

生成器模式

1. 需求背景

将信息以多种形式导出

2. 直接实现的问题

如果直接实现, 那么每一种形式的导出都要重新写一遍实现方式, 而每种实现方式的步骤都一样, 形成了思维上的重复(同样需要封装)或者叫步骤的重复, 方法的重复.

其实可以说很多的模式都是封装过程成对象或者抽象成接口, 有的是封装构造过程, 有的是封装动作过程.

3. 其实该模式也是工厂, 只不过该工厂是用来生产步骤的.

4. 具体实现:

生成器模式测试类

```
public static void main(String[] args) {  
    IBuilder builder=new Builder();  
    Director director=new Director(builder);  
    director.getResult();  
    IBuilder builder2=new Builder2();  
    director.setBuilder(builder2);  
    director.getResult();  
}
```

可以看出

对象主要有: Builder, Builder2, IBuilder, Director

然后Director的构造方法里有IBuilder , Director通过getResult方法即可实现导出文件

对象之间的关系很明确了, Builder, Builder2实现了IBuilder接口, 通过Director的构造方法来确定选择用的生成器, 而Director的getResult方法则封装了实现的步骤即IBuilder方法的执行顺序.

其实可以进一步把Director弄成单例模式, 因为一般来说这种对象有一个实例就行了, 如果要切换生成器的话, 可以用set方法直接注入进去.