



Übung 10: Interfaces, Comparable, Comparator

Das Interface `Comparable` wird für die „natürliche“ Ordnung von Objekten definiert. Falls benötigt, werden alternative Ordnungen über das Interface `Comparator` definiert.

Hinweis: Wie in der Vorlesung besprochen wird die Verwendung von Generics empfohlen. Falls Sie dennoch auf Generics verzichten, können Sie *Raw Types Warnings* ignorieren.

Aufgabe 1: Klasse CD

Schreiben Sie eine Klasse `CompactDisc` mit den Attributen `Interpret`, `Titel`, `Erscheinungsjahr` und `Plattenlabel`. Die Objekte der Klasse sollen **immutable** sein!

Implementieren Sie ferner sinnvolle *Konstruktoren* und *Getter*-Methoden und überschreiben Sie die Methoden `toString`, `equals` und `hashCode`. Zwei CDs sollen *gleich* sein, wenn `Interpret` und `Titel` übereinstimmen (ohne Berücksichtigung der Groß- und Kleinschreibung).

Aufgabe 2: Interface Comparable

Das Interface `java.lang.Comparable` ist in der Java-Bibliothek vordefiniert:

```
package java.lang;
import java.util.*;
public interface Comparable<T> {
    public int compareTo(T o);
}
```

Implementieren Sie das Interface `Comparable` für die Klasse `CompactDisc`. Ordnen Sie dabei CDs wie im Plattenladen, das heißt nach `Interpret`, wobei jedoch führende „der“, „die“, „das“, „ein“, „eine“, „einer“, „eines“, „the“ und „a“ nicht berücksichtigt werden. Zum Beispiel:

- „Die Toten Hosen“ steht unter „T“
- „The Who“ unter „W“.

Testen Sie Ihre Implementierung mit dem vorgegebenen JUnit Testtreiber `TestCompactDisc`!

Hinweise:

- Gehen Sie davon aus, dass das Attribut für den Interpreten niemals null ist.
- Verwenden Sie ggfs. geeignete Hilfsmethoden der Klasse `String`, wie z.B. die Methode `trim()`, um führende oder abschließende Leerzeichen zu entfernen.
<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>
- Die Klasse `Arrays` verfügt über eine Reihe von (statischen) Methoden, die je nach gewähltem Lösungsansatz hilfreich sein können, z.B. `sort()`, `copyOf()`, `binarySearch()`
<https://docs.oracle.com/javase/8/docs/api/java/util/Arrays.html>

Aufgabe 3: Interface Comparator – Vergleich nach Erscheinungsjahr

- Schreiben Sie eine Klasse `YearOfPublicationComparator`, die einen `Comparator` für `CompactDiscs` implementiert, der diese nach dem **Erscheinungsjahr** ordnet.
- Testen Sie mit der Methode `testYearOfPublicationComparator()` der JUnit4 Testklasse `TestCompactDisc.java`. Versuchen Sie zu verstehen, wie und was diese Methode testet.

Aufgabe 4: Interface Comparator – Vergleich nach Interpret und Titel

Schreiben Sie eine Klasse `InterpreterTitleComparator`, der die CDs (wie in der natürlichen Ordnung) nach `Interpret` ordnet, und zusätzlich die CDs bei gleichem `Interpreten` nach `Titel` ordnet.

Einen JUnit Testtreiber `TestInterpreterTitleComparator` finden Sie in der Vorgabe.