**Administratorendokumentation**

**Ein Bild, das Text, Uhr, Schrift, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.**

**Vokabeltrainer-App**  
*Dokumentenversion:*  
1.0 – Stand: 09.05.2025

Inhaltsverzeichnis

[**Inhaltsverzeichnis** **Fehler! Textmarke nicht definiert.**](#_Toc197694274)

[1. Allgemeine Informationen 3](#_Toc197694275)

[2. Systemanforderungen 4](#_Toc197694276)

[**Java-Laufzeitumgebung** 4](#_Toc197694277)

[**Abhängigkeiten** 4](#_Toc197694278)

[**Build-Tool** 4](#_Toc197694279)

[**Empfohlene Systemumgebung** 4](#_Toc197694280)

[3. Installation 5](#_Toc197694281)

[4. Start der Anwendung 6](#_Toc197694282)

[**Manuell** 6](#_Toc197694283)

[java -jar target/vokabeltrainer-1.0.0.jar 6](#_Toc197694284)

[**Über Skript** 6](#_Toc197694285)

[start.bat → java -jar target/vokabeltrainer-1.0.0.jar 6](#_Toc197694286)

[**macOS/Linux:** 6](#_Toc197694287)

[start.sh → java -jar target/vokabeltrainer-1.0.0.jar 6](#_Toc197694288)

[5. Konfigurationsmöglichkeiten 7](#_Toc197694289)

[**Speicherpfade anpassen** 7](#_Toc197694290)

[**Benutzeroberfläche** 7](#_Toc197694291)

[6. Fehlerbehandlung und Logs 8](#_Toc197694292)

[**Fehlerausgabe** 8](#_Toc197694293)

[**Optionales Logging** 8](#_Toc197694294)

[7. Backup & Wiederherstellung 9](#_Toc197694295)

[**Backup** 9](#_Toc197694296)

[**Wiederherstellung** 9](#_Toc197694297)

[8. Wartung & Updates 10](#_Toc197694298)

[**Java & Gradle** 10](#_Toc197694299)

[**Anwendung aktualisieren** 10](#_Toc197694300)

[**Wartungsintervalle** 10](#_Toc197694301)

[9. Kontakt 11](#_Toc197694302)

# 1. Allgemeine Informationen

Die Vokabeltrainer-App ist eine leichtgewichtige Java-Anwendung zur Verwaltung und Durchführung individueller Lerneinheiten für Vokabeln. Sie erlaubt es den Nutzern, eigene Karten zu erstellen und mithilfe eines Bewertungssystems zu lernen.

Die Administration umfasst die Installation, Konfiguration, Datensicherung sowie das Management von Fehlern und Benutzeranfragen.

# 2. Systemanforderungen

**Java-Laufzeitumgebung**

* **Version:** Java SE 21
* **Empfohlen:** OpenJDK 11 (LTS)
* **JAVA\_HOME:** korrekt gesetzt

**Abhängigkeiten**

* Jackson Databind (z.B. zur JSON-Verarbeitung)
* Apache Commons IO (für Dateizugriffe)

**Build-Tool**

* **Gradle:** Version 8.12 oder höher

**Empfohlene Systemumgebung**

* **Betriebssystem:** Windows 10/11, macOS ≥ 10.14, Linux (Ubuntu/Fedora)
* **RAM:** ≥ 2 GB (4 GB empfohlen)
* **Speicherplatz:** ≥ 100 MB
* **Auflösung:** Mind. 1024 × 768

# 3. Installation

1. Java Version 21 installieren
2. Gradle installieren
3. Projekt in gewünschten Ordner entpacken (z. B. C:\VokabelTrainer\)
4. Im Terminal ins Projektverzeichnis navigieren:  
     cd C:\VokabelTrainer\
5. Anwendung bauen:  
     gradle jlinkzip
6. Die Zip Datei liegt im build Ordner. Die Ausführbare Datei ist aber in build/Pentalingo/bin-Ordner abgelegt, mit dem Namen Pentalingo.bat.

# 4. Start der Anwendung

**Manuell**

bin/Pentalingo.bat

### **Über Skript**

**Windows:**

.\bin\Pentalingo.bat

# 5. Konfigurationsmöglichkeiten

### **Speicherpfade anpassen**

* Zu finden in StorageService.java
* Standard: Speicherung im lokalen Projektverzeichnis
* Anpassbar auf Netzlaufwerke oder benutzerdefinierte Pfade

### **Benutzeroberfläche**

* FX-basiertes GUI
* Farben, Schriftarten und Icons anpassbar in UI-Klassen
* Layout-Anpassungen in SettingsScreen.java, DeckDisplayScreen.java etc.

# 6. Fehlerbehandlung und Logs

### **Fehlerausgabe**

* Standard: Konsolenausgabe (z. B. bei JSON-Parsing oder Datei-I/O)

### **Optionales Logging**

* Integration über SLF4J oder Log4j möglich
* Konfiguration in Main.java

# 7. Backup & Wiederherstellung

### **Backup**

* Sicherung der Datenverzeichnisse (z. B. data/decks.json, data/settings.json)
* Empfehlung: regelmäßige Backups mit Zeitstempel
* Speicherort: externes Medium oder Netzwerkfreigabe

### **Wiederherstellung**

1. Anwendung beenden
2. Verzeichnisse ersetzen
3. Anwendung neu starten

# 8. Wartung & Updates

### **Java & Gradle**

* Java SE regelmäßig aktualisieren (z. B. neuer als Version 21)
* Gradle-Version updaten mit gradle […]

### **Anwendung aktualisieren**

* Neue Version mit gradle install bauen
* Alte .jar ersetzen

### **Wartungsintervalle**

* **Monatlich:** System- & Toolupdates
* **Vierteljährlich:** Backup-Kontrolle, Log-Analyse
* **Bei Release:** Austausch der App-Dateien

# 9. Kontakt

**Support-Anfrage an:**  
**Anna Fischer**  
**Firma:** DresdenBytes GmbH  
**E-Mail:** kontakt@dresdenbytes.de  
**Tel.:** +49 123 456789  
**Supportzeiten:** Mo–Fr, 9:00–17:00 Uhr