**《软件工程》**

**实验报告四 ：面向对象的系统建模B**

**姓 名： 李唯 学 号： 2020105426**

**院 系： 计算机与信息学院 专 业：**计算机科学与技术

**实 验 室： 实验日期： 2022/10/31**

**总评成绩： 审阅教师： 杨青**

### 一、实验目的

1.掌握状态图的绘制

2.理解状态图模型的意义

3.能够从实际问题中构建状态图模型

### 二、实验环境

### Rational Rose

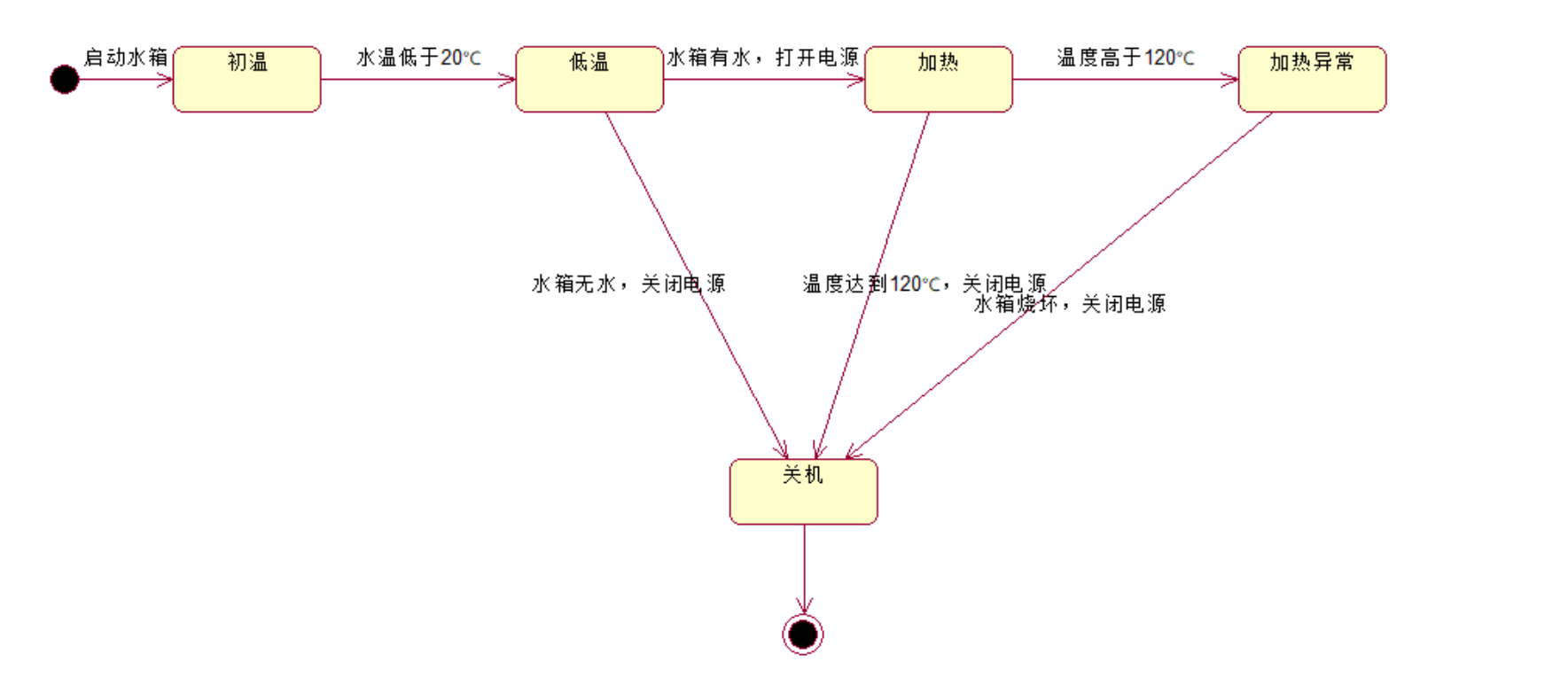
### 实验要求

1. 通过Rational Rose绘制基本状态图模型
2. 掌握绘制状态图的基本步骤和操作
3. 从实际问题中构建状态图模型

### 实验内容

## 实验一：

绘制恒温水箱20°~120°的状态图。恒温水箱启动之后，当温度传感器检测温度低于20°时，如果水量传感器检测到水箱中有水，则打开继电器电源，开始烧水；如果水量传感器检测到当前无水，则不打开电源烧水。当温度传感器检测到水箱中水的温度达到120°，则断开电源。如果发现水箱烧坏，则进行维修。



## 实验二：

阅读教材SafeHome项目说明(教材122页第八版，教材102页第九版），针对控制面板ControlPanel的状态建模，绘制ControlPanel状态图。SafeHome控制面板的行为说明参考如下：

在SafeHome的安全功能中控制面板类ControlPanel的状态如下：

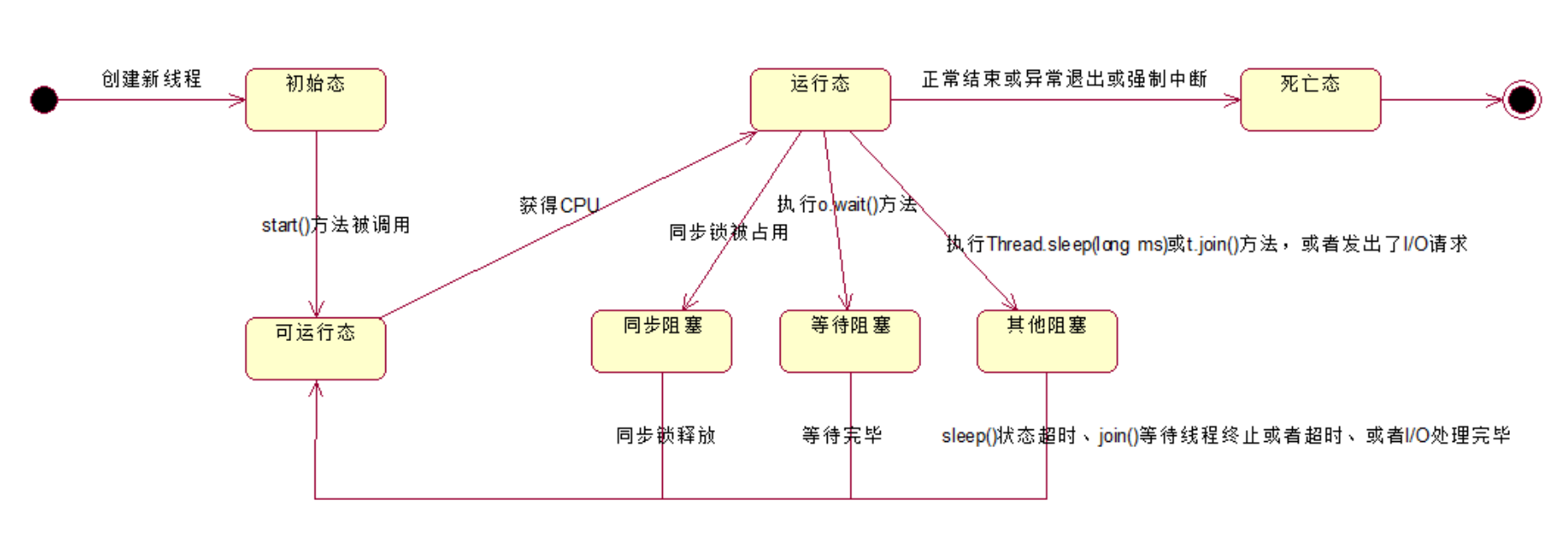
户主敲击控制面板的键盘按键输入密码时，控制面板处于读取键入字符状态。每输入一个字符则执行输入密码与设置的密码比较，验证密码是否正确；如果输入密码错误3次，则锁定控制面板；锁定时间超过120秒之后，再次允许户主输入；如果输入的密码正确，控制面板进入系统功能选择状态。

### 

## 实验三：

根据以下Java线程状态说明，使用Rose绘制Java线程状态机图：

1. 新建(new)：新创建了一个线程对象。
2. 可运行(runnable)：线程对象创建后，其他线程(比如main线程）调用了该对象的start()方法。该状态的线程位于可运行线程池中，等待被线程调度选中，获取cpu 的使用权 。
3. 运行(running)：可运行状态(runnable)的线程获得了cpu 时间片（timeslice） ，执行程序代码。
4. 阻塞(block)：阻塞状态是指线程因为某种原因放弃了cpu 使用权，也即让出了cpu timeslice，暂时停止运行。直到线程进入可运行(runnable)状态，才有机会再次获得cpu timeslice 转到运行(running)状态。阻塞的情况分三种：
5. (一)等待阻塞：运行(running)的线程执行o.wait()方法，JVM会把该线程放入等待队列(waitting queue)中。
6. (二)同步阻塞：运行(running)的线程在获取对象的同步锁时，若该同步锁被别的线程占用，则JVM会把该线程放入锁池(lock pool)中。
7. (三)其他阻塞：运行(running)的线程执行Thread.sleep(long ms)或t.join()方法，或者发出了I/O请求时，JVM会把该线程置为阻塞状态。当sleep()状态超时、join()等待线程终止或者超时、或者I/O处理完毕时，线程重新转入可运行(runnable)状态。
8. 死亡(dead)：线程run()、main() 方法执行结束，或者因异常退出了run()方法，或者用interrupt方法强制中断则该线程结束生命周期。死亡的线程不可再次复生。



### **五、实验结论**

在面向对象的软件系统中，一个对象无论多么简单或者多么复杂，都必然会经历一个从开始创建到最终消亡的完整过程，这个过程通常被称为对象的生命周期。一般来说，对象在其生命周期内是不可能完全孤立的，它必然会接受消息来改变自身，或者发送消息来影响其他对象。而状态图(State Diagram)用来描述一个特定对象在其生命周期中的各种状态以及状态之间的转换。这些对象可以是类、接口等等。通过对状态图的绘制，我们可以清楚的了解一个对象状态之间的转换顺序，通过状态的转换顺序可以清晰看出事件的执行顺序，对于我们在实际开发前对各个对象的了解具有重大意义。