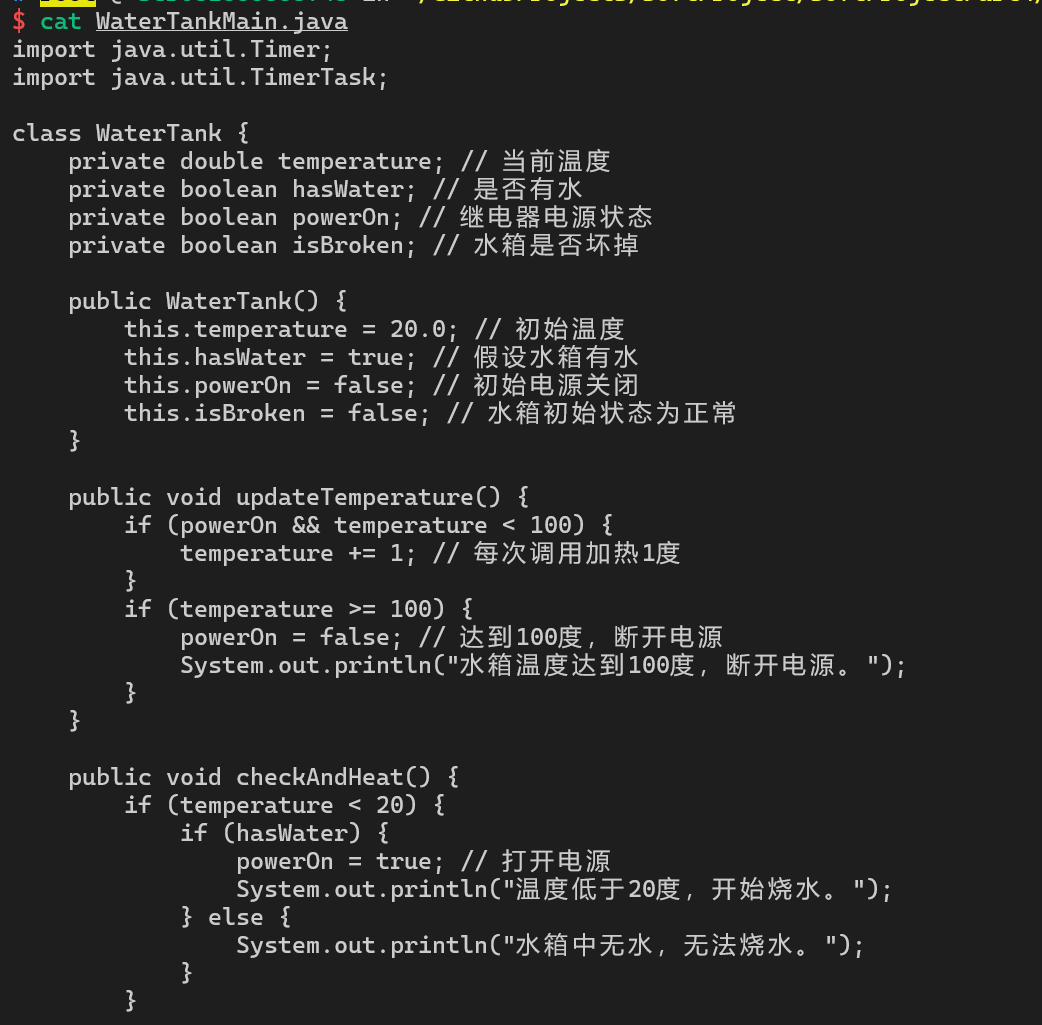
**《软件工程》**

**实验报告四：面向对象的系统建模B**

**姓 名： 胡国昌 学 号： 202210120518**

**院 系： 计算机与信息学院 专 业： 计算机科学与技术**

**实 验 室： J1-306 实验日期： 2024.10.21**

**总评成绩： 审阅教师： 杨青**

# 课程目标

1. 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，充分理解状态图的三要素
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

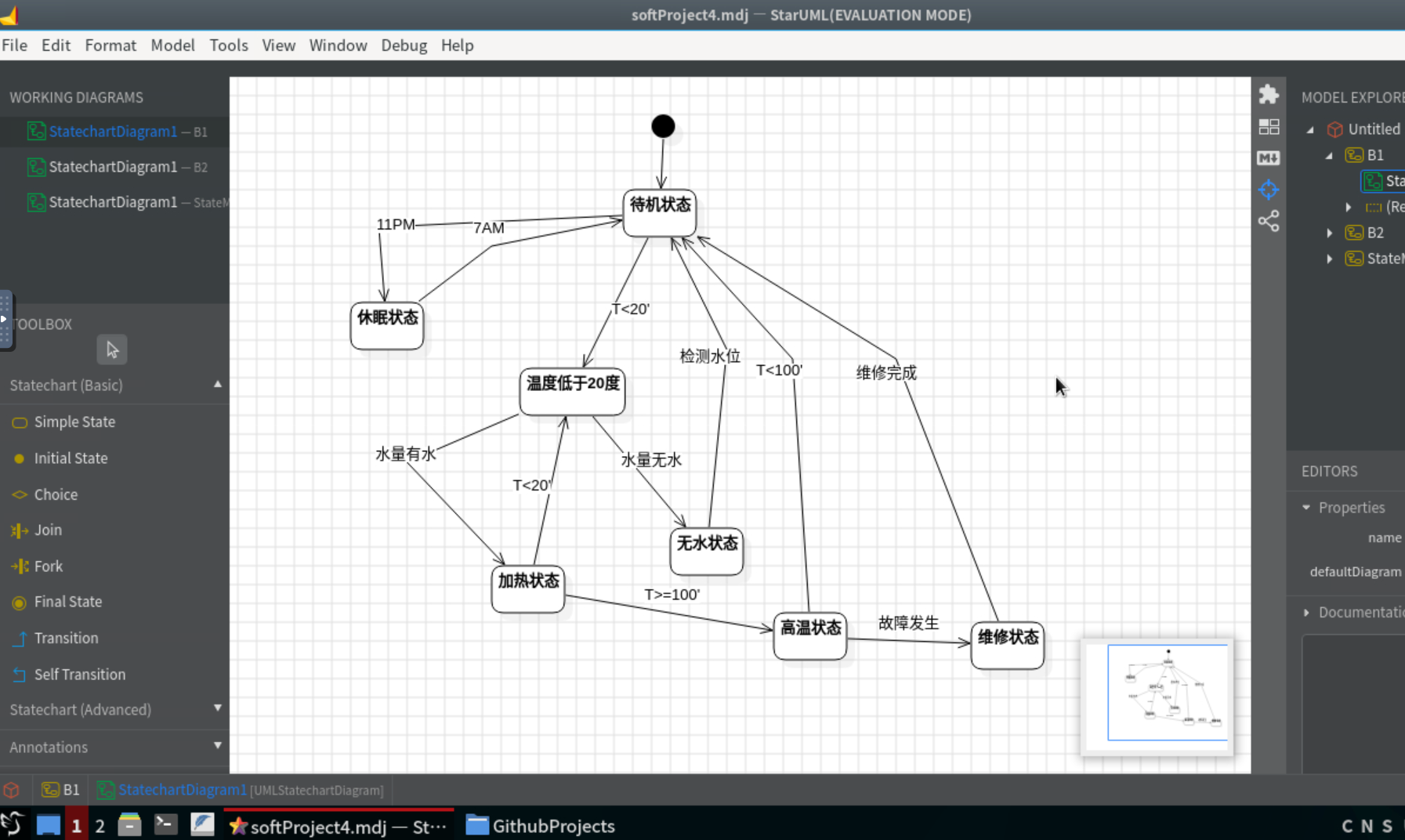
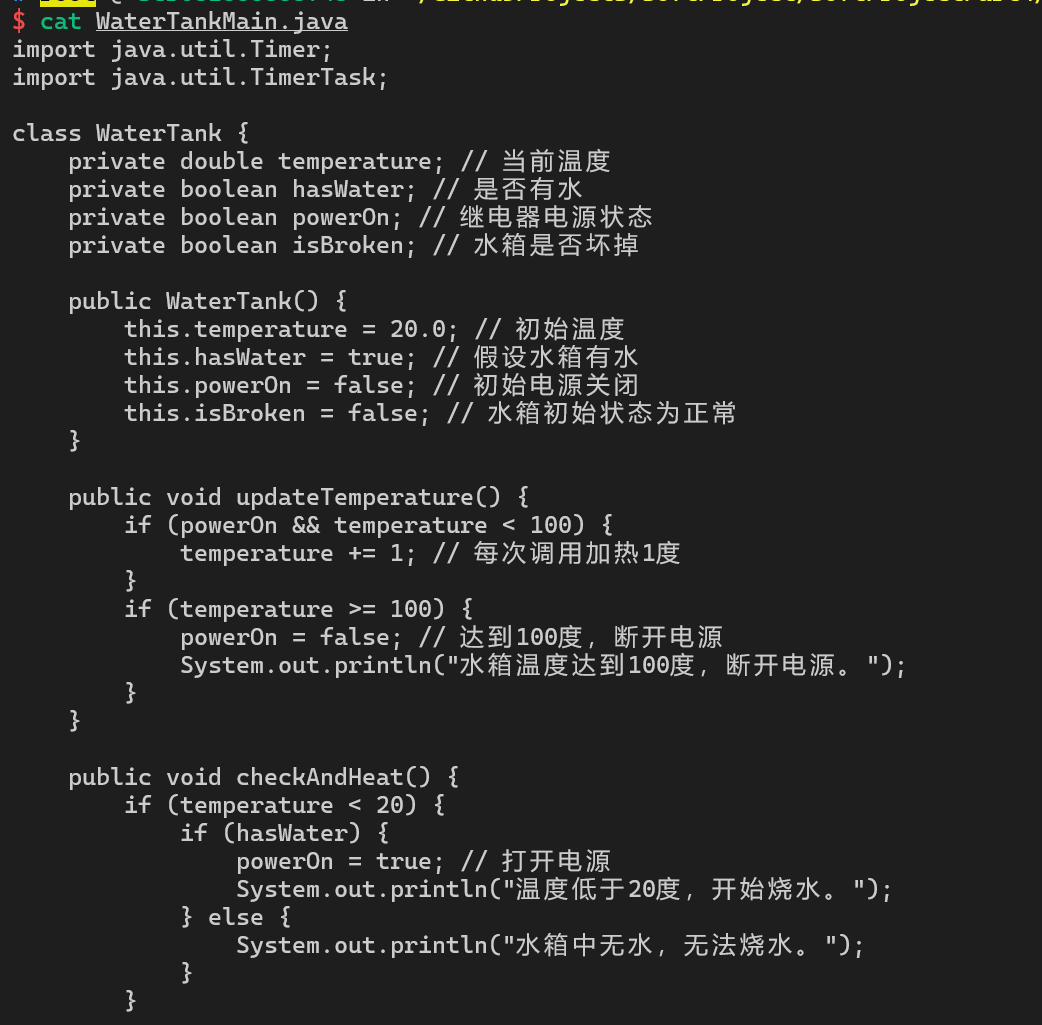
1. 按照实验报告模板编写实验报告，
2. 模型图贴到文档的适当位置；
3. 使用自己熟悉的语言，实现状态机模型，源代码的gitee地址写到实验报告中；
4. 文件以“学号-姓名-软件工程实验四.doc”的方式命名，提交到长江雨课堂软件工程实验四；

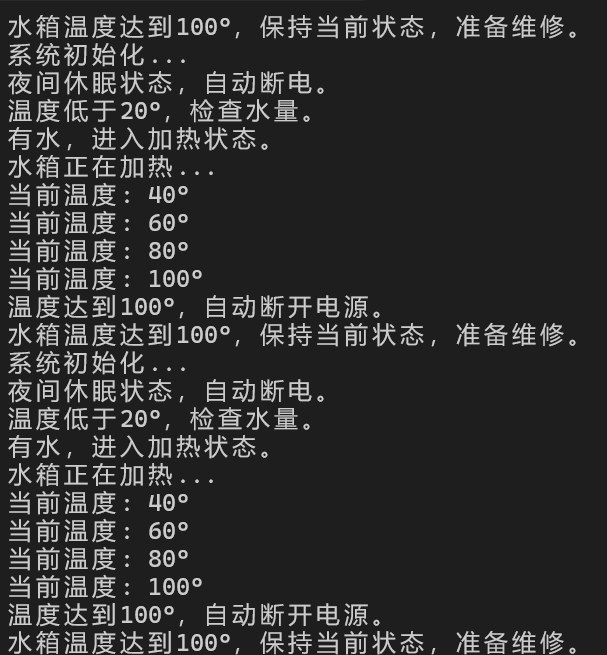
# 附加说明

Java 语言实现的状态机可以参考课堂上简化的的恒温烧水器代码示例，<https://gitee.com/izxj/seassignments.git> 。采用其他语言实现的同学可以自行网上搜索相关资料实现之。  
使用Visual Paradigm绘制状态机图可参考[pc访问](https://star.jmhui.com.cn/p1/938.html" \o "pc访问)，[移动端访问](https://star.jmhui.com.cn/m1/938.html)  
[========]

# 实验内容

## B1

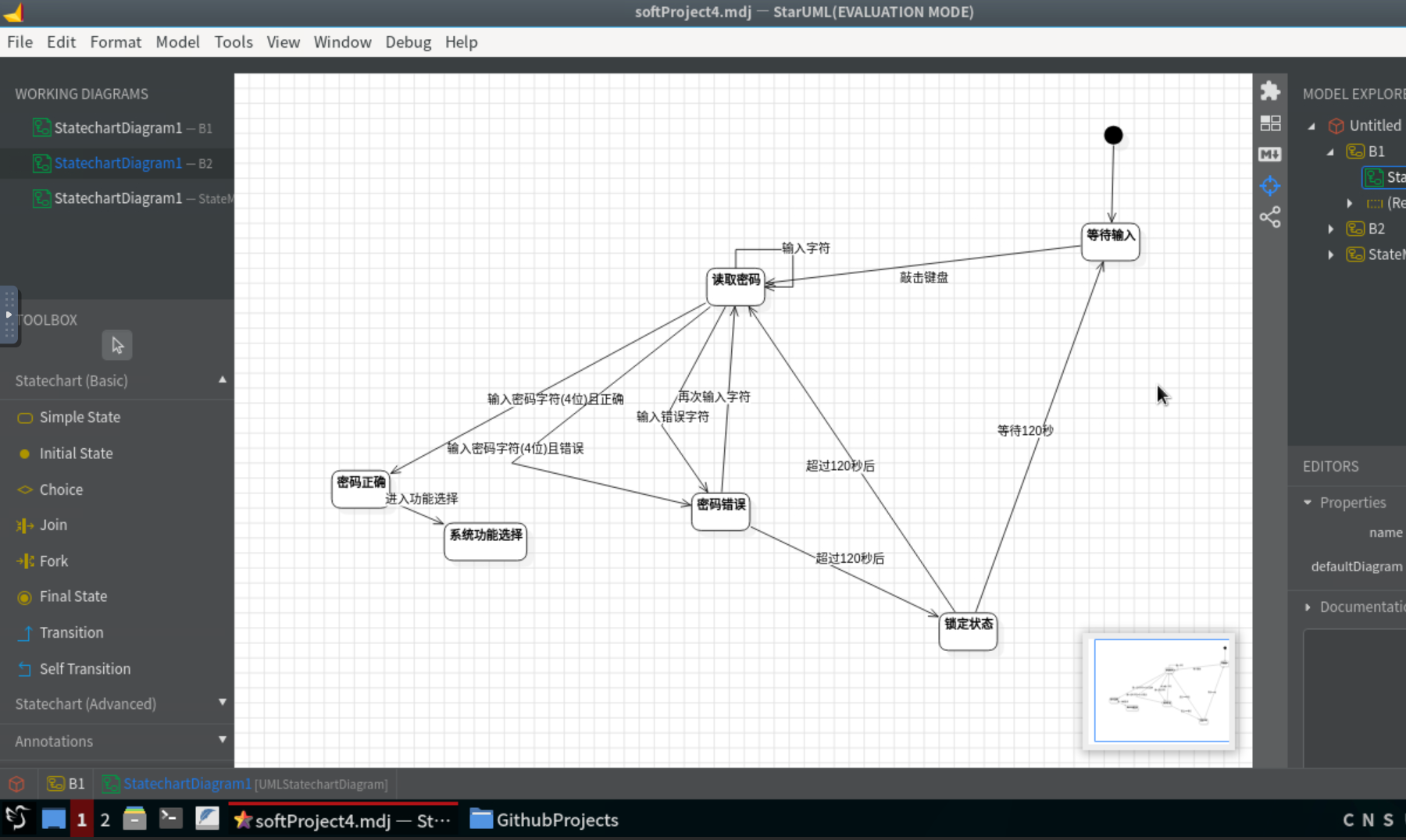
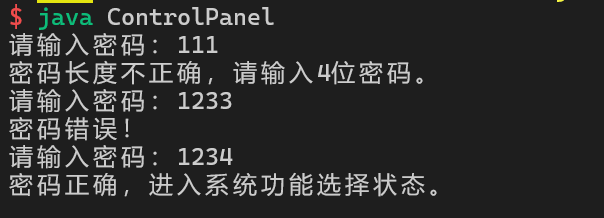
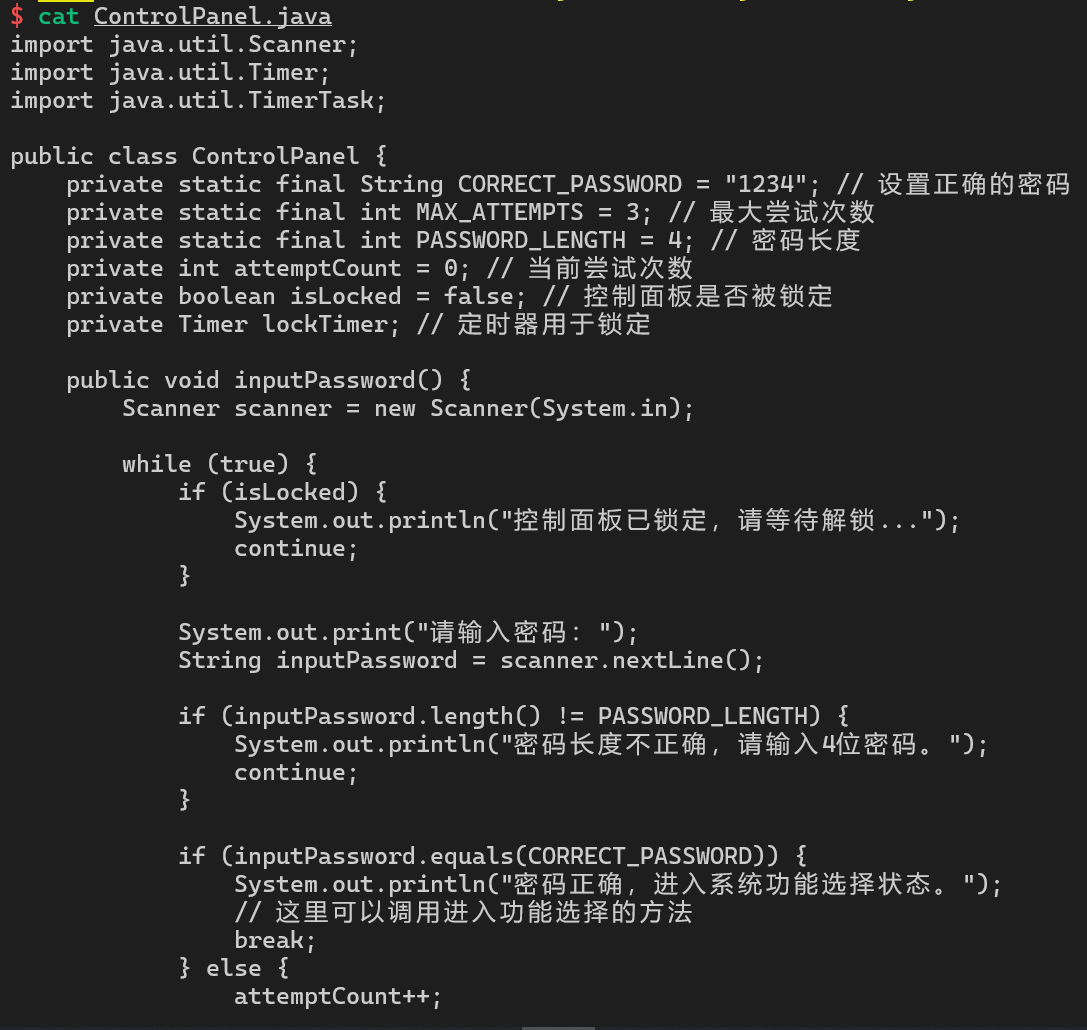
绘制恒温水箱20°~100°的状态图。恒温水箱启动之后，当温度传感器检测温度低于20°时，如果水量传感器检测到水箱中有水，则打开继电器电源，开始烧水；如果水量传感器检测到当前无水，则不打开电源烧水。当温度传感器检测到水箱中水的温度达到100°，则断开电源。如果水箱烧坏，则进行维修。当晚上11点整时，自动断开电源，进入休眠状态；早上7点整，自动进入上述恒温烧水过程。



## B2

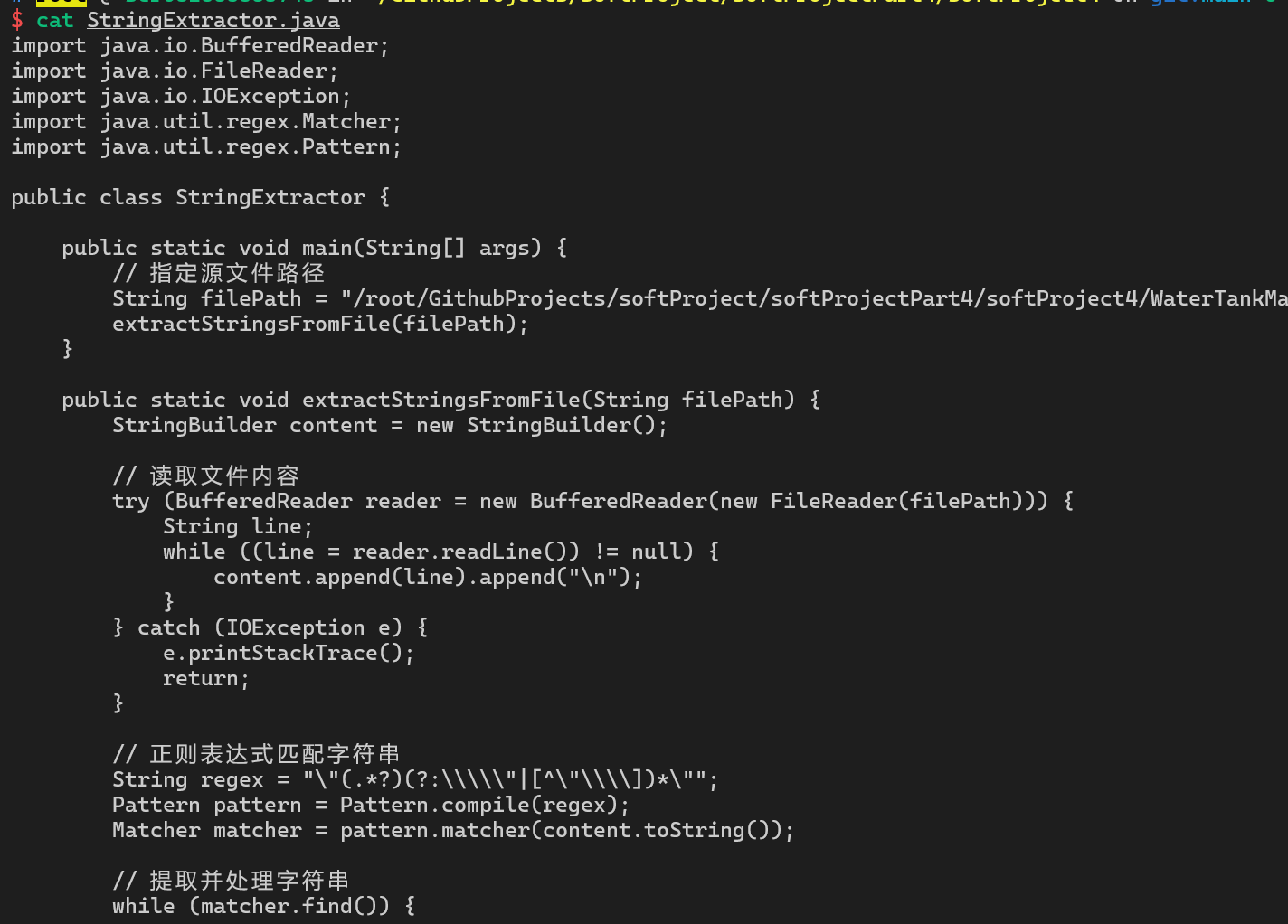
阅读教材《软件工程-实践者的方法》第八版 SafeHome项目说明(教材122页第八版，教材102页第九版），针对控制面板ControlPanel的状态建模，绘制ControlPanel

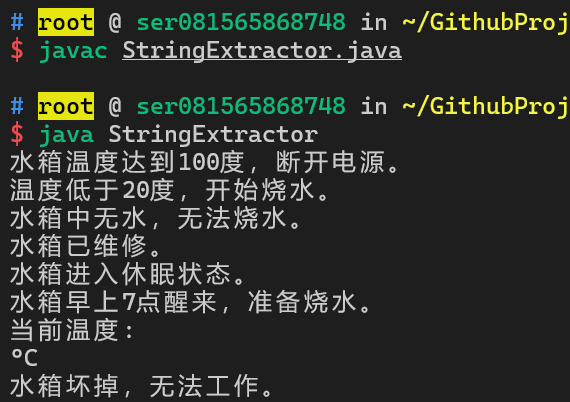
状态图。SafeHome控制面板的行为说明参考如下：  
在SafeHome的安全功能中控制面板类ControlPanel的状态如下：  
户主敲击控制面板的键盘按键输入密码时，控制面板处于读取键入字符状态。每输入一个字符则执行输入密码与设置的密码长度（4位）比较，如果达到密码长度则验证密码是否正确；如果输入密码错误3次，则锁定控制面板；锁定时间超过120秒之后，再次允许户主输入；如果输入的密码正确，控制面板进入系统功能选择状态。

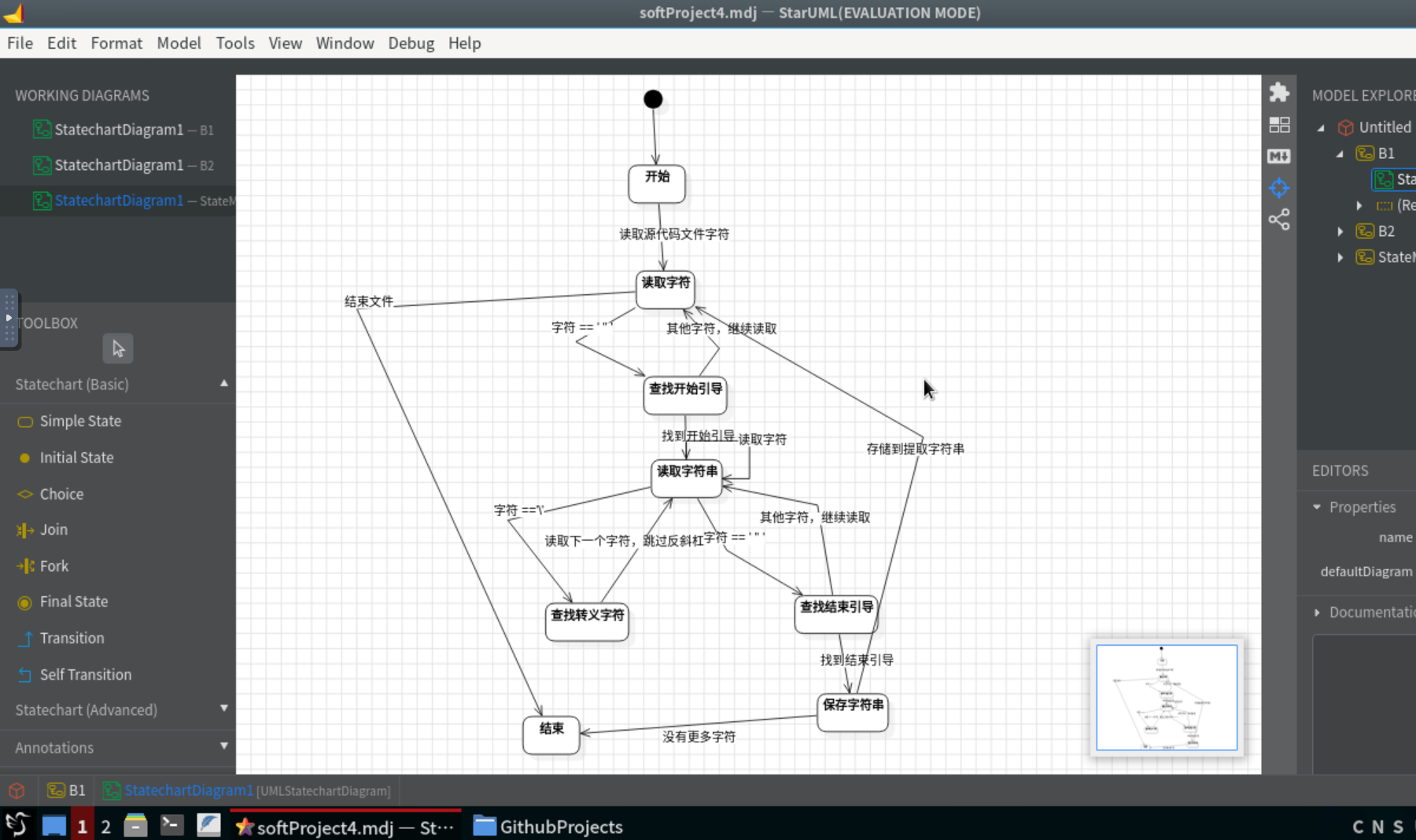


## B3

根据以下问题描述，采用状态图/状态机建模，并实现之：  
用户需要从源代码文件中提取源文件中的所有字符串。字符串是双引号之间的文本，但是字符串中的反斜杠会转义下一个字符，比如对于字符串：”一般学习一门编程语言的时候，基本上第一个程序是输出\”Hello World!\””,应该输出：一般学习一门编程语言的时候，基本上第一个程序是输出”Hello World!”。







# 实验结论

通过本次实验，我们对状态图及其在系统设计中的应用有了深入的理解。以下是此次实验的主要总结和反思：

**1. 理论与实践结合**

实验提供了一个将状态图理论应用于实际问题的机会。我们通过对恒温水箱、ControlPanel和字符串提取的建模，不仅理解了如何识别和绘制系统的状态和事件，还认识到状态图在描述系统行为时的清晰性和有效性。理论知识的学习使我们在绘制状态图时能够更好地识别状态、事件和转换条件。

**2. 状态图的重要性**

状态图在系统设计中具有以下几个重要的优势：

直观性：通过图形化的方式展示系统状态和状态间的转换，使设计逻辑一目了然。

系统化：状态图中明显的状态和转换关系使得系统在不同条件下的反应被系统地记录和呈现，从而更容易理解系统的整体运作。

可维护性：通过明确的状态和条件记录，后续对系统的维护和升级变得更加容易，因为开发人员能够快速识别系统的工作流程及其关键信息。

**3. 过程中的挑战**

在绘制状态图的过程中，我们也遇到了一些挑战：

状态与事件的识别：在复杂系统的状态转移中，必需明确每个状态的切换条件，避免遗漏。对于如字符串提取等复杂场景，识别所有可能的字符和转义情况需要仔细分析。

状态图的复杂性：对于某些系统，如ControlPanel，需要处理多种输入状态与锁定机制，绘制时需要有效管理状态图的复杂度，确保状态之间的过渡逻辑清晰。

**4. 改进与未来学习方向**

通过本次实验，我们认识到了结构化思维的重要性。在将来进行复杂系统设计时，可以采取以下改进措施：

分阶段建模：在设计复杂系统时，可以通过分阶段构建状态图，先建立每个模块的状态图，再进行综合，将各个模块结合起来，增强系统的整体性。

使用更高级的建模工具：可以考虑使用更高级的建模工具和方法，如UML状态机、时序图和活动图等，帮助更全面地展示系统的动态行为。

实际应用场景：将此类状态图应用于实际开发和项目管理中，帮助团队更好地理解需求的动态变化和约束条件。

**5. 总结经验**

最后，综上所述，本次实验显著增强了我们对状态图的理解，掌握了状态建模的基本方法和技巧。通过参与和完成这一研究任务，我们也体会到了理论知识在实际软件工程项目实现中的重要性。希望在今后的学习和工作中，能够继续运用这些技能，进一步深化对软件工程原理和实践的理解。。

**仓库地址**

[https://github.com/xieyangyuyue /softProject.git](https://github.com/xieyangyuyue%20/softProject.git)