

### Aufgabe: Mehrere Sortierungen (Comparator, anonyme Klasse)

Ein **TreeSet** ist eine generische Klasse, die das **Set**-Interface implementiert. Das **Set** wird dabei als Baum abgespeichert.

In dem Baum wird eine natürliche Ordnung sichergestellt, in der die Elemente mit Hilfe der **compareTo**-Methode geordnet eingefügt werden.

Soll eine andere Ordnung der Elemente hergestellt werden, ist dies möglich, indem eine Klasse erstellt wird, die das **Comparator**-Interface implementiert.

Das generische **Comparator**-Interface sieht (ausschnittsweise) wie folgt aus:

```
public interface Comparator<T>
{
    /*Compares its two arguments for order. Returns a negative integer,
       zero, or a positive integer as the first argument is less than, equal
       to, or greater than the second.
       ...
    */
    int compare(T o1, T o2);
}
```

Die untenstehende Klasse **Kreuzwortraetsel** zeigt ein Beispiel für ein **TreeSet**, das die natürliche Ordnung verwendet.

```
public class Kreuzwortraetsel
{
    static String[] alleWoerter = {"Bienenschwarm", "Buch", "Bibel",
        "Beige", "Barriere", "Bein", "Beil", "Christ", "Christian", "Carmen"};

    public static void main(String[] args)
    {
        TreeSet<String> t = new TreeSet<>();
        t.addAll(Arrays.asList(alleWoerter));
        for(String wort : t)
            System.out.println(wort);
    }
}
```

### Anmerkungen:

Mit **Arrays.asList(array)** lässt sich aus einem Array eine Liste erstellen.

Mit **t.addAll** lassen sich alle Elemente einer **Collection** in **t** einfügen.

- Implementieren Sie einen **Comparator**, der es ermöglicht, die Wörter des Kreuzworträtsels zunächst der Länge nach und bei gleicher Länge nach dem Alphabet zu sortieren.
- Die Klasse **TreeSet** stellt einen Konstruktor zur Verfügung, der einen **Comparator** entgegennimmt, der die Ordnung für den Baum festlegt. Modifizieren Sie obiges Beispiel so, dass das **TreeSet** Ihren **Comparator** aus Aufgabenteil a) verwendet.
- Implementieren Sie Ihren **Comparator** noch einmal mit Hilfe einer anonymen Klasse.
- Was sind in diesem Beispiel die Vor- und Nachteile der Verwendung der anonymen Klasse?