

# Seminar 4

Eine auf einem Server laufenden Anwendung schreibt in verschiedenen Dateien für Sicherheitsgründe mehrere Logs. Logs werden sehr oft verwendet, wenn man wissen soll, was während der Ausführung passiert ist. Es gibt verschiedene Typen von Logs, die in Dateien aufgenommen werden:

- SEVERE
- WARNING
- INFO
- CONFIG
- DEBUG

```
Logger.log(severity=severe, message="ssds",...)
```

Beispiel für eine JSON Log Datei:

```
[
  {timestamp: 2014-07-02 20:52:39, severity: DEBUG, "message":
  {"file":"HelloExample:19", "text": "This is debug : mkyong"}, visibility:[low, high]},
  {timestamp: 2014-07-02 20:52:39, severity: INFO, message: {file:HelloExample:23,
  text: This is info : mkyong}, visibility:[low]},
  {timestamp: 2014-07-02 20:52:39, severity: WARN, message: {file:HelloExample:26, text:
  This is warn : mkyong}, visibility:[high]},
  {timestamp: 2014-07-02 20:52:39, severity: ERROR, message: {text: This is error :
  mkyong}, visibility:[low]},
  {timestamp: 2014-07-02 20:52:39, severity: FATAL, message: {text: This is fatal :
  mkyong}, visibility:[low, medium]}
]
```

- Implementiere eine Klasse Message mit zwei Attribute: file (String) und text (String).
- Implementiere eine Klasse Log mit folgenden Attribute: timestamp (DateTime), visibility (List), severity (Enum) und message (Message).
- Implementiere eine Klasse LogParser. Die Klasse stellt eine statische Methode ParseLogFile(String path) bereit, die eine solche Log Datei (JSON Datei, wie oben) parst und eine Liste von Logs zurückgibt.
- Implementiere eine Klasse LogStatistics. Diese Klasse soll folgende Funktionalität ermöglichen:
  - Anzeige Logs
  - Sortieren von Logs nach Severity
- Implementiere die Klasse LogParser als Singleton (es darf nur eine einzige Instanz der Klasse geben).
- Wir haben bisher nur eine Klasse LogParser für JSON-Files implementiert, aber in der Tat muss man in der Lage sein, mit verschiedenen Arten von Files zu arbeiten: CSV, XML, TSV, etc. Schreibe eine Klasse, die das Factory Pattern implementiert und ermöglicht, verschiedene Arten von Parsern zu erstellen.