1. 提炼子类

如果类的一些行为在某种情况下才会用到，那么提炼子类，将这些行为移到子类

1. 提炼超类

如果某些类有一些相同的行为，那么提炼超类

// 重构前

class 人{

    吃饭();

    睡觉();

    玩手机();

}

class 猴子{

    吃饭();

    睡觉();

    爬树();

}

// 重构后

class 动物{

    吃饭();

    睡觉();

}

class 人: 动物 {

    玩手机();

}

class 猴子: 动物{

    爬树();

}

1. 提炼接口

某些类有相同的子集，那么为这个子集提炼接口，前提接口要有意义

// 重构前

class 人{

    吃饭();

    睡觉();

    玩手机();

}

class 猴子{

    吃饭();

    睡觉();

    爬树();

}

// 重构后

interface I动物{

    吃饭();

    睡觉();

}

class 人: I动物 {

    吃饭();

    睡觉();

    玩手机();

}

class 猴子: I动物{

    吃饭();

    睡觉();

    爬树();

}

该示例与如上示例不同之处在于，该示例假设人与动物的“吃饭”“睡觉”方式不同

1. 以组合取代继承

如果子类只使用超类中的一部分行为，那么改为组合模式吧，不明白组合就百度吧，不想示例了