1. 类图介绍

1.关联关系（被箭头所指的类作为另一个类的成员）

单向关联



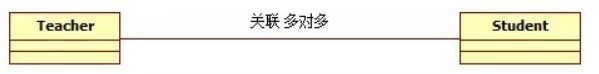
单向关联：水与气候是关联的

双向关联



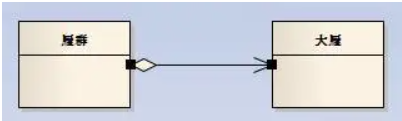
单向关联：水与气候是互相关联的

多重关联



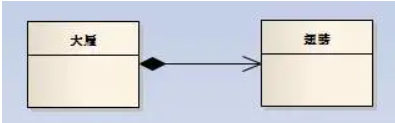
多重关联：一个老师负责教多个学生，一个学生可以学习多个老师的课程

聚合关系



聚合：大雁是雁群一部分，而大雁具有独立意义

组合关系

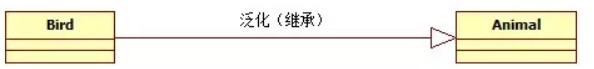


组合：翅膀是大雁一部分，而翅膀与大雁密不可分

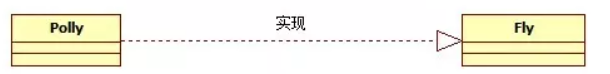
2.依赖关系（被箭头所指的类作为另一个类的方法的函数参数，耦合最弱）



3.继承关系（泛化关系）

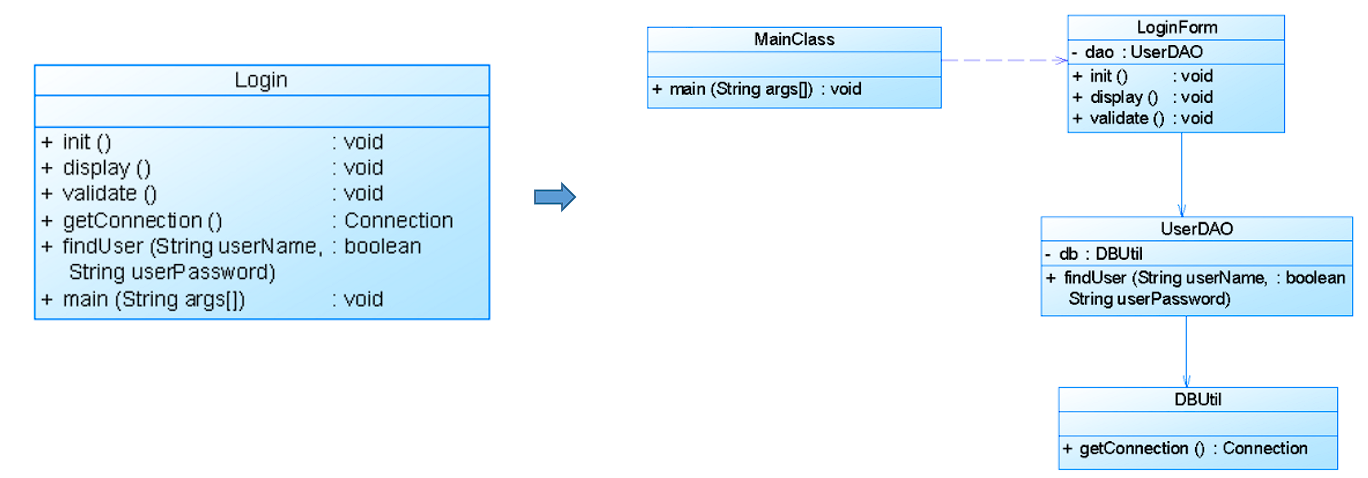


4.实现关系（C++中可以继承于没有成员变量的纯虚类，表示实现关系）



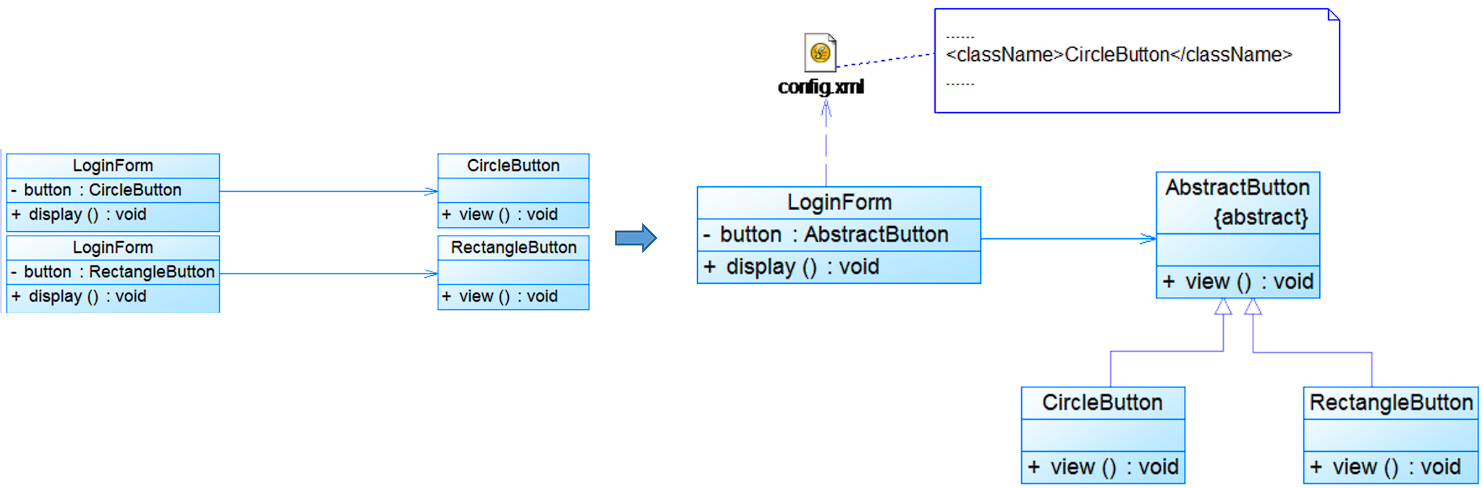
1. 设计原则
2. 单一职责原则

一个对象应该只包含单一的职责,并且该职责被完整地封装在一个类中



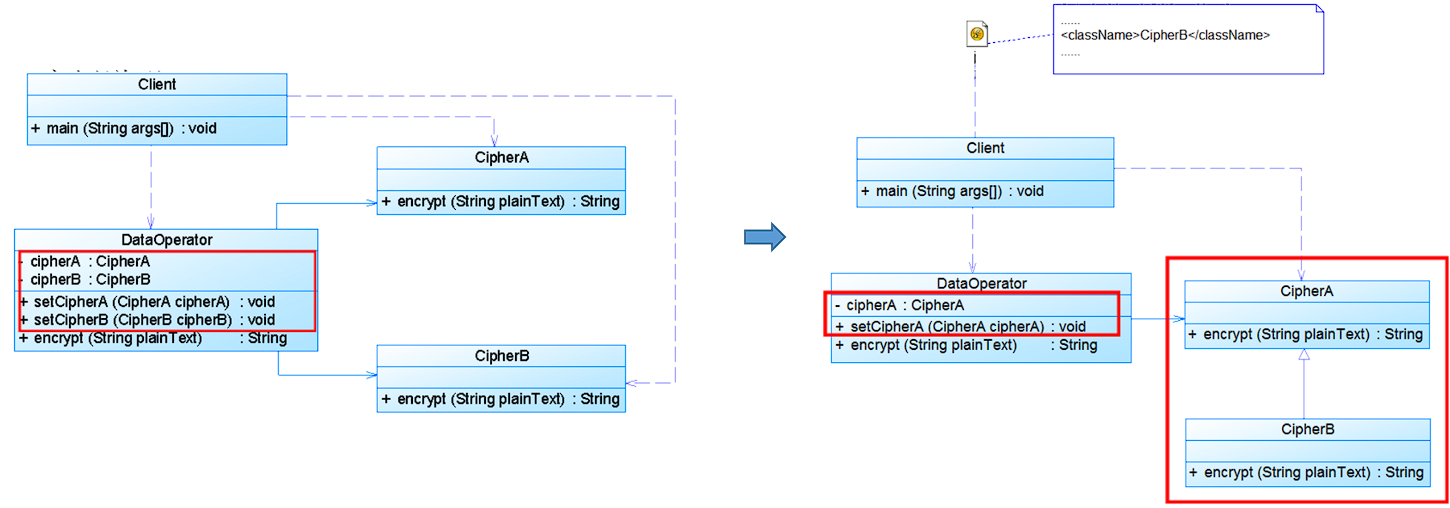
1. 开闭原则

对于扩展是开放的，但是对于修改是封闭的”，这意味着一个实体是允许在不改变它的源代码的前提下变更它的行为



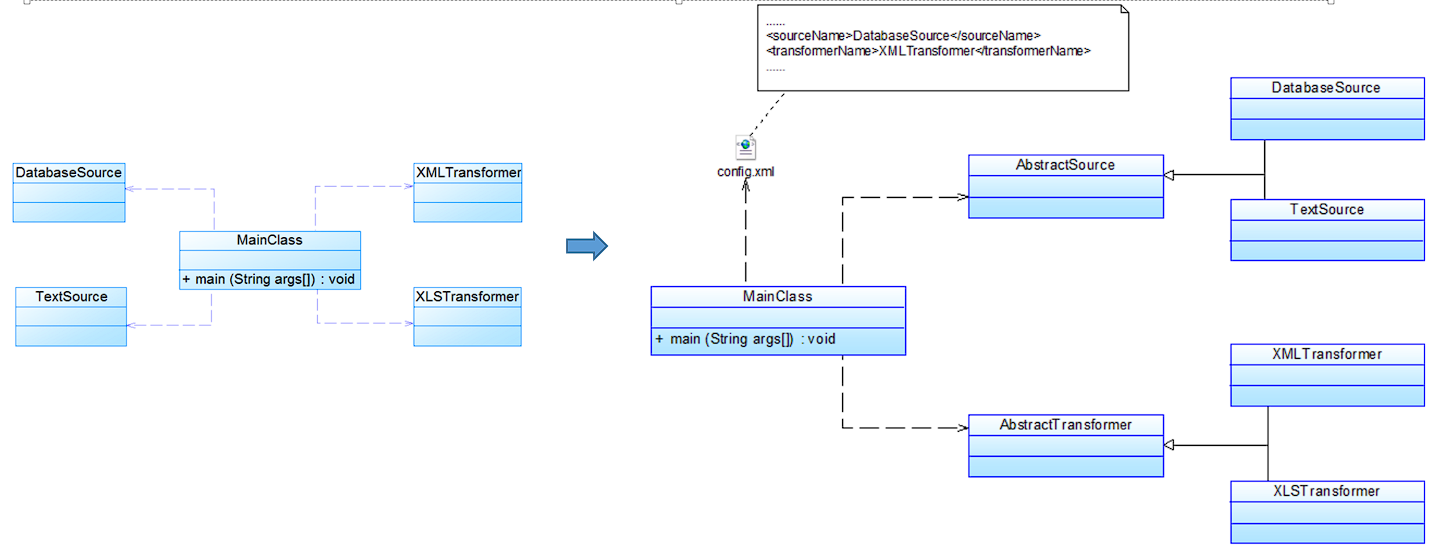
1. 里氏替代原则

所有使用基类的地方都能使用其子类对象，就是可以理解为多态特性



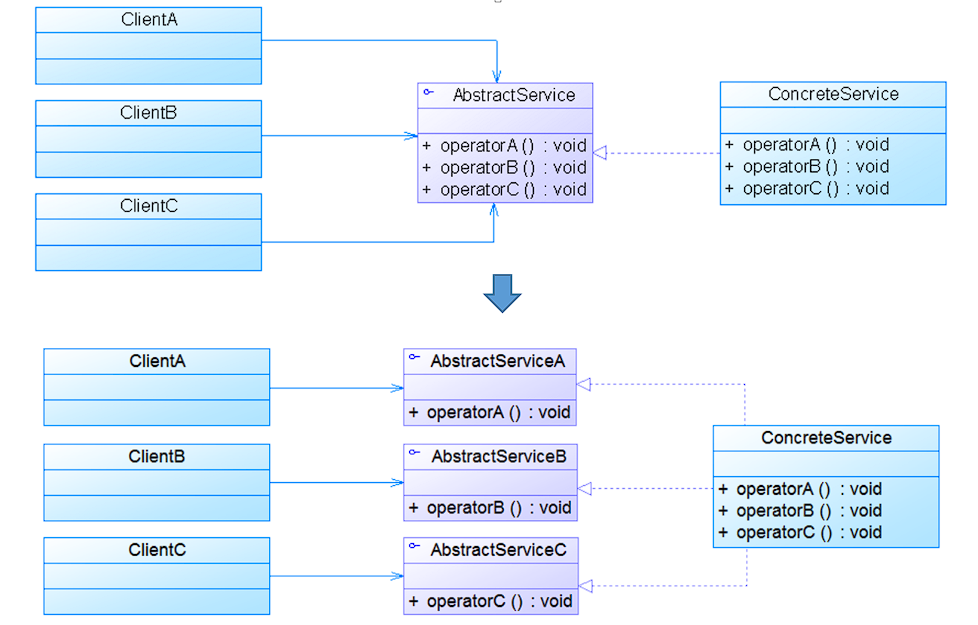
1. 依赖倒转原则

依赖抽象，不依赖细节。针对接口编程，而不针对实现编程。



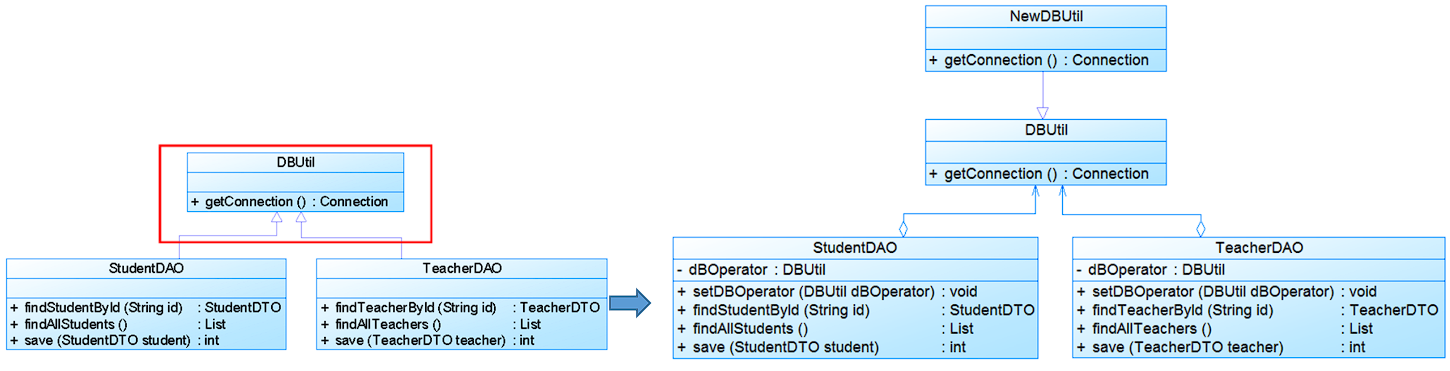
1. 接口隔离原则

一个类不应该依赖它不需要的接口，应把接口继续细分



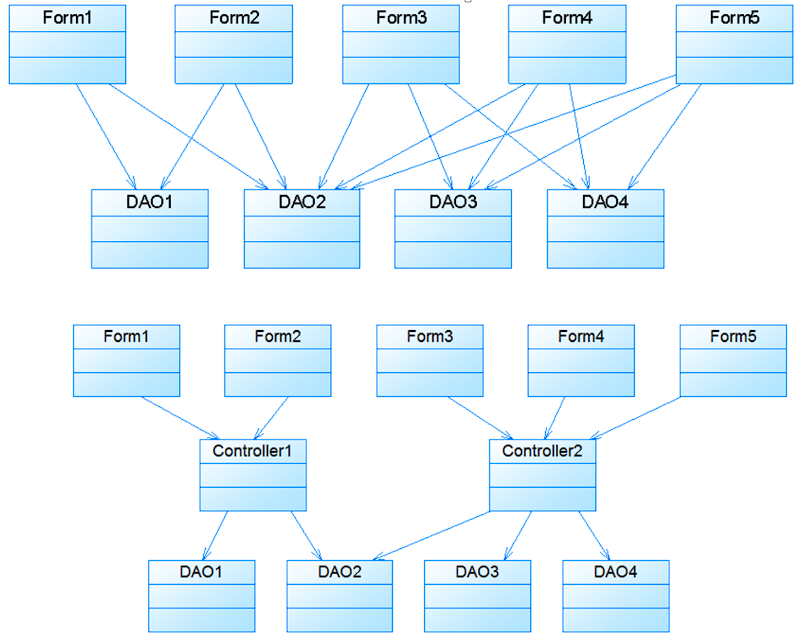
1. 合成复用原则

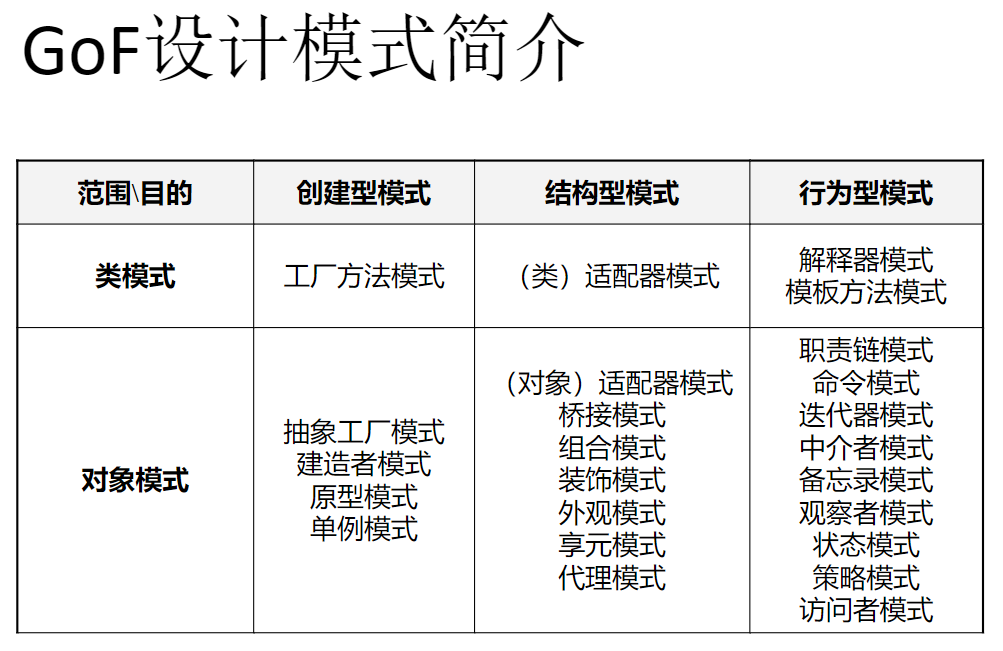
尽量使用聚合、组合，少使用继承



1. 迪米特法则

引入中间类，降低系统耦合度



1. GOF的23种设计模式

使用频繁程度：

抽象工厂模式(Abstract Factory) ★★★★★

建造者模式(Builder) ★★☆☆☆

工厂方法模式(Factory Method) ★★★★★

原型模式(Prototype) ★★★☆☆

单例模式(Singleton) ★★★★☆

适配器模式(Adapter) ★★★★☆

桥接模式(Bridge) ★★★☆☆

组合模式(Composite) ★★★★☆

装饰模式(Decorator) ★★★☆☆

外观模式(Facade) ★★★★★

享元模式(Flyweight) ★☆☆☆☆

代理模式(Proxy) ★★★★☆

职责链模式(Chain of Responsibility) ★★☆☆☆

命令模式(Command) ★★★★☆

解释器模式(Interpreter) ★☆☆☆☆

迭代器模式(Iterator) ★★★★★

中介者模式(Mediator) ★★☆☆☆

备忘录模式(Memento) ★★☆☆☆

观察者模式(Observer) ★★★★★

状态模式(State) ★★★☆☆

策略模式(Strategy) ★★★★☆

模板方法模式(Template Method) ★★★☆☆

访问者模式(Visitor) ★☆☆☆☆