

## Checkliste: LISTEN

	Ich kann ...	Beispiele
1	... Klassendiagramme, Objektdiagramme, Struktogramme und Sequenzdiagramme zeichnen und vorgegebene Diagramme in Code umschreiben und umgekehrt.	<p>Eine Musikwiedergabe App eines Smartphones soll die Wiedergabe einer Reihe von Titeln anhand einer Liste ermöglichen. Zu jedem Titel sollen Interpret, Name des Songs und die Länge des Titels in Sekunden gespeichert werden. Die Playliste soll es ermöglichen, dass man Titel am Ende der Playliste einfügen kann und für das Abspielen des nächsten Titels eine Methode vorhanden ist, die den ersten Titel der Liste zurückgibt.</p> <p>Zeichnen Sie ein Klassendiagramm, welches eine mögliche Implementierung der Liste darstellt.</p>
2	... verschiedene Implementierungen der Liste bewerten und erläutern wie man eine Liste mit Feld in eine dynamische Liste umbaut.	<p>Ein Entwickler stellt folgende Implementierung der Playliste von (1) vor:</p> <div style="display: flex; align-items: center; justify-content: center; margin: 10px 0;"> <div style="border: 1px solid black; padding: 5px; width: 250px; background-color: #ffff00;"> <p style="text-align: center; margin: 0;"><b>PLAYLISTE</b></p> <hr/> <p style="text-align: center; margin: 0;">TITEL[] titelliste int anzahl</p> <hr/> <p style="margin: 0;">PLAYLISTE() void Einfuegen(TITEL) TITEL Entfernen() void TitellisteAusgeben()</p> </div> <div style="margin: 0 20px; text-align: center;">enthält &gt;</div> <div style="border: 1px solid black; padding: 5px; width: 250px; background-color: #ffff00;"> <p style="text-align: center; margin: 0;"><b>TITEL</b></p> <hr/> <p style="margin: 0;">String interpret String name int laengeSekunden</p> <hr/> <p style="margin: 0;">TITEL(String i,String n, int l) void TitelInformationAusgeben()</p> </div> </div> <p>a) Nennen Sie zwei Probleme, die die vorgeschlagene Implementierung hat. Gehen Sie davon aus, dass die Playliste sehr lang sein kann.</p> <p>b) Erklären Sie, welche Teile der Liste oben man umbauen muss, um eine dynamische Liste zu bekommen.</p>
3	...darstellen, wie Einfügen und Entfernen in der Liste ablaufen.	Schreiben Sie in Pseudocode die Methoden „Einfuegen“ und „Entfernen“ ihrer Playliste.
4	... Struktur und Datenbereich der Liste gegeneinander abgrenzen und entsprechende Kommunikationswege und Klassendiagramme zeichnen.	<p>a) Stellen Sie dar, welche Bereiche Ihres Klassendiagrammes zum Datenbereich der Liste gehören und welche zur Struktur. Gehen Sie dabei auf Wiederverwendbarkeit und Aufgaben der jeweiligen Attribute und Methoden ein.</p> <p>b) Erweitern Sie (falls nötig) Ihr Klassendiagramm um eine Trennung von Struktur und Daten und Implementieren Sie die Klassen in Java.</p> <p>Hinweis: Zur Übung soll die Methode TitelInformationAusgeben() einfach per System.out.println(..) Interpret, Name und Länge auf die Konsole ausgeben.</p>
5	... Schnittstellen in Java implementieren und deren Funktion und Anwendungsbereich erläutern.	<p>a) Erläutern Sie, welche Aufgaben eine Schnittstelle erfüllt.</p> <p>b) Geben Sie anhand eines Beispiels an, welche Schlüsselwörter man braucht, um eine Schnittstelle in Java zu erstellen und zu verwenden. Gehen Sie dabei darauf ein, in welcher Klasse die jeweiligen Codes stehen müssen.</p>
6	... den Begriff rekursive Methode definieren und rekursive Methoden der Liste implementieren.	<p>a) Stellen Sie eine rekursive Methode, die die Gesamtlänge aller Titel in der Playliste berechnet, anhand eines Sequenzdiagrammes dar.</p> <p>b) Geben Sie Pseudocode an, der die Implementierung der nötigen Methoden in TITEL und PLAYLIST darstellt. Sie dürfen dabei davon ausgehen, dass die Liste nicht leer ist.</p>
7	... rekursive Methoden außerhalb der Liste anwenden.	<p>Eine Methode folgt folgenden Pseudocode:</p> <pre>void Geheim(int i){     if(i&lt;=0)     {         return 1;     }     else     {         return i*Geheim(i-1);     } }</pre> <p>Erläutern Sie, warum es sich um eine rekursive Methode handelt und welche Berechnung die Methode durchführt.</p>