

## Übung 7 – Vergleiche

### Aufgabe 1 (Theorie)

Schauen Sie in der Java API zu Comparable und Comparator nach <https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html> <https://docs.oracle.com/javase/8/docs/api/java/util/Comparator.html>

Beschreiben Sie mit eigenen Worten:

- Was der Unterschied zwischen `Comparable` und `Comparator` ist.
- Welche Unterschiede und Gemeinsamkeiten die Methoden `compareTo` und `compare` haben.
- Was es laut der API bei den Implementierungen von beiden Methoden für Restriktionen gibt.

### Aufgabe 2

Wir betrachten nochmals die Klassen `Person`, `Student` und `Boxer` aus einer vorherigen Übung. Definieren Sie jeweils eine natürliche Ordnung für die Klassen `Student` und `Boxer`, indem Sie diese `Comparable` implementieren lassen. Die natürliche Ordnung der Klasse `Student` soll über die Matrikelnummer definiert sein. Die natürliche Ordnung der Klasse `Boxer` soll über Gewicht und Name, Vorname (in der Reihenfolge) definiert und sein.

*Was würde passieren, wenn Sie nur/auch die Klasse `Person` `Comparable` implementieren lassen und warum?*

### Aufgabe 3

Schreiben Sie einen Comparator `ComparatorPersonVornameName`, der Personen zuerst anhand Ihres Vornamen und dann anhand des Namen vergleicht.

Schreiben Sie einen Comparator `ComparatorBoxerGewicht`, der Boxer nur anhand des Gewichts vergleicht.

### Aufgabe 4

Wir wollen in einer generischen Klasse `Suchbaum` einen Suchbaum nun so implementieren, dass wir auch andere Elemente als Integer (siehe `IntSuchbaum`) verwenden können. Ausserdem wollen wir für Vergleiche die natürliche Ordnung der Elemente oder einen Comparator nutzen können. Der Comparator soll als private Referenz in der Klasse gespeichert werden. Stellen Sie zwei Konstruktoren zur Verfügung. Einen parameterlosen Konstruktor, der den Effekt hat, dass die Referenz auf das Comparator-Objekt `null` ist.

Der zweite Konstruktor nimmt eine Referenz auf einen Comparator an und weist dieser der privaten Comparator-Variable zu.

Nehmen Sie hierfür Ihre Klasse `IntSuchbaum` als Grundlage. Passen Sie ihre Methoden wo nötig so an, dass statt `Integer` ein Typparameter verwendet wird. Ändern Sie Ihre Methoden so, dass Vergleiche statt mit Vergleichsoperatoren, nun mittels natürlicher Ordnung oder Comparator stattfinden. Ist die Comparator-Variable `null`, soll die natürliche Ordnung (`compareTo`) verwendet werden, ansonsten soll der Comparator (`compare`) verwendet werden. **Hinweis:** private Hilfsmethode, `cast`.

*Was folgt daraus gemäß des Substitutionsprinzips für die Daten, die Sie in dem Suchbaum speichern können? Was sind Vor- bzw. Nachteile?*

*Was passiert wenn Sie zwei Boxer-Objekte mit gleichem Gewicht, aber unterschiedlichen Namen in diesen Suchbaum einfügen wollen und dabei den `ComparatorBoxerGewicht` für den Vergleich verwenden?*