【用的不多】

factory method模式：**模板方法模式+简单工厂模式**，简单工厂模式稍微复杂一点的变种【每个类都用工厂模式产生，但有通用逻辑在产生前执行 ---- 将工厂的通用逻辑用模板方法实现】

场景：

（1）我们现在要生产的产品有多种

（2）每种产品的生产过程，都有一些是共同的逻辑，但是也有一些是特殊的逻辑

（3）现在要生产出来各种产品

【工厂通用逻辑】

public **abstract class Creator** {

   public Product create() {

       System.out.println("生产一个产品的**通用**基础逻辑");

       return factoryMethod();

  }

**protected abstract Product factoryMethod**();

}

【具体工厂】

public class **ConcreteCreator extends Creator** {

   public Product **factoryMethod**() {

       System.out.println("生产自己特有产品的特殊逻辑");

       return new ConcreteProduct();

  }

}

说实话，这种设计模式算是较为复杂一点的了，适用的是比较复杂的场景了

一般就是在有复杂工厂生产场景的情况下，比如就是需要多种工厂生产不同的产品，但是产品都是类似的，同时不同的工厂有一定的相同逻辑，那么就用这种模式

平时在项目中使用相对频率较低，但是在阶段一中，我们会找一个合适的case来为大家演示如何使用这种设计模式，初步定的是会跟命令模式结合起来使用

853769620