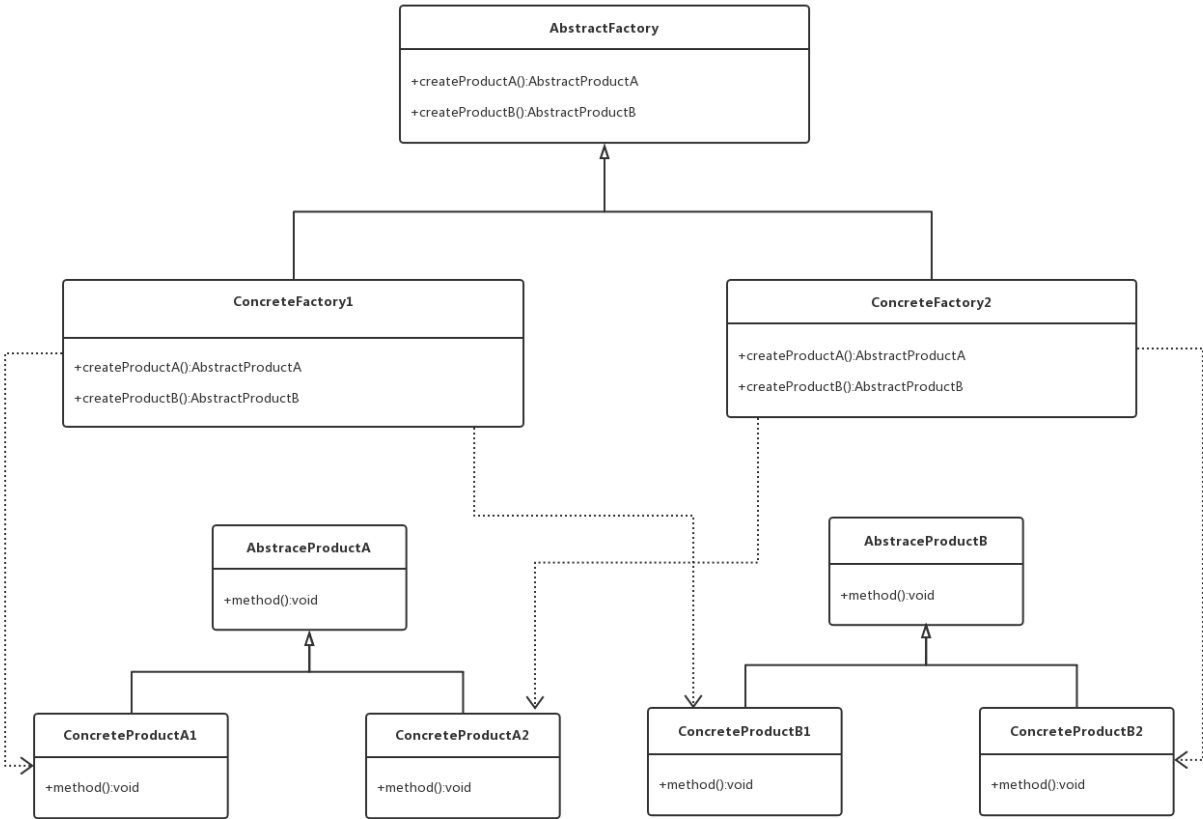
类图比较：

工厂方法：



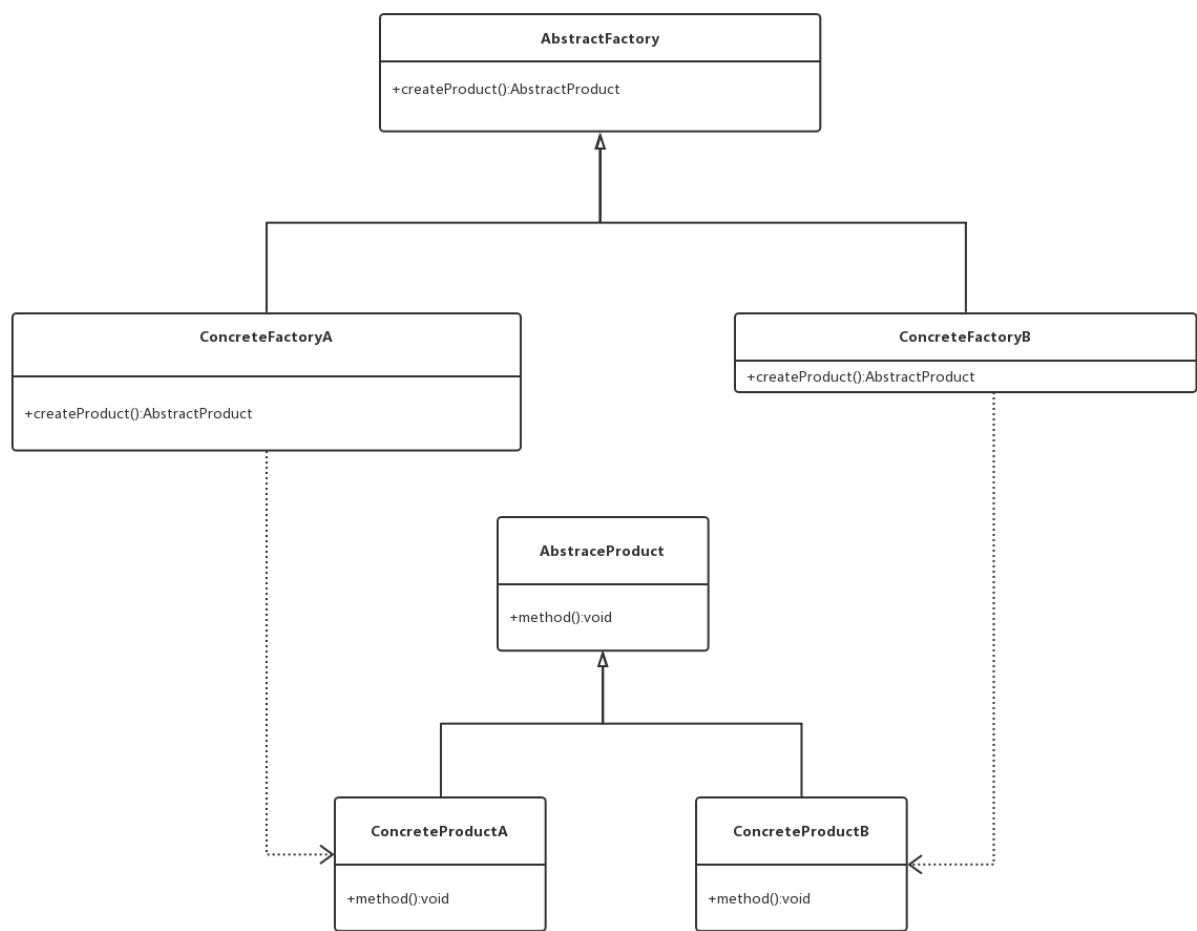
<https://blog.csdn.net/xyzso1z>

抽象工厂：



<https://blog.csdn.net/xyzso1z>

下面把工厂方法模式的类图修改一下：



可以看到工厂方法只生产一个产品，而抽象工厂创建一组产品。