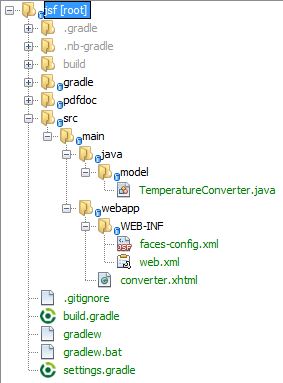
**Aufgabe**

Für den Anfang wurde das alte "hello\_jsf"-Projekt kopiert und umbenannt. Anschließend konnte das Projekt in NetBeans (mit installierten Gradle-Plugin) importiert werden.   
Files wie "hello.xhtml" werden nicht mehr verwendet und können deshalb gelöscht werden.

Danach werden die neuen Dateien für den Converter erstellt, dazu gehören "converter.xhtml" (die View) und "TemperatureConverter.java" (das Model).

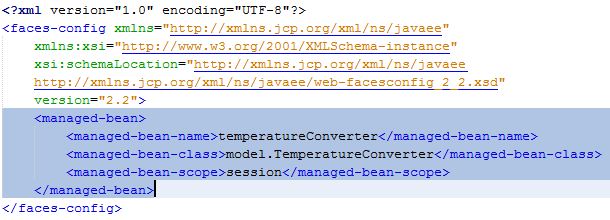
Ordner/Projektstruktur



Dann wurde "web.xml" bearbeitet. Der Wert des <display-name>-Tags wurde zu "Temperature Converter" geändert. In das <welcome-file>-Tag wurde die Datei "converter.xhtml" geschrieben. Der Rest bleibt gleich.

In der Datei "faces-config.xml" stehen alle Daten über die Managed Beans. Diese werden ebenfalls geändert.

Managed Bean Konfiguration



Anschließend wird in "converter.xhtml" die ganze Struktur aufgebaut:

Die View



Danach können im Model ("TemperatureConverter.java") die nötigen Funktionen erstellt werden:

|  |
| --- |
| package model;  import java.io.Serializable;  public class TemperatureConverter implements Serializable{  private static final long serialVersionUID = 1L;  private double convert;  private double converted;  private boolean initial;  private String unit;    public double getConvert() {  return convert;  }    public void setConvert(double convert) {  this.convert = convert;  }    public double getConverted() {  return converted;  }    public String getUnit() {  return unit;  }    public boolean getInitial() {  return initial;  }    public void init(){  initial = true;  converted = 0;  convert = 0;  unit="";  }    public String reset() {  init();  return "reset";  }    public void celsiusToFahrenheit() {  this.initial = false;  this.unit="Fahrenheit";  this.converted = (convert \* 1.8) + 32;  }    public void fahrenheitToCelsius() {  this.initial = false;  this.unit = "Celsius";  this.converted = (convert - 32) / 1.8;  }  } |

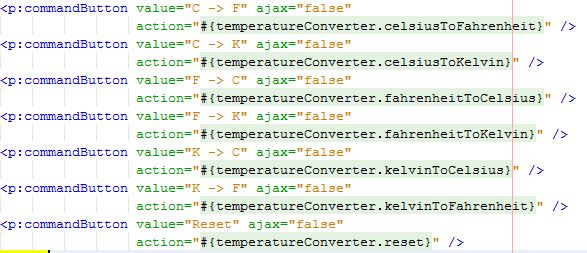
Das Model

Danach wird das Projekt mit "gradle jettyrun" in der Konsole gebuildet und ausgeführt, anschließend kann über den Browser darauf zugegriffen werden.

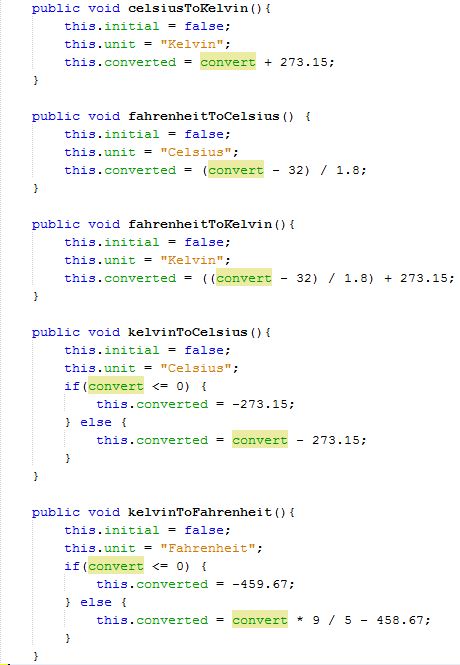
**Erweiterung**

Um neue Buttons hinzuzufügen müssen neue <p:commandButton> erstellt werden, jeweils mit der richtigen Beschriftung und Convert-Funktion:

converter.xml



Die Funktionen müssen auch der TemperatureConverter-Klasse zugefügt werden:



TemperatureConverter.java

Anschließend kann das Projekt wieder mit "gradle jettyrun" gebuildet und ausgeführt werden.



Das Endergebnis

Dieses Projekt auf GitHub:   
<https://github.com/dchlebicki/tgm-aufgaben/tree/master/15-16/SEW/tempconv>