实验九 UML，逻辑， 软件体系结构设计（一）

实验目的：

1. 深入理解UML

2. 了解计算机学科中的逻辑

3. 学习对比软件体系结构设计GB和IEEE最新SAD (Software Architecture Document)的标准

4. 研究经典软件体系结构案例

3. 完成自己项目的SRS

实验内容：

1. 阅读“The Unified Modeling Language Reference Manual”，进一步学习UML知识，理解如何应用UML对系统进行建模

《The Unified Modeling Language Reference Manual》是由James Rumbaugh、Ivar Jacobson和Grady Booch三位著名计算机科学家共同撰写的一本关于统一建模语言（UML）的权威指南。这本书为软件开发者、架构师、项目经理、系统工程师、程序员、分析师、合同官员、客户以及任何需要对复杂软件系统进行规范、设计、构建或理解的专业人士提供了全面且实用的参考资料。

**UML概述**

UML是一种通用的可视化建模语言，用于规范、可视化、构建和记录软件系统的工件。它捕获了必须构建的系统的决策和理解，用于理解、设计、浏览、配置、维护和控制这些系统的信息。UML适用于所有开发方法、生命周期阶段、应用领域和媒介。

**UML的目标**

为开发者提供一个通用的建模语言。

支持现有的面向对象开发过程。

统一过去关于建模技术的经验，并将当前的软件最佳实践整合到一个标准方法中。

**UML的概念区域**

静态结构：定义了应用领域的关键概念、它们的内部属性以及它们之间的关系。

动态行为：通过状态图和活动图来描述对象随时间变化的行为。

实施结构：包括组件和部署视图，用于描述系统的物理实现。

模型组织：通过包的概念来组织和管理系统模型。

**UML视图**

静态视图：关注系统的结构化分类，包括类图、用例图等。

动态视图：关注系统随时间的行为，包括状态图、活动图、序列图等。

实施视图：关注系统的物理实现，如组件图和部署图。

模型管理视图：关注模型本身的组织，如包和子系统。

**UML的扩展机制**

UML提供了有限的扩展能力，包括约束、标记值和刻板印象（stereotypes），以适应日常需要的扩展，而无需对基本语言进行更改。

**UML环境**

UML模型在环境中使用，其目的和解释受到环境其他部分的影响。UML是通用的建模语言，包括语义和表示法，但可以与不同的工具和实现语言一起使用。

**学习UML的应用**

要应用UML对系统进行建模，可以遵循以下步骤：

理解需求：与利益相关者沟通，明确系统的需求和目标。

创建用例图：识别系统的功能需求，并用用例图来表示。

构建类图：定义系统中的类、它们的属性、方法和关系。

设计动态行为：使用状态图和活动图来描述对象的状态变化和活动流程。

实现视图：设计组件图和部署图，展示系统的物理结构和运行时部署。

模型管理：使用包来组织和管理模型，确保模型的可维护性和可扩展性。

应用扩展机制：根据需要使用UML的扩展机制来定制模型，以适应特定的应用领域。

2. 浏览“LOGIC IN COMPUTER SCIENCE--Modelling and Reasoning about Systems”，了解常用逻辑及其在计算机学科中的应用

《逻辑在计算机科学中的应用——建模与推理系统》是一本由Michael Huth和Mark Ryan合著的教科书，深入探讨了逻辑在计算机系统设计、规范和验证中的应用。这本书涵盖了多种逻辑系统，包括命题逻辑、一阶逻辑、时序逻辑、模态逻辑等，并讨论了它们的语义、证明理论和应用。

1. 命题逻辑（Propositional Logic）：

基础概念，如命题、合取（∧）、析取（∨）、否定（¬）、蕴含（→）等。

自然演绎（Natural Deduction）证明方法。

命题逻辑的语义，包括真值表和模型。

命题逻辑的完备性和可判定性。

1. 一阶逻辑（Predicate Logic）：

引入量词（如全称量词∀和存在量词∃）来表达更丰富的语言。

一阶逻辑的证明理论和模型理论。

一阶逻辑的不可判定性，以及其表达能力。

1. 模型检查（Model Checking）：

线性时序逻辑（Linear-time Temporal Logic, LTL）和计算树逻辑（Computation Tree Logic, CTL）。

模型检查的算法，包括系统建模、属性编码和模型检查器的运行。

1. 程序验证（Program Verification）：

软件验证框架，包括核心编程语言和Hoare三元组。

部分正确性和完全正确性的证明演算。

1. 模态逻辑（Modal Logics）和代理（Agents）：

不同模式的真实性，基本模态逻辑。

逻辑工程，包括有效公式的库存和可及性关系的重要性质。

1. 二进制决策图（Binary Decision Diagrams, BDDs）：

布尔函数的表示，包括命题公式和真值表。

减少的有序BDDs（Reduced Ordered BDDs）的算法。

1. 其他主题：

包括软件微模型、程序设计中的契约式编程等。

这本书不仅提供了逻辑理论的深入讨论，还涉及了它们在实际系统建模和验证中的应用。书中包含了大量的实例、练习和参考文献，适合作为计算机科学专业学生的教材，也适合研究人员和专业人士参考。通过学习这些逻辑系统，读者可以更好地理解计算机科学中的规范语言、证明助手和模型检查器等工具的工作原理。

1. 分工协作，参考国标“13 - 软件(结构)设计说明(SDD)”等资料，对比参考SAD最新标准IEEE-42010.pdf，针对自己的项目设计SAD初稿。

4. 分工协作，学习、检索研究经典软件体系结构案例。

On-the-Criteria-To-Be-Used-in-Decomposing-Systems-into-Modules.pdf

http://www.cs.cmu.edu/~ModProb/index.html

5. 完成软件需求规格说明SRS

**下周五（含）前将软件需求规格说明提交给相应的助教**

项目跟踪，建立能反映项目及小组每个人工作的进度、里程碑、工作量的跟踪图或表，将其保存到每个小组选定的协作开发平台上，每周更新。