

Homework: 阅读《Quality Characteristics for Software Architecture》，简述软件体系结构的质量属性。

答：

| Characteristics | Description |
|-----------------|--|
| Functionality | The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions (what the software does to fulfil needs) |
| Reliability | The capability of the software product to maintain its level of performance under stated conditions for a stated period of time |
| Usability | The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions (the effort needed for use) |
| Efficiency | The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions |
| Maintainability | The capability of the software product to be modified. Modifications may include corrections, improvements or adaptations of the software to changes in the environment and in the requirements and functional specifications (the effort needed to be modified) |
| Portability | The capability of the software product to be transferred from one environment to another. The environment may include organizational, hardware or software environment |

文章中对于软件体系结构的质量属性的描述，主要是基于 ISO 9126-1 质量模型并对其进行适配和定制。ISO 9126-1 是一个国际标准，用于描述和评估软件产品的质量。该标准将软件质量定义为六个主要特性：功能性、可靠性、可用性、效率、维护性和可移植性，以及这些特性的若干子特性。文章通过调整这一标准模型，以适应特定的软件体系结构需求，进而详细探讨了每个质量属性在软件体系结构中的具体应用和重要性。

1. 功能性

功能性关注软件能否提供合适的功能以满足既定的需求，包括其正确性、适应性、互操作性、安全性。在软件体系结构领域中，功能性意味着架构需要设计成能够支持所有预期功能的实现，同时确保这些功能能够在不同的环境和条件下正常工作，确保数据和操作的安全。

2. 可靠性

可靠性强调软件的稳定性和错误恢复能力，涵盖了成熟度、容错性、可恢复性等子特性。软件体系结构必须能够确保系统即使在面对硬件故障、软件缺陷或外部攻击时，也能保持一定的性能水平，或能够在出现故障时迅速恢复。

3. 可用性

可用性包括用户对软件的易用性、理解性、学习性和吸引性的感知。对于软件体系结构而言，设计应考虑到如何使系统界面直观易用，确保用户能够轻松学习和使用系统功能，以及系统对于不同用户群体的可使用性。

4. 效率

效率关注系统在给定条件下对资源的使用率，包括时间效率和资源效率。在体系结构设计中，这意味着需要优化数据处理流程、算法和资源分配策略，以确保系统能够在满足功能和服务质量要求的同时，最小化对计算和存储资源的需求。

5. 维护性

维护性强调软件的可修改性、稳定性和测试性。软件体系结构应当易于分析、修改和扩展，支持错误定位和修复，以及适应环境变化和新需求的添加。这通常涉及到

高内聚、低耦合的设计原则，以及良好的文档和代码标准。

6. 可移植性

可移植性是指软件从一个环境迁移到另一个环境的能力，包括适应性、安装性和替换性。在软件体系结构层面，这意味着设计应当考虑到不同操作系统、硬件平台和第三方软件间的兼容性，以便软件能够轻松迁移或集成。

文章通过案例研究的形式，展示了如何将这些质量属性应用于实际的软件体系结构设计中。文章分析了一个股票交易监控系统的设计需求，探讨了不同体系结构方案如何满足这些质量要求，并比较了各方案的优缺点。这种方法不仅展示了质量属性在体系结构设计中的应用，也强调了在软件开发过程中评估和优化这些属性的重要性。