

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 2 班</u>
实验名称	<u>睡眠小伙伴</u>	教导教师	<u>曾少宁</u>
姓名	<u>方锦祥</u>	学号	<u>1414080903231</u>
		日期	<u>2017.5.19</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://zeng.shaoning.net/uml/>。

四、实验内容、程序清单及运行结果

题目：睡眠小伙伴

主要功能：1.播放睡眠歌曲

2.设置定时闹钟

1、实验一：需求建模 - 用例模型

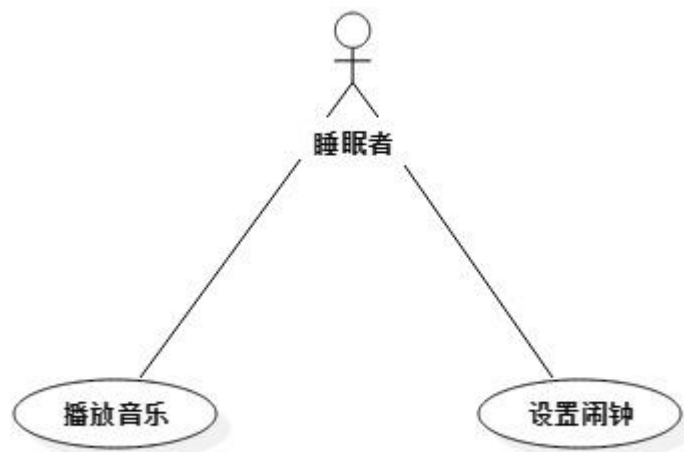


图 1：睡眠小伙伴用例图

注：用例规约内容及项目可自行增加。

用例编号:	UC001
用例名称:	播放音乐
用例描述:	
前置条件:	
基本流程:	<ol style="list-style-type: none"> 1. 睡眠者在睡眠曲界面选择一首本软件提供的睡眠曲; 2. 睡眠曲控制器检查选择睡眠曲是否成功; 3. 睡眠曲控制器显示睡眠曲的参数设定界面; 4. 睡眠者设置睡眠曲播放时间, 音效, 次数。点击“确定”按钮; 5. 睡眠曲控制器将设置的参数信息保存到文件中; 6. 根据参数信息, 睡眠曲控制器自动播放停止睡眠曲。
扩展流程:	<ol style="list-style-type: none"> 2.1 若该睡眠曲文件存在, 提示“文件选择成功”; 2.2 若该睡眠曲文件不存在或损坏, 提示“文件不能打开, 请重新选择”;
后置条件:	

用例编号:	UC002
用例名称:	设置闹钟
用例描述:	
前置条件:	
基本流程:	<ol style="list-style-type: none"> 1. 睡眠者在闹钟界面新建一个闹钟; 2. 闹钟控制器显示闹钟的参数设定界面; 3. 睡眠者设置闹钟的时间, 音乐, 重复次数, 点击“确定”按钮; 4. 闹钟控制器把设置的参数信息保存到参数文件中;
扩展流程:	<ol style="list-style-type: none"> 4.1 若存储空间不足, 参数文件不能保存, 提示“存储空间不足”; 4.2 若参数文件保存成功, 提示“闹钟创建成功”;
后置条件:	

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。

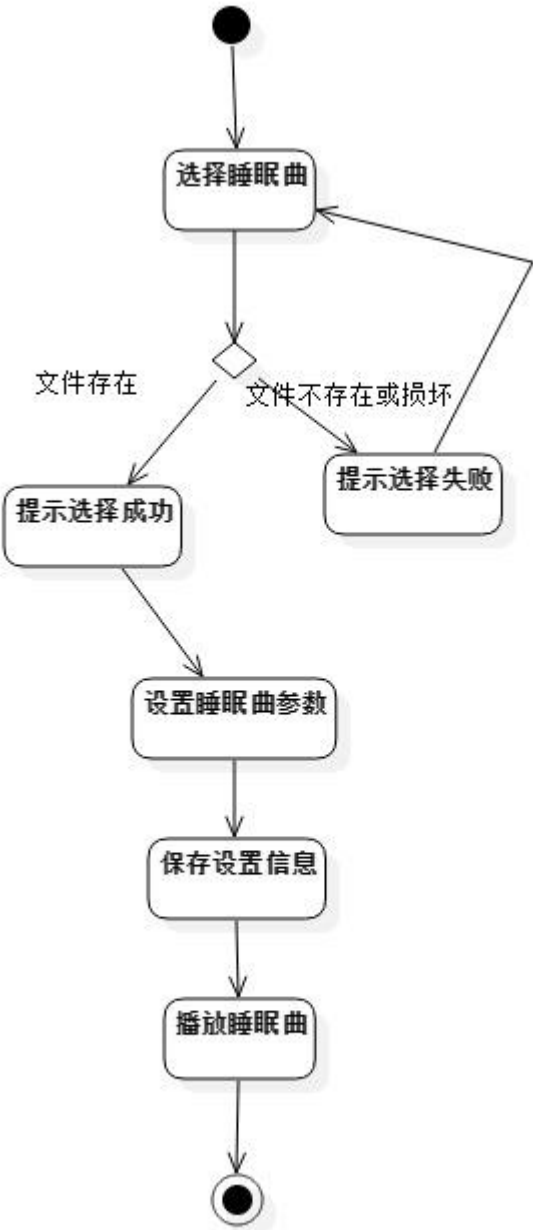


图 1：播放音乐活动图

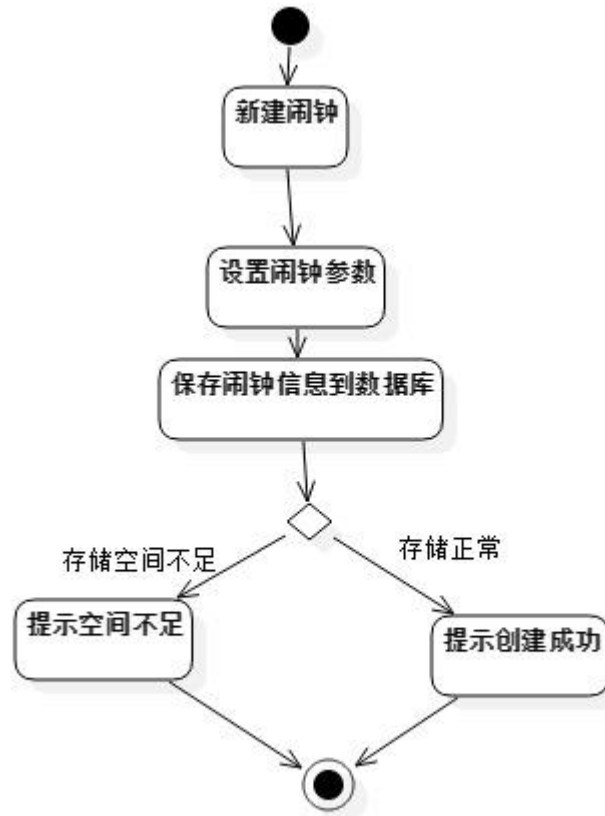


图 2：设置闹钟活动图

3、实验三：逻辑建模 – 类模型

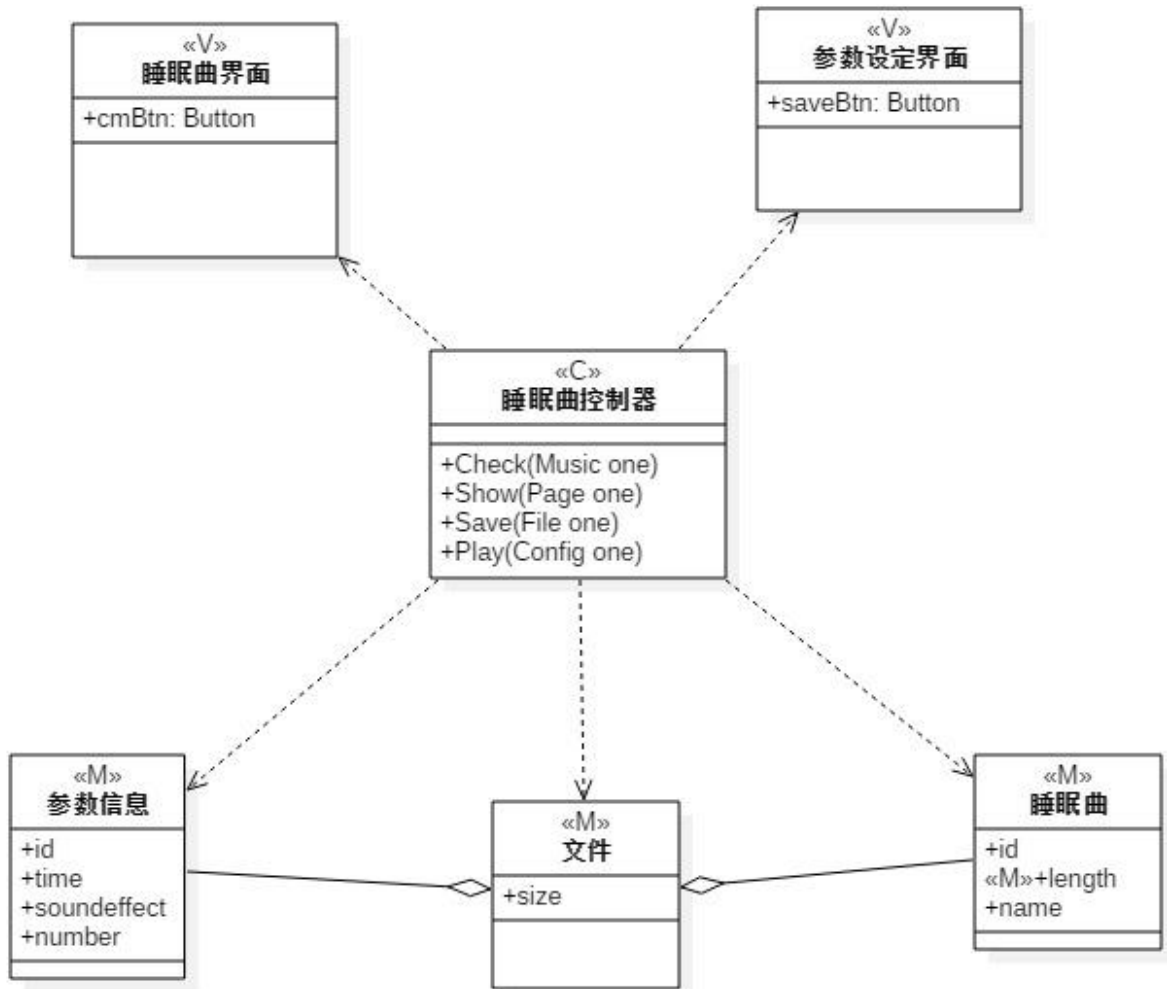
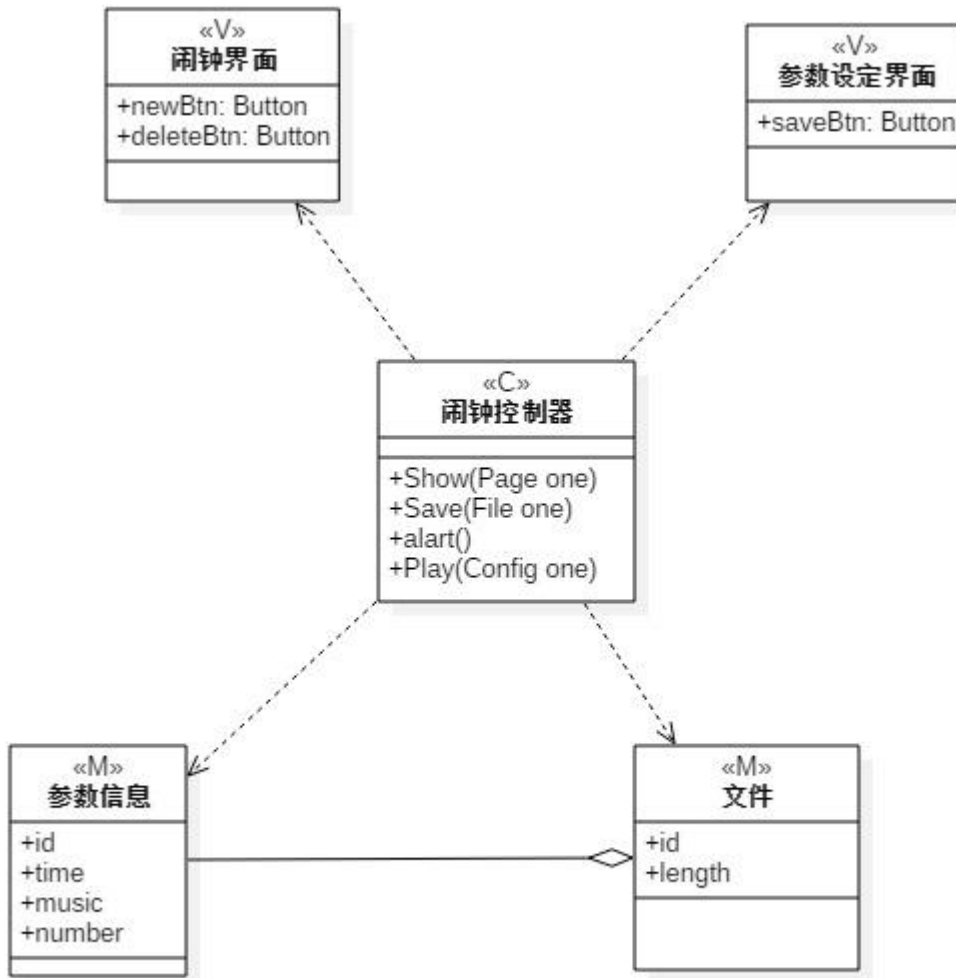


图 1：播放音乐类



图

图 2: 设置闹钟类图

4、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。

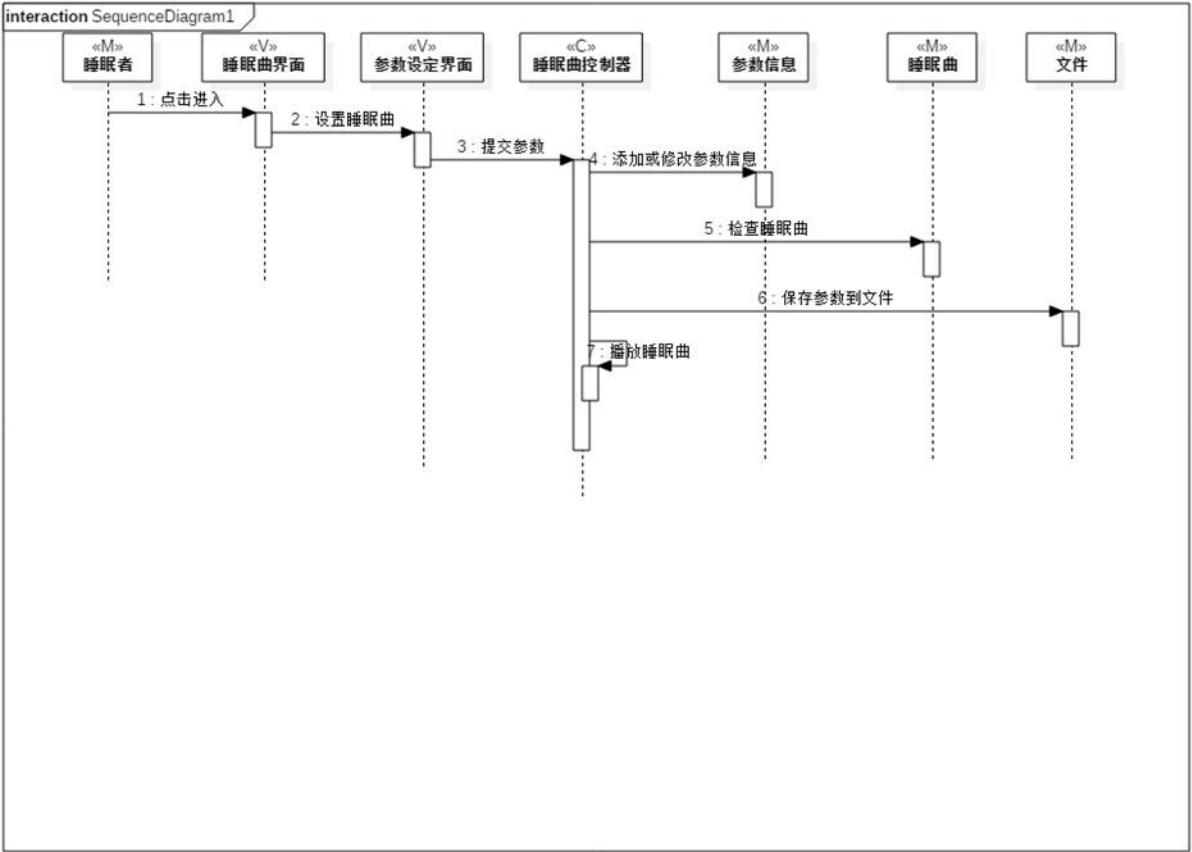


图 1：播放音乐顺序图

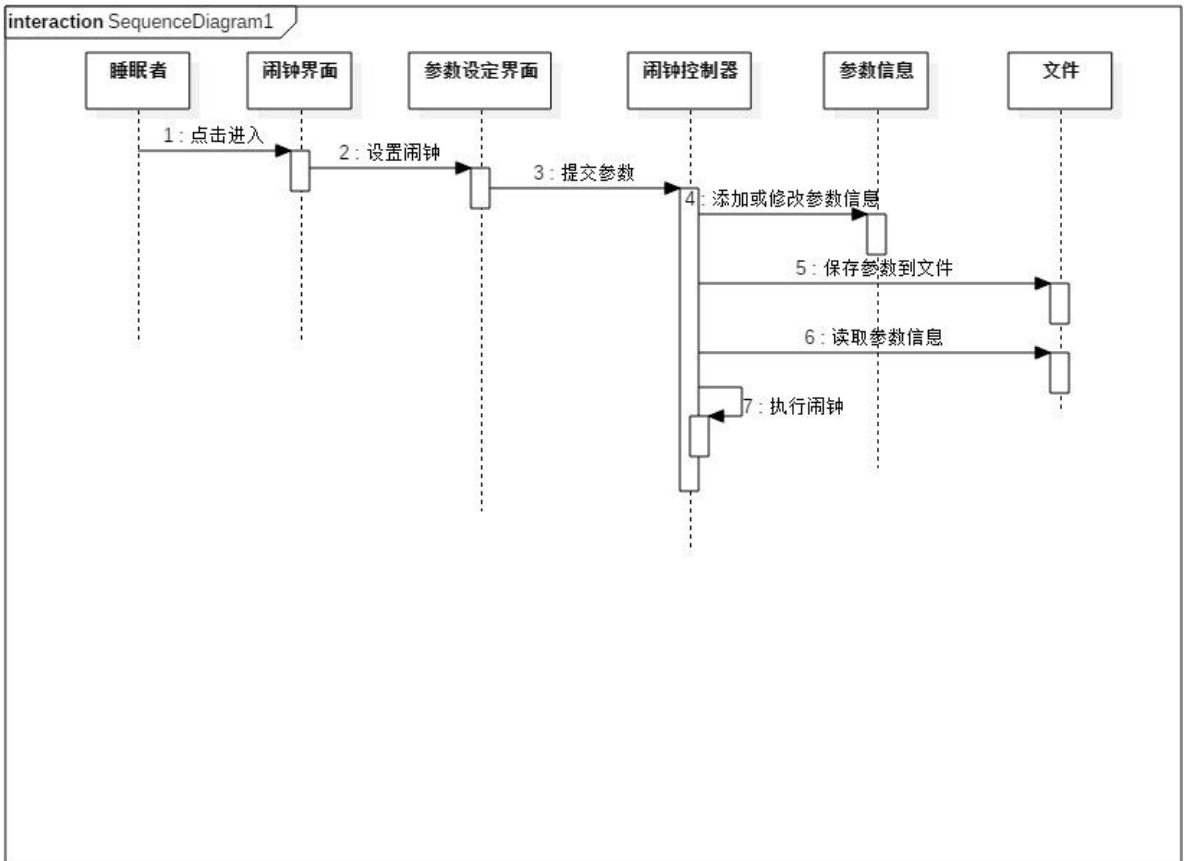


图 2：设置闹钟顺序图

5、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。

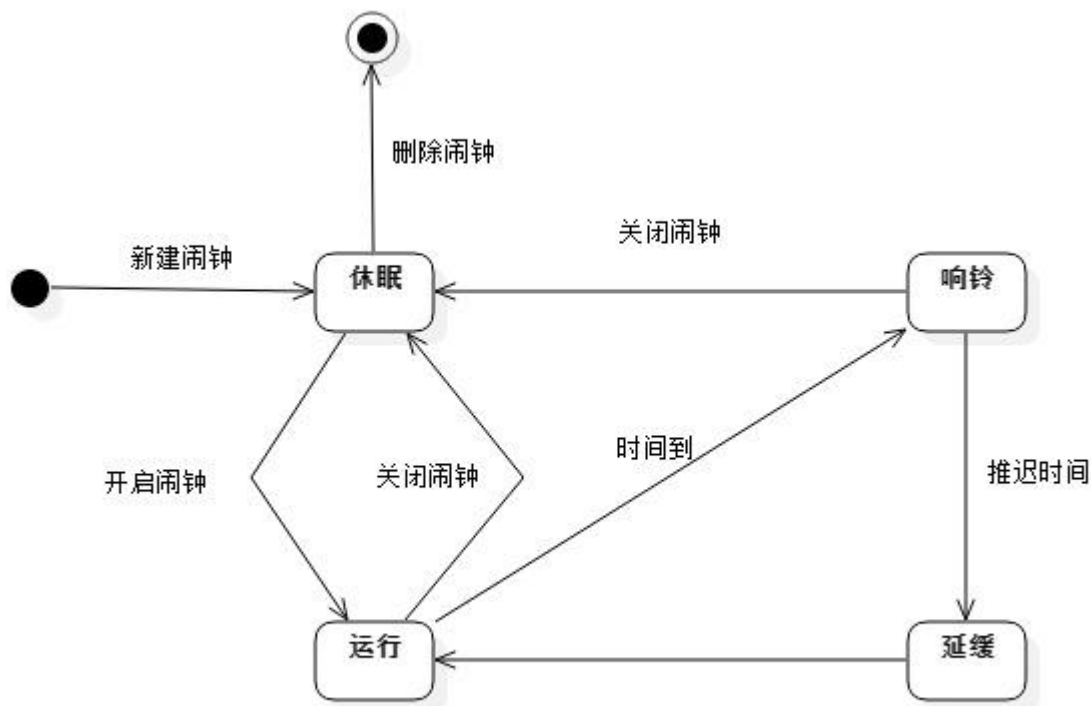


图 1：闹钟状态图

五、实验体会

实验一：在实验中画出了用例图，明白了用户和用例之间的关系。还写出了用例规约，扩展流程就是对基本流程的补充，是基本流程另一个条件的执行。

实验二：这次实验是画出活动图，活动图中有且只有一个开始，结束。而且每个活动都是由动宾短语组成。

实验三：类图要在上方添加一个标签，显示它的类型（视图 V，控制器 C，模型 M）。写出它们之间的关系。为接下来的做准备。

实验四：顺序图是描述各个对象之间的消息及其顺序。每个顺序图的第一个为 Actor，在后面才加上

MVC。

实验五：这次试验是画出状态图，首先应该先选出一个比较重要的对象（应该包含多个状态的对象）。
然后先写出各种状态，再把它们之间的关系切换写好。