**Experiment Report - 61 - test8\_SmartGasStove**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | line 22 | Semantic | √ | × | × |
| E02 | line 26 | Semantic | √ | × | × |
| E03 | line 39 | Logic | √ | √ | √ |

Additional Errors Found by Self: 0

Self-Review Detection Rate: 100%

Peer 1 Detection Rate: 33%

Peer 2 Detection Rate: 33%

1. **Source Code**
2. package a;
3. public class c100\_SmartGasStove {
4. private double minFlameIntensity; // 最低火焰强度阈值
5. private double maxFlameIntensity; // 最高火焰强度阈值
6. private double currentFlameIntensity; // 当前火焰强度
7. private double gasFlowRate; // 当前燃气流量（单位：L/min）
8. // 构造函数：初始化火焰强度阈值、初始火焰强度和燃气流量
9. public c100\_SmartGasStove(double minFlameIntensity, double maxFlameIntensity, double initialFlameIntensity, double initialGasFlowRate) {
10. this.minFlameIntensity = minFlameIntensity;
11. this.maxFlameIntensity = maxFlameIntensity;
12. this.currentFlameIntensity = initialFlameIntensity;
13. this.gasFlowRate = initialGasFlowRate;
14. }
15. // 更新火焰强度，并根据数值条件调整燃气流量
16. public void updateFlameIntensity(double newFlameIntensity) {
17. currentFlameIntensity = newFlameIntensity;
18. adjustGasFlow();
19. System.out.printf("Current Flame Intensity: %.2f, Gas Flow Rate: %.1f L/min%n",
20. currentFlameIntensity, gasFlowRate);
21. if (currentFlameIntensity < minFlameIntensity) {
22. System.out.println("ALETR: Flame intensity too low! Increasing gas flow.");
23. } else if (currentFlameIntensity > maxFlameIntensity) {
24. System.out.println("ALERT: Flame intensity too high! Decreasing gas flow.");
25. } else {
26. System.out.println("Flame intensity is optimal. No adjustment needed.");
27. }
28. }
29. // 根据火焰强度调整燃气流量
30. private void adjustGasFlow() {
31. if (currentFlameIntensity < minFlameIntensity) {
32. // 增加燃气流量 20%
33. gasFlowRate = gasFlowRate \* 1.2;
34. } else if (currentFlameIntensity >= maxFlameIntensity) {
35. // 降低燃气流量 20%
36. gasFlowRate = gasFlowRate \* 0.8;
37. }
38. // 当火焰强度在最佳范围内时，不改变燃气流量
39. }
40. // 获取当前火焰强度
41. public double getCurrentFlameIntensity() {
42. return currentFlameIntensity;
43. }
44. // 获取当前燃气流量
45. public double getGasFlowRate() {
46. return gasFlowRate;
47. }
48. public static void main(String[] args) {
49. // 初始化智能燃气灶系统：最低火焰强度 50，最高火焰强度 80（任意单位），初始火焰强度 70，初始燃气流量 1.0 L/min
50. c100\_SmartGasStove stove = new c100\_SmartGasStove(50.0, 80.0, 70.0, 1.0);
52. // 测试用例
53. stove.updateFlameIntensity(45.0); // 低于最低阈值，增加燃气流量
54. stove.updateFlameIntensity(85.0); // 高于最高阈值，降低燃气流量
55. stove.updateFlameIntensity(65.0); // 在最佳范围内，保持燃气流量不变
56. }
57. }