

## 1. 分层体系结构风格

### 1.1 概述

分层体系结构风格是一种常见的软件体系结构风格，它的设计原则是将一个复杂的系统分成多个层次，并将每个层次之间的通信限制为只在相邻层之间进行。

### 1.2 特点

下面是分层体系结构风格的一些特点：

- 每个层次都有一个特定的责任，层与层之间的接口是标准化的，并遵循严格的规则。
- 较高层次的模块不知道较低层次的细节，只需要知道如何与它交互，而较低层次的模块也不知道上层模块的使用方式，只需要按照规定的接口提供服务。

### 1.3 优点

分层体系结构风格的优点包括：

- 可维护性：分层结构的模块易于调试和修改，因为每个层次都独立于其他层次。
- 可扩展性：可以轻松添加新的层次或调整现有的层次。
- 可重用性：每个层次的模块都可以独立地进行测试和重用。
- 可靠性：分层结构的模块可以更容易地进行测试和验证，并且容错性更好。

### 1.4 缺点

分层体系结构风格的缺点包括：

- 性能问题：由于数据必须在各层之间传递，因此分层结构可能会导致较低的性能。
- 复杂性：随着层次的增加，系统变得更加复杂。
- 通信开销：每个层次都需要进行通信，这可能会增加通信开销。

### 1.5 历史背景

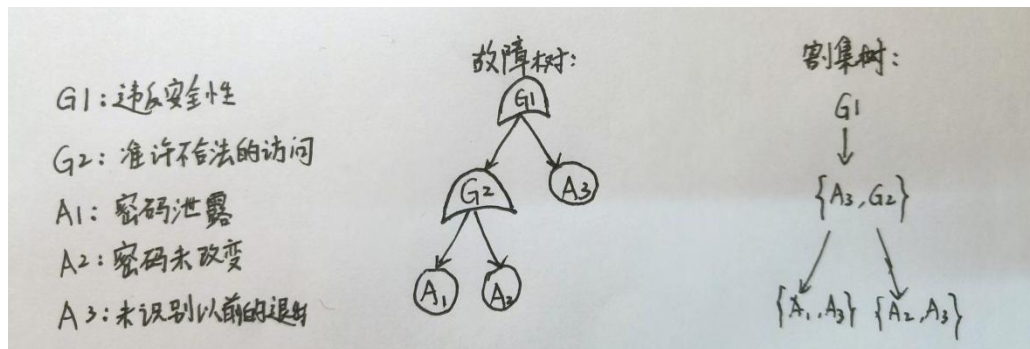
分层体系结构风格的历史背景可以追溯到 20 世纪 50 年代，当时 IBM 公司的研究员 D. L. Parnas 提出了模块化编程的概念，这奠定了分层体系结构的基础。随着计算机技术的发展，分层体系结构得到了广泛的应用，例如在大型数据库系统和分布式应用程序中都可以看到分层体系结构的应用。

### 1.6 应用场景

分层体系结构风格的实际应用非常广泛。例如，分层体系结构在大型 Web 应用程序中得到了广泛的应用，其中应用程序的各个层次可以分别部署在不同的服务器上。此外，分层体系结构也被用于设计和实现许多其他类型的系统，例如大型企业应用程序和嵌入式系统。

### 3.

#### 5.14 习题:



### 4.

属性	优先级	数据流风格	调用 / 返回风格	独立部件风格	虚拟机风格	数据共享风格
易于改变算法	1	5	1	4	3	1
易于改变数据表示	4	1	4	4	4	1
易于改变功能	3	5	1	1	2	4
好的性能	3	5	4	3	3	4
有效的数据表示	3	1	5	3	4	5
易于复用	5	4	5	3	2	1
总计		62	72	56	56	49