阅读C++ 编程思想 第10 章并总结关键内容

1. 模式分类

（1）创建型（Creational）：包括单件（Singleton）、工厂（Factory）和构建器（Builder）模式，用于对象创建，隔离细节，增加新类型时不需改代码。

（2）结构型（Structural）：代理（Proxy）和适配器（Adapter）模式，影响对象连接，确保变化不影响连接。常由工程限制决定。

（3）行为型（Behavioral）：包括命令（Command）、模板方法（Template Method）、状态（State）、策略（Strategy）、职责链（Chain of Responsibility）、观察者（Observer）、多派遣（Multiple Dispatching）和访问者（Visitor）模式，处理特定操作类型对象，封装操作过程。

2.简化习语

（1）信使

将消息封装到一个对象中到处传递， 而不是将消息的所有片段分开进行传递。

（2）收集参数

作用是从传递给它的函数中获取信息。

3.各模式介绍

（1）单件模式

最简单的设计模式，允许一个类有且仅有一个实例。

1. 命令模式

作用：消除代码间的耦合；

主要特点：允许向一个函数或者对象传递一个想要的动作。

1. 代理模式和状态模式

相同点：都提供一个代理类；

不同点：代理模式只有一个实现类，状态模式有多个实现。

1. 适配器模式

适配器模式使不兼容的接口能够交互。当两个库或代码段具有相同思想但表达不同时，适配器模式能快速找到解决方案，有效解决接口兼容性问题。

1. 模板方法模式

它的定义在基类中，并且不能改动。

1. 策略模式

在程序运行时，可以插入变化的代码。

1. 职责链模式

是一种使用策略对象的动态一般化，链中的每个策略都尝试满足调用，直到成功或链末，终止条件是达到链底或成功找到策略。

1. 工厂模式

使用通用工厂创建对象，不允许散布创建代码，所有创建对象的代码转到工厂执行，增加新对象时只需修改工厂。由于每个面向对象程序都需要创建对象，且可能通过添加新类型扩展程序，工厂模式可能是最有用的模式之一。

1. 抽象工厂

使用若干工厂方法模式，每个工厂方法模式创建一个不同类型的对象。当创建一个工厂对象时，要决定如何使用由那个工厂创建的所有对象。

（10）构造器模式

用于创建复杂对象，将对象的创建和它的“表示法”分开，创建过程保持原状，产生对象的表示法可能不同。