**10.12homework**

**List three characteristics that can serve as a guide to evaluate design quality.**

Firmness: A program should not have any bugs that inhibit its function. (坚固)可靠

Commodity: A program should be suitable for the purposes for which it was intended.（适用）实现所有用户需求功能

Delight: The experience of using the program should be pleasurable one.（愉悦）良好的用户体验

**Explain how effective modular design is achieved through functional independence of the individual modules?**

模块化设计的核心在于将一个复杂的系统分解成多个较小、更易管理的部分（即模块）。每个模块都围绕着特定的功能进行构建，并且尽可能地保持功能上的独立性。这意味着一个模块的变化不会直接影响到其他模块的工作，从而降低了整个系统的复杂度，提高了开发效率和系统的可维护性。

**Describe the principle of information hiding as it applies to software design.**

每个模块对其它所有模块都隐蔽自己的设计决策。即封装本模块中包含的信息（算法和数据），不要被（不需要这些信息的）其它模块访问，并且使用尽量少的全局变量。可以有效地控制模块间交互的方式，增强系统的安全性和稳定性，同时减少不同模块之间的耦合度。

**What is design refactoring?**

重构就是仅修改软件（模块）内部结构，而不改变其外部行为（接口属性）

**List the four design models required for a complete specification of a software design and the role of each.**

User model — 用户模型

Design model — 设计模型

Mental model (system perception) —用户的心理模型(系统感知)

Implementation model — 实现模型

**10.19homework**

**1.Why is architecture important?**

Representations of software architecture are an enabler for communication between all parties (stakeholders) interested in the development of a computer-based system.-有助于各方的交流

The architecture highlights early design decisions that will have a profound impact on all software engineering work that follows and, as important, on the ultimate success of the system as an operational entity.-突出早期的设计，对其后工作很重要（打好设计的根基）

Architecture “constructs a relatively small, intellectually graspable (understandable) model of how the system is structured and how its components work together”.-建立了一个相对小、易理解的模型，其描述了系统如何构成及构件如何一起工作

**2. What are the elements that make up a software architectural style?**

**描述一组构件，以及它们的连接关系，以及约束条件。**

Each style describes a system category that encompasses:

1. a set of components (e.g., a database, computational modules) that perform a function required by a system, 完成系统需要的某种功能的一组构件

(2) a set of connectors that enable “communication, coordination and cooperation” among components, 能够使构件间实现通信、协作的一组连接件

(3) constraints that define how components can be integrated to form the system, and

定义构建如何成为系统的约束

(4) semantic models 语义模型that enable a designer to understand the overall properties of a system by analyzing the known properties of its constituent parts.

**3. What is an archetype?**

An archetype原型 is an abstraction (similar to a class) that represents one element of system behavior, it’s core abstract classes or patterns of system)

原型（Archetype）是一个抽象（类似于类），它代表系统行为的一个元素，即系统的核心抽象类或模式。

**10.26homework**

**1.How does the object-oriented view of component-level design differ from the traditional view?**

面向对象的组件级设计强调的是将系统分解成具有封装、继承和多态特性的对象。每个对象都包含一组协作的类，组件内的每个类都已完全细化，包括所有与其实现相关的属性和操作，并且可以通过接口的消息传递与其他对象进行通信。这种设计方法提高了复用、灵活性和可维护性。

**2.Describe the differences between the software engineering terms coupling and cohesion?**

耦合：指的是不同模块或组件之间的相互依赖程度。高耦合意味着一个模块的更改可能会影响到其他模块，增加了系统的复杂性和维护难度。

内聚：是指单个模块内部各部分之间的相关程度。高内聚表示模块内的元素紧密相关，执行单一职责，这样的模块更容易理解、测试和维护。

好的设计应该是高内聚低耦合的，这样能提高系统的模块化程度，减少错误传播的风险。

**3.What are the advantages of component-based software engineering**

缩短前置时间。从现有的组件库中构建完整应用程序的速度更快。 更高的投资回报率（ROI）。有时通过购买组件而不是在内部重新开发相同的功能可以节省成本。 组件开发成本的杠杆效应。在多个应用程序中重用组件可以使成本分散到多个项目上。 提升质量。组件在许多不同的应用程序中被重用和测试。 基于组件的应用程序的维护。通过精心的工程设计，可以用新的或增强的组件相对容易地替换过时的组件。

**4.What are the steps used to complete the component-level design for a software development project?**

**标识、详细设计、部署类**

**标识类**（从需求模型和架构设计中，提取、标识类）

标示所有问题域相对应的设计类（主要是功能方面的）

标示所有基础设施域相对应的设计类（在分析模型中可能没有涉及到），主要是一些接口类（GUI,OS,DATABASE等）

**详细设计类**(类的协作、 接口、属性、操作)

设计数据库、文件管理和处理需要的类

设计（细化）类的行为表示（从行为模型中导出，如果有）

设计（细化）部署图(软硬件部署)

**检查、重构与部署**

**5.Is it reasonable to say that problem domain components should never exhibit external coupling? If you agree, what type of component would exhibit external coupling?**

从理论上讲，问题领域组件（Domain Component）确实应该尽量避免外部耦合，因为它们主要负责处理业务逻辑，应该保持较高的内聚性和较低的耦合度。但是，在实际应用中，某些情况下还是会出现一定程度的外部耦合，比如：

技术组件（Technical Component）：这类组件通常用于提供通用的技术支持，如数据库访问、网络通信等，可能会与外部系统或服务有较多的交互，因此存在外部耦合。

适配器组件（Adapter Component）：用于将不同系统或组件连接起来，实现不同接口之间的转换，自然就会有外部耦合。

**10.26homework**

**1. List three principles that should be applied when building any user interface.**

1. 把控制权交给用户

2. 减轻用户的记忆负担

3. 保持界面一致

**2.What framework activities are completed when following an evolutionary (or spiral) user interface development process?**

1. 界面分析和建模 （Interface analysis and modeling）

2. 界面设计 （Interface design）

3. 界面构建 （Interface construction）

4. 界面验证 （Interface validation）

**3.List four things you need to do in user experience analysis.**

1. 用户研究（user analysis）：理解通过该界面使用软件的用户

2. 任务研究（task analysis）：理解最终用户在工作中需要进行的任务

3. 内容研究（content analysis）：理解在界面上需要向用户展示的内容

4. 工作环境分析（environment analysis）：理解任务将要实施时所处的环境

**4.What elements of a user interface design can be evaluated before a prototype is built?**

布局合理性：检查界面布局是否符合逻辑，信息层次是否清晰。

导航结构：评估导航系统的有效性，确保用户能够轻松找到他们需要的信息或功能。

视觉设计：审查颜色搭配、字体选择等视觉元素是否和谐，是否有助于提升用户体验。

可访问性：考虑界面设计是否考虑到残障人士的需求，如提供足够的对比度、支持键盘导航等。

**5. List four interface design issues present in the development of most user interfaces.**

a. 自适应用户配置文件 b. 图形错误处理 c. 显示系统 d. 响应时间