**Debugging Report for Clock.java**

**Identified Bugs and Fixes**

**1. Constructor Validation**

**Bug**: The constructor allowed negative values for hours, minutes, and seconds, which could lead to unexpected behavior.

**Fix**: Modify the constructor to throw an IllegalArgumentException for any negative values.

public Clock(int hours, int minutes, int seconds) {

if (hours < 0 || minutes < 0 || seconds < 0) {

throw new IllegalArgumentException("Negative values not permitted");

}

this.hours = hours;

this.minutes = minutes;

this.seconds = seconds;

}

**2. addHour Rollover Bug**

**Bug**: The addHour() method incorrectly reset hours to 0 when hours >= 24. It should instead reset when hours == 23.

**Fix**: Adjust the rollover condition to check if (hours == 23).

public void addHour() {

if (hours == 23) {

hours = 0;

} else {

hours++;

}

}

**3. addMinute Rollover Bug**

**Bug**: The addMinute() method mistakenly checked seconds >= 60 instead of minutes for the rollover condition. This led to recursion and incorrect minute handling.

**Fix**: Correct the logic to check if (minutes == 59), then reset minutes to 0 and call addHour().

public void addMinute() {

if (minutes == 59) {

minutes = 0;

addHour();

} else {

minutes++;

}

}

**4. addSecond Method Implementation**

**Bug**: The Clock class lacked an addSecond method, making it incomplete for second-based time increments.

**Fix**: Implement an addSecond method to increment seconds, reset it at 59, and call addMinute() for rollover.

public void addSecond() {

if (seconds == 59) {

seconds = 0;

addMinute();

} else {

seconds++;

}

}

**5. get12HourFormat for AM/PM Representation**

**Bug**: The get12HourFormat() method may have failed to correctly display 12 AM and 12 PM, which are special cases.

**Fix**: Modify the method to handle 12-hour transitions, ensuring that 00 is displayed as 12 AM and 12:00 PM is displayed correctly.

public String get12HourFormat() {

String period = (hours >= 12) ? "PM" : "AM";

int hour12 = (hours == 0 || hours == 12) ? 12 : hours % 12;

return String.format("%d:%02d:%02d %s", hour12, minutes, seconds, period);

}

**6. Edge Case Testing and Validation**

**Bug**: Without specific edge case testing, certain time transitions, such as at midnight or minute/hour boundaries, could have unpredictable behavior.

**Fix**: Add comprehensive tests (as in clockTest.java) to ensure that edge cases are covered, verifying behavior at boundaries like 23:59:59, 00:00:00, and 12-hour transitions.

**OUTPUT SCREENSHOTS**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**