



INFORMATIQUE ET INGÉNIERIE DU WEB (L3)

PERFECTIONNEMENT EN JAVA

---

# Wetterstation

---

*Eingereicht von:*  
Julian KRUG  
Hanna SCHALL

November 2020

Design basierend auf einem LaTeX-Template von WikiBooks,  
modifiziert von Elsa Slattegård für Uppsala university  
(License: CC BY-NC-SA 3.0)

# Kapitel 1

## Projektbeschreibung

### 1.1 Kurzbeschreibung

Resultat dieses Projekts ist eine Java-Applikation, die im Wesentlichen aus einer Wetterstation und einem Sensor besteht. Die Wetterstation kann sich mit einem Sensor verbinden, Daten anfordern und diese ausgeben. Zudem wurde diese Projektbeschreibung mit anschließender Anforderungsanalyse, Unit-Tests für die Wetterstation und ein UML-Klassendiagramm konzipiert.

### 1.2 Kontext / Hintergrund

Die Vorlesung *Perfectionnement en Java* des Wintersemesters 2020, gehalten von Herrn Prof. Brocks, vertieft das Verständnis von Software-Entwicklung, insbesondere in der Programmiersprache Java.

Im September und Oktober wurden der Aufbau von UML-Klassendiagrammen, die Datenübertragung mittels UDP und TCP, das Erstellen von Projektbeschreibungen und Anforderungsanalysen, Unit Tests, Test-Driven Development und Nebenläufigkeit behandelt.

Um die erlernten Elemente zusammenzuführen, wurde dieses Projekt realisiert. Es vereinigt in der Vorlesung gestellte Teilaufgaben wie die Kommunikation über TCP, inspiriert von der Logik eines klassischen Telefons, und das Erstellen eines Thermometers.

### 1.3 Aufgabenstellung

Die Details zur Aufgabenstellung wurden in einem Word-Dokument im Moodle-Kurs zur Vorlesung dargestellt. Hier nun zusammenfassend einige Worte dazu:

- Ein Sensor verfügt über eine Vendor Id, eine Produkt Id und einen Standort, der zu Beginn definiert werden muss. Er misst Temperaturen im Bereich von -20 und 50°C.
- Eine Wetterstation kann sich mit einem Sensor verbinden und dessen Dienste anfordern. Der Benutzer kann sich das Minimum und Maximum

der übermittelten Temperaturen des aktuellen Zeitraums anzeigen lassen. Der Zeitraum kann resettet werden.

- Die Dienste des Sensors umfassen INFO (Senden der drei oben genannten Informationen über den Sensor), DATA (periodisches Übermitteln von aktueller Temperatur inklusive Uhrzeit) und STOP (Stoppen der periodischen Übermittlung).
- Sowohl die Sensor-Applikation als auch die Wetterstation-Applikation werden über die Konsole gestartet und gestoppt.

## 1.4 Ausgangszustand (Ist-Zustand)

Die Umsetzung obiger Aufgabenstellung in Java-Code basiert auf den in der Vorlesung behandelten Implementierungen

- eines Thermometers, dessen Messwerte über die Konsole eingegeben werden können und welches eine Min/Max-Funktion besitzt,
- eines Telefons, dessen Benutzer über TCP *hören* und *in den Hörer sprechen* kann
- und einer Konsolen-Applikation, anhand derer die Funktionsweise von Threads erläutert werden.

Zudem baut die Anforderungsanalyse (siehe unten) auf die während der Vorlesung in Kleingruppen angefertigte Analyse der Projektanforderungen auf.

## 1.5 Anlass / Projektmotivation

Das vorliegende Projekt wurde aufgrund des Wunsches, die Vorlesungsinhalte in die Praxis umzusetzen und eine saubere Arbeit im Rahmen des *Contrôle continue 1* abzuliefern, umgesetzt. Stakeholder sind in diesem Fall der korrigierende Professor, Herr Prof. Brocks, und die programmierenden Studierenden, Julian Krug und Hanna Schall. Zudem könnten eventuelle Benutzer der Applikation als weitere Interessensgruppe betrachtet werden.

## 1.6 Ziele (Soll-Zustand)

**Primäres Ziel** des Projekts ist das Funktionieren der Temperaturmessung des Sensors, der Datenübermittlung an die Wetterstation und die korrekte Implementierung der Min/Max- sowie der Reset-Funktion.

**Speziell** in der hier dargestellten Lösung kann eine Wetterstation alle empfangenen Temperaturen darstellen, sofern sie im korrekten Format ankommen. Ein Sensor wird gestartet und wartet auf das Verbinden einer Wetterstation. Wird auch diese gestartet, verbindet sie sich automatisch mit dem Sensor, sofern dieser bereit ist.

In der Wetterstation kann der Benutzer von verschiedenen Funktionen Gebrauch machen, die im Menü beschrieben sind. Wird das Übermitteln der aktuellen Temperatur (zufällig erzeugt) mit einer Periodizität von bspw. 5 Sekunden angefordert, so aktualisiert die Wetterstation ihre Temperaturanzeige im

5-Sekunden-Takt.

Der Benutzer kann die Wetterstation stoppen oder auch lediglich den Sensor bitten, das regelmäßige Senden der Temperatur (DATA STOP) zu unterbinden.

## **1.7 Rahmenbedingungen und Vorgaben**

Das Projekt wurde in einer Zweiergruppe erstellt. Die ursprüngliche Abgabefrist war der 02.11.2020, die jedoch aufgrund der Nicht-Erreichbarkeit des Moodle-Servers auf den 05.11.2020 verschoben wurde.

Da die direkte persönliche Zusammenarbeit durch die Corona-Pandemie nicht möglich war, fand die Realisierung des Projekts dank Besprechungen über Discord, Austauschen von Kurznachrichten über WhatsApp und Versionieren des Codes über GitHub statt.

## Kapitel 2

# Anforderungsanalyse

Im Folgenden werden die Anforderungen an das Projekt analysiert.

### 2.1 Funktionale Anforderungen (functional requirements)

Die funktionalen Anforderungen sind gruppiert in die Bereiche *Wetterstation* und *Sensor*.

#### 2.1.1 Sensor

- FR1** Der Sensor bietet einen Info-Dienst, der das Senden von Produkt Id, Vendor Id und Standort beinhaltet, an.
- FR2** Der Sensor bietet einen Data-Dienst, der periodisch Zeit und aktuelle Temperatur an die Wetterstation schickt, an.
- FR3** Die Zeitangabe gibt Datum und exakte Uhrzeit (inklusive Sekunden) an.
- FR3** Der Benutzer kann den Data-Dienst aktivieren und die Messfrequenz angeben.
- FR4** Eine Messfrequenz wird in Sekunden angegeben und muss mindestens 1 Sekunde betragen, hat aber keine Obergrenze.
- FR5** Der Sensor bietet einen Stop-Dienst, der das periodische Senden beendet, an.
- FR6** Der Admin muss den Sensor über die Konsole starten.
- FR7** Der Admin muss den Standort des Sensors über die Konsole eingeben.
- FR8** Wetterstation und Sensor können miteinander kommunizieren (Datenaustausch).

### 2.1.2 Wetterstation

- FR9** Der Admin kann die Wetterstation über die Konsole starten.
- FR10** Die Wetterstation hat eine MinMax-Funktion, die das Minimum und das Maximum der Temperatur eines Zeitraums angibt.
- FR11** Der Zeitraum ist begrenzt durch den Start des Programms oder Reset von Minimum und Maximum und dem aktuellen Zeitpunkt.
- FR12** Der Benutzer kann die MinMax-Funktion der Wetterstation benutzen.
- FR13** Der Benutzer kann das Minimum und Maximum zurücksetzen (Reset).
- FR14** Die Wetterstation kann sämtliche Temperaturen anzeigen.

## 2.2 Nicht-Funktionale Anforderungen (non-functional requirements)

Die nicht-funktionalen Anforderungen sind gruppiert in die Bereiche *Wetterstation*, *Sensor* und *Administrativ*.

### 2.2.1 Sensor

- NFR1** Der Sensor muss eine Produkt Id, eine Vendor Id und einen Standort haben.
- NFR2** Der Sensor kann im Bereich -20 bis 50°C messen.
- NFR3** Der Sensor muss durch Unit-Tests getestet werden.

### 2.2.2 Wetterstation

- NFR4** Die Wetterstation kommuniziert mit dem Sensor über TCP.
- NFR5** Die Wetterstation kann sich mit genau einem Sensor verbinden.
- NFR6** Der Sensor muss durch Unit-Tests getestet werden.

### 2.2.3 Administrativ

- NFR7** Es muss ein UML-Klassendiagramm erstellt werden.
- NFR8** Es muss eine Projektbeschreibung verfasst werden.
- NFR9** Die Abgabefrist des Projekts ist am 05.11.2020 um 6 Uhr.
- NFR10** Die Programmiersprache ist Java.
- NFR11** Das Programm muss auf einem durchschnittlichen PC (mind. 1 GB RAM) lauffähig sein.

## Kapitel 3

# UML-Klassendiagramm

