# 函数

## 过长函数

Extract Method

当函数过长，必须函数分解变小

## 过多变量

Replace Method with Method Object

将一个函数放进一个单独对象中，如此一来局部变量就成了对象内的字段。

## 参数赋值

Remove Assignments to Parameters

如果要对参数赋值，以一个临时变量取代该参数。

## 内联临时变量

Inline Temp

临时变量，可以省略的选择内联

## 内联函数

Inline Method

直接调用函数，避免引入临时变量

## 引入解释性变量

Introduce Explaining Variable

布尔表达式过于复杂时，引入中间布尔变量

## 分解临时变量

Split Temporary Variable

临时变量应该只被赋值一次，如果它们被赋值超过一次，就意味着它们在函数中承担了一个以上的责任。

循环变量、结果收集变量，例外。

## 单向关联与双向关联

一对一

一对多

多对一

多对多

## 以类取代类型码

Replace Type Code with Class

Replace Type Code with State/Strategy

## 以多态取代switch

Replace Conditional with Polymorphism

Replace Conditional with Subclass

Push Down Field

Push Down Method

## 去掉null的检查

Introduce Null Object

## 合并重复的条件片段

Consolidate Duplicate Conditional Fragments

把相同的代码移到条件表达式之前或者之后。

## 移掉控制标记

Remove Control Flag

以break或return语句取代

## 函数合并

Parameterize Method

不同函数合并到一个函数，使用参数来标识

Replace Parameter with Explicit Methods

一个函数分解为不同函数

## 参数

Introduce Parameter Object

# 类

## 移动代码

Move Field

Move Method

在该函数最常引用的类中建立一个有着类似行为的新函数。

Extract Class

Inline Class

## 以工厂方法取代构造函数

Replace Constructor with Factory Method

## 集成和委托

Pull Up Field

Pull Up Method

Push Down Field

Push Down Method

Extract Subclass（一个类放不下，添加一个子类）

Extract Interface（提取公共接口）

Extract Superclass（提取父类）

## 折叠集成体系

Collapse Hierarchy

超类和子类之间没有太多区别，合成一个。

## 塑造模板函数

Form Template Method

将一个原函数分解为多个函数，让子类分别去实现这些函数，然后把原函数上移至超类。

## 以委托取代继承

Replace Inheritance with Delegation

去掉继承，改成添加一个字段来调用方法。

## 以继承取代委托

Replace Delegation with Inheritance

当发现使用了委托中很多函数，而且都是简单的委托函数，则改为继承更好。

如果委托是为了数据共享，则必须使用委托。

# 大型重构

## 梳理并分解继承体系

Tease Apart Inheritance

一个继承体系同时承担了两项责任，建立两个继承体系，并通过委托关系让其中一个调用另一个。

## 将过程化设计转化为对象设计

Convert Procedural Design to Objects

将数据记录变成对象，将大块的行为分成小块，并将行为移入相关对象中。

## 将领域和表述/显示分离

Separate Domain from Presentation

将领域逻辑分离出来，为它们建立独立的领域类。

## 提炼继承体系

Extract Hierarchy

某个类做了太多工作，其中一部分工作是以大量条件表达式完成的。

建立继承体系，以一个子类表示一种特殊情况。