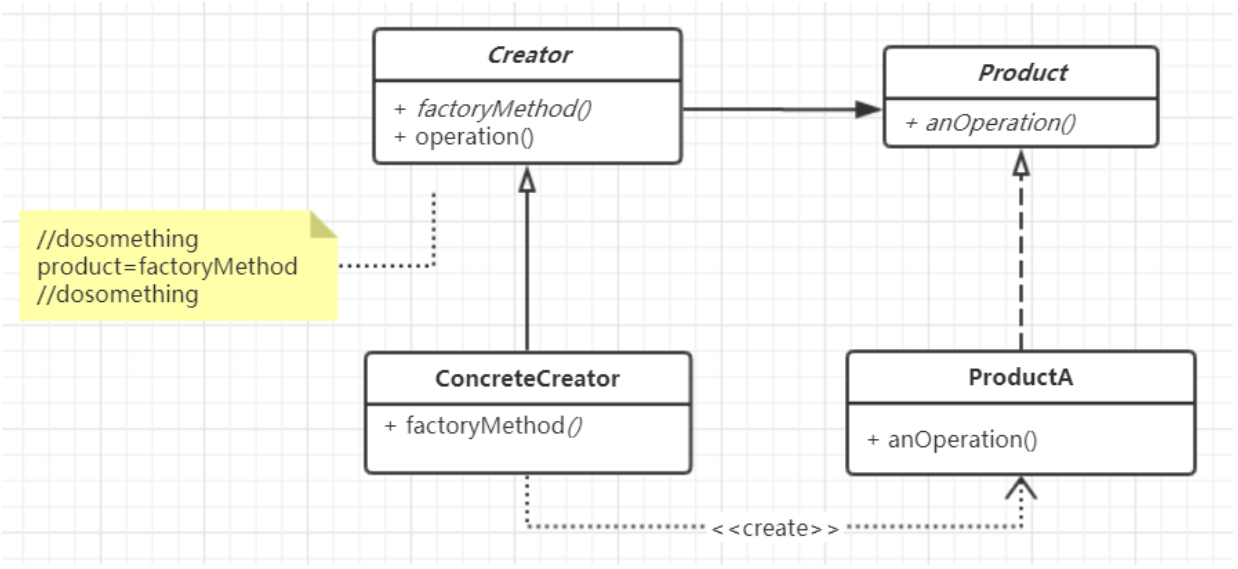
工厂方法模式

Factory Method

郭嘉

模式定义:

定义一个用于创建对象的接口，让子类决定实例化哪一个类。Factory Method 使得一个类的实例化延迟到子类



简单工厂：

```
class SimpleFactory {
 public static Product createProdcut(String type) {
 if (type.equals( "0" )) {
 return new ProductA();
 } else if (type.equals( "1" )) {
 return new ProductA1();
 } else {
 return null;
 }
  }
}
```

工厂方法：

```
// 稳定接口
interface Product {
 public void method1();
}
// 具体实现
class ProductA implements Product {

 public void method1() {
 System.out.println( "ProductA.method1 executed. " );
  }
}

class ProductA1 implements Product {

 public void method1() {
 System.out.println( "ProductA1.method1 executed. " );
  }
}



abstract class Application {
```

```
23
24   abstract Product createProduct();
25
26   Product getObject() {
27   Product product=createProduct();
28   // ...
29   // ...
30   return product;
31   }
32
33 }
34 // 工厂方法具体实现类
35 class ConcreteProductA extends Application {
36   @Override
37   Product createProduct() {
38   // ....
39   return new ProductA();
40   }
41 }
42
43 class ConcreteProductA1 extends Application {
44   @Override
45   Product createProduct() {
46   //...
47   return new ProductA1();
48   }
49 }
```

应用场景

1.当你不知道改使用对象的确切类型的时候
2.当你希望为库或框架提供扩展其内部组件的方法时

主要优点：
1. 将具体产品和创建者解耦

2. 符合单一职责原则

3. 符合开闭原则

源码中的应用：

```
// java api
// 静态工厂方法

Calendar.getInstance()
java.text.NumberFormat.getInstance()
java.util.ResourceBundle.getBundle()

// 工厂方法
 java.net.URLStreamHandlerFactory
javax.xml.bind.JAXBContext.createMarshaller
```