

1. SEVERE:

- **Beschreibung:** Dies ist der höchste Schweregrad für eine Log-Nachricht. Meldungen auf diesem Level signalisieren schwerwiegende Probleme, die die Stabilität der Anwendung ernsthaft beeinträchtigen können.
- **Beispiel:** Ein kritischer Systemfehler, wie etwa ein Datenbankausfall oder eine nicht behandelbare Ausnahme, die zum Absturz der Anwendung führt.
- **Anwendungsfall:** SEVERE-Level-Nachrichten werden verwendet, wenn ein Teil der Anwendung abstürzt oder ein nicht wiederherstellbarer Fehler auftritt. Typischerweise werden diese Fehler von Entwicklern und Betreibern überprüft, um größere Probleme zu beheben.

Beispielcode:

```
logger.severe("Kritischer Fehler: Verbindung zur Datenbank verloren.");
```

2. WARNING:

- **Beschreibung:** Ein Log-Level, das auf eine potenziell gefährliche Situation hinweist, die möglicherweise Probleme verursacht, aber nicht sofort zu einem Fehler oder Absturz führt.
- **Beispiel:** Wenn eine Datei oder Ressource, auf die die Anwendung zugreifen möchte, nicht verfügbar ist, aber die Anwendung dennoch weiterhin funktionieren kann.
- **Anwendungsfall:** WARNING-Level-Nachrichten sind wichtig, um auf problematische Zustände hinzuweisen, die bald behoben werden müssen, bevor sie zu ernsteren Fehlern führen.
- **Beispielcode:**

```
logger.warning("Warnung: Verbindung zum Server ist instabil.");
```

3. INFO:

- **Beschreibung:** Dies ist der Standard-Log-Level für informative Nachrichten. INFO-Level-Nachrichten geben dem Benutzer oder Entwickler Hinweise auf den normalen Ablauf der Anwendung, wie z. B. Start- oder Endzeitpunkte bestimmter Operationen.
- **Beispiel:** Eine Nachricht darüber, dass eine Datei erfolgreich geladen wurde oder dass ein bestimmter Prozess abgeschlossen ist.
- **Anwendungsfall:** Verwende INFO-Nachrichten, um den regulären Ablauf der Anwendung aufzuzeichnen, z. B. wenn bestimmte Operationen wie das Einlesen von Daten oder das Starten von Diensten erfolgreich durchgeführt wurden.
- **Beispielcode:**

```
logger.info("Die Anwendung wurde erfolgreich gestartet.");
```

4. CONFIG:

- **Beschreibung:** Dieser Level wird verwendet, um Konfigurationsinformationen aufzuzeichnen. Dies sind Nachrichten, die nützliche Informationen über den Zustand der Anwendungskonfiguration bieten.
- **Beispiel:** Eine Nachricht, die die aktuellen Einstellungen oder Konfigurationsparameter protokolliert, wenn ein Dienst startet.
- **Anwendungsfall:** CONFIG-Level-Nachrichten werden protokolliert, um sicherzustellen, dass alle Systeme korrekt konfiguriert sind und mit den richtigen Parametern gestartet wurden. Dies hilft bei der Überprüfung der Systemumgebung.
- **Beispielcode:**

```
logger.config("Die Anwendung läuft im Debug-Modus.");
```

5. FINE:

- **Beschreibung:** Dieser Level wird verwendet, um detaillierte Debugging-Informationen zu protokollieren. FINE ist für Situationen gedacht, in denen du Informationen über den Ablauf der Anwendung auf niedrigerer Ebene brauchst, die für den normalen Betrieb nicht notwendig sind.
- **Beispiel:** Protokollierung der Schritte in einem Algorithmus oder der Start und das Ende einer Methode.
- **Anwendungsfall:** FINE-Level-Nachrichten sind für Entwickler hilfreich, um detaillierte Einblicke in den Ablauf der Anwendung zu erhalten. Diese Nachrichten werden in der Regel nicht in Produktionsumgebungen protokolliert, sondern eher während der Entwicklung oder bei Tests.
- **Beispielcode:**

```
logger.fine("Die Methode 'berechneErgebnis' wurde aufgerufen.");
```

6. FINER:

- **Beschreibung:** Dieser Level bietet noch detailliertere Informationen als FINE. Er wird verwendet, um tiefere Einblicke in den Ablauf des Programms zu gewinnen, insbesondere wenn du Fehler oder unerwartetes Verhalten aufdecken möchtest.
- **Beispiel:** Protokollierung des Zustands von Variablen während eines komplexen Algorithmus.
- **Anwendungsfall:** FINER-Level-Nachrichten werden verwendet, um auf sehr granularer Ebene nach Fehlern oder Optimierungspotenzial zu suchen, insbesondere während des Debuggings komplexer Probleme.
- **Beispielcode:**

```
logger.finer("Aktueller Wert der Variable 'x' ist: " + x);
```

7. FINEST:

- **Beschreibung:** Dies ist der niedrigste Level im Logger und bietet die detailliertesten Debugging-Informationen. FINEST-Nachrichten werden nur in extrem detaillierten Debugging-Sitzungen verwendet, wenn du den genauen Ablauf und die Ausführungsreihenfolge aller Teile der Anwendung verstehen musst.
- **Beispiel:** Protokollierung von jedem Schritt und jeder Operation in einem kritischen Algorithmus oder System.
- **Anwendungsfall:** FINEST-Level-Nachrichten sind für tiefe, oft performance-kritische Debugging-Sitzungen hilfreich, in denen es wichtig ist, jeden Schritt der Anwendung zu verfolgen.
- **Beispielcode:**

```
logger.finest("Betrete Schleife: i = " + i);
```

8. OFF:

- **Beschreibung:** Der OFF-Level deaktiviert das Logging vollständig. Sobald ein Logger auf den Level OFF gesetzt wird, werden keine Nachrichten mehr protokolliert.
- **Anwendungsfall:** Dies wird in Situationen verwendet, in denen Logging für eine bestimmte Komponente oder Anwendung deaktiviert werden soll, etwa in hoch optimierten Umgebungen, in denen die Log-Ausgabe überflüssig ist.

9. ALL:

- **Beschreibung:** Der ALL-Level aktiviert alle Log-Nachrichten, unabhängig von ihrem Schweregrad. Das bedeutet, dass alle Nachrichten, von SEVERE bis FINEST, protokolliert werden.
- **Anwendungsfall:** Dies wird normalerweise während der Entwicklungs- oder Debugging-Phase verwendet, um sicherzustellen, dass wirklich alle Informationen gesammelt werden.

Zusammenfassung der Levels in Bezug auf ihre Priorität:

- SEVERE (höchste Priorität)
- WARNING
- INFO
- CONFIG
- FINE
- FINER
- FINEST (niedrigste Priorität)
- OFF (kein Logging)
- ALL (alle Nachrichten loggen)