



Aufgabenblatt 1: XML, Generics

Aufgabe 1.1: XML

- 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 1.2: Generics

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

<<Java Class>>
ArrayList<T>
-anzahlElemente: int
-elemente: Object[]
+ArrayList()
+hinzufuegen(T):void
+get(int)
+entfernen(T):void
+entferneElementAnIndex(int):void
+getAnzahlElemente():int
+toString():String
+getKleinstesElement()

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

- Die Elemente in der Liste sollen vergleichbar sein
- 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 ArrayList<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

Testen Sie allen Code!