**Debugging Report for Clock.java**

**Identified Bugs and Fixes**

1. **Incorrect Condition in addMinute() Method**
   * **Bug**: The addMinute() method currently checks seconds >= 60 instead of minutes >= 60. This causes the method to incorrectly increment the seconds field rather than the minutes field, leading to inaccurate time representation.
   * **Original Code**:

public void addMinute() {

if (seconds >= 60) {

seconds = 0;

addMinute();

} else {

seconds++;

}

}

* + **Fix**: Update the condition to check if minutes == 59. This ensures that when minutes reach 59, they reset to 0, and an hour is added.
  + **Corrected Code**:

public void addMinute() {

if (minutes == 59) {

minutes = 0;

addHour();

} else {

minutes++;

}

}

1. **Direct Modification of hours in get12HourFormat()**
   * **Bug**: The get12HourFormat() method directly modifies the hours field, which can lead to unintended side effects by altering the Clock instance’s internal state. For example, calling get12HourFormat() would change the stored hours value unexpectedly.
   * **Original Code**:

public String get12HourFormat() {

if (hours == 0) {

hours = 12;

} else if (hours > 12) {

hours = hours - 12;

}

return hours + ":" + pad(minutes) + ":" + pad(seconds) + " " + getSuffix();

}

* + **Fix**: Use a temporary variable, displayHours, to store the calculated 12-hour representation without altering the original hours value.
  + **Corrected Code**:

public String get12HourFormat() {

int displayHours = hours;

if (displayHours == 0) {

displayHours = 12; // Midnight in 12-hour format

} else if (displayHours > 12) {

displayHours -= 12; // PM times in 12-hour format

}

return displayHours + ":" + pad(minutes) + ":" + pad(seconds) + " " + getSuffix();

}

1. **Improper Range Validation in Constructor**
   * **Bug**: The constructor for Clock only checks for non-negative values but does not validate whether they fall within the correct ranges (e.g., hours should be between 0 and 23, and minutes and seconds should be between 0 and 59). This can lead to invalid time values being accepted, causing incorrect behavior.
   * **Original Code**:

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

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

throw new IllegalArgumentException("Non-negative values not permitted");

}

this.hours = hours;

this.minutes = minutes;

this.seconds = seconds;

}

* + **Fix**: Add range checks for hours (0-23), minutes (0-59), and seconds (0-59). This ensures that only valid values are accepted.
  + **Corrected Code**:

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

if (hours < 0 || hours > 23 || minutes < 0 || minutes > 59 || seconds < 0 || seconds > 59) {

throw new IllegalArgumentException("Invalid time values. Ensure hours (0-23), minutes (0-59), and seconds (0-59).");

}

this.hours = hours;

this.minutes = minutes;

this.seconds = seconds;

}

1. **Potential Error in getSuffix() Method for AM/PM**
   * **Bug**: The getSuffix() method could produce incorrect results for edge cases like midnight (12:00 AM) and noon (12:00 PM). The method currently checks if hours < 12 to determine if it’s "AM" or "PM", which doesn’t account for these boundaries.
   * **Original Code**:

private String getSuffix() {

return (hours < 12) ? "AM" : "PM";

}

* + **Fix**: Adjust the logic to specifically handle cases for midnight (AM) and noon (PM), ensuring accuracy for boundary cases.
  + **Corrected Code**:

private String getSuffix() {

return (hours < 12 || hours == 24) ? "AM" : "PM";

}

1. **Incorrect Condition in pad() Method for Single-Digit Padding**
   * **Bug**: The pad() method had a condition that would not pad values less than 10 correctly. The code should ensure values less than 10 have a leading zero in the returned string.
   * **Original Code**:

private String pad(int value) {

return (value < 9) ? "0" + value : String.valueOf(value);

}

* + **Fix**: Adjust the condition to value < 10 to ensure that single-digit values (0-9) are padded correctly.
  + **Corrected Code**:

private String pad(int value) {

return (value < 10) ? "0" + value : String.valueOf(value);

}

**Summary of Fixes**

1. **In addMinute()**: Corrected if (seconds >= 60) to if (minutes == 59) for proper minute incrementation.
2. **In get12HourFormat()**: Used a temporary variable displayHours to avoid modifying the hours field.
3. **In Constructor**: Added validation to ensure time values are within valid ranges (hours 0-23, minutes and seconds 0-59).
4. **In getSuffix()**: Adjusted logic to handle specific cases for midnight (AM) and noon (PM).
5. **In pad()**: Updated the condition to value < 10 to ensure single-digit values are zero-padded correctly.

**SCREENSHOTS OF OUTPUT**

**A screenshot of a computer

Description automatically generated**