# 第11讲 软件设计

问题越复杂解决问题所需要的花费更多，将复杂问题分解成富多个子问题解决会更加容易。

寻找最佳的模块数目

**过程抽象：把用文字形成的功能描述演变成代码。**

**数据抽象：**文字表达的数据结构**演变成数据库表和存储过程。数据抽象与过程抽象一样，能使设计者按不同的详细程度表示数据对象。**

信息隐藏是实现**抽象、模块化机制**的基本支撑

**耦合度**的等级：

① 非直接耦合（两个模块之间没有直接的关系——独立性最强）

② 数据耦合 （通**过参数**传递数据）

③ 特征耦合 ——标记耦合（一组模块通过**参数表**传递记录信息，就是标记耦合）

④ 控制耦合

⑤ 外部耦合 （**同一全局简单变量**）

⑥ 公共耦合（**同一公共数据环境**：例如全局数据结构、共享的通信区、内存的公共覆盖区、外存上的文件、物理设备）——————属于比**较 严重的耦合**

⑦ 内容耦合

应该应**尽量使用数据耦合，少用控制耦合，限制公共耦合的范围，完全不用内容耦合。**

**内聚性**

* 偶然性内聚（没有联系）
* 逻辑性内聚（传给模块的判定参数）
* 时间性内聚
* 过程性内聚（必须以特定次序执行）
* 通讯性内聚（使用了相同的输入数据，在同一数据结构上）
* 顺序性内聚（高内聚模块）
* 功能性内聚（**功能内聚是内聚度最高的类型**。）

**软件复用级别：**

几种形式的**代码复用**

**设计结果**复用（重用某个设计模型）

**分析结果**复用（重用分析模型）

**设计模式**

单一职责原则：**一个类只负责一项职责，**若一个类包含多个职责，则根据不同的职责重新划分类，使得每个类与其职责一一对应。

开放-封闭原则：要求设计的软件实体应该是可**以扩展但是不可以修改**

里式替换原则：子类可以**扩展**父类的功能，但不能**改变**父类原有的功能

接口隔离原则：客户端**不应该依赖**它不需要的接口，一个类对另一个类的**依赖应**该建立**在最小**的接口上，在建立单一接口时，不需要建立庞大臃肿的接口，应该尽量细化接口，并且接口中的方法尽量少。

依赖倒置原则：高层不能依赖低层，两者都应该依赖其抽象，抽象不应该依赖细节，细节应该依赖抽象

合成复用原则

迪米特法则