

计算机科学系实验报告

课程名称	<u>UML 与可视化建模</u>	班级	<u>14 网络 1 班</u>		
实验名称	<u>选课系统</u>	指导教师	<u>曾少宁</u>		
姓名	<u>方思特</u>	学号	<u>1414080903135</u>	日期	<u>2017.4.14</u>

一、实验目的

掌握基于 UML 2.0 的建模概念与方法，掌握各种 UML 图的概念与画法，其中包括用例图、活动图、类图、顺序图、组件图和状态图等。

二、实验设备与环境

操作系统：Windows 7；建模工具：StarUML。

四、实验要求

1. 实验及实验报告以增量方式完成，每次作业都在上一次作业的基础上完成，作业提交网站不提供报告下载，所以请同学们自行保管好自己的实验报告；
2. 请将实验报告中“占位符”信息替换为自己的实验相关信息；
3. 请认真撰写实验体会，**实验课结束时**立即上传实验报告：<http://www.zsn.cc/uml/>。

四、实验内容、程序清单及运行结果

选课系统

主要针对两类人群，分别是管理员与学生。

- 1、管理员——选课系统需提供课程管理，包含添加课程与修改课程。
- 2、学生——选课系统需提供选课服务。

1、实验一：需求建模 - 用例模型

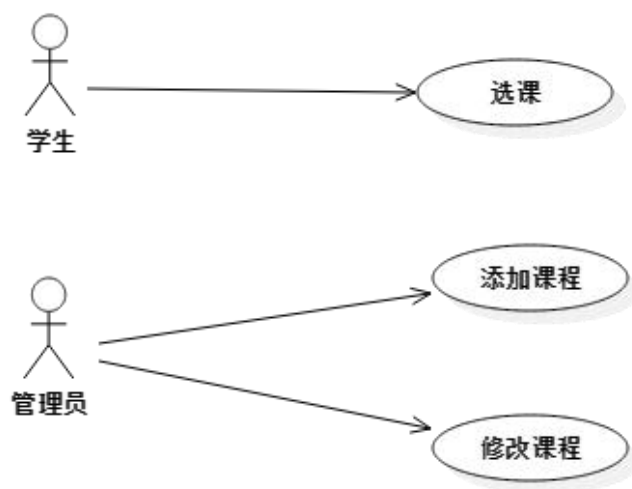


图 1：选课系统用例图

注：用例规约内容及项目可自行增加。

用例编号：	UC001
用例名称：	添加课程
用例描述：	管理员通过系统界面进入，添加所要开设的课程，确认无误后将其信息保存到数据库中，以供学生选择。
前置条件：	管理员成功登录至管理界面
基本流程：	<ol style="list-style-type: none"> 1. 进入管理界面，系统显示目前所建立的全部课程信息系统。 2. 管理员选择添加课程。 3. 系统提示输入新课程信息。 4. 管理员输入信息。 5. 系统验证是否和已有的课程冲突。 6. 系统添加新课程，提示课程添加成功。 7. 系统重新进入管理界面，显示所有课程。
扩展流程：	<ol style="list-style-type: none"> 5.1 课程已经存在，有冲突，则系统重新进入输入界面并提示课程有冲突 5.2 用户重新输入，验证无误后进入第 6 步
后置条件：	添加课程成功后，选课系统保存该课程的信息。

用例编号：	UC002
用例名称：	修改课程
用例描述：	管理员可以通过系统来修改已存在的课程的课程信息
前置条件：	管理员成功登录至管理界面
基本流程：	<ol style="list-style-type: none"> 1. 进入修改主界面，系统显示目前所建立的全部课程信息。 2. 管理员选择要修改的课程。 3. 是否确定修改该课程。 4. 管理员输入新信息。 5. 验证是否有课程冲突。

	6. 系统修改课程，提示修改成功。 7. 系统重新进入主界面，显示所有课程。
扩展流程:	3.1 系统提示重新选择。 3.2 用户确认后进入第 4 步 5.1 系统提示冲突，显示冲突的课程信息 5.2 用户重新输入
后置条件:	无

用例编号:	UC003
用例名称:	选课
用例描述:	学生进入选课系统界面，浏览的课程，最后选择一门自己喜欢的课程并提交。
前置条件:	学生成功登录了选课系统
基本流程:	1. 进入选课主界面。 2. 学生点击选课。 3. 系统显示所有课程信息。 4. 学生选择课程。 5. 系统验证课程是否可选。 6. 系统提示课程选择成功。 7. 用例结束。
扩展流程:	5.1 系统提示课程不可选及原因 5.2 学生重新选课 5.3 验证成功后进入第 6 步
后置条件:	学生选课并成功提交后，系统中新增学生的选课记录。

2、实验二：过程建模 – 活动模型

使用活动图描述系统的业务过程。

方法：将用例规约中的基本流程与扩展流程抽象为过程步骤（Action），画出对应的活动图。

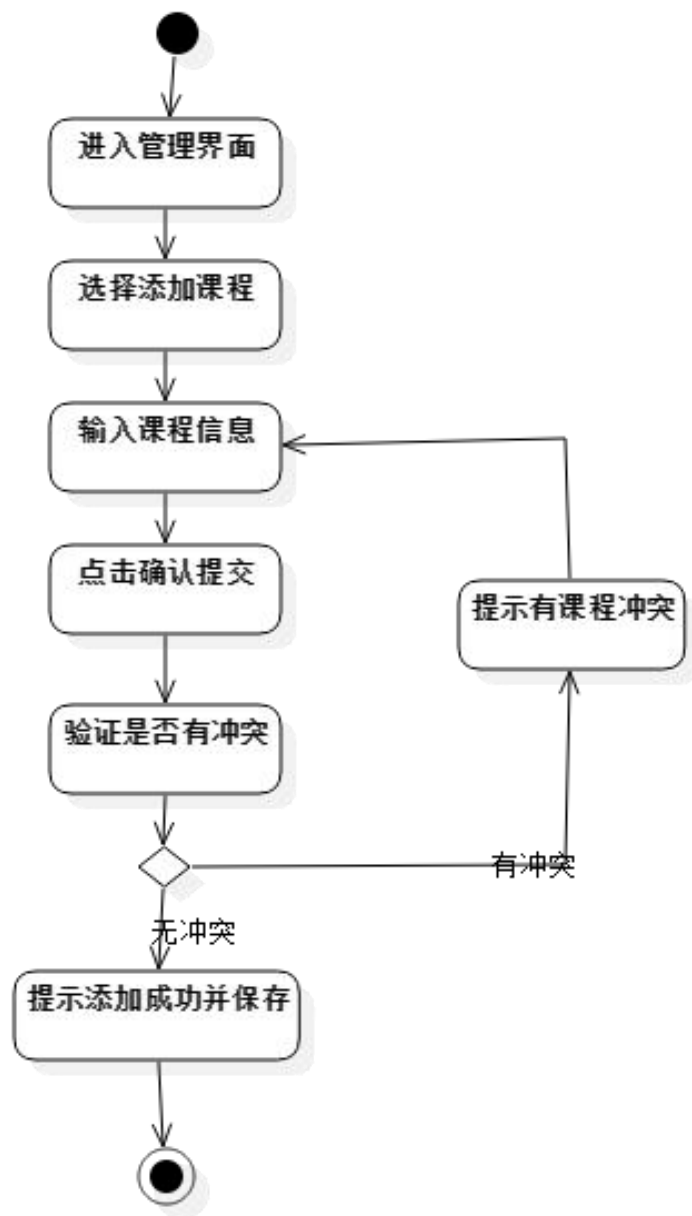


图 2: 添加课程活动图

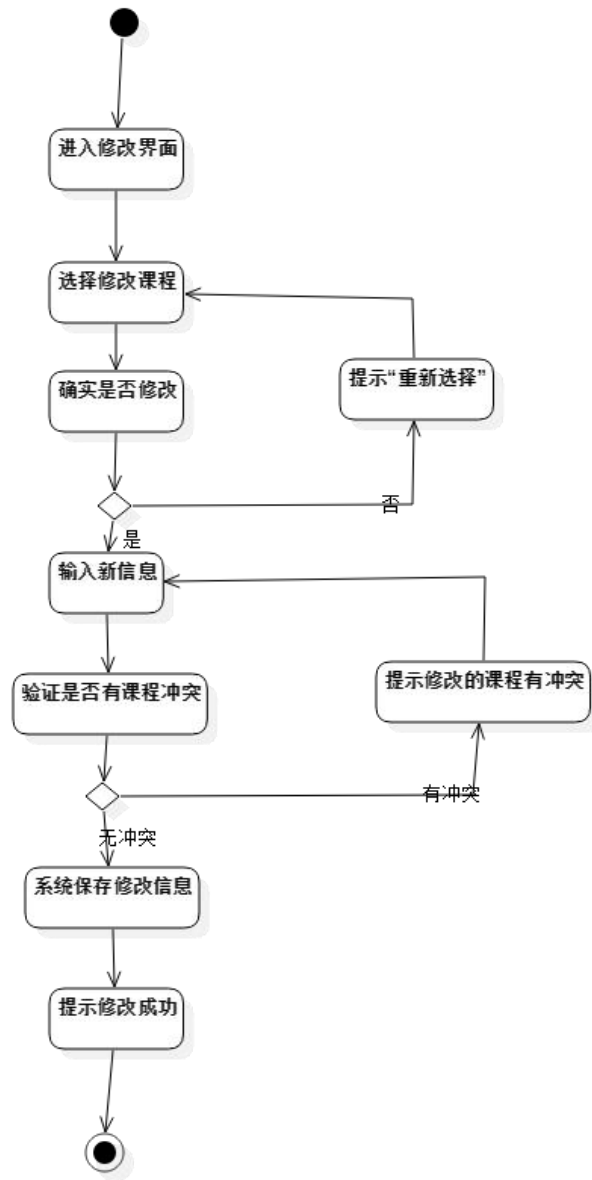


图 3 修改课程活动图

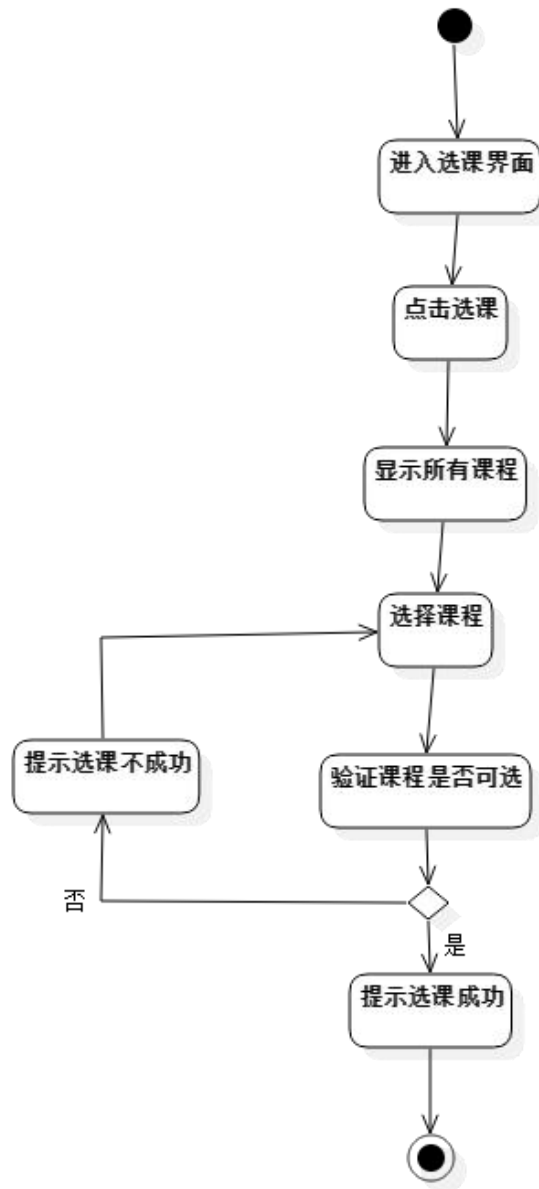


图 4：选课活动图

实验三：逻辑建模 – 类模型

基于 MVC 设计模式找出实现用例的类。

方法：分别找出实现用例的模型（Model）、视图（View）和控制器（Controller）类，确定类之间的关系及其关键属性，画出类图。

参考：讲义 P26 页。

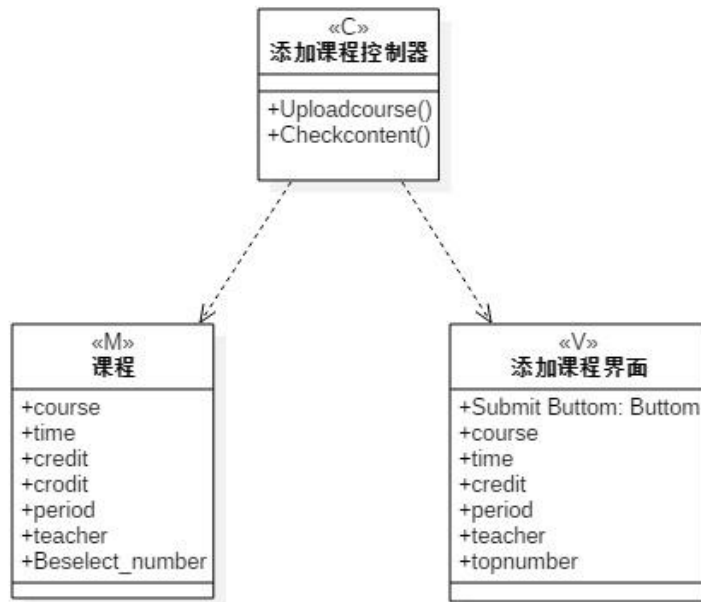


图 5: 添加课程类图

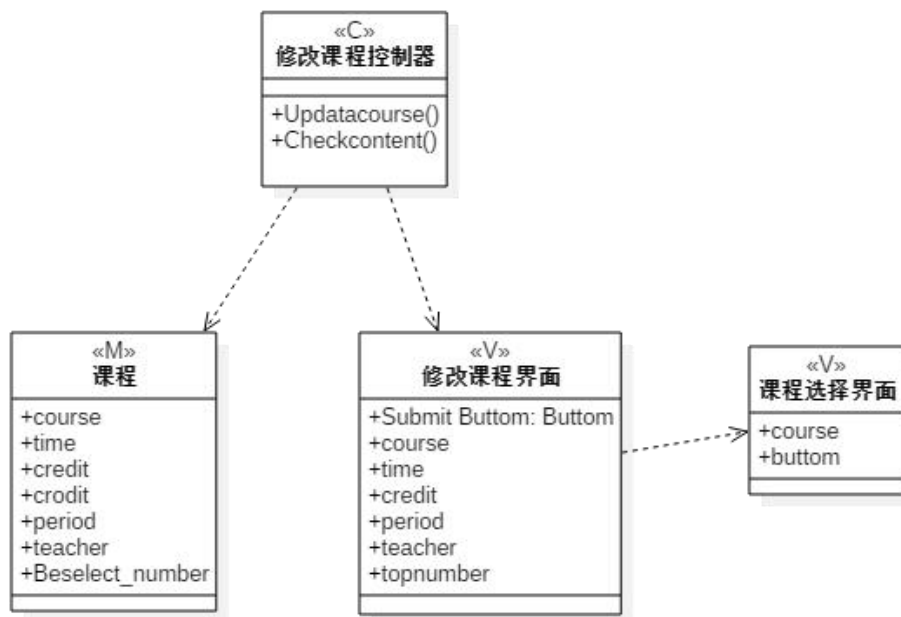


图 6 修改课程类图

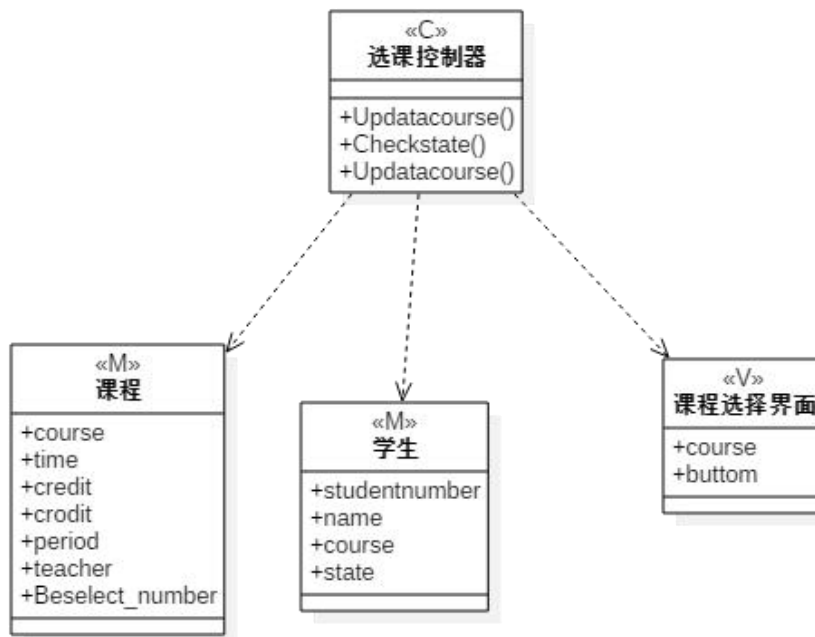


图 7：选课类图

3、实验四：交互建模 – 顺序模型

创建各个类（MVC 及 Actor）的对象，并描述对象之间的交互。

方法：分别创建参与者（Actor）、界面类（View）、控制器类（Controller）和模型类（Model）的对象，描述各个对象之间的消息及其顺序，画出顺序图。

参考：讲义 P33 页 8.7.2。

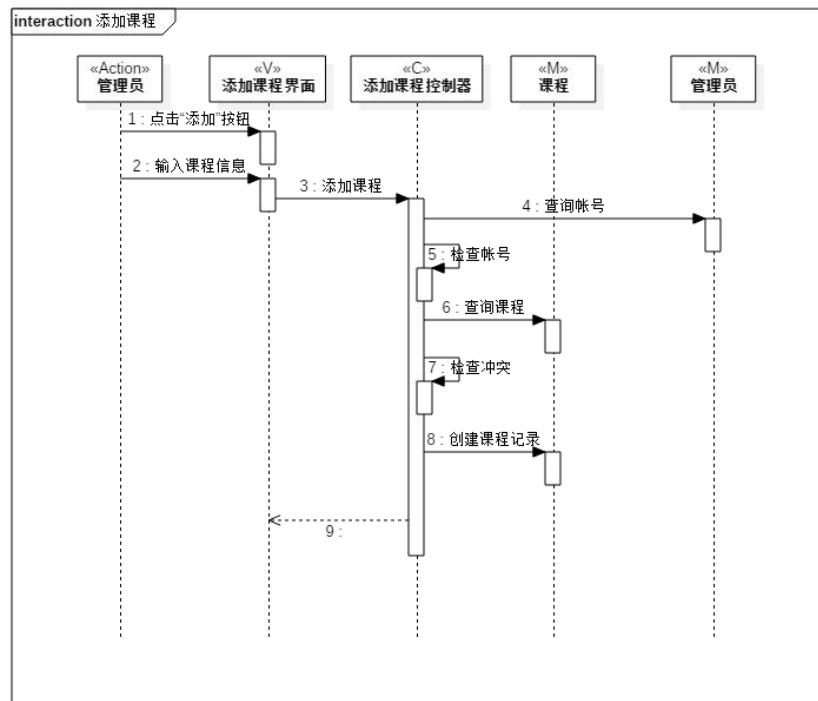


图 8 添加课程——顺序图

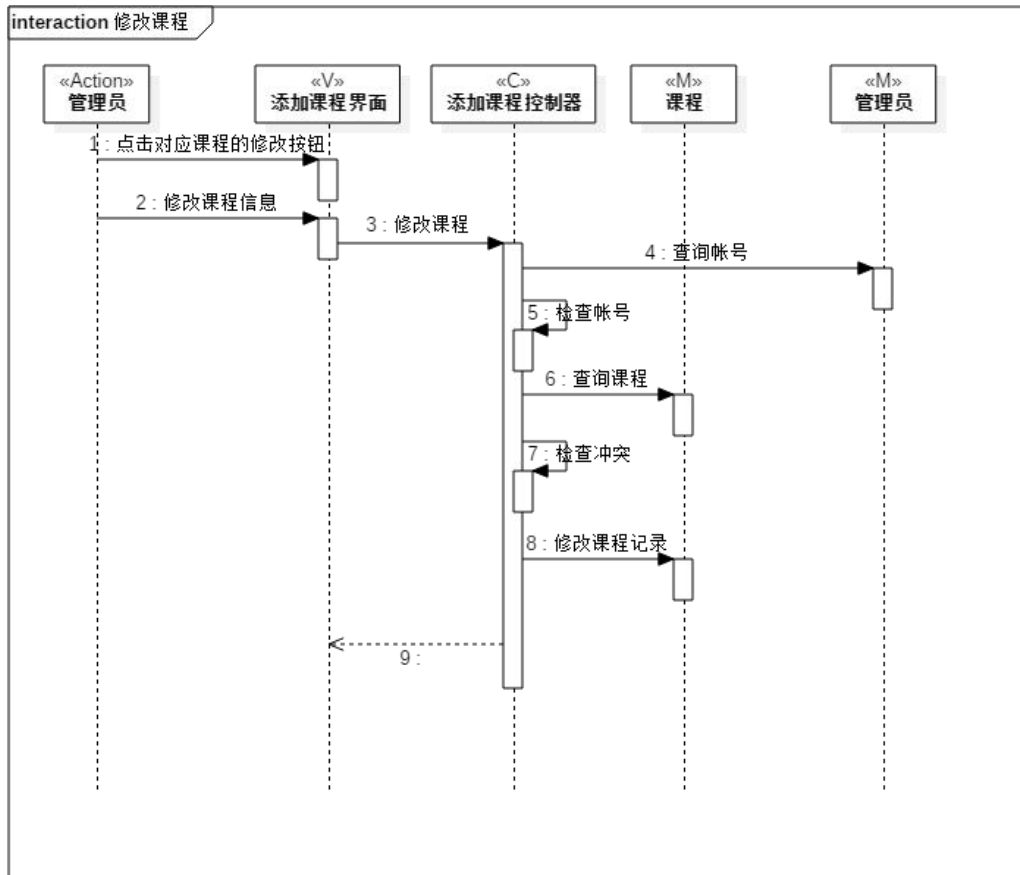


图 9 修改课程——顺序图

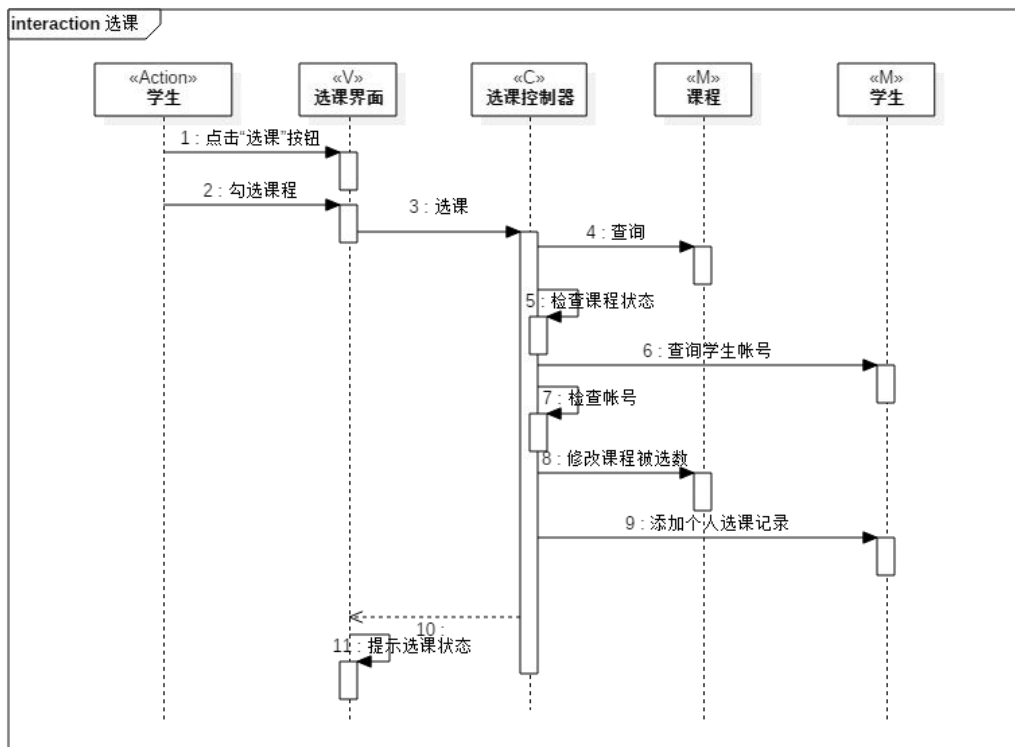


图 10 选课——顺序图

4、实验五：状态建模 – 状态模型

对系统中最重要的对象进行状态建模。

方法：选择一种对象，定义该对象的状态，描述状态之间的切换及条件，画出状态图。

参考：讲义 P9 和 P10 页。

五、实验体会

实验一：

本次实验完成对所选题目的需求建模，通过 StarUML 工具绘制用例图，并给出用例规约。在绘制用例图的过程中，因为是首次使用 StarUML，使用起来不太顺畅。通过网上搜索，最后还是顺利地绘制出简单的用例图。希望在接下来的学习中能更好地掌握 StarUML 的使用技巧以及 UML 建模的相关知识。

➤ 概念解释：用例

用例图主要用来描述“用户、需求、系统功能单元”之间的关系。它展示一个外部用户能够观察到的系统功能模型图。用例图多用于静态建模阶段（主要是业务建模和需求建模），帮助开发团队以一种可视化的方式理解系统的功能需求。

实验二：

本次实验主要完成活动图的绘制，以描述系统业务流程。实验通过活动图的绘制，将用例规约中的基本流程与扩展流程抽象为过程步骤，以活动图的形式呈现。

➤ 概念解释：活动图

活动图是 UML 用于对系统的动态行为建模的另一种常用工具，它描述活动的顺序，展现从一个活动到另一个活动的控制流。活动图在本质上是一种流程图。活动图着重表现从一个活动到另一个活动的控制流，是内部处理驱动的流程。

实验三：

本次实验基于 MVC 设计模式绘制了类图。通过找出用例的模型（Model）、视图（View）和控制器（Controller）类，确定类之间的关系及其关键属性，画出类图。

本次实验难点在于各个类之间关系的理解，大体分为依赖关系、关联关系、聚合关系、组合关系、继承关系五种。类图用来定义系统中的类，包括描述类的结构和类之间的关系，主要作用是描述系统的静态结构。本次实验让我对类的概念有了进一步的理解，初步掌握在 UML 建模语言中如何表示类之间的关系。

➤ 概念解释：类图

类图展示了系统的逻辑结构，类和接口的关系。类图表示不同的实体(人、事物和数据)如何彼此相关，显示了系统的静态结构。类图可用于表示逻辑类，逻辑类通常就是业务人员所谈及的事物种类，类图还可用于表示实现类，实现类就是程序员要编写的类。

实验四：

在本次实验中，创建了参与者（Actor）、界面类（View）、控制器类（Controller）和模型类（Model）对象，并描述各个对象之间的消息及其顺序，给出对象之间的顺序图。

➤ 概念解释：顺序图

顺序图用于显示对象是如何在一个给定的情境下进行交互的。顺序图的一个重要特征是时间从顶部到底端：交互从图例顶端开始，终止于底部，也就是说，越低越晚。顺序图的一个典型用法是用来演示面向对象的系统中的动态演化。对于每个关键的协作而言，就会为之建立一些顺序图来演示该协作各个场景中对象是如何进行交互的。

实验五: