



# **Administratordokumentation**

Litera Book Catalog

**Dokumentenversion:**  
**2.0 – Stand: 03.06.2025**

## Inhaltsverzeichnis

<b>1. Allgemeine Information.....</b>	<b>3</b>
<b>2. Systemanforderungen.....</b>	<b>3</b>
<b>3. Installationsanleitung für den Litera Book Catalogue .....</b>	<b>5</b>
<b>4. Konfiguration und Anpassungsmöglichkeiten .....</b>	<b>5</b>
<b>5. Fehlerbehandlung und Logs.....</b>	<b>7</b>
<b>6. Backup- und Wiederherstellungsmöglichkeiten.....</b>	<b>8</b>
<b>7. Wartungshinweise und Updates .....</b>	<b>9</b>
<b>8. Kontaktperson:.....</b>	<b>10</b>

## 1. Allgemeine Information

Die Desktop-Anwendung **Litera Book Catalogue** wurde entwickelt, um sowohl Administratoren als auch Anwendern eine komfortable und effiziente Verwaltung eines Buchkatalogs zu ermöglichen. Ziel der Software ist die strukturierte Erfassung, Verwaltung und Präsentation von Büchern sowie deren Rezensionen. Hauptfunktionen sind eine schnelle und intuitive Suchfunktion nach Buch-Titeln und Filtern, eine detaillierte Anzeige aller relevanten Buchinformationen sowie die Möglichkeit, direkt Rezensionen hinzuzufügen und zu verwalten. Die Benutzeroberfläche ist übersichtlich gestaltet, um eine einfache Navigation zu gewährleisten und den Administrationsaufwand gering zu halten.

Um die Anwendung **Litera Book Catalogue** effizient und störungsfrei nutzen zu können, sollten die untenstehenden Systemvoraussetzungen beachtet werden.

Diese Desktop-Software ist für die eigenständige Einrichtung und Ausführung auf lokalen Rechnern konzipiert. Bei etwaigen Schwierigkeiten wenden Sie sich bitte an unser Support-Team oder kontaktieren Sie direkt Ihre zuständige Ansprechperson per E-Mail.

## 2. Systemanforderungen

### Technische Voraussetzungen

Um die reibungslose Installation, Ausführung und Wartung der Java-Desktop-Anwendung **Litera Book Catalogue** sicherzustellen, sind folgende technische Rahmenbedingungen einzuhalten:

### Java-Laufzeitumgebung

Die Anwendung basiert auf der Programmiersprache Java und benötigt daher eine entsprechende Laufzeitumgebung:

- **Java-Version:** Erforderlich ist **Java SE 11 oder höher**, idealerweise eine LTS-Version (Long Term Support), z. B. OpenJDK 11 oder Oracle JDK 11.
- **Installation:** Die Java-Umgebung kann über die offizielle Webseite von Oracle oder alternativ über Anbieter wie [Adoptium](https://adoptium.net/) bezogen und installiert werden.
- Nach erfolgreicher Installation sollte die Umgebungsvariable `JAVA_HOME` korrekt gesetzt sein.

## Abhängigkeiten (Libraries)

Die Anwendung nutzt externe Bibliotheken, die über das Build-Tool automatisch eingebunden werden:

- **Jackson Databind (ab Version 2.15.2):** Dient zur Serialisierung und Deserialisierung von Objekten, insbesondere für das Einlesen und Speichern von Buch- und Rezensionsdaten.
- **JUnit 5:** Wird ausschließlich für automatisierte Tests während der Entwicklungs- und Wartungsphasen verwendet. Für Endnutzer ist keine direkte Nutzung erforderlich.

## Build-Tool

Die Anwendung wird mit dem Build-Management-Werkzeug **Apache Maven** gepflegt und kompiliert:

- **Empfohlene Version:** Maven 3.6.x oder neuer
- **Funktion:** Verwaltung von Abhängigkeiten, Kompilierung des Codes, Paketierung der Anwendung als JAR-Datei, Ausführung von Tests, und ggf. Obfuskation (Codeverschleierung mit ProGuard).

## Systemanforderungen (empfohlene Hardware/Software)

Die Software ist ressourcenschonend und kann auf gängigen Betriebssystemen installiert werden:

- **Betriebssysteme:**
  - Windows 10 oder 11
  - macOS (ab Version 10.14 Mojave)
  - gängige Linux-Distributionen (z. B. Ubuntu, Fedora)

### Hardware-Empfehlungen:

- **Arbeitsspeicher (RAM):** Mindestens 2 GB (4 GB oder mehr empfohlen für parallele Nutzung anderer Programme)
- **Festplattenspeicher:** Mindestens 200 MB verfügbarer Speicherplatz für die Anwendung und lokale Datendateien
- **Bildschirmauflösung:** Mindestens 1024 × 768 Pixel, idealerweise Full HD für optimale Darstellung der Benutzeroberfläche

## 3. Installationsanleitung für den Litera Book Catalogue

### 1. Voraussetzungen

Stellen Sie sicher, dass Java JDK 11 auf Ihrem Computer installiert ist.

→ Wenn Sie es noch nicht haben, können Sie es hier herunterladen:

<https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>

### 2. Anwendung herunterladen

Entpacken Sie den Ordner LiteraBookCatalogue (z. B. aus einer ZIP-Datei) an einen beliebigen Ort, z. B. auf den Desktop.

### 3. Anwendung starten

1. Öffnen Sie den entpackten Ordner.
2. Öffnen Sie darin den Unterordner final.
3. Doppelklicken Sie auf die Datei **buchkatalog-1.0.0-all-obf.jar**.
4. Die Anwendung startet sofort – es sind keine weiteren Installationsschritte erforderlich.

### 4. Falls das Doppelklicken auf die JAR-Datei nicht funktioniert:

1. Öffnen Sie ein Terminal (z. B. auf Windows: cmd)
2. Navigieren Sie zum Ordner target:  
**cd Pfad\zu\LiteraBookCatalogue\final**
3. Starten Sie die Anwendung manuell mit:  
**java -jar buchkatalog-1.0.0-all-obf.jar**

## 4. Konfiguration und Anpassungsmöglichkeiten

Die Anwendung **Litera Book Catalogue** lässt sich in mehreren Bereichen an individuelle Anforderungen und Systemumgebungen anpassen. Nachfolgend werden die wichtigsten Konfigurationsmöglichkeiten erläutert.

### 1. Speicherpfade konfigurieren

Die Speicherorte für Buch- und Rezensionsdaten sind im Quellcode der Datei StorageService.java festgelegt. Die beiden relevanten Konstanten befinden sich am Anfang der Klasse:

```
private static final String BOOKS_FILE = "src/main/resources/books_short.json";
```

```
private static final String REVIEWS_FILE = "src/main/resources/reviews.json";
```

## Standard-Speicherung

Standardmäßig werden die Daten im Verzeichnis `src/main/resources/` des Projekts gespeichert. Diese Pfade sind relativ zum Projektverzeichnis angegeben.

## Anpassung durch Administratoren

Wenn Sie die Daten z. B. auf einem Netzlaufwerk oder in einem externen Ordner speichern möchten, können Sie die Pfade folgendermaßen anpassen:

```
private static final String BOOKS_FILE = "D:/Daten/Litera/books.json";
```

```
private static final String REVIEWS_FILE = "D:/Daten/Litera/reviews.json";
```

Alternativ ist auch ein relativer Pfad zum Benutzerverzeichnis möglich:

```
private static final String BOOKS_FILE = System.getProperty("user.home") + "/litera/books.json";
```

```
private static final String REVIEWS_FILE = System.getProperty("user.home") + "/litera/reviews.json";
```

## Wichtig:

- Achten Sie darauf, dass die Pfade sowohl **zur Lesezeit (ladeBuecher/ladeRezensionen)** als auch **zur Schreibzeit (speichereRezension)** gültig und beschreibbar sind.
- Bei einer Änderung ist ein Rebuild des Projekts erforderlich, damit die neue Konfiguration übernommen wird.

## Anpassung der Benutzeroberfläche

Die grafische Oberfläche der Anwendung basiert auf dem Java Swing-Framework. Gestaltungselemente wie Farben, Schriftgrößen und Icons können über zentrale Ressourcen individuell angepasst werden:

- **Farben und Schriftarten:** Werden in Form von Konstanten (z. B. Color BACKGROUND\_COLOR, Font DEFAULT\_FONT) in den UI-Klassen definiert und können dort geändert werden.
- **Icons und Bilder:** Im Ressourcenordner (images/) hinterlegte Bilder (z. B. für Buttons oder Banner) lassen sich durch eigene Grafiken im gleichen Format ersetzen.
- **Layout-Anpassungen:** Kleinere strukturelle Änderungen an der Anordnung von UI-Elementen können direkt in den Panel-Klassen (StartPanel, BookListPanel, etc.) vorgenommen werden.

So lässt sich die Anwendung bei Bedarf an ein Unternehmensdesign (Corporate Identity) anpassen oder optisch auf bestimmte Zielgruppen ausrichten.

## 5. Fehlerbehandlung und Logs

Die Anwendung **Litera Book Catalogue** verfügt über grundlegende Mechanismen zur Fehlerausgabe und bietet darüber hinaus die Möglichkeit zur Integration von Logging-Frameworks für eine erweiterte Fehleranalyse.

### 1. Fehlerausgabe

Im Standardbetrieb werden auftretende Fehler und Ausnahmen direkt über die **Kommandozeile (Terminal)** ausgegeben. Dies ermöglicht eine schnelle Sichtkontrolle während der Laufzeit und ist insbesondere für Entwickler sowie Administratoren im Test- und Wartungsumfeld hilfreich.

- **Beispiel:** Fehler beim Laden von Dateien oder Parsing-Probleme in der JSON-Verarbeitung werden im Terminal protokolliert.
- **Empfehlung:** Beim Auftreten von Fehlverhalten sollte die Konsole systematisch auf Ausgaben überprüft werden.

### 2. Logging (optional)

Für eine dauerhafte und strukturierte Protokollierung kann die Anwendung mit einem Logging-Framework ausgestattet werden. Dies ist aktuell nicht aktiv, jedoch vorbereitet:

- **Möglichkeit:** Integration von Bibliotheken wie java.util.logging, Log4j oder SLF4J.

- **Ort der Integration:** Die Einstiegsklasse `ApplicationWindow.java` eignet sich als zentrale Stelle zur Initialisierung eines Loggers.
- **Beispielhafte Log-Inhalte:** Benutzerinteraktionen, Ladeprozesse von Dateien, Ausnahmen mit Stacktraces.

## 6. Backup- und Wiederherstellungsmöglichkeiten

Die Anwendung **Litera Book Catalogue** speichert alle Buch- und Rezensionsdaten lokal auf dem Dateisystem. Um einen Datenverlust durch Systemfehler, versehentliches Löschen oder Hardwareausfälle zu vermeiden, wird eine regelmäßige Sicherung der Daten empfohlen.

### Backup-Erstellung

Die einfachste Methode zur Sicherung besteht darin, das Verzeichnis zu kopieren, in dem sich die Anwendungsdaten befinden.

#### Empfehlung:

- Erstellen Sie tägliche oder wöchentliche Backups – abhängig von der Nutzungsintensität.
- Speichern Sie die Sicherungen auf einem externen Medium oder Netzlaufwerk, um zusätzliche Ausfallsicherheit zu gewährleisten.
- Benennen Sie Sicherungen mit Datum, z. B. `backup_2025-05-09/`, um Wiederherstellungen zeitlich einordnen zu können.

### Wiederherstellung

Im Falle eines Datenverlustes oder fehlerhafter Daten genügt es, das Backup wieder an die ursprüngliche Position zurückzukopieren:

1. **Anwendung beenden** (um Dateizugriffe zu vermeiden).
2. Das beschädigte oder gelöschte Verzeichnis löschen oder umbenennen.
3. Die zuvor erstellte Sicherung an dieselbe Stelle kopieren.
4. **Anwendung erneut starten** – die Daten werden automatisch geladen, sofern die Struktur korrekt ist.



## Automatisierung (optional)

Fortgeschrittene Benutzer oder Administratoren können Skripte zur **automatischen Datensicherung** einrichten (z. B. mit Bash, PowerShell oder über Cronjobs), um menschliche Fehler auszuschließen.

## 7. Wartungshinweise und Updates

Eine regelmäßige Wartung der Anwendung **Litera Book Catalogue** gewährleistet einen stabilen Betrieb, reduziert Sicherheitsrisiken und stellt sicher, dass neue Funktionen effektiv genutzt werden können.

### Aktualisierung der Java-Umgebung und Build-Tools

- **Java Runtime (JDK):**  
Überprüfen Sie regelmäßig, ob neue Versionen von Java SE verfügbar sind (idealerweise LTS-Versionen wie Java 17 oder 21). Neue Releases beinhalten Sicherheitsverbesserungen und Leistungsoptimierungen.
- **Apache Maven:**  
Auch das verwendete Build-Tool sollte auf dem aktuellen Stand gehalten werden. Verwenden Sie den Befehl `mvn -v`, um Ihre derzeitige Version zu prüfen. Updates finden Sie auf der offiziellen [Maven-Website](#).

### Anwendung aktualisieren

- **Neues Release installieren:**  
Wenn neue Versionen der Anwendung bereitgestellt werden, laden Sie diese aus dem Projekt-Repository herunter, führen Sie `mvn clean install` aus und ersetzen Sie die alte .jar-Datei durch die neue Version.

### Performance-Monitoring und Wartung

- **Ressourcenverbrauch prüfen:**  
Überwachen Sie CPU- und Speicherverbrauch der Anwendung bei längerer Nutzung.

- **Reaktionszeiten beobachten:**

Sollte die Anwendung deutlich langsamer reagieren, kann dies auf eine Fragmentierung der Datendateien oder auf eine zu große Datenmenge hinweisen.

- **Logs auswerten:**

Analysieren Sie regelmäßig die Terminal-Ausgaben und (wenn aktiviert) die Log-Dateien, um Fehlerquellen frühzeitig zu identifizieren.

### **Empfehlung für Wartungszyklen**

- **Monatlich:** Systemupdates und Abhängigkeiten prüfen.
- **Vierteljährlich:** Backups kontrollieren, Logs durchsehen, Anwendung testen.
- **Bei neuen Releases:** Anwendung ersetzen und neue Features dokumentieren.

## **8. Kontaktperson:**

Marija Ratnikova

**E-Mail:** marija.ratnikova@example.com

**Supportzeiten:** Montag bis Freitag, 9:00 – 17:00 Uhr

Bitte geben Sie bei Anfragen möglichst genaue Informationen zur Systemumgebung, Fehlermeldung und zur durchgeführten Aktion an, um eine schnelle Bearbeitung zu ermöglichen.

Firma NOVATECH SOLUTIONS

Technologiepark 15,

22303 Hamburg, Deutschland

Ansprechpartner: Anna Fischer

Kontaktdaten:

Telefon: +49 40 1234567

E-Mail: info@novatech-solutions.de

