

# Aufgabenblatt 3

## Allgemeines zum Lösen von Aufgabenblättern

Verwenden Sie Ihr existierendes IntelliJ-IDEA-Projekt und lösen Sie darin die Aufgaben. Eine gute Lösung erfüllt die Vorgaben der Aufgabenbeschreibung, ist kurz und einfach gehalten, ist mit informativen Kommentaren versehen und wurde gut getestet. Objektvariablen sind `private`.

Das Projekt mit den gelösten Aufgaben muss rechtzeitig vor der Deadline als ZIP-Datei in TUWEL hochgeladen werden. Programmtext für das Testen soll im Projekt enthalten sein. Bei mehrfachem Hochladen zählt die zuletzt hochgeladene ZIP-Datei. Es gibt keine andere Möglichkeit zur Abgabe.

Jede gelöste Aufgabe muss in TUWEL angekreuzt werden. Lösungen angekreuzter Aufgaben müssen in der Übungseinheit präsentiert werden können. Nach der Deadline ist das Ändern der Kreuzchen nicht möglich.

## Aufgabe 1 (30%, 1.5 Punkte)

Schreiben Sie eine Klasse `Playlist2` mit dem (abgesehen vom Klassennamen) gleichen Konstruktoraufruf und den gleichen Methoden wie `Playlist`, wobei die Songs diesmal in einer doppelt verketteten Liste (Variante beliebig) abgespeichert werden sollen. Implementieren Sie außerdem

```
void addBefore(String title, Song song)
```

in `Playlist2`. Diese Methode fügt `song` in der `Playlist2` vor dem ersten Song mit dem Titel `title` ein, bzw. am Ende, wenn es keinen Song mit dem Titel gibt.

### Fragen

Welche Vor- und Nachteile hat diese Implementierung für diese Aufgabe im Vergleich zu Arrays und einfach-verketteten Listen?

## Aufgabe 2 (verpflichtend, 30%, 1.5 Punkte) unbedingt lösen

Diese Aufgabe wird in der Ad-hoc-Aufgabe und in späteren Aufgabenblättern erweitert. Wenn Sie sie nicht machen, werden Sie die Ad-hoc-Aufgabe vermutlich nicht schaffen.

Schreiben Sie eine Klasse `SongTree`, die Songs in einem binären Suchbaum verwaltet, wobei die Titel der Songs als Suchschlüssel verwendet werden. Mit `new SongTree()` kann der Benutzer einen leeren `SongTree` erzeugen. Implementieren Sie folgende nicht-statische Methoden in `SongTree`:

```
void add(Song song)
void print()
```

Dabei soll `add` `song` in den `SongTree` einfügen, und `print` die Songs in aufsteigender alphabetischer Reihenfolge (eigentlich der Reihenfolge, die durch `CompareTo` definiert ist) ausgeben, je ein Song pro Zeile, in dieser Hinsicht wie bei `Playlist`.

Einführung in die  
Programmierung 2

LVA-Nr. 185.A92  
2018 S  
TU Wien

### Thema:

doppeltverkettete Liste;  
baumförmige, rekursive  
Datenstruktur

### Ausgabe:

16. 4. 2018

### Abgabe (Deadline):

23. 4. 2018, 6:00 Uhr

Lösungen hochladen und  
gelöste Aufgaben  
ankreuzen (TUWEL)

### Skriptum:

Seiten 60–67  
Aufgaben 2.27–2.32

## Fragen

- Der abstrakte Datentyp von `SongTree` unterscheidet sich stark genug von dem von `Playlist`, dass er trotz gewisser Ähnlichkeiten nicht mehr als `Playlist`-Variante gelten kann. Dass `getLaenge()` und der Parameter des Konstruktors nicht verlangt werden, ist nicht der Grund, sondern eine Konsequenz daraus (also: weil es keine `Playlist`-Variante ist, brauchen wir auch kein `getLaenge()`). Welche Unterschiede sind gemeint?
- Welche Einschränkung ergibt sich durch die Implementierung als Suchbaum? Was macht ihre Implementierung, wenn der Benutzer diese Einschränkung nicht einhält? Wie könnte man diese Einschränkung beheben, ohne den Suchbaum komplett aufzugeben?

## Aufgabe 3 (20%)

Fügen Sie zur Klasse `SongTree` eine nicht-statische Methode

```
Song lookupTitle(String title)
```

hinzu, die den Song in `SongTree` zurückgibt, dessen Titel gleich `title` ist, bzw. `null`, wenn es keinen solchen Song gibt.

## Fragen

Diese Methode kann man rekursiv, aber auch ohne große Verrenkungen iterativ implementieren. Welche Variante ist einfacher, oder sind sie gleich kompliziert?

## Aufgabe 4 (20%)

Auch wenn es inhaltlich nicht wirklich sinnvoll ist, ist es doch eine gute Übungsaufgabe: Fügen Sie zur Klasse `SongTree` eine nicht-statische Methode

```
long getLaenge()
```

hinzu, die die Summe der Längen der Songs zurückgibt. Diese Methode soll die Länge erst beim Aufruf ausrechnen und nicht auf eine vorberechnete Länge in einer Objektvariablen zugreifen.

## Fragen

Vergleichen Sie `getLaenge()` mit `print()`. Was sind die Unterschiede, was die Gemeinsamkeiten?