

Labor 4 – Logger Anwendung mit Java 8 (Deadline – 11te Woche – 12-16 Dezember 2016)

In unserem Tagen, eine große Anwendung, die läuft auf einem Server schreibt für Sicherheitsgründe mehrere Logs in verschiedene Dateien. Logs sind sehr oft verwendet, wenn man soll kennen was schlecht passiert in der Implementierung. Es gibt verschiedene Typen von Logs, die sind in Dateien aufgeschrieben:

- SEVERE – Wenn etwas die Anwendung kaputt gemacht hat;
- WARNING – Die Anwendung ist nicht kaputt, aber vielleicht etwas schlecht passiert;
- INFO – Statistische Infos (Benutzer Anmeldungsdatum, Benutzer Aktionen)
- CONFIG – Verbindungen zum Datenbank, Partnersysteme.
- DEBUG – Wenn man im Debug Modus ist.

Beispiel für ein Log Datei:

```
2014-07-02 20:52:39 DEBUG HelloExample:19 - This is debug : mkyong
```

```
2014-07-02 20:52:39 INFO HelloExample:23 - This is info : mkyong
```

```
2014-07-02 20:52:39 WARN HelloExample:26 - This is warn : mkyong
```

```
2014-07-02 20:52:39 ERROR HelloExample:27 - This is error : mkyong
```

```
2014-07-02 20:52:39 FATAL HelloExample:28 - This is fatal : mkyong
```

In diesem Labor, muss es eine Anwendung umsetzen, die macht folgendes:

1. Es parst ein Log Datei (Text Datei, wie oben) und bringt eine Liste von Logs zurück;
2. Anzeige die Logs bei Typ;
3. Anzeige der am häufigsten Log Meldungen für Type = SEVERE;
4. Anzeige die Logs, die in den letzten 10 Tagen geschrieben wurden;
5. Kategorisierung von Fehler (z.B. bekannte Fehler, unbekannter Fehler, für Go-Live relevant, etc.)

Implementierung Herausforderungen:

- Konsole oder GUI Anwendung mit Menu;
- OOP Struktur (verteilen im mehrere Packages; z.B. Model, Repository, Kontroller, App)
- JUnit Testen für alle Methoden von Repository, Kontroller
- **MUSS** Java 8 Funktionalität (Streams, Predicate, Lambda Expressions, Map And Reduce)
- **OHNE** for / while von Alte Java Versionen