

# DesignPatterns\_iOS

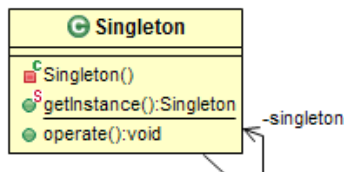
Personal scribble demos while learning design patterns.

设计模式主要分三个类型:创建型、结构型和行为型。

## 创建型

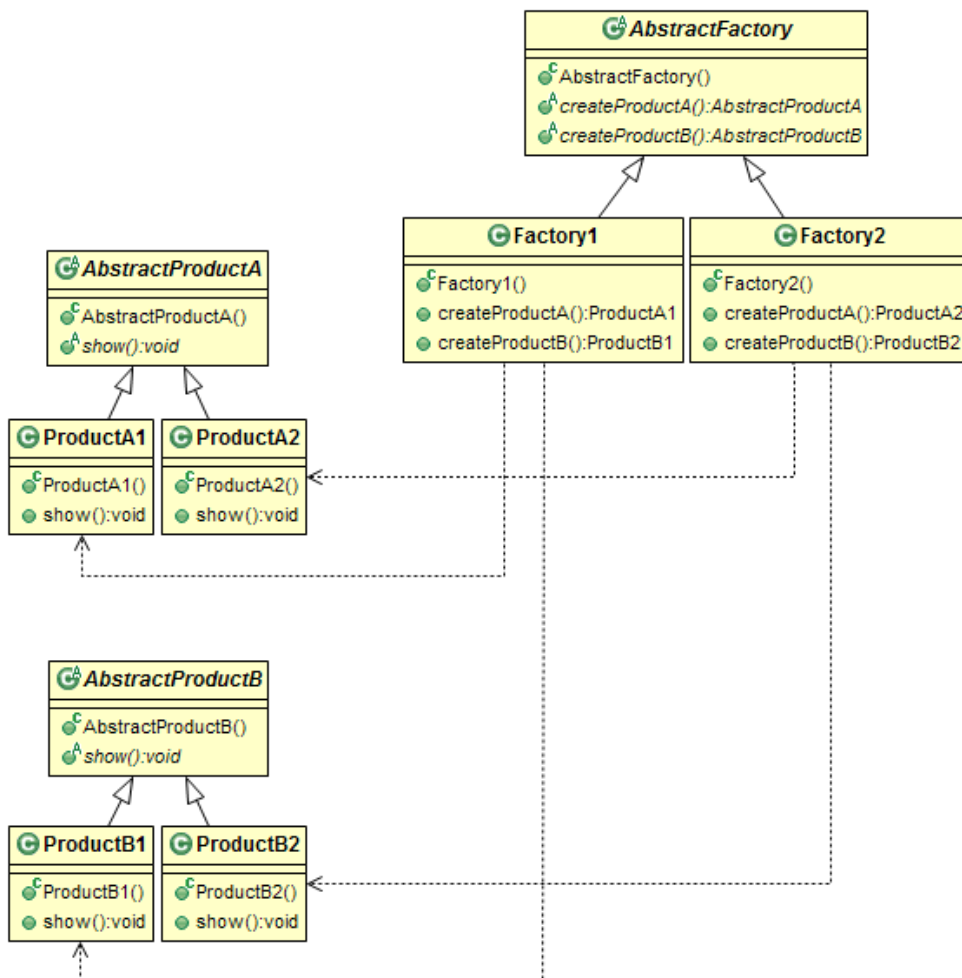
### 一、Singleton，单例模式

保证一个类只有一个实例，并提供一个访问它的全局访问点



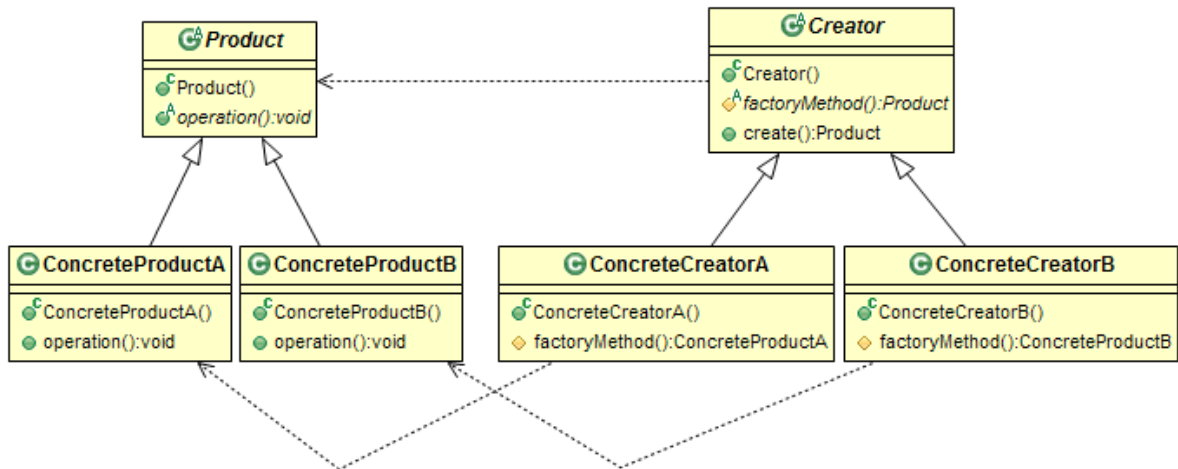
### 二、Abstract Factory，抽象工厂

提供一个创建一系列相关或相互依赖对象的接口，而无须指定它们的具体类。



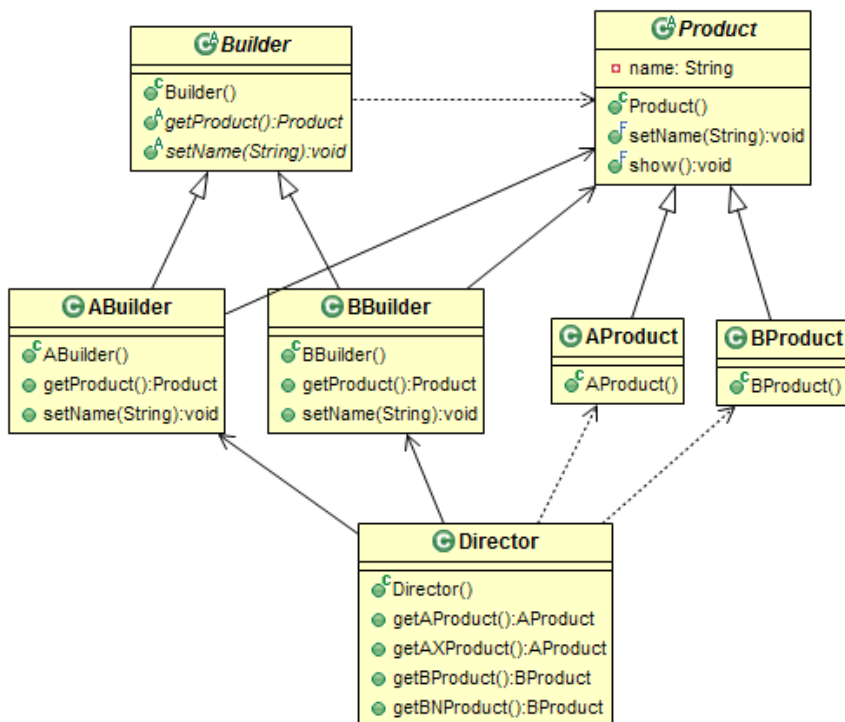
### 三、Factory Method, 工厂方法

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使一个类的实例化延迟到了子类。



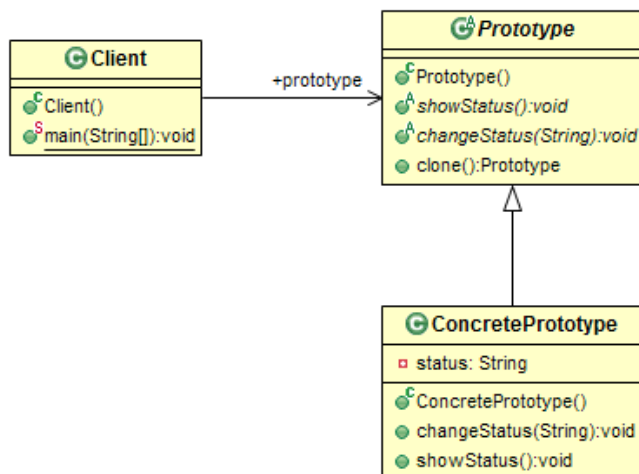
### 四、Builder, 建造模式

将一个复杂对象的构建与他的表示相分离，使得同样的构建过程可以创建不同的表示。



### 五、Prototype, 原型模式

用原型实例指定创建对象的种类，并且通过拷贝这些原型来创建新的对象。



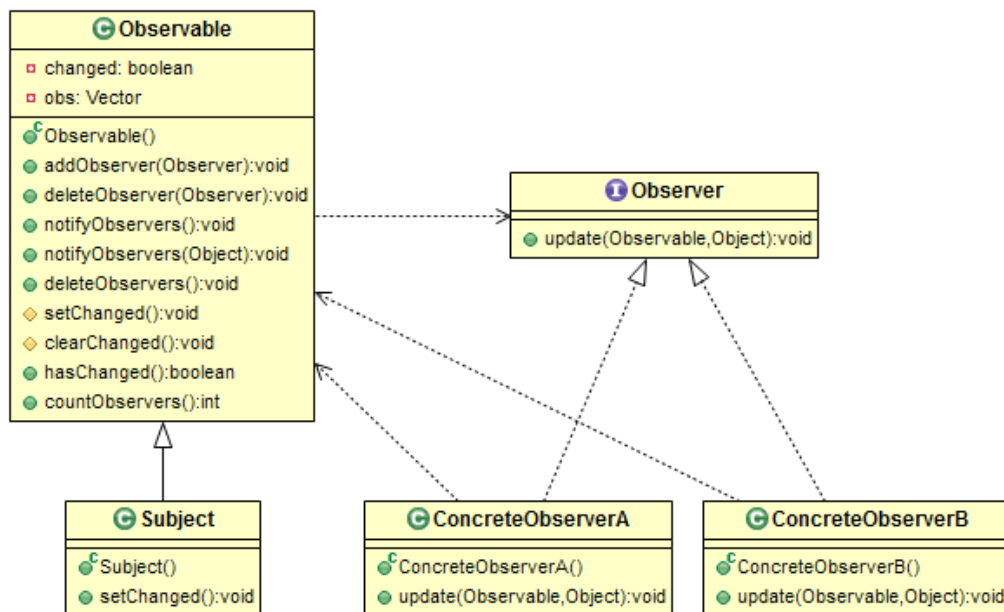
## 行为型

### 六、Iterator，迭代器模式

提供一个方法顺序访问一个聚合对象的各个元素，而又不需要暴露该对象的内部表示。

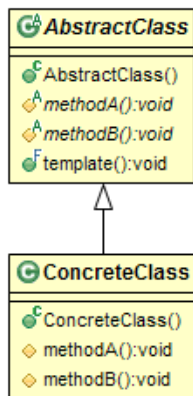
### 七、Observer，观察者模式

定义对象间一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知自动更新。



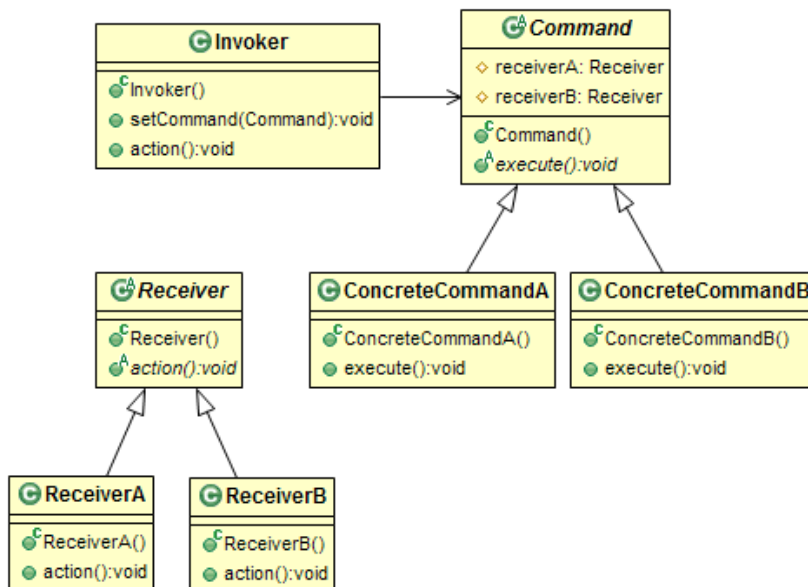
### 八、Template Method，模板方法

定义一个操作中的算法的骨架，而将一些步骤延迟到子类中，TemplateMethod使得子类可以不改变一个算法的结构即可以重定义该算法得某些特定步骤。



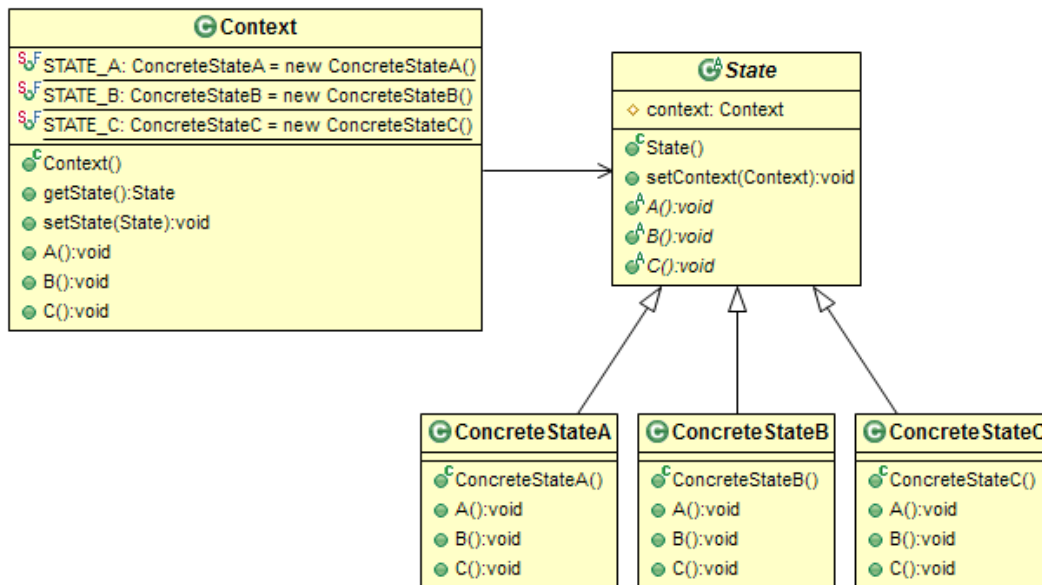
## 九、Command，命令模式

将一个请求封装为一个对象，从而使你可以用不同的请求对客户进行参数化，对请求排队和记录请求日志，以及支持可撤销的操作。



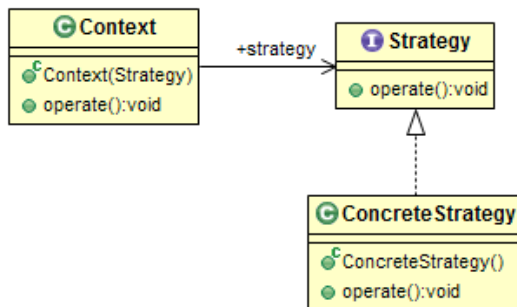
## 十、State，状态模式

允许对象在其内部状态改变时改变他的行为。对象看起来似乎改变了他的类。



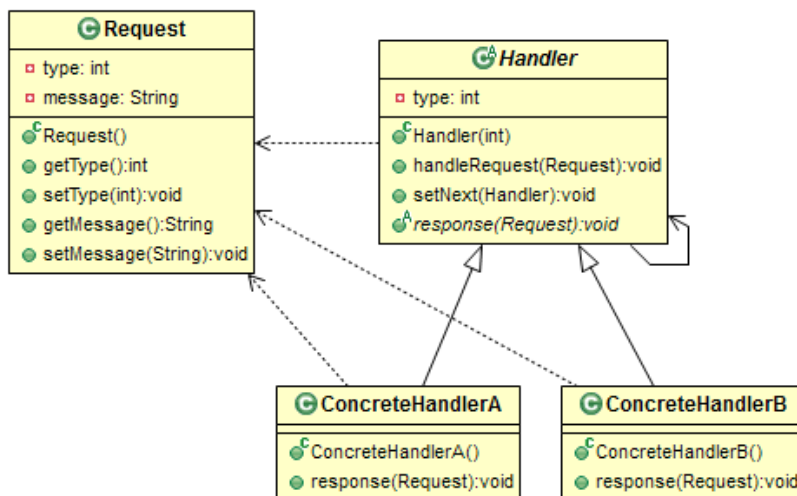
## 十一、Strategy，策略模式

定义一系列的算法，把他们一个个封装起来，并使他们可以互相替换，本模式使得算法可以独立于使用它们的客户。



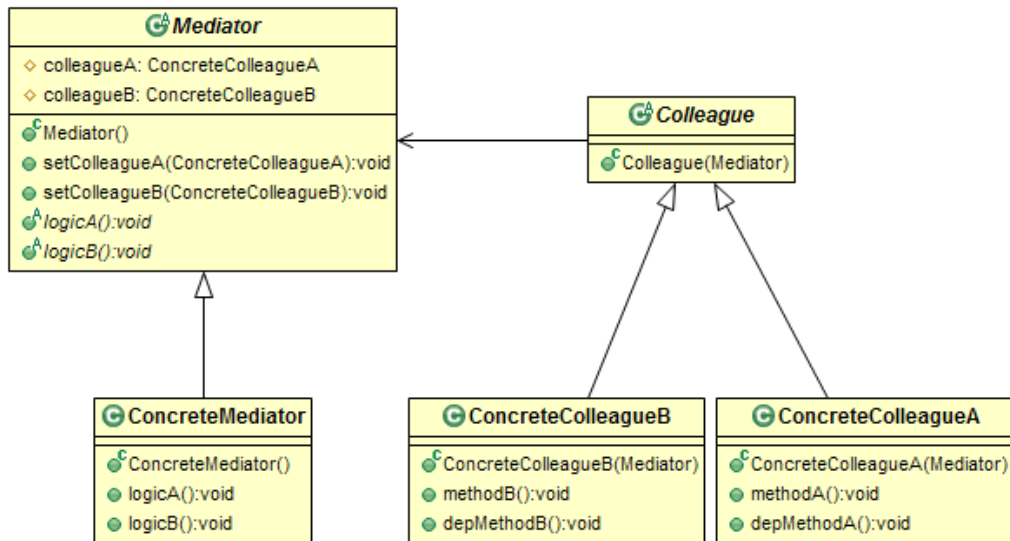
## 十二、Chain of Responsibility，职责链模式

使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系



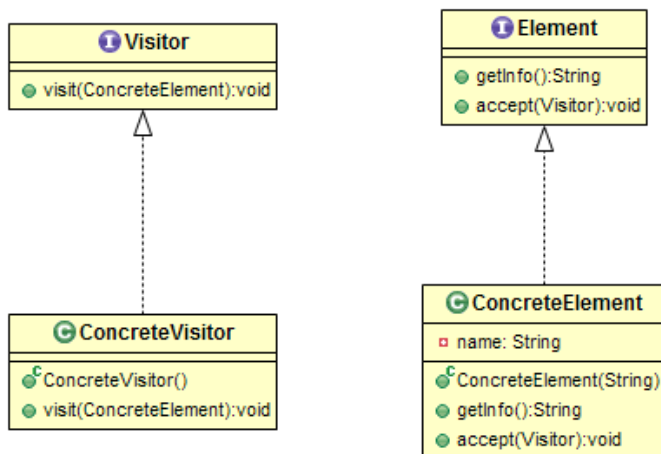
## 十三、Mediator，中介者模式

用一个中介对象封装一些列的对象交互。



#### 十四、Visitor，访问者模式

表示一个作用于某对象结构中的各元素的操作，它使你可以在不改变各元素类的前提下定义作用于这个元素的新操作。

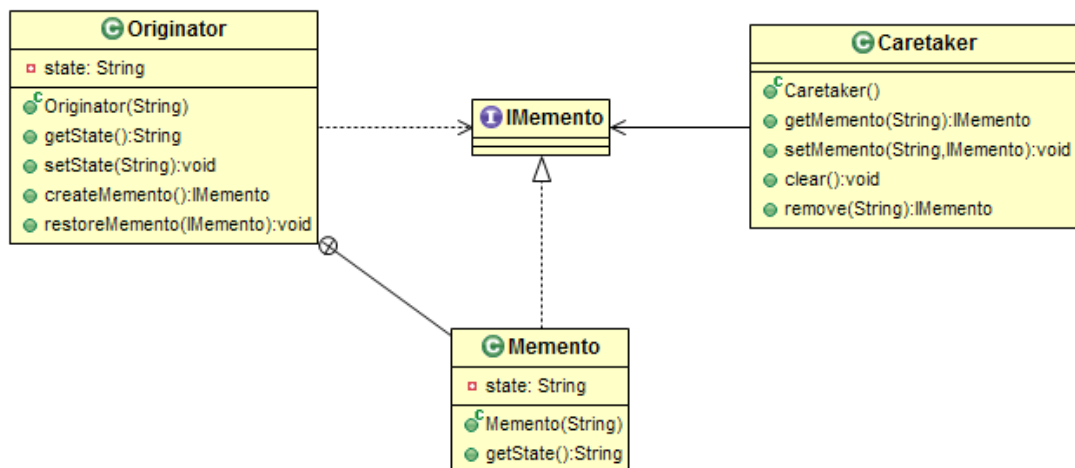


#### 十五、Interpreter，解释器模式

给定一个语言，定义他的文法的一个表示，并定义一个解释器，这个解释器使用该表示来解释语言中的句子。

#### 十六、Memento，备忘录模式

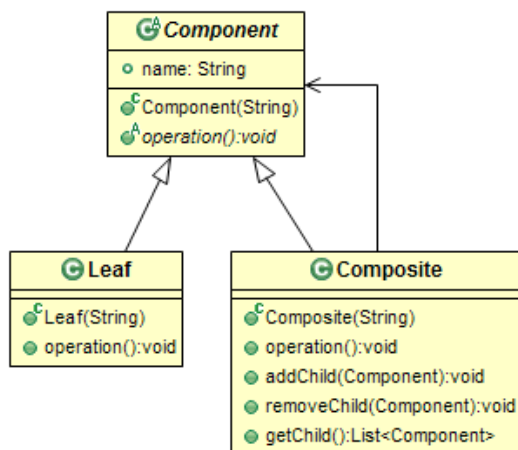
在不破坏对象的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。



结构型有

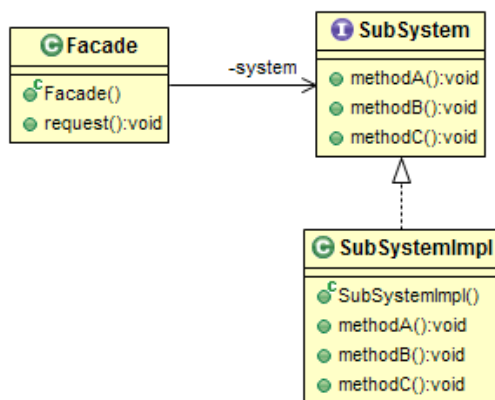
## 十七、Composite，组合模式

将对象组合成树形结构以表示部分整体的关系，Composite使得用户对单个对象和组合对象的使用具有一致性。



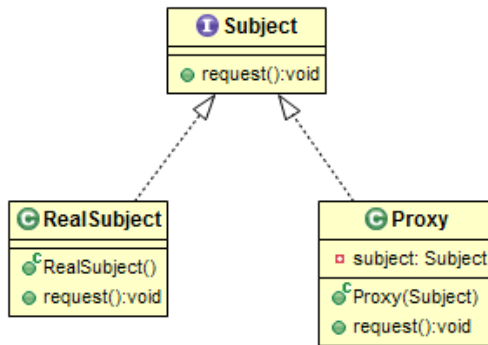
## 十八、Facade，外观模式

为子系统的一组接口提供一致的界面，facade提供了一高层接口，这个接口使得子系统更容易使用。



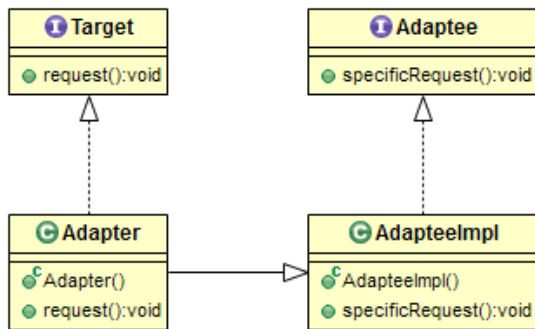
## 十九、Proxy，代理模式

为其他对象提供一种代理以控制对这个对象的访问



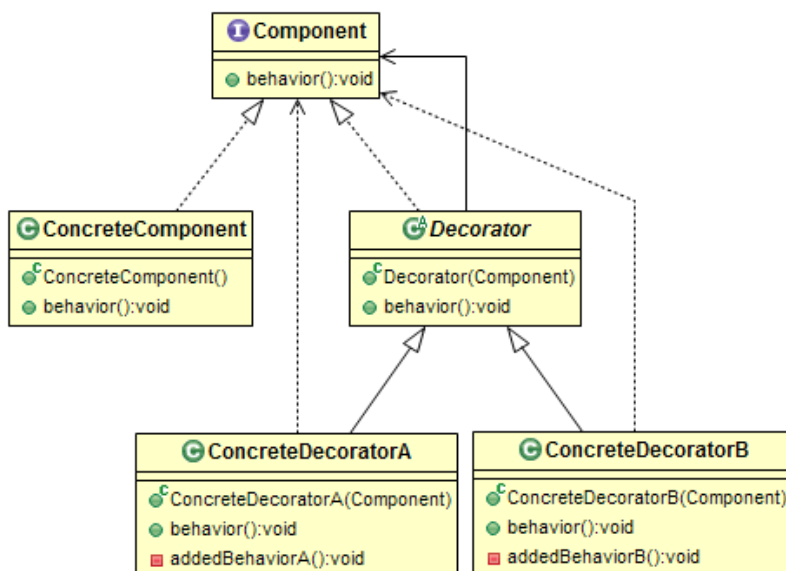
## 二十、Adapter,适配器模式

将一类的接口转换成客户希望的另外一个接口，Adapter模式使得原本由于接口不兼容而不能一起工作那些类可以一起工作。



## 二十一、Decorator，装饰模式

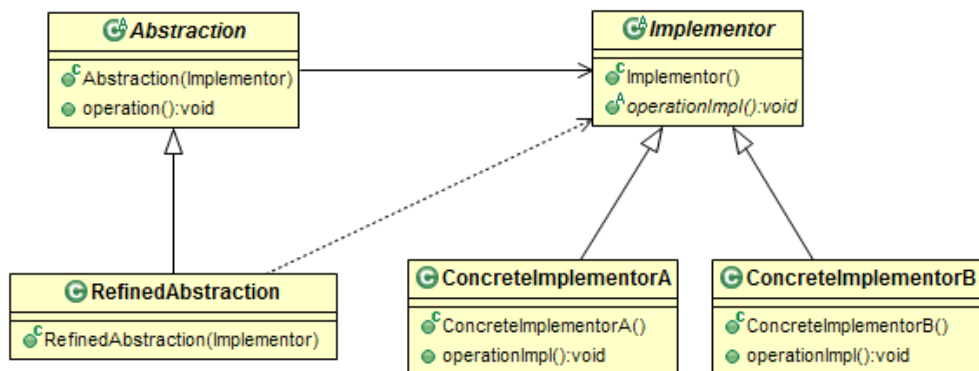
动态地给一个对象增加一些额外的职责，就增加的功能来说，Decorator模式相比生成子类更加灵活。



## 二十二、Bridge，桥模式

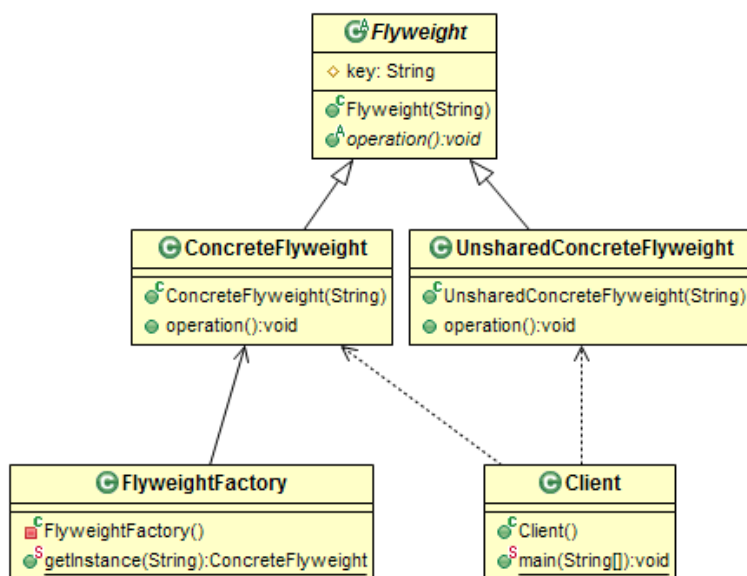
将抽象部分与它的实现部分相分离，使他们可以独立的变化。





## 二十三、Flyweight, 享元模式

运用共享技术有效的支持大量细粒度的对象。



除了以上23个模式之外，根据《java与模式》中的分类，还有4种模式，DefaultAdapter 缺省适配器模式，SimpleFactory 简单工厂模式，Multiton 多例模式，Immutable 不变模式。