1. **工厂方法 Factory Method**



Fig 1: Factory Method

抽象类Creator中createProduct()方法被子类ConcreteCreator重写，工厂实际生产的产品类型是由子类决定的。

当多个不同类型对象有相同的业务逻辑时，可以使用工厂方法构建对象来减少代码修改量：把共同的业务逻辑放在父类方法someOperation()中，具体的业务逻辑由继承工厂类决定。

所生产的对象需要实现统一的接口，这样子工厂子类才可以重写createProduct()方法。

e.g.

|  |
| --- |
| Creator creator = new ConcreteCreatorA();  creator.someOperation();  //如果此时业务需要新增一个类，其业务逻辑与ConcreteCreatorA类似，我们就不需要重新写一个类，只需要写一个新的工厂子类，子类生产对应的对象。  Creator creator = new ConcreteCreatorB();  //类似creator.someOperation();等业务语句都不需要修改。 |

1. **抽象工厂方法 Abstract Factory**