工厂模式

### 作用

1. 同意管理对象的创建。如果修改时方便维护。例如：某个类需要子类替换时，只需要修改一处即可。
2. 进一步解耦。

**分类：简单工厂模式、工厂方法模式、抽象工厂模式**

### 简单工厂模式

简单工厂直接通过需要生产产品的name或type类区分即可。

**缺点**：不符合开闭原则（对扩展开放；对修改封闭），每当要添加一个产品类型时，需要修改工厂类，加上新产品的if判断。解决方法，将工厂进行抽象，每个子工厂只生产一种产品。添加一个产品时，只需添加一个子工厂类，而不用修改工厂类。

**public** **class** SimpleFactory {

**public** **static** Doll getInstance(String name){

Doll doll=**null**;

**if**("cloth".equals(name)){

doll=**new** ClothDoll();

}

**if**("barbie".equals(name)){

doll=**new** BarbieDoll();

}

**return** doll;

}

}

### 工厂方法模式

添加一个产品时，只需添加一个子工厂类，而不用修改工厂类。**更加符合开闭原则**（对扩展开放；对修改封闭）。

缺点：会有很多工厂类

### 抽象工厂模式

一个工厂生成相配套的几种产品。