可复用性（Reuseability）, 顾名思义, 是指可以重复使用的程度. 对于我们的论坛项目, 主要考虑后端代码复用, 前端代码复用和线程复用.

## 后端代码复用

### 使用接口和抽象类

先定义接口或者抽象类, 而不是根据现实对象直接定义类. 用抽象类的实例去访问各种派生对象.

证据实例: 对于整个论坛系统, 有三种角色, 分别是游客, 注册用户, 管理员. 我们可以定义接口或抽象类User, 然后再定义Guest, RegisteredUser, Administrator, 这三个类都从User派生, 有公共的方法签名. 这样定义之后, 使用基类User的指针使用同样的逻辑操作用户, 管理员和游客. 比如, 现在有一个游客注册成了论坛用户, 则只需改变对象引用, 无需再写用户转换代码.

User(Abstract)

RegisteredUser(继承)

Guest(继承)

Admin(继承)

### 类之间相互关联依赖时的方法划分

论坛系统的各个实体, 不可能是相互独立的, 比如发帖这个动作, 必须要用到用户类和帖子类, 在这种情况下, 要合理划分方法, 尽可能与事件独立是一个能让代码最大限度复用的好办法.

具体实例: 用户发帖后, 可以给用户增加积分/经验/人气等, 该代码逻辑应该写在帖子类内部还是用户类内部? 如果写在帖子类内部, 则用户回复帖子时也有相关操作, 则需要重新再写一遍, 显然出现了冗余. 所以该逻辑应该写在用户类内部. 复用原则之一: 公用方法与事件分离.

## 前端代码复用

### 使用CSS

HTML的样式如果直接写在html文件中, 同样的样式会写很多遍, 显然属于无用功, 可以把样式分离到CSS文件中, 使用时调用.

具体实例: 论坛是Web应用, 则论坛的整个主题是确定的, 因而颜色基调是确定的, 很多界面的风格相似, 完全可以使用独立的CSS文件表示网页样式, 再结合Servlet动态生成每个页面的特殊格式, 可以达到最大程度的样式复用.

## 线程池

任何一个优秀的服务器(容器)必然要实现线程池, 尽可能最大限度地复用线程, 因为创建和销毁一个线程的代价太高了, 这样能省去大量服务器资源.

证据: 任何高并发服务器都是这么做的, 比如Apache Tomcat和Nginx.