# 一、类之间的关系

关联关系：使用实例变量实现；

聚合关系：整体和个体之间的关系；

合成关系：整体和部分之间的关系；

依赖关系：体现为局域变量、方法的参量，以及对静态方法的调用；

# 二、面向对象的设计原则

1、开-闭原则

一个软件实体应当对扩展开发，对修改关闭；

2、里氏代换原则

任何基类可以出现的地方，子类一定可以出现；

3、依赖倒转

要依赖于抽象，不要依赖于实现；

4、接口隔离

使用多个专门的接口比使用单一的总接口要好；

5、组合/聚合复用原则

要尽量使用合成/聚合，尽量不要使用继承；

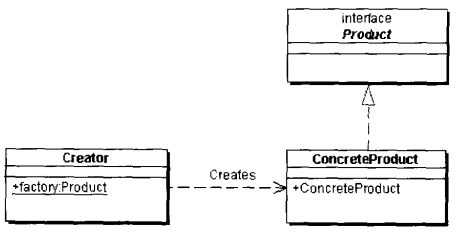
6、迪米特法则

一个对象应当对其他对象有尽可能少的了解；

# 三、创建模式

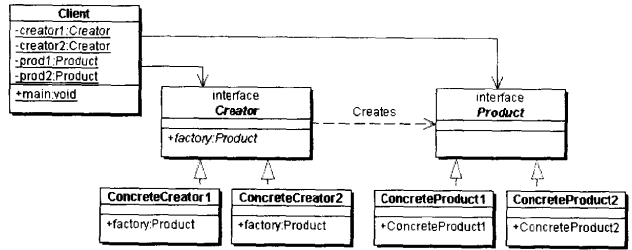
## 1、简单工厂模式/静态工厂方法模式

由一个工厂类根据传入的参量决定创建出哪一种产品类的实例。

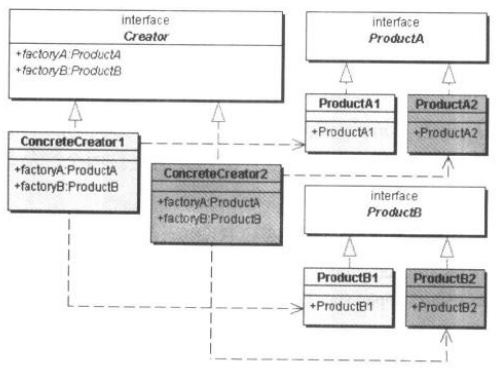


## 2、工厂方法模式/虚拟构造子模式

工厂方法模式的核心是一个抽象工厂类，



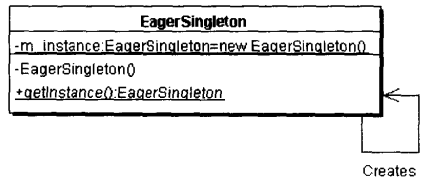
## 3、抽象工厂模式



## 4、单例模式

确保某一个类只有一个实例，而且自行实例化并向整个系统提供这个实例。

（1）、饿汉式单例类：类在加载时就被实例化；



（2）、懒汉式单例类：第一次被引用时将自己实例化；

## 5、多例模式

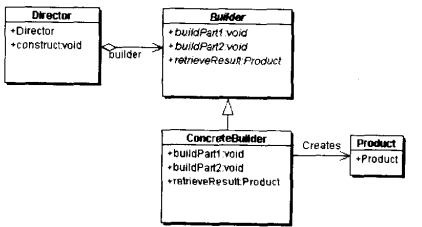
多例类：允许有限多个实例，并向整个JVM提供自己实例的类；

特点：（1）、可以有多个实例；（2）、多例类必须自己创建、管理自己的实例，并向外界提供自己的实例；

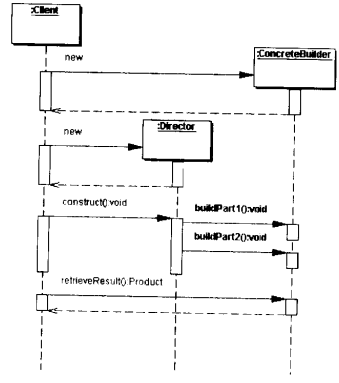
## 6、建造模式

是对象的创建模式，可以将一个产品的内部表现与产品的生成过程分割开来。将一个复杂的构建与其表示相分离，使得同样的构建过程可以创建不同的表示。 [构建与表示分离，同构建不同表示]

建造模式是将复杂的内部创建封装在内部，对于外部调用的人来说，只需要传入建造者和建造工具，对于内部是如何建造成成品的，调用者无需关心。



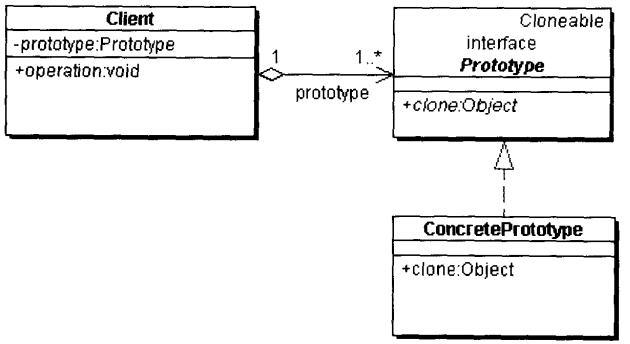
UML序列图：



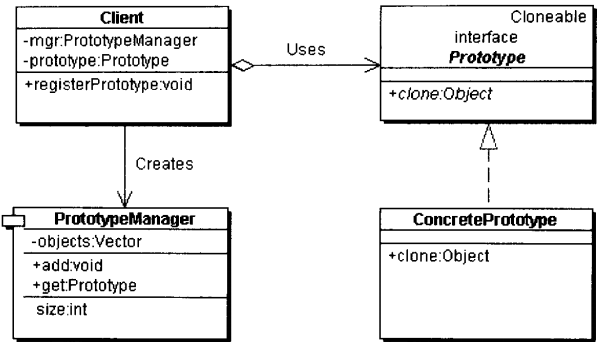
## 7、原始模型模式

属于对象的创建模式，通过给出一个原型对象来指明所要创建的对象的类型，然后用复制这个原型对象的方法创建出更多同类型的对象。

（1）、简单形式：原型对象数目较少而且比较固定；



（2）、登记形式：要创建的原型对象数目不固定的话；



模式的实现：深复制和浅复制

（1）、浅复制：仅仅复制所考虑的对象，而不复制它所引用的对象；

（2）、深复制：把要复制的对象所引用的对象都复制了一遍；

# 四、结构模式

# 五、行为模式