**《软件工程》**

**实验报告三 ：面向对象的系统建模B**

**姓 名： 郝舒森 学 号： 202210120305**

**院 系： 计算机与信息学院 专 业： 计算机科学与技术**

**实 验 室： J-1307 实验日期： 11.5**

**总评成绩： 审阅教师：**

### 一、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

### 二、实验环境

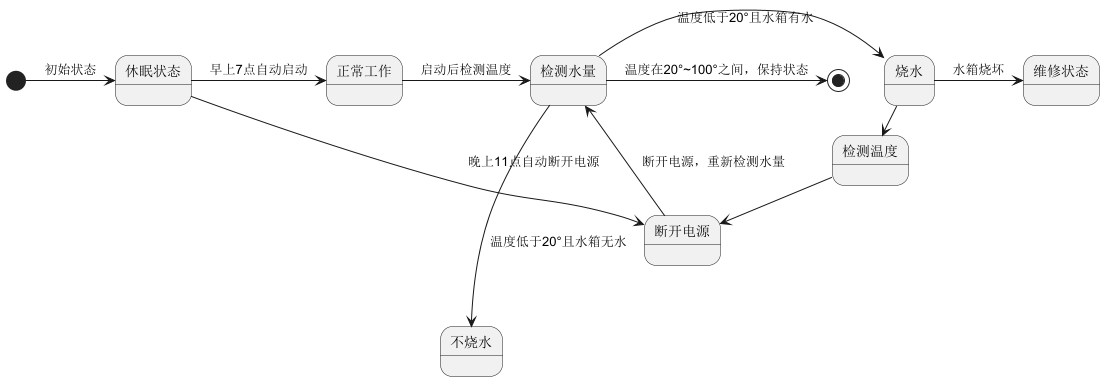
PlantUML，idea，java17

### 三、实验要求

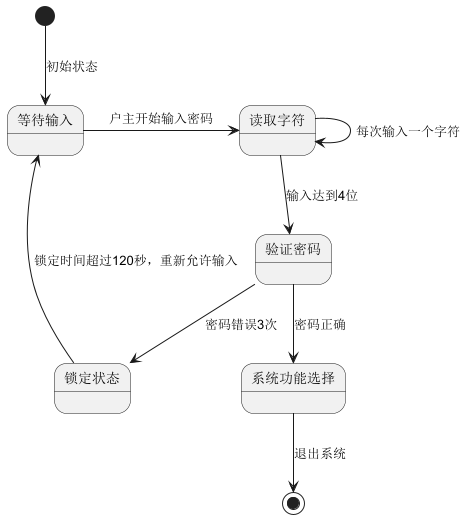
1. 1.按照实验报告模板编写实验报告，
2. 2.模型图贴到文档的适当位置；
3. 3.使用自己熟悉的语言，实现状态机模型，源代码的gitee地址写到实验报告中；
4. 4.文件以“学号-姓名-软件工程实验四.doc”的方式命名，提交到长江雨课堂软件工程实验四

### 四、实验内容

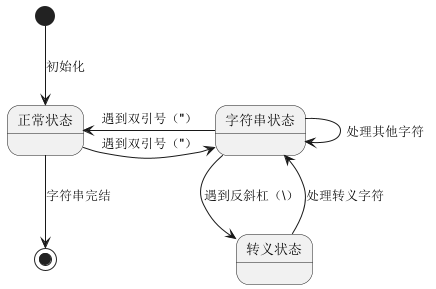
B1



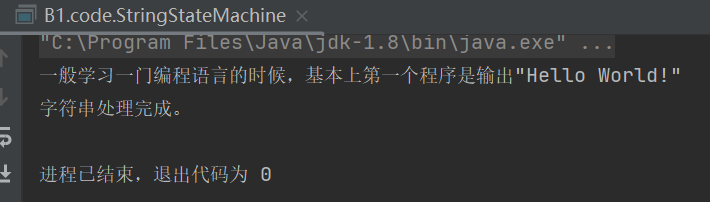
B2



B3



package B1.code;  
  
// 状态接口  
interface State {  
 void handle(Character input, StateMachine context);  
}  
  
// 状态机上下文类  
class StateMachine {  
 private State currentState; // 当前状态  
  
 public StateMachine(State initialState) {  
 this.currentState = initialState; // 初始化状态  
 }  
  
 public void setCurrentState(State state) {  
 this.currentState = state; // 设置当前状态  
 }  
  
 public void handleInput(Character input) {  
 currentState.handle(input, this); // 处理输入字符  
 }  
}  
  
// 具体状态类：正常状态  
class NormalState implements State {  
 @Override  
 public void handle(Character input, StateMachine context) {  
 if (input == '“') {  
 // 遇到双引号，进入字符串状态  
 context.setCurrentState(new StringState());  
 } else {  
 // 处理其他字符  
 System.*out*.print(input);  
 }  
 }  
}  
  
// 具体状态类：字符串状态  
class StringState implements State {  
 @Override  
 public void handle(Character input, StateMachine context) {  
 if (input == '\\') {  
 // 遇到反斜杠，进入转义状态  
 context.setCurrentState(new EscapeState());  
 } else if (input == '”') {  
 // 遇到双引号，返回到正常状态  
 context.setCurrentState(new NormalState());  
 } else {  
 // 在字符串状态下处理其他字符  
 System.*out*.print(input);  
 }  
 }  
}  
  
// 具体状态类：转义状态  
class EscapeState implements State {  
 @Override  
 public void handle(Character input, StateMachine context) {  
 // 打印转义字符  
 System.*out*.print(input);  
 // 返回字符串状态  
 context.setCurrentState(new StringState());  
 }  
}  
  
// 主类，运行状态机  
class StringStateMachine {  
 public static void main(String[] args) {  
 StateMachine stateMachine = new StateMachine(new NormalState());  
  
 // 测试字符串  
 String testString = "“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!\\\"”";  
  
 for (char ch : testString.toCharArray()) {  
 stateMachine.handleInput(ch); // 逐字符处理输入  
 }  
  
 // 最终输出换行，表示字符串处理完成  
 System.*out*.println("\n字符串处理完成。");  
 }  
}



### 五、实验结论

通过本次软件工程实验“面向对象的系统建模”，我深入理解了状态机在系统行为建模中的重要性，并熟练使用了PlantUML工具完成实验任务。在实验中，我构建了恒温水箱的状态图，清晰展示了各个状态之间的转换关系以及系统在不同条件下的反应。通过对状态的细致分析，我学会了如何用UML图形化语言准确描述系统的动态行为，尤其是在处理水量和温度传感器输入时，确保系统的健壮性与自动控制能力。实验过程中，我掌握了状态之间的转换条件及其影响，能够有效应对不同的运行环境。这次实验不仅加深了我对状态机模型的理解，还提升了我在实际项目中进行动态行为建模的能力，为今后的系统设计与实现奠定了坚实的基础。

### 六、仓库地址

https://github.com/Senwwwwww/Software