# Testat 2 der Vorlesung Programmierung

In diesem Testat sollen Sie die Klassen **Playlist, PlaylistItem, Song** und **Album** anpassen. Die Schnittstelle **Lasting** ist vorgegeben und darf nicht verändert werden. Die Klasse **Main** enthält eine main-Methode, mit der Sie Ihre Implementierung testen können.

**Aufgabe 1** (60 Punkte)
Ergänzen Sie die Klasse **PlaylistItem** wie folgt:

- Fügen Sie der Klasse 2 private Instanzvariablen hinzu: "title" vom Typ String und "duration" vom Typ int.
- Initialisieren Sie diese Felder im vorgegebenen Konstruktor mit den Werten, die dieser als Parameter erhält.
- Implementieren Sie die Schnittstelle **Lasting**, indem Sie den Wert der Instanzvariable "duration" zurückgeben.
- Überschreiben Sie die toString() Methode aus java.lang.Object, sodass diese eine Zeichenkette in folgendem Format zurückgibt: <title> (<duration> s)
  (also Wert der Instanzvariable title, Leerzeichen, Klammer auf, Wert der Instanzvariable duration, Leerzeichen, kleines s, Klammer zu).

Ergänzen Sie die Klassen **Song** und **Album** wie folgt:

- Ändern Sie die Vererbungsbeziehungen so, dass beide Klassen von PlaylistItem erben.
- Fügen Sie den Klassen den Konstruktor ihrer neuen Superklasse hinzu und geben Sie die Parameter an die Superklasse weiter.
- Überschreiben Sie die toString() Methode, sodass vor den Wert der Implementierung aus der Superklasse noch der Name der Klasse gefolgt von einem Doppelpunkt und einem Leerzeichen angefügt wird. D.h. analog zu PlaylistItem soll die Ausgabe wie folgt aussehen:
  - Album: <title> (<duration> s) (für toString() in Album)
  - Song: <title> (<duration> s) (für toString() in Song)

Nachdem Sie diese Änderungen gemacht haben, sollten alle Kompilerfehler beseitigt sein.

**Aufgabe 2** (40 Punkte)
Die Klasse **Playlist** speichert eine begrenzte Menge an **PlaylistItem** mit Hilfe eines "item"-Arrays.Die containsItem-Methode prüft, ob sich ein PlaylistItem bereits in der Playlist (d.h. im itemsArray) befindet. Die addItem-Methode fügt ein PlaylistItem hinzu, sofern dieses nicht null ist und es noch Platz gibt (also items.length < maxItems). Erweitern Sie die Klasse wie folgt:

- Implementieren Sie die Schnittstelle **Lasting** und geben Sie über die getDuration-Methode die Summe der Dauer aller PlaylistItems in der Playlist zurück.
- Ergänzen Sie die Prüfungen in der addItem-Methode, sodass diese zusätzlich noch sicherstellt, dass ein PlaylistItem nur dann hinzugefügt wird
  - wenn es sich noch nicht in der Playlist befindet (Tipp: containsItem)
  - wenn dadurch die maximale Dauer (gespeichert in der Instanzvariable "maxDuration") nicht überschritten wird (Tipp: getDuration).
- Implementieren Sie die Methode lastIndexOf(title,startIndex) so dass diese das PlaylistItem mit dem Titel "title" im items-Array ab der Stelle "startIndex" bis 0 sucht. Wird ein PlaylistItem mit dem Titel gefunden, soll der höchste Index zurückgegeben werden. Ist der Name nicht im Bereich von "startIndex" bis 0 enthalten, soll -1 zurückgegeben werden. **Einschränkung:** Nutzen Sie für die Implementierung dieser Methode **keinerlei Schleifen, sondern Rekursion.**