《软件构架实践》这本书主要分为四个部分：预想构架（1-3章）、创建构架（4-10章）、分析构架（11-13章）、从一个系统到多个系统（14-19章）。我将主要针对第二部分——创建构架来撰写学习笔记。

这本书第二部分要了解的重点是“设计师如何创建构架”，主要包括7章内容，分别是：理解质量属性、实现质量属性、空中交通管制：高可用性案例分析、设计构架、飞行模拟：构架可集成性案例分析、构架编档、软件构架重构。

质量属性：

系统从设计、实现到部署的整个过程中考虑质量属性的实现。质量属性包括下列三类：

（1）系统的质量属性。（可用性、可修改性、性能、安全性、可测试性和易用性）

（2）受架构影响的商业属性。（上市时间、成本和收益、所希望的系统生命期的长短、目标市场、推出计划、与老系统的集成）

（3）与架构本身相关的一些质量属性。（概念完整性、正确性与完整性、可构建性）

战术与架构模式的关系：

Active Objcet设计模式将方法执行从方法调用中分离出来，以增强并发，并简化对驻留在其自身控制线程中的对象的同步访问。

该模式由6个元素组成：代理，它提供了允许客户对主动对象调用公共访问方法的接口；方法请求，它定义了用于执行主动对象的方法的一个接口；激活接口，它维持了挂起方法请求的一个缓冲器；调度程序，它决定接下来执行什么方法请求；附属，他定义可建模为主动对象的行为和状态；将来，它允许客户获得方法调用的结果。

该模式的动机就是增强并发性——这是一个性能目标。因此其主要目的就是实现“引入并发“性能战术。然而，还要注意该模式包含的其他战术。

（1）信息隐藏（可修改性）。每个元素都选择了它将实现的责任，并将其实现隐藏在接口后面。

（2）仲裁者（可修改性）。该代理充当着把变化缓冲到方法调用中的仲裁者。

（3）绑定时间（可修改性）。主动对象模式假定对该对象的请求在运行时到达该对象。然而，并没有确定客户机与代理的绑定时间。

（4）调度策略（性能）。调度程序实现一些调度策略。

对设计师来说，分析过程包括理解嵌入在实现中的所有战术；设计过程包括在关于哪些战术最和将实现系统期望的目标方面，做出一个明智的选择。

架构模式和样式：

软件中架构模式与建筑物中的架构样式类似，它由几个将他们组合起来以维持架构完整性的关键特性和规则组成。架构模式由以下几个因素确定：

（1）一组元素类型（如数据存储库或计算数学函数的组件）

（2）指出其相互关系的元素的拓扑布局。

（3）一组语义限制（如管道——过滤器样式中的过滤器是纯数据转化器——他们以增量形式将其输入流转换为输出流，但并不控制上游流或下游元素）。

（4）一组交互机制（如子例程调用、事件——调阅者、黑板）、他们确定元素将如何通过允许的拓扑进行协调。

架构模式和战术之间是什么关系呢？正如已经说明的那样，我们把战术看作是设计的基本“构建块”，并根据该战术创建架构模式和策略。

设计架构：

几乎在我们遇到的所有成功的面向对象系统中都具有但失败的系统中缺少的两个特性是：存在一个强大的构架构想，应用管理良好的迭代式增量开发周期。

功能、质量和商业需求的某个集合“塑造”了构架。我们把这些塑造需求称为“构架驱动因素”。

构架设计必须按需求分析进行，但不需要在需求分析完成后在开始构架设计。实际上，在确定了关键的构架驱动因素后，就可以开始构架设计了。当设计了构架的足够多的部分后，就可以开始开发骨架系统了。该骨架系统在上面进行迭代开发（以及其在任何一个点交付的能力）的框架。组织结构对构架产生影响。

属性驱动的设计(ADD)是一个用于设计构架以满足质量需求和功能需求的方法。ADD把一组质量属性场景作为输入，并使用对质量属性实现和构架之间的关系的了解，对构架进行设计。

ADD步骤：

（1）选择要分解的模块

（2）根据这些步骤对模块进行求精

（3）对需要进一步分解的每个模块重复上述步骤

构架编档：

构架编档是创建构架的最有价值的一步。即使构架非常完美，但如果没有人理解它，或主要的涉众误解了它，它也没有什么用处。如果您创建了一个非常强大的构架，那么，就必须用足够的细节明确地描述它，并以一种其他人可以快速找到所需要信息的方式对其进行组织。否则，构架不会有什么用处，您所付出努力就白费了。捕获构架信息的描述方式采用UML表示法。

软件构架重构：

构架重构是一种解释、交互和迭代的过程，涉及许多活动；它不是自动进行的。它需要反向工程专家和设计师具备相关技能并投入精力，这在很大程度上是因为没有在源代码中清楚地表示构架构件。

软件构架重构由以下活动组成，这些活动以迭代的方式进行：

（1）信息提取

（2）数据库构造

（3）视图融合

（4）重构

