

# Pflichtenheft: Lizenzmanagement für Dependencies

---

## Problemstellung

Bei der Auslieferung kommerzieller Software müssen **alle genutzten Bibliotheken** sowie **deren Lizenzen** mit ausgeliefert werden. Da jedes Softwareprojekt **unterschiedliche Formate** zur Verwaltung dieser Lizenzinformationen nutzt, ist eine einheitliche Lösung erforderlich. Zudem müssen die Informationen in unterschiedlichen Formaten **bereitgestellt** werden, beispielsweise als **HTML** für eine Website oder als Markdown für die Dokumentation der Software-Auslieferung.

Um diese Anforderungen zu erfüllen, soll eine Anwendung entwickelt werden, die Lizenzinformationen aus verschiedenen Dateiformaten einlesen, verarbeiten und in unterschiedlichen Ausgabeformaten darstellen kann.

---

## Zielsetzung

Die zu entwickelnde Anwendung soll:

- Lizenzinformationen aus verschiedenen Dateiformaten (DAT, JSON, XML) einlesen können.
- Die Abhängigkeiten und transitiven Dependencies der Libraries erfassen und analysieren.
- Die erfassten Informationen in unterschiedlichen Formaten ausgeben (z. B. HTML, Markdown, Konsole).
- Eine einfache Erweiterbarkeit für zukünftige Anpassungen bieten.
- Lizenzprüfungen durchführen und Warnungen ausgeben, falls unerlaubte Lizenzen gefunden werden.

---

## Funktionale Anforderungen

### 1. Einlesen der Lizenzinformationen

- Implementierung für das DAT-Format sowie ein weiteres Format nach Wahl (JSON oder XML).
- Definitionen für alle weiteren Formate bereitstellen, Implementierung ist nicht zwingend erforderlich.
- Verarbeitung von Informationen zu Libraries, Versionen, Lizenzen und deren Abhängigkeiten (transitive Dependencies).

### 2. Verarbeitung der Lizenzinformationen

- Speicherung der Daten in einer geeigneten Datenstruktur zur effizienten Verwaltung.
- Unterstützung einer hierarchischen Verarbeitung, um auch transitive Dependencies korrekt zu erfassen.

### 3. Ausgabe der Lizenzinformationen

- Bereitstellung der Informationen in verschiedenen Formaten:
  - **Konsolenausgabe**
  - **Dateiausgabe:**
    - HTML
    - Markdown
- Unterstützung folgender Ausgabe-Sortierungen:
  - **Tree-Ausgabe:** Hierarchisch nach Libraries sortiert.
  - **Flat-Ausgabe:** Alphabetisch nach Libraries sortiert.
  - **Flat-Ausgabe nach Lizenztyp:** Gruppierung aller Libraries mit derselben Lizenz.
- Mindestens eine Kombination aus Ausgabeformat, Destination und Sortierung muss vollständig implementiert werden. Für weitere Kombinationen sind Definitionen ausreichend.

### 4. Lizenzprüfung

- Automatische Überprüfung der Lizenzen beim Einlesen der Daten.
- Ausgabe einer Warnung, falls eine unerlaubte Lizenz (z. B. GPL) oder keine Lizenzangabe gefunden wird.

### 5. Main-Klasse & CLI-Parameter

- Die Anwendung wird über die Kommandozeile mit zwei Parametern aufgerufen:
  - `plugin` : Dateiformat der Lizenzinformationen (`dat`, `json`, `xml`)
  - `outputFormat` : Format der Ausgabe (`tree`, `flat_lib`, `flat_license`)

---

## Technische Anforderungen

- **Objektorientierte und modulare Architektur**
- **Nutzung bewährter Design-Patterns**
- **Erweiterbarkeit** für zukünftige Anpassungen
- **Fehlertoleranz / Fehlerhandling**
- **Technologie-Stack:**
  - Reines Java
    - keine externen Bibliotheken
    - keine Frameworks
  - Ausnahmen:
    - GSON

## Abgabe

- Mögliche Formate:
  - ZIP-Archiv
  - Link zum GitHub Clone
  - GitHub Pull Request
- Inhalt:
  - Quellcode der Anwendung
  - Kurz-Dokumentation
  - README-Datei mit Anweisungen zur Nutzung

*Die Aufgabe ist angesetzt auf ca. 5 Stunden. Die eingereichte Lösung muss selbstständig erarbeitet worden sein.*

Viel Erfolg!