

实验四：面向对象的系统建模B

[分享](#)[收藏](#)[点赞 0](#)[评论0](#)

2021-10-13 20:12:39

来源：本站编辑

阅读量：133

“可运行状态(runnable)的线程获得了cpu 时间片 (timeslice) 直到线程进入可运行(runnable)状态运行(running)的线程在获取对象的同步锁时JVM会把该线程置为阻塞状态该状



目录

[课程目标](#)[实验目的：](#)[实验要求：](#)[实验内容：](#)[B1](#)[B2](#)[B3](#)

课程目标

课程目标1：能够使用Rose 等常用UML建模工具构建用例模型、行为模型、设计模型。

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

实验要求：

- 1.按照实验报告模板编写实验报告，
- 2.模型图贴到文档的适当位置；
- 3.文件以“学号-姓名-实验四.doc”的方式命名，提交到长江雨课堂实验四；

.....

实验内容：

B1

绘制恒温水箱20°~120°的状态图，可参考讲课PPT；

B2

阅读教材SafeHome项目说明，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。

B3

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：
(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。

(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。

(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。

5. 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。

< 上一篇：实验三：面向对象的...

下一篇：没有了 >

评论

已有0条评论

0/150

提交

热门评论





还没有任何评论互动呢~