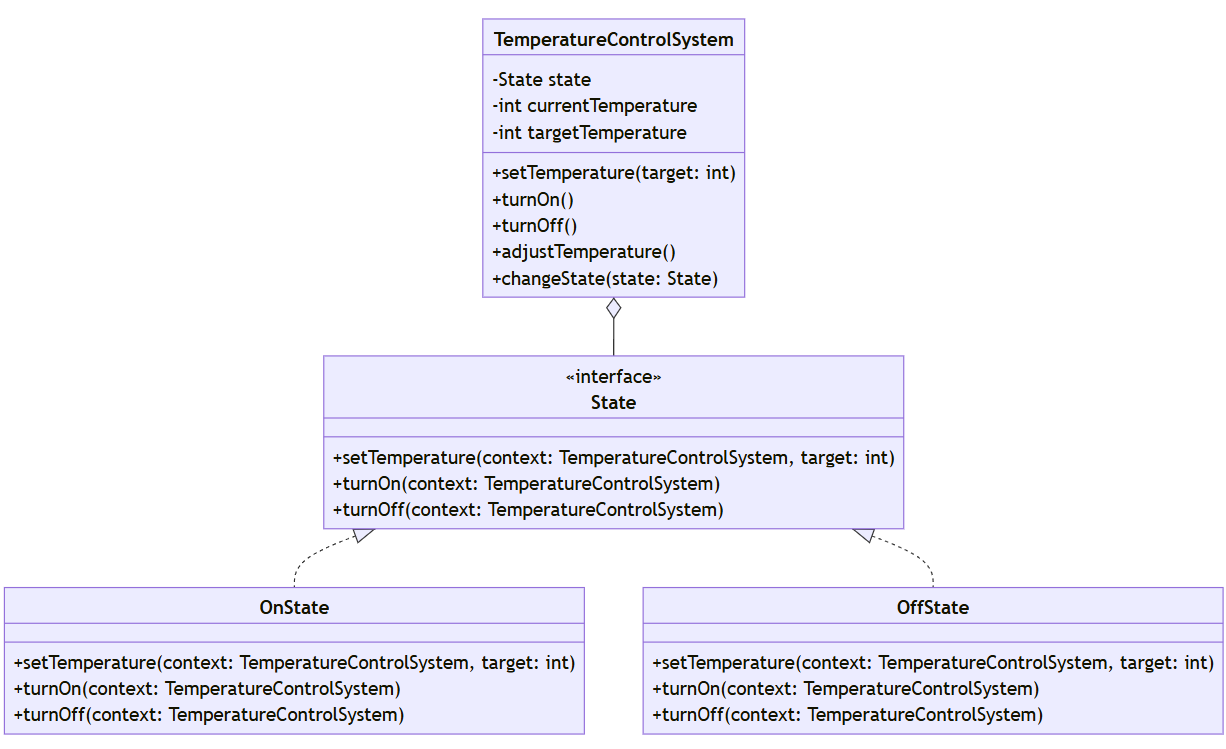
**题目：**

小明正在开发一个温度自动调节系统，该系统能够根据用户设置的温度值自动调节温度。当系统开启时，它会读取用户设置的期望温度值T，若当前温度低于T，它将温度提升到T，若当前温度高于T，它将温度降为T。当系统关闭时，它将停止调节温度。当系统开启时，它可以接受温度设置及关闭系统的指令，当系统关闭时，它只接受开启系统的指令。请用恰当的设计模式设计该系统，画出类图，给出核心代码。

**模式：**

状态模式

**类图：**



|  |
| --- |
| **代码** |
| *// 状态接口*  interface State {      void setTemperature(TemperatureControlSystem context, int target);      void turnOn(TemperatureControlSystem context);      void turnOff(TemperatureControlSystem context);  }  *// 开启状态类*  class OnState implements State {      private static OnState instance = new OnState();      private OnState() {}      public static OnState getInstance() {          return instance;      }      @Override      public void setTemperature(TemperatureControlSystem context, int target) {          System.out.println("Setting target temperature to " + target + " degrees.");          context.setTargetTemperature(target);          context.adjustTemperature();      }      @Override      public void turnOn(TemperatureControlSystem context) {          System.out.println("System is already ON.");      }      @Override      public void turnOff(TemperatureControlSystem context) {          System.out.println("Turning off the system.");          context.changeState(OffState.getInstance());      }  }  *// 关闭状态类*  class OffState implements State {      private static OffState instance = new OffState();      private OffState() {}      public static OffState getInstance() {          return instance;      }      @Override      public void setTemperature(TemperatureControlSystem context, int target) {          System.out.println("System is OFF. Cannot set temperature.");      }      @Override      public void turnOn(TemperatureControlSystem context) {          System.out.println("Turning on the system.");          context.changeState(OnState.getInstance());      }      @Override      public void turnOff(TemperatureControlSystem context) {          System.out.println("System is already OFF.");      }  }  *// 温度控制系统上下文类*  class TemperatureControlSystem {      private State state;      private int currentTemperature;      private int targetTemperature;      public TemperatureControlSystem() {          state = OffState.getInstance();  *// 默认初始状态为关闭*          currentTemperature = 20;         *// 假设默认当前温度为20度*      }      public void setTargetTemperature(int target) {          targetTemperature = target;      }      public void setTemperature(int target) {          state.setTemperature(this, target);      }      public void turnOn() {          state.turnOn(this);      }      public void turnOff() {          state.turnOff(this);      }      public void adjustTemperature() {          if (currentTemperature < targetTemperature) {              System.out.println("Increasing temperature to " + targetTemperature + " degrees.");              currentTemperature = targetTemperature;          } else if (currentTemperature > targetTemperature) {              System.out.println("Decreasing temperature to " + targetTemperature + " degrees.");              currentTemperature = targetTemperature;          } else {              System.out.println("Temperature is already at the target: " + targetTemperature + " degrees.");          }      }      public void changeState(State state) {          this.state = state;      }  }  *// 测试类*  public class TemperatureControlTest {      public static void main(String[] args) {          TemperatureControlSystem system = new TemperatureControlSystem();  *// 尝试设置温度（系统关闭状态）*          system.setTemperature(25);  *// 开启系统并设置温度*          system.turnOn();          system.setTemperature(25);  *// 再次设置温度*          system.setTemperature(18);  *// 关闭系统*          system.turnOff();  *// 再次尝试设置温度（系统关闭状态）*          system.setTemperature(22);      }  } |