



# Arbeitsauftrag Kompetenzcheck Python

KMS+/Python/KC\_06



## Ziel des Arbeitsauftrages:

Unser Kunde, die Musikschule "EasyChords", hat uns beauftragt, eine Akkord-Management-Applikation zu entwickeln, die ihren Schülern das Erlernen verschiedener Akkorde und Songstrukturen erleichtern soll. Die Applikation soll eine benutzerfreundliche Oberfläche bieten, Songs mit ihren Akkordfolgen darstellen und eine interaktive Lernumgebung für Musiktheorie bieten.



## Kontext:

Erstellen Sie ein Programm mit einer grafischen Benutzeroberfläche (GUI), das Songs und deren Akkorde aus einer SQL-Datenbank laden, präsentieren und interaktiv mit Akkorden arbeiten kann. Die Anwendung soll folgende Funktionen bieten:

1. Implementierung der Datenbankbindung und Datenverarbeitung
2. Erstellung einer benutzerfreundlichen GUI mit zwei Hauptbereichen
3. Implementierung der Song-Auswahl und Akkord-Anzeige
4. Implementierung eines interaktiven Akkord-Generators

Bevor Sie mit dem Programmieren beginnen, erstellen Sie mit einem UML-Tool Ihrer Wahl (z.B. draw.io) ein Ablaufdiagramm des Programms. Dieses Diagramm soll Ihnen helfen, die Struktur Ihres Codes zu planen und zu visualisieren.



## Detaillierte Anforderungen:

### **Zu Punkt 1: Implementieren Sie die Datenbankbindung und Datenverarbeitung**

- Stellen Sie eine Verbindung zur bereitgestellten SQL-Datenbank "song\_chords\_db" her
- Implementieren Sie Funktionen zum Auslesen der Songs und deren Akkordfolgen
- Stellen Sie eine sinnvolle Datenstruktur für die Verwaltung der Songs und Akkorde bereit
- Implementieren Sie eine Fehlerbehandlung für die Datenbankbindung

### **Zu Punkt 2: Erstellen Sie eine benutzerfreundliche GUI mit zwei Hauptbereichen**

- Entwickeln Sie eine GUI, die in zwei Hauptbereiche unterteilt ist:
  - Oberer Bereich: Song-Auswahl und Akkordfolgen-Anzeige
  - Unterer Bereich: Interaktiver Akkord-Generator
- Sorgen Sie für eine klare visuelle Trennung der Bereiche
- Verwenden Sie die vorgegebene Farbpalette: #ECEBDE, #C1BAA1, #D7D3BF, #ECEBDE
- Gestalten Sie die Benutzeroberfläche übersichtlich und intuitiv bedienbar

### **Zu Punkt 3: Implementieren Sie die Song-Auswahl und Akkord-Anzeige**

- Erstellen Sie ein Dropdown-Menü zur Auswahl der verfügbaren Songs
- Zeigen Sie die vollständige Akkordfolge des ausgewählten Songs an
- Implementieren Sie eine automatische Aktualisierung der Akkordanzeige bei Songwechsel
- Stellen Sie sicher, dass die Anzeige auch bei längeren Akkordfolgen übersichtlich bleibt

### **Zu Punkt 4: Implementieren Sie einen interaktiven Akkord-Generator**

- Erstellen Sie eine Reihe von interaktiven Buttons für die Grundtöne (C bis H/B)
- Implementieren Sie Buttons für verschiedene Akkordtypen (Dur, Moll, 7, sus2, sus4)
- Gestalten Sie die Buttons optisch ansprechend
- Zeigen Sie für den ausgewählten Akkord die zugehörigen Töne an (z.B. C Dur: C - E - G)
- Implementieren Sie die Logik zur Berechnung der Akkordtöne basierend auf Musiktheorie



## Zusätzliche Anforderungen:

- Stellen Sie sicher, dass die Anwendung robust gegen Fehleingaben und Verbindungsprobleme ist
- Achten Sie auf eine konsistente Farbgebung und ein ansprechendes Design
- Verwenden Sie moderne Tkinter-Widgets für eine zeitgemäße Darstellung

### Erlaubte Hilfsmittel

Es ist den Teilnehmenden erlaubt, zur Erfüllung der Anforderungen das Internet bzw. eigene Unterlagen zu benutzen. **NICHT ERLAUBT ist ChatGPT od. eine andere KI.** Alle in der jeweiligen Programmiersprache mitgelieferten Bibliotheken/Konstrukte sowie externe Bibliotheken, die zur Erfüllung notwendig sind und nicht durch einen anderen Teilnehmenden erstellt wurden, sind erlaubt.

### Abgabe:

Erstellen Sie ein neues Repository auf Github und laden Sie Ihre Trainer:innen dazu ein. zusätzlich zippen Sie den gesamten Programmordner und laden ihn auf Moodle hoch

### **Abgabezeitraum: 14.00 - 14.15 Uhr auf Moodle**

Zippen Sie Ihren Programmordner und laden ihn auf Moodle hoch.

**Die Abgabe öffnet und schließt automatisch. Sollte in diesem Zeitraum keine Abgabe erfolgen, wird der Kompetenzcheck automatisch mit nicht bestanden gewertet.**



### **Arbeitszeit:**

**Die Zeitvorgabe zur Erarbeitung des Arbeitsauftrages ist täglich von 07.00 - 14.00 Uhr.**

Sie teilen sich Ihre Pausenzeiten selbst ein und besprechen Ihre Pausen im Vorhinein mit dem Trainerteam! Wenn Sie eine Mittagspause machen wollen, halten Sie sich bitte an die Zeitvorgaben des SZF derzeit, 11.30 -12.00 Uhr und teilen Sie das dem Trainerteam im Voraus mit. Die Mittagspause wird nicht auf die Arbeitszeit eingerechnet.

## **Allgemeine Beurteilungskriterien**

Bei der Beurteilung der Umsetzung des Arbeitsauftrages in ein sind zwei Kriterien wichtig:

### **Funktionalität**

Das Software-Programm soll die in der Angabe stehenden Anforderungen durch die Bearbeitung von möglichen Eingabedaten bzw. durch die Ausführung von Funktionen/Algorithmen lösen, sodass das gewünschte Endergebnis, welches in der jeweils geforderten Form samt den korrekten Daten erbracht werden soll, erzielt wird. *Hierbei wird die technische Umsetzung außer Acht gelassen. Es zählt nur der Vergleich des erbrachten Endresultates zu den gegebenen Anforderungen.*

### **Technische Umsetzung**

Als Software Entwickler ist es nicht nur wichtig, das gewünschte Ergebnis zu erzielen. Es ist auch von essenzieller Bedeutung, dass die technische Umsetzung gewissen konzeptionellen und strukturellen Anforderungen entspricht. Der Code oder das Schema wird dadurch testbarer, übersichtlicher/lesbarer und auch bei Fehlerbehebungen wartbarer. Des Weiteren kann zum Beispiel durch den Einsatz von technischen Konstrukten in Software-Lösungen (Schleifen, Entscheidungen, ...) möglicherweise auch der Verbrauch des benötigten Speicherbedarfs minimiert bzw. eine effizientere(schnellere) Ausführung des Programms erzielt werden.

Hierbei geht es **nicht um das erzielte Endergebnis**, sondern um die Analyse der technischen (Teil-)Umsetzung der Anforderungen.



## Bewertungen

Die Bewertung Ihrer Arbeitsaufträge erfolgt durch das begleitende Trainerteam direkt im Anschluss an Ihre Präsentation anhand der festgelegten Bewertungskriterien. Für eine positive Beurteilung ist eine Mindestleistung von 60 % erforderlich. Sollte ein Arbeitsauftrag diese Vorgabe nicht erfüllen, wird der gesamte Kompetenzcheck als nicht bestanden gewertet. Im Falle eines zweiten Antritts müssen lediglich die negativ bewerteten Arbeitsaufträge wiederholt werden.

## Bewertungskriterien:

Planung	Max. 10 Pkt
Codequalität	Max. 20 Pkt
Funktionalität	Max. 20 Pkt
Usability	Max. 10 Pkt
Struktur	Max. 10 Pkt
Clean Coding / Konventionen	Max. 15 Pkt
Präsentation	Max. 15 Pkt
<b>Gesamt</b>	<b>Max. 100 Pkt</b>

**Erreicht wurden:**                      / 100 Pkt

## Bewertungsschlüssel:

0% – 59 % - Nicht bestanden

60% – 89% - Bestanden

90% - 100% - Mit Auszeichnung bestanden