Clean-Code Prinzipien die berücksichtigt wurden und deren Nutzen:

1. Deskriptive Namensgebung: Die verwendeten Klassennamen, Methodennamen und Variablennamen sind aussagekräftig und beschreiben die Funktionen und den Zweck der Elemente.

2. Einrückung und Lesbarkeit: Der Code ist gut eingerückt und formatiert, was die Lesbarkeit verbessert und die Struktur des Codes klarer macht.

3. Kommentare: Es werden Kommentare verwendet, um die Funktionen und den Ablauf des Codes zu erklären. Dadurch wird das Verständnis erleichtert, insbesondere für Entwickler, die den Code später lesen oder warten müssen.

4. Verwendung von Leerzeilen: Der Code enthält Leerzeilen, um logische Abschnitte voneinander zu trennen und die Lesbarkeit zu verbessern.

5. Vermeidung von übermäßig langen Methoden: Die Methoden sind relativ kurz und erfüllen jeweils eine spezifische Aufgabe. Dadurch wird der Code besser lesbar, leichter wartbar und ermöglicht eine bessere Wiederverwendbarkeit.

6. Single Responsibility Principle (SRP): Die Klasse "WeatherController" hat eine klare Verantwortung, nämlich die Verarbeitung der Daten und die Kommunikation mit der View. Sie erfüllt das SRP, indem sie nur für eine Aufgabe zuständig ist.

7. Dependency Injection: Die Abhängigkeit des "WeatherController" von der "WeatherStation" wird über den Konstruktor injiziert. Dadurch wird die Abhängigkeit gelöst und die Klasse ist flexibler und leichter testbar.

8. Vermeidung von zu viele Übergabeparametern: In einigen Fällen werden ähnliche Operationen wie das Aktualisieren der Temperaturstatistiken (Minimum, Maximum, Durchschnitt) in separaten Methoden durchgeführt, um den Code übersichtlicher zu machen