# System Analysis and Design

## Homework Assignment 5 (lecture 12-16)

学号：19335286 姓名：郑有为

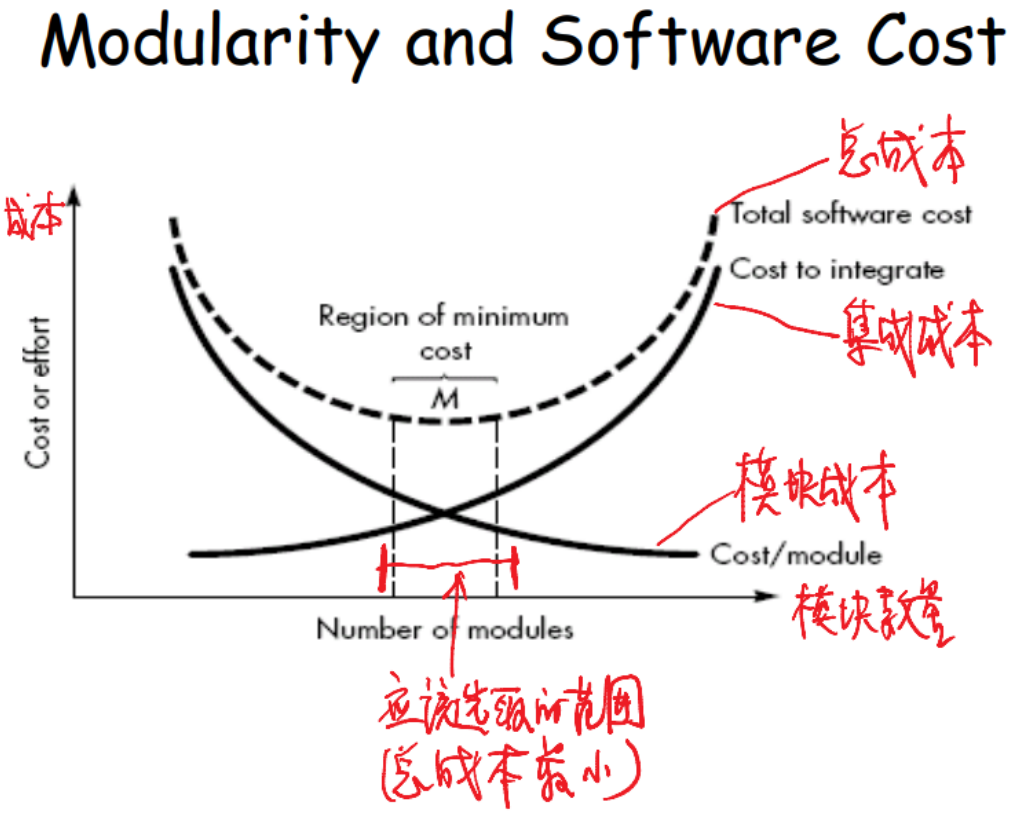
**1. Software is divided into separately named and addressable components called modules. What is the idea of Modularity in software design? How can we design good software modules?**

软件设计中的模块化思想是什么?

模块化思想是一种分治的思想，即为了解决软件的复杂性问题或者降低软件的复杂性，软件整体划分成块，这些块都相对独立，彼此之间用接口（协议）通信，每个块完成特定功能，多个块组合可以完成一系列功能。通过模块化使得整个软件可控，易于维护和扩展。

如何设计好软件模块?

从课件上的模块数量与成本的关系图可以看到，软件的模块数量越多，虽然模块成本会降低，当集成成本会越高，因此模块数量以及大小应该适中才能保证总成本达到最低。此外，在划分模块的时候要考虑业务逻辑并根据任务需求归类划分，同时还要兼顾模块之间的关系，确保模块独立性以及模块之间接口的简单统一。




**2. SOLID stands for 5 OOD principles. What are they? What ideas on OOD do they each reveal?**

S.O.L.I.D五项OOD原则：

1. S - Single-responsibility principle - 单一职责原则

   一个类应该只有一项工作，且满足模块化、高内聚、低耦合的设计要求。

2. O - Open-closed principle：开放封闭原则

   对象或实体应该对扩展开放，对修改关闭。

3. L - Liskov substitution principle - Liskov替换原则

   在继承关系中，父类定义的属性和方法若在子类中使用，则两者应该具有相同的语义解释；

   所有子类应该兼容父类，每个子类都应该可以替换它的父类。

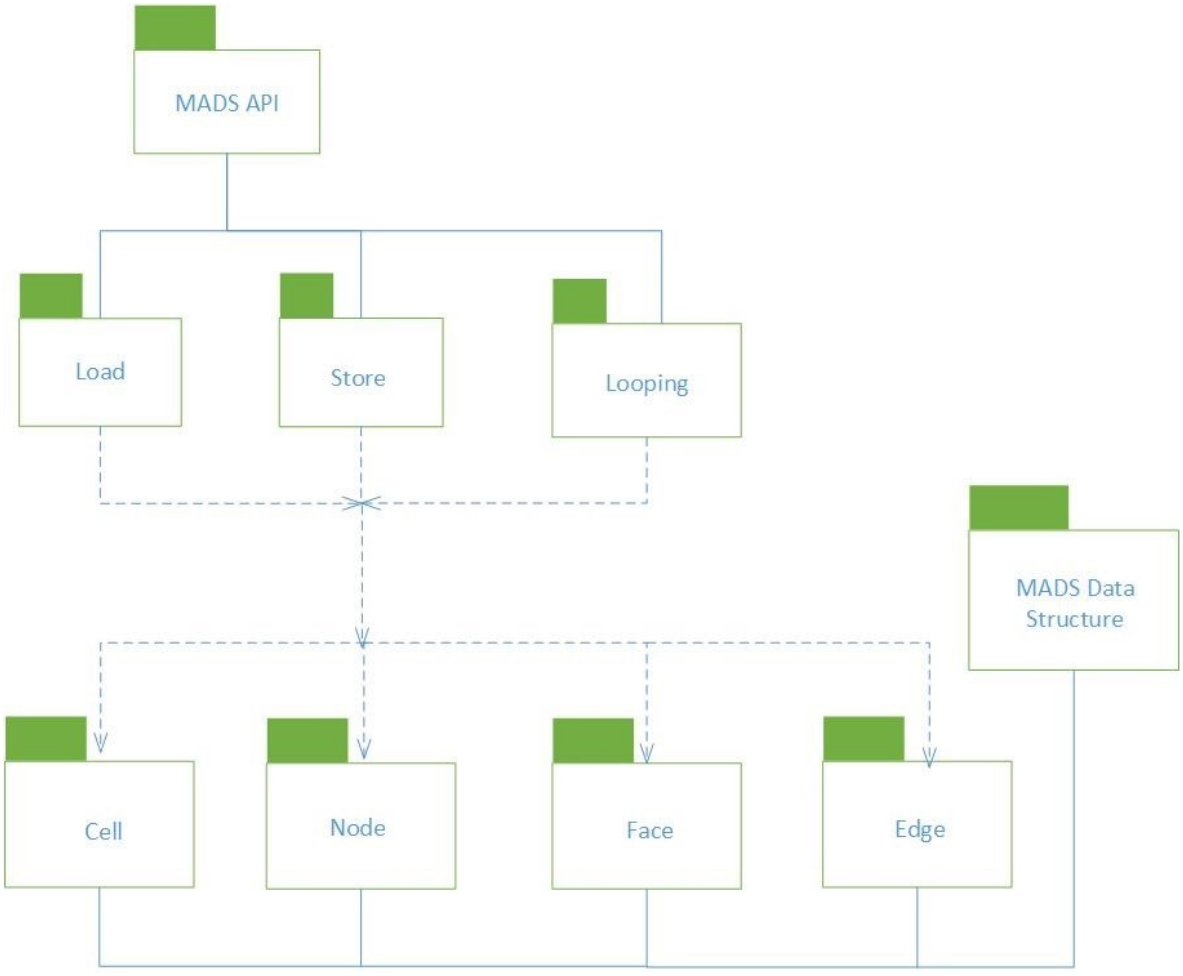
4. I - Interface segregation principle - 接口分离原则

   永远不应该强迫客户端实现它不使用的接口，也不应该强迫客户端依赖它们不使用的方法。即无用的依赖应该从接口中分离出去。

5. D - Dependency inversion principle - 依赖递置原则

   实体必须依赖于高层抽象而不是底层的具体实现。

**3. What kind of UML diagrams are suitable for architectural modeling? Please draw a UML diagram to show the architecture of your course project.**




**4. There are two kinds of object models in object design in an iterative and evolutionary development with agile method. What are they? What are the differences between the two?**

对象模型有两种类型：动态和静态。

二者的区别是：

- 动态模型有助于设计逻辑，代码行为或方法体，例如UML交互图（顺序图或通信图）。
- 动态模型倾向于创建更为有益、困难和重要的图形。

- 静态模型有助于设计包，类名，属性和方法特征标记（但不是方法体）的定义，例如UML类图。

静态和动态建模之间具有关系，敏捷建模对此的实践是并行创建模型：花费较短的时间创建交互图（动态），然后转到对应的类图（静态），交替进行。

**5. Why the object design skill is much more valuable than knowing UML notation?**

- 对象设计技巧是最重要的，从对象设计到绘制UML表示图是一个自内而外的过程。掌握设计技巧意味着你需要知道和熟练如何基于对象进行思考和设计，需要考虑如何将最佳的实践模式运用到对象设计中。而绘制UML图只是对象设计中做出的决策的一个反映。
- 在绘制UML对象图时，我们要回答以下关键问题：对象的职责是什么？对象在与谁协作？应该应用什么设计模式？回答这些问题远比了解UML表示法之间的差异重要。
- 了解对象设计技术并不一定要了解如何绘制UML，但是绘制UML需要先理解对象设计技术并回答其中的关键问题。

**6. What are the required knowledge for Fundamental Object Design?**

基本的对象设计需要了解的是：

- 职责分配原则
- 设计模式

**7. A guideline for dynamic object modeling states that “Spend time doing dynamic modeling with interaction diagrams, not just static object modeling with class diagrams”. What does this statement mean?**

这段陈述的意思是：

- 动态模型有助于设计逻辑、代码行为或方法体，例如UML交互图（顺序图或通信图）。
- 动态模型倾向于创建更为有益、困难和重要的图形。
- 静态模型有助于设计包、类名、属性和方法特征（但不是方法体）的定义，例如UML类图。
- 静态和动态建模之间具有关系，敏捷建模推荐并行创建模型：花费较短的时间创建交互图（动态），然后转到对应的类图（静态），交替进行。
- “花时间用交互图进行动态建模，而不仅仅是用类图进行静态对象建模”告诉了我们，不能只专注于静态建模，而忽略了动态建模的重要性。因为事实上，大部分具有挑战性、有益和有效的设计工作都会在绘制UML动态视图的交互图的时候发生，只有通过动态建模才能真正落实这些准确和详细的结论。

**8. Find possible software objects (possible classes you think important) for mesh applications and draw CRC cards for each object.**

CRC Card

| Mesh | **Collaborator** |
| --- | --- |
| **Responsibilities** | |
| - 载入VTK格式文件数据<br>- 储存数据到VTK格式文件<br>- 循环执行核函数 | - Mesh_API<br>- Cell<br>- Node<br>- Face<br>- Edge<br>- Face |

CRC Card

| Mesh_API | Collaborator |
| --- | --- |
| **Responsibilities** | |
| - 提供用户接口 | - Mesh |

CRC Card

| Node | Collaborator |
| --- | --- |
| **Responsibilities** | |
| - 数据修改<br>- 载入VTK结点数据 | - Mesh<br>- Cell<br>- Edge<br>- Face |

CRC Card

| Cell | Collaborator |
| --- | --- |
| **Responsibilities** | |
| - 载入Mesh单元数据<br>- 查看组成单元的各节点<br>- 单元数据修改 | - Mesh<br>- Node<br>- Edge |

CRC Card

| Edge | Collaborator |
| --- | --- |
| **Responsibilities** | |
| - 载入Mesh边数据<br>- 查看组成边的各节点<br>- 查看与边关联的各边<br>- 边数据修改 | - Mesh<br>- Node<br>- Cell |

CRC Card

| Face | Collaborator |
| --- | --- |
| **Responsibilities** | |
| - 载入Mesh面数据<br>- 查看组成面的各节点<br>- 面数据修改 | - Mesh<br>- Node |

**9. Explain (or summarize) in your own words on how to do Dynamic Object modeling.**

- 工具使用：绘制交互图（顺序图或者通信图）、状态机图或活动图。

  其中，顺序图是一种栅栏格式的描述交互的图，并在右侧添加新创建的对象；而通信图以图或者网格格式描述对象交互，对象可位于任何位置。它们各有优劣，前者对工具的支持更友好，阅读方便，后者更适合在墙上进行绘制、易于修改、有效利用空间。

  - 顺序图通常使用生命线框图表示参与者，对消息表达式应使用标准的语法，可以使用执行规格条来表示控制期等。顺序图在应答与返回、发送给自身的消息、实例的创建、对象的生命线和对象的销毁、循环图框、有条件图框、集合迭代图框、图框嵌套、关联交互图、调用静态方法、同步异步调用上都有详细的规定，遵循这些规则有利于做出规范的动态建模图表。
  - 通信图使用链来指明对象间的导航和可见性，也是用消息进行跨对象的传递、与顺序图不同的是，消息需要进行编号。

**10. Explain (or summarize) in your own words on how to do Static object modeling.**

- 工具使用：绘制类图、包图或部署图。其中类图，用来表示静态对象建模分析过程中的类、接口与关联。对于UML类图表示法，使用类图进行静态建模时须遵循以下规则：

  使用属性文本表示法或者关联线表示法表示UML属性，表示的内容包括：

  - 可见性（+表示公共，-表示私有）
  - 导航性箭头（源对象指向目标对象）
  - 角色名（表示属性名称），不需要标注关联名称。
  - 类关联：要注意何时使用属性文本、何时使用关联线，对数据类型对象使用属性文本表示法，对其它对象使用关联线。
  - 注解：使用规定的注解符号来表示注解、注释、约束和方法体。
  - 操作和方法：UML在类框的一个分栏里表示操作特征标记，包含以下准则：操作是声明，包含名称、参数、返回类型、异常列表、可能的前后置条件约束；可见性省略时表示操作是公共的，通常对构造析构操作加上书名号、对属性的访问操作通常省略。
  - 关键字：使用关键字对模型元素进行类别上的修饰，格式通常为书名号。
  - 特性串：用特性来表示元素特征的已命名的值，格式通常为大括号。
  - 类图还提供对泛化、抽象类、抽象操作、依赖、接口、聚合、关联类、模板类的特殊表示。
- 设计思想：从交互图中产生类图，二者交替进行。

**11. Revise the collection of the following artifacts you have done for your course project.**

- Glossary
- Supplementary Specification
- SSD
- System Operation Contracts
- Domain Model (Conceptual class diagrams)
- System Architecture
- Interaction diagram for dynamic object modeling
- Domain class diagram

Notice that some sections may be not complete in your document at this moment.