

Praktikum Programmiermethodik 2 (Technische Informatik)

WS 2016/2017, Hochschule für Angewandte Wissenschaften (HAW), Hamburg
Prof. Dr. Philipp Jenke, Kasperczyk-Borgmann



Für dieses Aufgabenblatt gelten die folgenden Regeln:

- Der Java Code Style (siehe EMIL) ist einzuhalten. Es werden keine Abgaben abgenommen, die diese Anforderungen nicht erfüllen.
- Pro Team muss ein gemeinsames Git-Repository für das Projekt vorhanden sein (z.B. auf GitHub, Bitbucket).
- Der Code muss vollständig getestet sein (JUnit-Tests).

Aufgabenblatt 1: Wiederholung Collections, XML, Generics

Aufgabe 1.1: Wiederholung PM1/PT

Lernziele: Umgang mit dem Collections-Framework, Vergleichen in Java, Basisklasse Object verwenden

Aufgaben:

- Schreiben Sie eine Klasse Student. Ein Student hat folgende Eigenschaften: Vorname, Nachname, Matrikelnummer (Ganzzahl), Liste von Prüfungsleistungen.
- Prüfungsleistungen sind ein eigener Typ Pruefungsleistung, der wiederum den Namen des Moduls und die Note beinhaltet.
- Student muss equals() und hashCode() überschreiben. Zwei Studenten sind genau dann gleich, wenn sie die gleiche Matrikelnummer haben
- Machen Sie Studenten vergleichbar, indem Sie das Interface Comparable<T> implementieren. Sortiert werden soll anhand der Matrikelnummer.
- Manchmal ist auch der Vergleich nach Nachname, Vorname gewünscht. Setzen Sie dies durch einen geeigneten Comparator<T> um.

Aufgabe 1.2: XML

Lernziele: XML-Dokumente lesen und schreiben, XML-DTD erstellen

Aufgabe:

- Wir betrachten folgende Fachlichkeit: Ein Sensor für skalare Messungen hat eine ID und beinhaltet eine Liste von Messungen (Wert: double und Zeitstempel: LocalDateTime)
- Schreiben Sie Java-Klassen, die Sensoren und Messungen repräsentieren
- Entwerfen Sie eine XML-DTD für XML-Dateien, die einen Sensor repräsentieren. Der folgende XML-Code soll gültig sein:

```
<?xml version="1.0" ?>
<!DOCTYPE Sensor SYSTEM "sensor.dtd">
<Sensor id="Temperatur Wohnzimmer">
    <Messung wert="23,2" zeitstempel="2016-06-23T16:51:40.408"/>
    <Messung wert="24,5" zeitstempel="2016-06-24T12:12:52.000"/>
</Sensor>
```

- Schreiben Sie Code zum Erzeugen eines Java-Sensor-Objektes aus einer Sensor-XML-Datei.
- Schreiben Sie Code zum Erzeugen einer Sensor-XML-Datei aus einem Java-Sensor-Objekt.

Hinweise: Konvertierung zwischen LocalDateTime und String: toString() und parse(), aktuelle Zeit: LocalDateTime.now()

Aufgabe 1.3: Generics

Lernziele: Erstellen generischer Typen, Arbeiten mit Typebounds und Wildcards,

Aufgabe:

- Schreiben Sie eine eigene Implementierung von ArrayListe<T>, die intern ein Array verwendet und folgende Schnittstelle bietet:

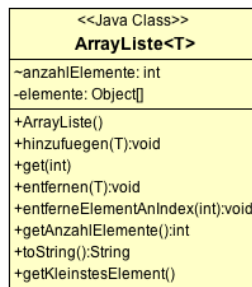


Abbildung 1: Klassendiagramm von ArrayListe. Die Methoden get() und getKleinstesElement() sollen Werte vom Typ T zurückliefern.

- Sollten sich durch erforderliche Type-Casts Compiler-Warnungen ergeben, ist das in dieser Aufgabe ausnahmsweise in Ordnung.
- Schreiben Sie eine statische Methode (außerhalb der Klasse ArrayListe<T>), die für eine beliebige Liste prüft, ob das erste Element (wenn es mindestens eins gibt) eine Zahl ist.
- Sie dürfen in Ihrer Implementierung natürlich keine Funktionalität einer anderen Collection verwenden