

#### 17.1.4 Schnittstelle `Comparator<T>` und Wildcards

Zum leichteren Auffinden von Teilen in einem Magazin, soll eine Möglichkeit geschaffen werden, mit der die Kleinteile der einzelnen Magazine nach ihrer Bezeichnung und Ausführung sortiert werden können.

Die Utility-Klasse `Collections` aus der Java-Klassenbibliothek bietet mit der statischen Methode `sort()` die Möglichkeit des Sortierens von Listen. In der Klasse `RostfreieKleinteileMagazin` soll die Liste vom Typ `ArrayList` mit Hilfe dieser Methode sortiert werden. Mit einer Klasse, die die Schnittstelle `Comparator<T>` implementiert, kann die Sortierreihenfolge für beliebige Klassen definiert werden.

Zu diesem Zweck soll eine generische Utility-Klasse `TeileUtils` im Paket `utils` geschaffen werden. Diese Klasse soll die Schnittstelle `Comparator<T>` implementieren und eine statische, eigenständig generische Methode `sortiereMagazin()` bereitstellen. Zur Sortierung der Liste soll die Variante der Methode `sort()` verwendet werden, die zwei Parameter annimmt.

Der Konstruktor dieser Klasse soll nicht öffentlich sein. Stattdessen soll bei der Verwendung der Methode `sortiereMagazin()` eine neue Instanz der Utility-Klasse erstellt werden.

Der formale Typ-Parameter `T` dieser Klasse, sowie der eigenständig generischen Methode `sortiereMagazin()` soll folgenden Einschränkungen unterliegen: `T` darf alle von `Teil` abgeleiteten Typen darstellen und dieser, von `Teil` abgeleitete Typ, darf wiederum mit einem beliebigen Typ parametrisiert worden sein.

Die Methode `sortiereMagazin()` soll nach ihrem Aufruf die Methode `sort()` der Klasse `Collections` aufrufen. Als erster Parameter soll die übergebene Liste weitergereicht werden, als zweiter Parameter soll eine neue Instanz der zu implementierenden Utility-Klasse übergeben werden. Die Methode `compare()` aus der Schnittstelle `Comparator<T>` soll – wie beschrieben – lexikografisch zunächst die Bezeichnung und im zweiten Schritt die Ausführung der jeweils zu vergleichenden Teile auswerten. Dazu kann die Methode `compareToIgnoreCase()` der Klasse `String` herangezogen werden. Der Rückgabewert der Methode `compare()` soll dabei den Bedingungen, die in der Java-Klassenbibliothek-Dokumentation beschrieben sind, genügen. Die Methode `equals()` aus der Schnittstelle `Comparator<T>` muss nicht implementiert werden.

Zur Implementierung der Klasse `TeileUtils` muss der folgende Programmcode ergänzt werden. Bevor Sie die Klasse übersetzen können, müssen Sie die Datei entsprechend umbenennen:

```
// Datei: TeileUtils_Vorlage.java

package utils;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
import teile.Teil;
```

```

public class TeileUtils<T . . . . > implements Comparator<T>
{
    private TeileUtils()
    {
    }
    public int compare (T t1, T t2)
    {
        // String-Vergleiche machen und Ergebnis zurückgeben
        . . . . .
    }

    public static<T . . . . > void sortiereMagazin(List<T> teile)
    {
        // Klassenmethode sort() der Klasse Collections<T> aufrufen
        . . . . .
    }
}

```

**Die Klasse** RostfreieKleinteileMagazin **soll um die Methode** sortiereMagazin() **erweitert werden:**

```

public void sortiereMagazin()

```

**Diese Methode soll lediglich die Methode** sortiereMagazin() **der geschaffenen Utility-Klasse aufrufen und ihr die intern verwendete Liste** magazin **übergeben.**

**Es sollen alle Klassen aus der vorherigen Aufgabe verwendet werden. Zum Testen steht die Klasse** KleinteileMagazinTest4 **zur Verfügung:**

```

// Datei: KleinteileMagazinTest4.java
package magazin;

import teile.GewindeTeil;
import teile.Mutter;
import teile.RostgarantieNichtErfuelltException;
import teile.Scheibe;
import teile.Schraube;

public class KleinteileMagazinTest4
{
    public static void main (String[] args)
    {
        // Anlegen zweier Magazine für Gewindeteile und Scheiben
        RostfreieKleinteileMagazin<GewindeTeil<Float>> gewindeteile =
            new RostfreieKleinteileMagazin<GewindeTeil<Float>>();
        RostfreieKleinteileMagazin<Scheibe<Float>> scheiben =
            new RostfreieKleinteileMagazin<Scheibe<Float>>();

        // Hinzufügen der unterschiedlichen Magazine zu einem Array
        RostfreieKleinteileMagazin<?>[] magazine =
            {gewindeteile, scheiben};

        try
        {
            // Anlegen dreier Gewindeteile
            gewindeteile.aufnehmen
                (new Mutter<> (1000, Mutter.NORMAL, 6, 5f, 0.01f));

```

```

gewindeteile.aufnehmen (new Mutter<>
    (1000, Mutter.SICHERUNG, 6, 6f, 0.01f, 18));
gewindeteile.aufnehmen (new Schraube<>
    (1000, Schraube.SENKKOPF, Schraube.KREUZSCHLITZ,
    6f, 0.01f, 18));
// Anlegen einer Scheibe
scheiben.aufnehmen (new Scheibe<>
    (1000, Scheibe.FEDER, 5.5f, 10.1f, 0.01f, 12));

// Das Magazin gewindeteile sortieren
gewindeteile.sortiereMagazin();

// Inhalt der Magazine ausdrucken
for (RostfreieKleinteileMagazin<?> magazin : magazine)
{
    System.out.println ("Gebe naechstes Magazin aus:");
    magazin.druckeMagazin();
}
}
catch (RostgarantieNichtErfuellteException e)
{
    System.out.println(e.getMessage());
}
}
}

```