

# Aufgabenblatt 5

## 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 (40%, 2 Punkte)

Schreiben Sie eine Klasse `SongTable`, die eine Hashtabelle mit folgenden nicht-statischen Methoden implementiert:

```
void add(Song song)
Song lookupTitle(String titel)
String toString()
void print()
```

mit einem Verhalten wie bei `SongTree`, nur dass `toString` und `print` die Songs in beliebiger Reihenfolge anordnen können.

Verwenden Sie für die Hash-Tabelle eine fixe Größe (die man im Konstruktor angeben kann), und zur Kollisionsbehandlung eine einfach verkettete Liste. Beachten Sie, dass Sie diesmal (im Gegensatz `Playlist1`) einfügen können, wo sie wollen.

### Fragen

- Vergleichen Sie diese Art der Kollisionsbehandlung und Elimination von Größenbeschränkungen mit der im Skriptum gezeigten. Was sind die Vor- und Nachteile der Methoden?
- Vergleichen Sie `SongTable` mit `PlayList1` und `SongTree`. Was sind die Gemeinsamkeiten und Unterschiede im Verhalten (nicht in der Implementierung). Überlegen Sie sich, wie Sie diese drei Klassen in einer Typ hierarchie organisieren würden; welche Typen übernehmen welche Eigenschaften von den übergeordneten Typen, und welche fügen sie hinzu?

**Einführung in die Programmierung 2**

LVA-Nr. 185.A92  
2018 S  
TU Wien

**Thema:**

Hash-Tabelle, Sortieren, Iterator

**Ausgabe:**

14. 5. 2018

**Abgabe (Deadline):**

28. 5. 2018, 6:00 Uhr  
Lösungen hochladen und gelöste Aufgaben ankreuzen (TUWEL)

**Skriptum:**

Seiten 84–98  
Aufgaben 3.17–3.29

Ein Eintrag in der Hashtabelle enthält eine Liste mit allen Songs, die den gleichen Index in der Hash-Tabelle haben.

## Aufgabe 2 (verpflichtend, 20%, 1 Punkte) unbedingt lösen

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

Definieren Sie Interfaces `SongIterable` und `SongIterator` analog zu Listings 3.17 und 3.18, nur dass `next` einen Song zurückgeben soll, und diese Interfaces daher auch `java.lang.Iterable<Song>` (bzw. `java.util.Iterator<Song>`) erweitern sollen.

Ändern Sie die Klasse `Playlist1` so, dass sie `SongIterable` implementiert, und implementieren Sie die Methode `iterator`.

Schreiben Sie `toString` für `Playlist1`, wobei Sie den Iterator verwenden.

## Fragen

Wie könnte ein Iterator für `SongTable` implementiert werden?

## Aufgabe 3 (40%, 2 Punkte)

Ändern Sie die Klasse `SongTree` so, dass sie `SongIterable` erweitert, und implementieren Sie die Methode `iterator` und alles was dazugehört; dabei soll der Iterator die Songs in lexikografisch aufsteigender Reihenfolge der Titel liefern.

Implementieren Sie den internen Zustand des Iterators dabei mit zwei Objektvariablen: `depth` gibt an, in welcher Tiefe das nächste Element gefunden wird (-1, wenn es keines mehr gibt). `path` ist eine Zahl, deren Bits den Pfad zu diesem Element angeben. Dabei gibt jedes Bit an, ob man im Baum nach links oder nach rechts gehen muss. Verwenden Sie `long` oder `BigInteger` für `path`.

## Fragen

- Welche Einschränkung ergibt sich aus der Implementierung von `path` als `long`?
- Ist es besser, dass das niederwertigste oder das höchstwertige Bit von `path` angibt, in welche Richtung man von der Wurzel aus gehen soll, oder ist es egal? Begründen Sie Ihre Antwort.
- Ist es besser, wenn 1 links und 0 rechts bedeutet, oder umgekehrt? Oder ist es egal? Begründen Sie ihre Antwort.
- Was sind die Vor- und Nachteile dieser Implementierung im Vergleich zu der von `OrderedTree` und `OTIter` im Skriptum?