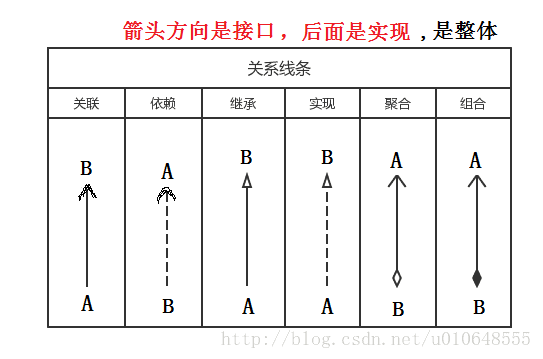
## 设计模式专题

1. 关系线条



1. 设计模式三大类型
2. 创建型
3. 工厂模式
4. 抽象工厂模式
5. 单例模式
6. 建造者模式
7. 原型模式
8. 结构型
9. 适配器模式
10. 装饰者模式
11. 代理模式
12. 外观模式
13. 桥接模式
14. 组合模式
15. 享元模式
16. 行为型
17. 策略模式 ② 模板模式 ③ 观察者模式 ④ 迭代子模式 ⑤ 责任链模式

⑥ 命令模式 ⑦ 备忘录模式 ⑧ 状态模式 ⑨ 访问者模式 ⑩ 中介模式

① 解释器模式

1. 设计模式六大原则
2. 开闭原则：对扩展开放、对修改闭合
3. 替换原则：父类出现的地方、子类一定能出现
4. 依赖倒置原则：具体依赖于抽象、底层依赖于顶层
5. 接口隔离原则：模块间要通过接口隔离开来，而不是通过具体的类来强耦合
6. 迪米特法则：又最少知识原则，一个对象应该对其他对象有尽可能少的了解。
7. 合成复用原则：尽量使用合成/聚合的方式，而不是继承
8. 设计模式详解
9. 观察者模式

在一个被观察的主体对象中，聚合了一个观察者的集合对象。当主体发生改变时，遍历这个集合通知所有观察者。

1. 模板模式

将一个流程中的一些方式定义成抽象方法，让子类继承实现，不同子类可以有不同的实现方式；和一些钩子方法，留待子类扩展应用；和一个不可重写重写的方法来实现整个流程功能。

1. 策略模式

在一个上下文中，依赖一个抽象的策略类对象。再对这一策略对象实现不同的算法。上下文类可以住人不同的实例策略对象，形成不同的算法。

1. 门面模式

将用户需要的接口全部定义到一个门面类中，客户只需要操作这一个门面类，即可满足所有需求。

1. 享元模式

用一个享元工厂聚合所有的享元对象，实现对象缓存功能。如shiro的缓存机制。

1. 代理模式

在一个代理类中依赖实际类，并实现一些增强的功能。

1. 装饰者模式

在装饰类中依赖抽象类，并实现一些增强功能，再实现装饰类的子类去具体实现方法。

1. 合成模式

在合成类中聚合一个抽象类集合对象，且这一合成类实现了抽象接口，用于数据遍历，如文件遍历。

1. 适配器模式

客户端所需要的接口，通过适配类适配成服务端有的接口，适配类为implement 客户端接口，extend服务端接口。

1. 原型模式

给出一个原型对象，来指明所有创建对象的类型，然后复制这个原型创建更多的类对象。

1. 建造模式

通过一个建造者对象，将具体建造对象建造出来的零件，组成完整的产品对象，返回给客户端，客户端不需要知道具体零件的生产过程。

1. 单例模式

确保一个类，只有一个实例。

1. 工厂模式

具有抽象产品和抽象工厂。将一个产品的创建分成2个逻辑。

1. 抽象工厂模式

具有抽象工厂，抽象产品（产品等级），产品族，将一个具体的产品族返回给客户端。

