

- 1 Software is divided into separately named and addressable components called modules. What is the idea of Modularity in software design?

模块化设计，简单地说就是将产品的某些要素组合在一起，构成一个具有特定功能的子系统，将这个子系统作为通用性的模块与其他产品要素进行多种组合，构成新的系统，产生多种不同功能或相同功能、不同性能的系列产品。设计好的软件模块,需要做到高内聚、低耦合；力求以少量的模块组成尽可能多的产品，并在满足要求的基础上使产品精度高、性能稳定、结构简单、成本低廉，模块间的联系尽可能简单；系列化模块，用有限的产品品种和规格来最大限度又经济合理地满足用户的要求。

How can we design good software modules?

使用分治的思想，将大问题分为各种小问题，小问题构成大问题

- 2 SOLID stands for 5 OOD principles. What are they?

S-单一职责原则，O-开放封闭原则，L-Liskov 替换原则，I-接口分离原则，D-依赖逆置原则。

What ideas on OOD do they each reveal?

S-一个类只能有一种职责，模块化、高内聚、低耦合

O-对象或实体应打开以进行扩展，但关闭以进行修改，允许扩展类型，却不允许修改接口

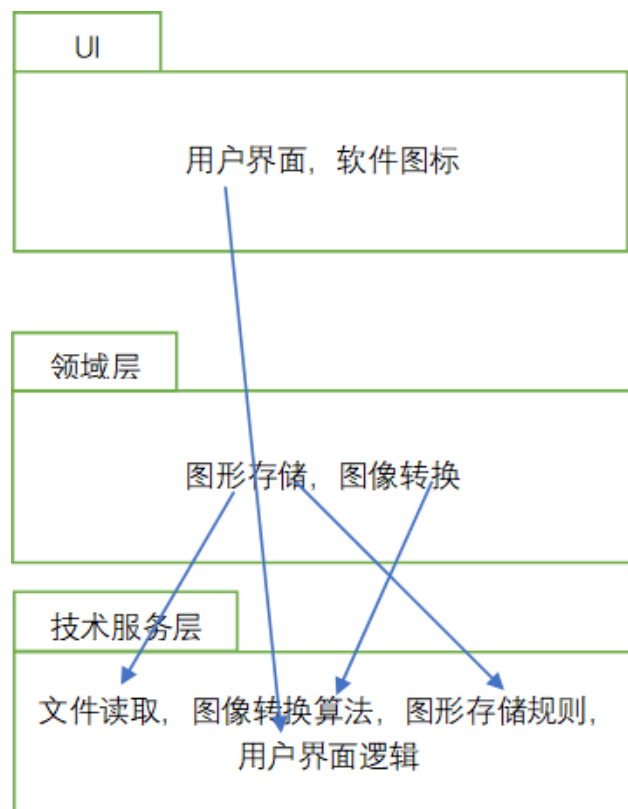
L-所有子类型应该兼容父类型，在继承关系中，父类型的方法若在子类中使用，应该具有相同的语义解释

I-不应强迫客户实现它不使用的接口，也不应强迫客户依赖它们不使用的方法，用户无用的方法应该从接口中分离出去

D-对象必须依赖于抽象，而不是具体实现。高级模块不能依赖于低级模块，而应该依赖于抽象。

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?

动态对象建模和静态对象建模

What are the differences between the two?

动态对象建模包括交互图，其中主要包括通信图和顺序图，主要是一系列对象行为的展示。

静态对象建模主要是类图，用例图等一系列不包括交互和行为的属性，而是对象的静态性质。

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

可以让人知道如何去构思与设计一个对象，，可以更好地将最佳实践方案应用于对象设计，绘制 UML 图只是设计决策的一个反映。

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

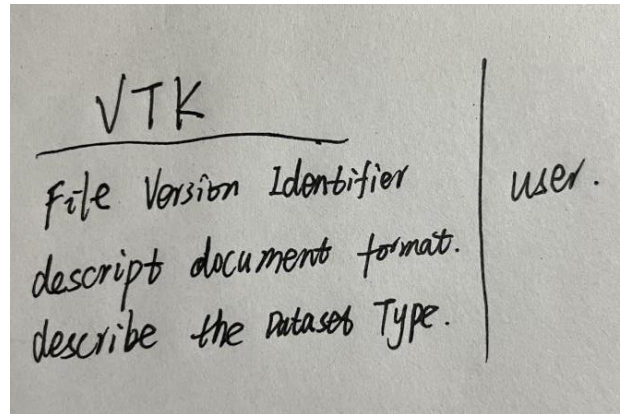
职责分配：CRC 卡是一种纸质索引卡，人们可以在上面写入类的责任和合作者。每张卡片代表一个类。每一个类设想各种场景探研对象地职责及其协作者。

设计模型：静态对象建模，动态对象建模。

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?

要花足够时间做动态建模，这样使你更深入地理解对象及其行为。

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



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

动态对象建模是面向对象建模中的一种方法，它有助于设计逻辑、代码行为和方法体，其中绘制 UML 动态视图交互图是最困难和最实用的工作，UML 包含交互图来说明对象如何通过消息进行交互，有两种常见的类型：顺序图和通信图，顺序图清晰地显示消息的顺序或时间顺序，使用的是水平的空间；通信图空间经济的灵活性，添加新的对象在两个维度，但是难以看到消息序列。顺序图相比于通信图又更多的符号标记，我们应该将大量时间用于绘制交互图而不仅仅是类图。

做动态对象建模的过程主要就是绘制顺序图和通信图的过程。

顺序图像栅栏的格式，在水平空间的右侧添加新创建的对象，在中间的下方描述继承的方法，通信图以图或网格格式描述对象的交互，对象可以置于图中的任何位置。

用生命线框图表示参与者，生命线向下垂直延伸，垂直的生命线之间有水平的箭头连接两个生命线，其上通过消息表达式表示对象之间传递的消息，带有实心箭头的虚线上的 create 消息通常表示调用 new 和其构造器。用虚线

+ 实心箭头表示同步消息，如果是异步则线条箭头。单实例类对象需要在对象的右上角注明，使用 return 的消息语法或在活动条末端使用应答/返回消息线表示应答或返回，用图框可以表示可选性、循环、并行、临界片段。在通信图中，链接是对象之间的连接路径，多条消息或两种方式的消息可以在同一链接线上流动，可以用序列号给消息排序，用顺序号+字母表示选择。。

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

静态建模是指对象之间通过属性互相联系，而这些关系不随着时间而转移。UML 的静态建模机制包括用例图、类图、对象图、包图、构件图、配置图。其中的对象类图与对象图表达了对象模型的静态结构，能够有效地建立专业领域的计算机系统对象模型。这里的类图由系统中使用的类以及他们之间的关系组成，分为长式和短式。同时类图也是构建其他图的基础。而对象是对象类的实例，用对象图来描述，亦分为长式和短式。而建模的关键即是类的识别，常用的方法有名词识别法、系统实体识别法、从用例中识别类、利用分解与抽象技术。在确定类时，常用的两种技术是分解技术和抽象技术。静态建模的关系不随时间转移，这些关系就包括关联、聚集、依赖和细化。那么进行静态对象建模，就是一个不断反复、不断细化的过程。过程中，应该完成若干张类图和对象图。而整个结构，应当包括构成系统的类、对象、它们的属性和操作，以及这些对象类之间的关系。

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

- Glossary

glossary 术语表			
	definition 定义	format 格式	aliases 别名
mesh application development system	网格应用程序开发系统		
mesh structures 网格结构	网格结构被认为是一种静态结构，是能够表示一个几何对象的离散模型。常用于将物体渲染到计算机上进行物理模拟。例如有限元分析或计算流体力学。网格可以用2d或3d来表示几何对象。		
nodes 节点	网格点		
cells 单元	域被分割成的控制体		
face 面	单元的边界		
edges 边	面的边界		
zone 域	节点、面和单元的集合		
domain 领域	节点、面和单位域的集合		
Structural mesh 结构网格	可以表示为一个数组，其中单元的关系可以通过他们的相对位置来存储解释，或者说用坐标轴的坐标来表示。		
Unstructured mesh 非结构网格	可以表示为一个图，通常是节点，边，面，单元格的集合，其中的单元以任意方式排列，单元和节点之间的连接性都必须显式存储。		
VTK (格式文件)	Vtk 是一个开源的免费软件系统，主要用于三维计算机图形学、图像处理和可视化。Vtk是在面向对象原理的基础上设计和实现的。		visualization toolkit
Gmsh	Gmsh是带有内置前后处理机制的三维有限元网格生成器，提供简单的耦合工具用以处理学术问题，具有参数输入和先进的可视化能力。其围绕四个单元：几何，网格，求解和后处理。		
CGNS	一种数据格式。使用包括两方面：从可以正确调用其数据库的语句接口或从源码编译CGNS的库文件。		CFD General Notation System
h5	H5文件是层次数据格式，用于存储科学数据的一种文件格式和库文件。它将文件结构简化成两个主要的对象类型： 1 数据集dataset,就是同一类型数据的多维数组 2 组group,是一种容器结构,可以包含数据集和其他组,若一个文件中存放了不同种类的数据集,这些数据集的管理就用到了group		Hierarchical Data Format, HDF5
set 集合	是mesh节点的数据结构，具有索引、名称和大小等属性，也有一些支持MPI的属性。		
map 映射	是两个节点集之间映射关系的数据结构。		
dat数据集	集合中节点属性的数据结构，包括维度、数据集大小、存储位置(在主机或设备上)、数据类型、缓冲区和标志位(驻点)为MPI。		
arg参数	是带有数据集(Dat)和节点连接(Map)的块的数据结构，此外还有访问选项(例如:只读)，以及MPI支持的一些属性。		
plan计划	是网格设计计划数据结构，它有一些基本的网络参数(类似于Arg)，但另外还有一些详细的执行计划，包括阶段、颜色设置、间接设置之间的连接、块(Arg)映射、MPI支持，以及内核调用的设置。		
MPI 信息传递接口	MPI是一个跨语言的通讯协议，用于编写并行计算机。支持点对点 and 广播。MPI是一个信息传递应用程序接口，包括协议和语义说明，他们指明其如何在各种实现中发挥其特性。MPI的目标是高性能、大规模性、和可移植性。		
Finite element calculation 有限元计算	有限元计算的任务是基于有限元模型完成有关的数值计算，并输出需要的计算结果。它的主要工作包括单元和总体矩阵的形成、边界条件的处理和特性方程的求解。		

- Supplementary Specification

1. Objectives

The purpose of this document is to define requirements of the Mesh Application. This Supplementary Specification lists the requirements that are not readily captured in the use cases of the use-case model. The

Supplementary Specifications and the use-case model together capture a complete set of requirements on the application.

2. Scope

This Supplementary Specification applies to the Mesh Application. This specification defines the non-functional requirements of the application; such as reliability, usability, performance, and supportability as well as functional requirements that are common across a number of use cases.

3. References

None.

4. Functionality

The mesh file should be in the form in VTK.

5. Usability

The user's computer should be able to run basic c++ code.

6. Reliability

None.

7. Performance

None.

8. Supportability

None.

9. Security

None.

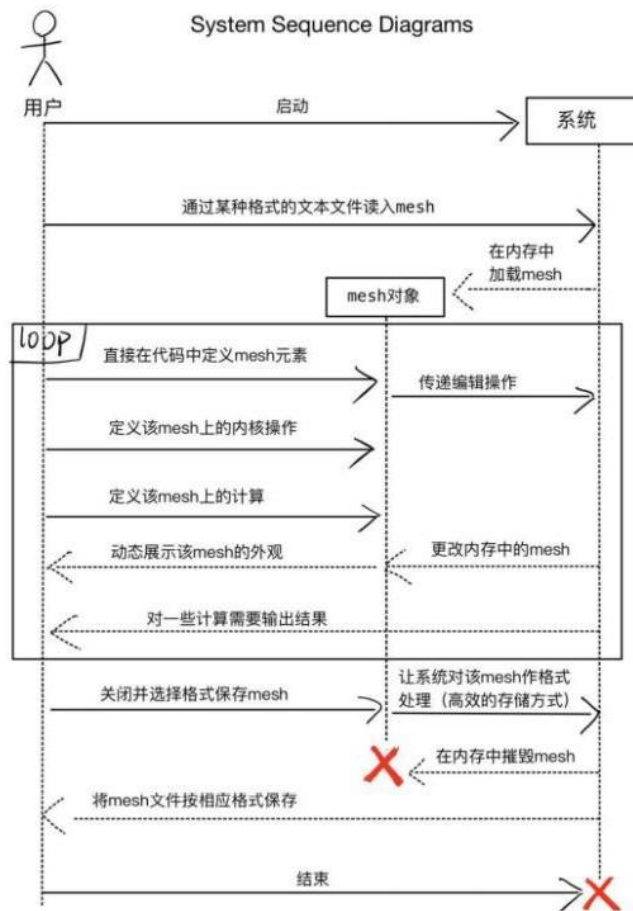
10. Design Constraints

The application should provide a simple GUI in windows system.

11. Others

The language is English.

- SSD



- System Operation Contracts

Contract CO1:openMesh

Operation:openMesh();

Cross References:Use Case:Edit Mesh

Precondition:none

Postcondition:

1、 The Mesh file s is open. 2、

The file is initialized.

Contract CO2:createMesh

Operation:createMesh();

Cross References:Use Case:Edit Mesh

Precondition:none

Postcondition:

1、 A Mesh file s is create. 2、

The file is initialized.

Contract CO3:editMesh

Operation:editMesh();

Cross References:Use Case:Edit Mesh

Precondition:createMesh/openMesh

Postcondition:

1 、 main operation to change the mesh.

Contract CO4:saveMesh Operation:saveMesh();

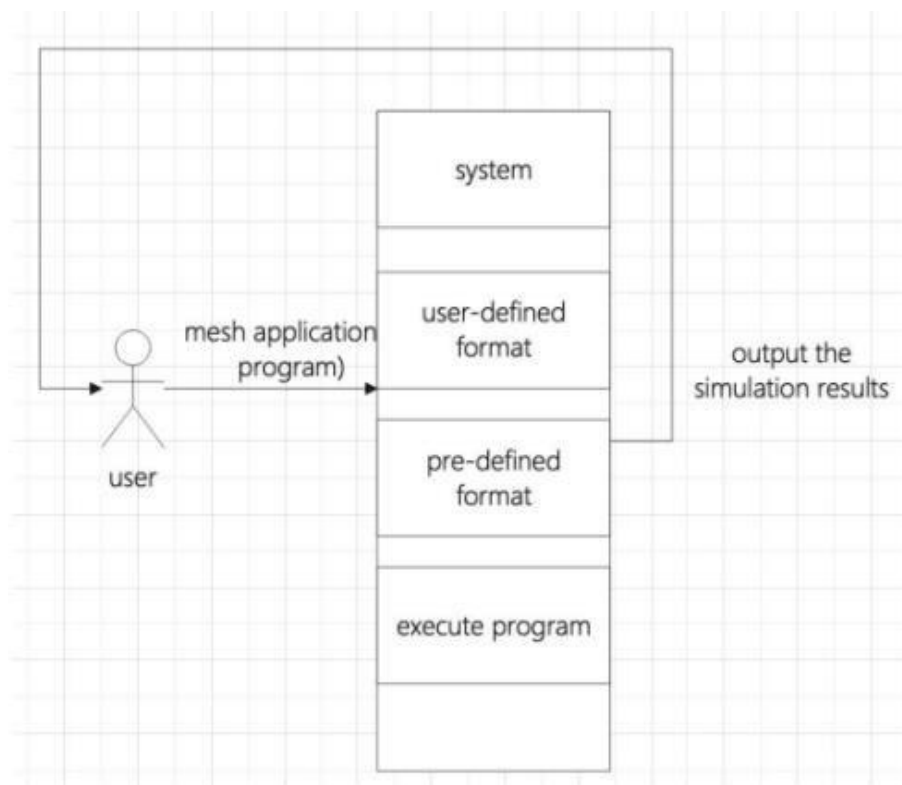
Cross References:Use Case:Edit Mesh

Precondition: createMesh/openMesh

Postcondition:

1、 save the mesh file in the spercific way.

- Domain Model (Conceptual class diagrams)



- System Architecture 无
- Interaction diagram for dynamic object modeling 无
- Domain class diagram 无