

## Aufgabenstellung

Unter dem Motto „Die Zeit läuft: Folgen Sie uns zur Energiewende“ versucht die österreichische Bundesregierung das Klima zu retten. Verschiedene Messstationen speichern kleinste Temperatur, größte Temperatur und die Niederschlagsmenge pro Tag in XML-Dateien. Für die Jahre 2000 bis 2009 sind die Einzelmessungen noch vorhanden. In der Datei „allData.xml“ sollten alle Messwerte seit 1948 bis 2009 gespeichert sein. Jeder braucht manchmal Hilfe, auch die Bundesregierung. Sie möchte alle Messdaten vermengen, doppelte Einträge entfernen, um anschließend richtig Analysen durchführen zu können.

Die Projekt-Vorlage gibt es auf dem Prüfungsmoodle. Im „resources“-Verzeichnis sind die Messdaten. Die Datei „max.html“ ist eine fertige Auswertung zur Kontrolle, ob eure Programme richtig funktionieren.

Alle Regeln der OOP sind einzuhalten. Vermeide doppelten Code. Das Laden und Speichern von den XML-Messdaten muss mit JAXB erfolgen. Keine Anti-Patterns!

### **Klasse UI**

#### a) Menüsteuerung via Konsole

```
0 ... Beenden
1 ... Daten vom Verzeichnis 'resources' laden
2 ... Daten von der Datei 'allData.xml' laden
3 ... Duplikate entfernen (mit WHILE Schleife)
4 ... Duplikate entfernen (mit Stream API)
5 ... Maximaler Temperaturwert von allen Stationen
```

```
Wahl [0,5] = 2
```

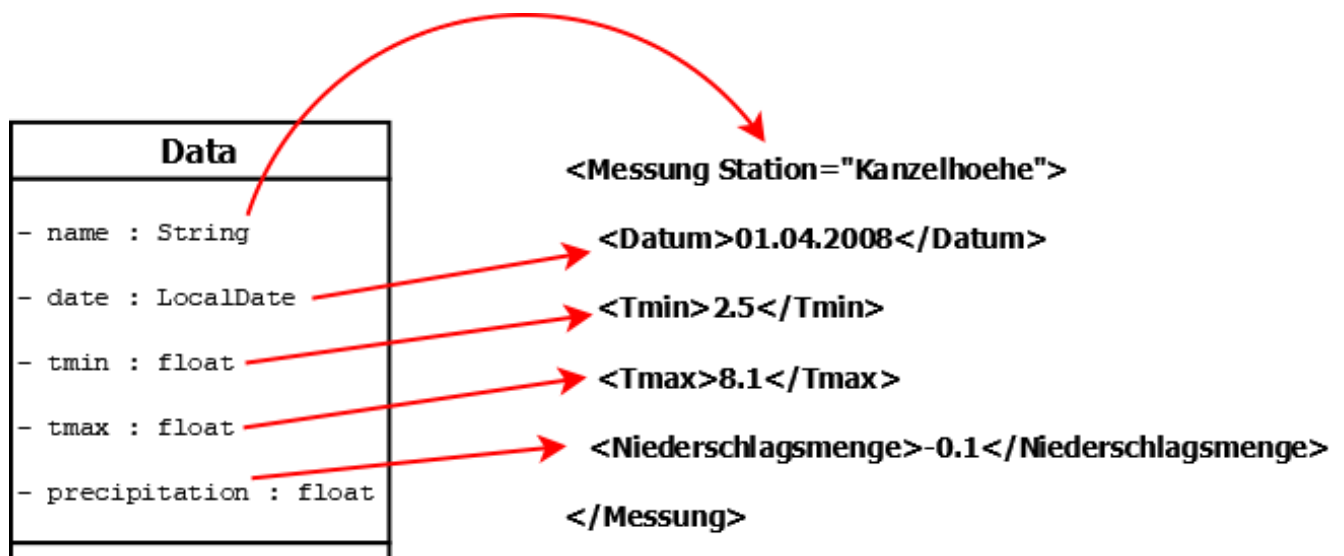
```
Gewählt wurde = 2
```

```
...
```

### **Klassen Data, DataCollection, DataComparator, LocalDateAdapter**

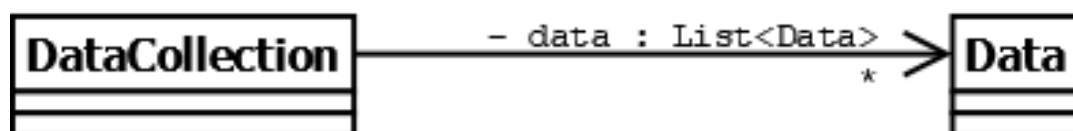
#### b) Klasse Data

Die Klasse Data repräsentiert eine Einzelmessung:



#### c) Klasse DataCollection

Die Klasse DataCollection wird verwendet, um alle Messwerte von der Datei „allData.xml“ mit JAXB zu laden. Diese Datei hat genau 113230 Messungen. Sie beinhaltet keine Duplikate.



Beim Laden werden immer alle Messdaten in der Collection „data“ gespeichert (immer nur hinzufügen). Somit können beim Laden doppelte Einträge entstehen. Doppelte Einträge entstehen auch, wenn die Menüpunkte 1 und 2 mehrmals aufgerufen werden.

#### d) Klasse DataComparator

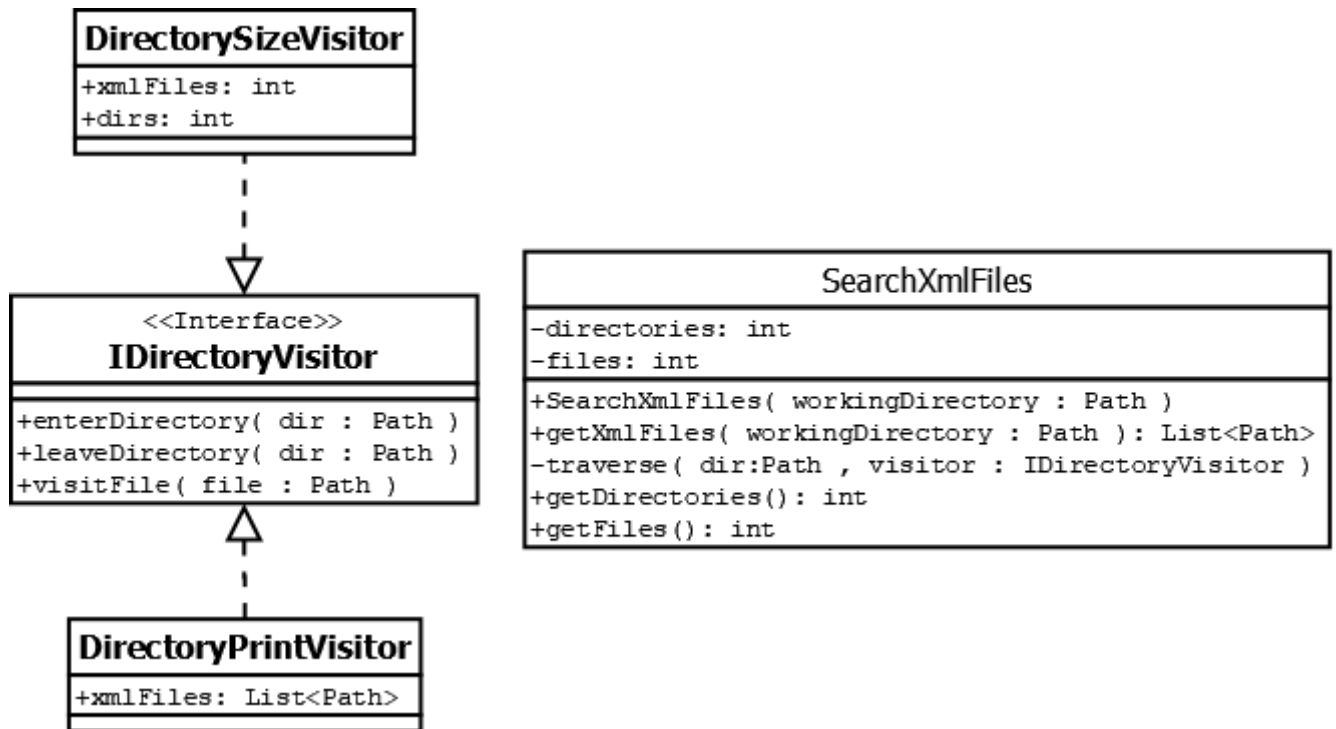
Um doppelte Einträge mit der WHILE Schleife zu entfernen, muss diese Klasse, das Interface java.util.Comparator implementieren. Duplikate existieren, wenn Name der Station und das Datum der Messung übereinstimmen.

#### e) Klasse LocalDateAdapter

Damit mit JAXB das Datum von dem XML-Dateien richtig gelesen werden kann, muss ein entsprechender XmlAdapter implementiert werden.

### **Design Pattern Visitor**

Damit beim Menüpunkt 1 alle XML-Dateien gefunden werden (auch von allen Unterverzeichnissen usw.) soll eine Lösung mit dem Design Pattern Visitor entwickelt werden:



#### f) IDirectoryVisitor, DirectorySizeVisitor, DirectoryPrintVisitor

Die Klasse `DirectorySizeVisitor` ermittelt wie viele XML-Dateien und Unterverzeichnisse es gibt. Anzahl der Unterverzeichnisse ist 11 (mit dem eigenen „resources“ Verzeichnis). Insgesamt gibt es 18267 XML-Dateien.

Die Klasse `DirectoryPrintVisitor` erstellt eine Liste mit allen XML-Dateien.

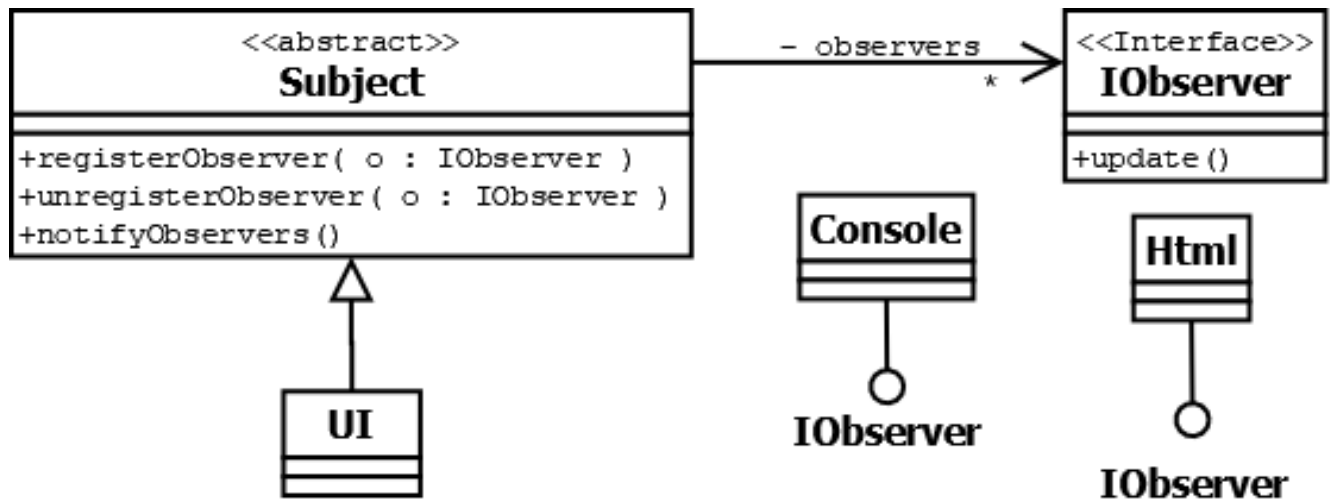
#### g) Klasse SearchXmlFiles

Der Konstruktor verwendet den `DirectorySizeVisitor` um mit der „traverse“ Methode, die Anzahl der XML-Dateien und Unterverzeichnisse zu bestimmen. Die Methode „getXmlFiles“ verwendet dann `DirectoryPrintVisitor` um mit der „traverse“ Methode, eine Liste mit allen XML-Dateien zu erzeugen.

### **Design Pattern Observer**

Nach dem Laden bzw. Entfernen der Duplikate, haben sich die Daten geändert. Diese Änderungen werden mit dem Design Pattern Observer gemeldet:

#### h) IObservable, Subject



Immer wenn sich die Anzahl der Datensätze geändert hat (nach dem Laden bzw. Entfernen der Duplikate) ruft die Klasse UI die Methode „notifyObservers“ auf, um die Änderungen an den Klassen Console und Html weiterzugeben.

#### i) Klasse Console

Die Klasse Console erzeugt folgenden Ausgabe (Anzahl der Messungen, Namen der Messstationen):

```
Anzahl der Datensätze = 113230
Kanzelhoehe, Kolbnitz, Landeck, Mayrhofen, Salzburg Flugh.
```

#### j) Klasse Html

Diese Klasse erzeugt die Datei „max.html“. Von allen vorhandenen Messstationen, wird der maximale Temperaturwert bestimmt. Es werden alle jene Messungen protokolliert, deren „tmax“ Wert maximal  $\pm 3\%$  vom maximalen Temperaturwert der jeweiligen Messstation abweicht. Hinweis: siehe „max.html“

### **Design Pattern Singleton**

#### k) Klasse Calculator

Verwaltet alle Messungen („dataList“). Alle Berechnungen werden in dieser Klasse durchgeführt.

<div style="text-align: center;"> <b>&lt;&lt;Singleton&gt;&gt;</b>  <b>Calculator</b> </div>
<pre>-dataList: List&lt;Data&gt; -workingDirectory: Path</pre>
<pre>+addDataFromXmlFiles( xmlFiles : List&lt;Path&gt; ) +addDataFromDataCollection( file : Path ) +removeDuplicates1() +removeDuplicates2() +getStationNamesAsString(): String +getMaxTemperatureOfAllStations(): Map&lt; String, List&lt;Data&gt;&gt;</pre>

Menüpunkt 3 verwendet die Methode „removeDuplicates1“ um alle Duplikate mit einer WHILE Schleife aus „dataList“ zu entfernen.

Menüpunkt 4 verwendet die Methode „removeDuplicates2“ um alle Duplikate mit der Stream API aus „dataList“ zu entfernen.

Im Menüpunkt 5 wird mit Hilfe der Methode „getMaxTemperatureOfAllStations“ von der Klasse UI folgende Ausgabe erzeugt (wenn alle Messdaten geladen sind, und wenn vorher alle Duplikate entfernt wurden):

```
Kanzelhoehe (2)
  Data { Kanzelhoehe, 1950-07-04, 14.5, 31.9, -999.0 }
  Data { Kanzelhoehe, 1983-07-27, 19.0, 31.4, -0.1 }
Kolbnitz (9)
  Data { Kolbnitz, 1962-06-18, 16.2, 35.6, 1.0 }
  Data { Kolbnitz, 1962-06-22, 20.3, 35.8, -0.1 }
  Data { Kolbnitz, 1983-07-26, 14.0, 35.4, -0.1 }
  Data { Kolbnitz, 1983-07-27, 14.5, 35.9, -0.1 }
  Data { Kolbnitz, 1983-07-28, 15.5, 36.1, -0.1 }
  Data { Kolbnitz, 1984-07-11, 14.1, 36.1, -0.1 }
  Data { Kolbnitz, 1984-07-12, 16.4, 35.9, -0.1 }
  Data { Kolbnitz, 1994-07-04, 14.7, 36.1, -0.1 }
  Data { Kolbnitz, 2009-07-23, 16.6, 35.5, -999.0 }
Landeck (5)
  Data { Landeck, 1952-07-02, 14.1, 36.8, -0.1 }
  Data { Landeck, 1952-07-04, 18.7, 35.8, -0.1 }
  Data { Landeck, 1952-07-05, 16.7, 36.8, 18.0 }
  Data { Landeck, 1983-07-27, 17.4, 36.0, -0.1 }
  Data { Landeck, 2003-08-13, 16.0, 35.8, -0.1 }
Mayrhofen (10)
```

```

Data { Mayrhofen, 1952-07-04, 17.6, 34.2, -0.1 }
Data { Mayrhofen, 1952-07-05, 16.6, 34.2, 2.5 }
Data { Mayrhofen, 1983-07-27, 17.4, 33.8, -0.1 }
Data { Mayrhofen, 1984-07-11, 17.4, 33.8, -0.1 }
Data { Mayrhofen, 1998-07-21, 16.5, 33.8, -0.1 }
Data { Mayrhofen, 2002-06-18, 15.9, 34.2, -0.1 }
Data { Mayrhofen, 2002-06-19, 17.8, 33.9, 0.1 }
Data { Mayrhofen, 2002-06-20, 18.2, 34.8, 2.1 }
Data { Mayrhofen, 2003-08-13, 16.2, 34.5, -0.1 }
Data { Mayrhofen, 2007-07-19, 17.7, 33.8, -0.1 }
Salzburg Flugh. (4)
Data { Salzburg Flugh., 1983-07-27, 20.0, 36.2, -0.1 }
Data { Salzburg Flugh., 1984-07-11, 19.1, 36.4, -0.1 }
Data { Salzburg Flugh., 2003-08-13, 19.8, 35.9, 0.0 }
Data { Salzburg Flugh., 2009-07-23, 18.6, 36.6, 4.1 }

```

Von jeder Messstation wird der maximale Temperaturwert gesucht. Alle Messungen die nur  $\pm 3\%$  vom jeweiligen Maximalwert abweichen, werden ausgegeben. Sortiert wird nur nach den Stationsnamen.

l) Klasse XmlDal

<<Singleton>> <b>XmlDal</b>	
<pre> +loadData( file : Path ): Data +loadDataCollection( file : Path ): DataCollection </pre>	

Zugriff auf die XML-Daten erfolgt immer mit dieser Klasse.