

Softwaretechnik

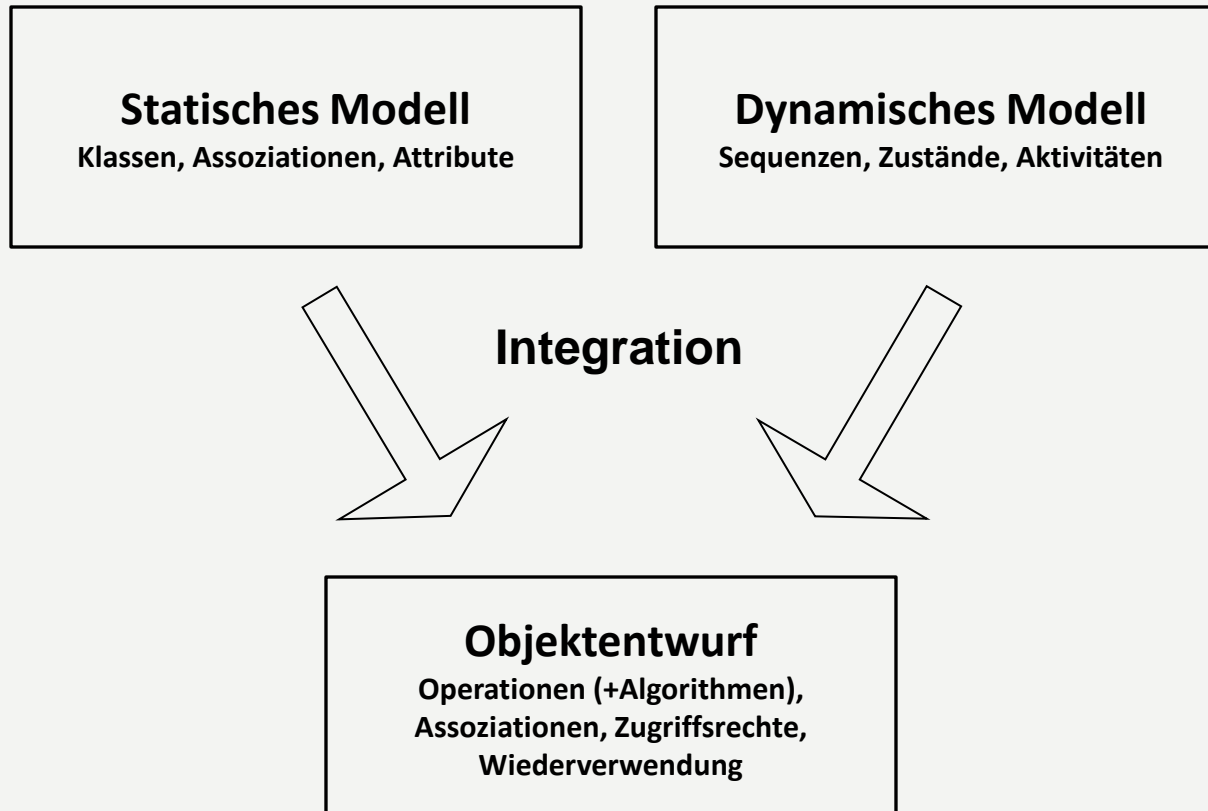
WiSe 09/10

Übungsblatt 9 (Objektentwurf)

Gegeben seien das in der Analyse erstellte statische und dynamische Modell für das Wetterstationssystem (vgl. Übung 6 und 7). Die beiden Modelle sollen in einen **Objektentwurf** integriert werden. Dabei sind die folgenden Punkte zu berücksichtigen:

1. Operationen zum Objektmodell der Analyse hinzufügen. Insbesondere Parameter und ggf. Ergebnistypen bestimmen und abstrakte Operationen identifizieren.
2. Algorithmen der Operationen beschreiben und in Pseudo-Code dokumentieren.
3. Assoziationen ggf. uni-direktional ausrichten; ggf. abgeleitete Assoziationen hinzunehmen.
4. Zugriffsrechte für Attribute, Rollen und Operationen bestimmen.
5. Klassendiagramm der Analyse dabei laufend überarbeiten.

Der Objektentwurf ergibt sich aus der Integration des statischen und dynamischen Modells der Analyse:





Algorithmus der Vorlesung in 5 Schritten:

1. Operationen

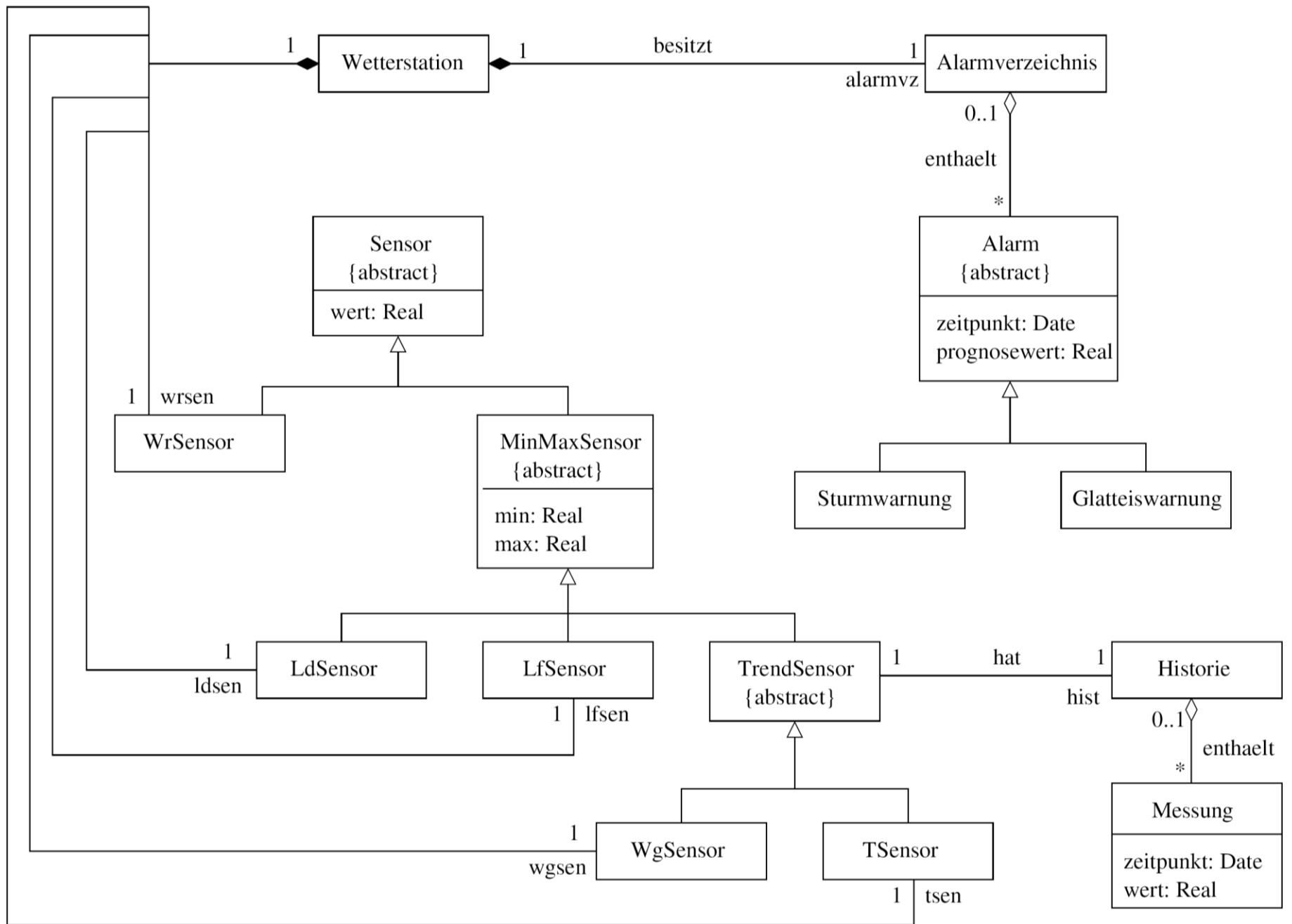
Heute

1. Operationen aus Sequenz-, Zustands-, und Aktivitätsdiagrammen übernehmen
2. Algorithmen definieren (=> führt ggf. zu weiteren Operationen)

2. Assoziationen ausrichten
3. Zugriffsrechte bestimmen
4. Mehrfachvererbung auflösen
5. Wiederverwendung von Klassen

- Die Vorlesung (IV, 4.1, Folien 5-8) definiert einen Algorithmus für Schritt 1. Für jede Klasse K:

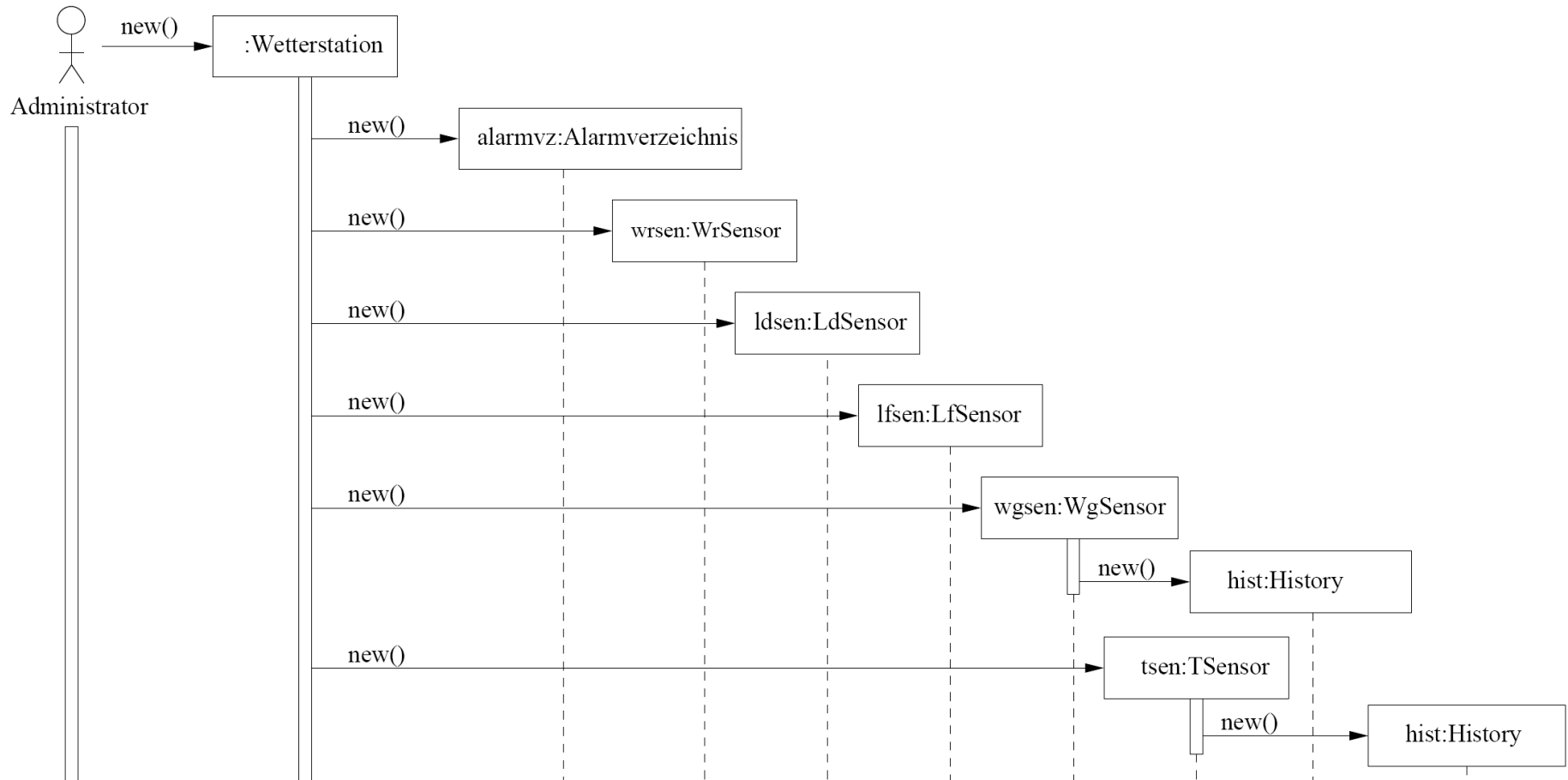
Aktion	Input
Füge Operation für jede an ein Objekt von K gesendete Nachricht hinzu (ggf. auch Konstruktoren!)	Interaktionsdiagramme , die K enthalten
Füge Operation für jedes Call Event, jeden (lokalen) Aktivitätsaufruf und jede Aktion hinzu (ggf. auch Konstruktoren!)	Zustandsdiagramme für K Aktivitätsdiagramme für Operationen von K
Füge benötigte Zugriffsoperationen (Getter/Setter) für Attribute und Rollen hinzu (zum Lesen und/oder Schreiben)	Klassendiagramm
Beschreibe Algorithmen der Operationen (führt ggf. zu weiteren Assoziationen und Operationen) – als Pseudocode, Interaktions- oder Aktivitätsdiagramme	Interaktionsdiagramme , die K enthalten Aktivitätsdiagramme für Operationen von K



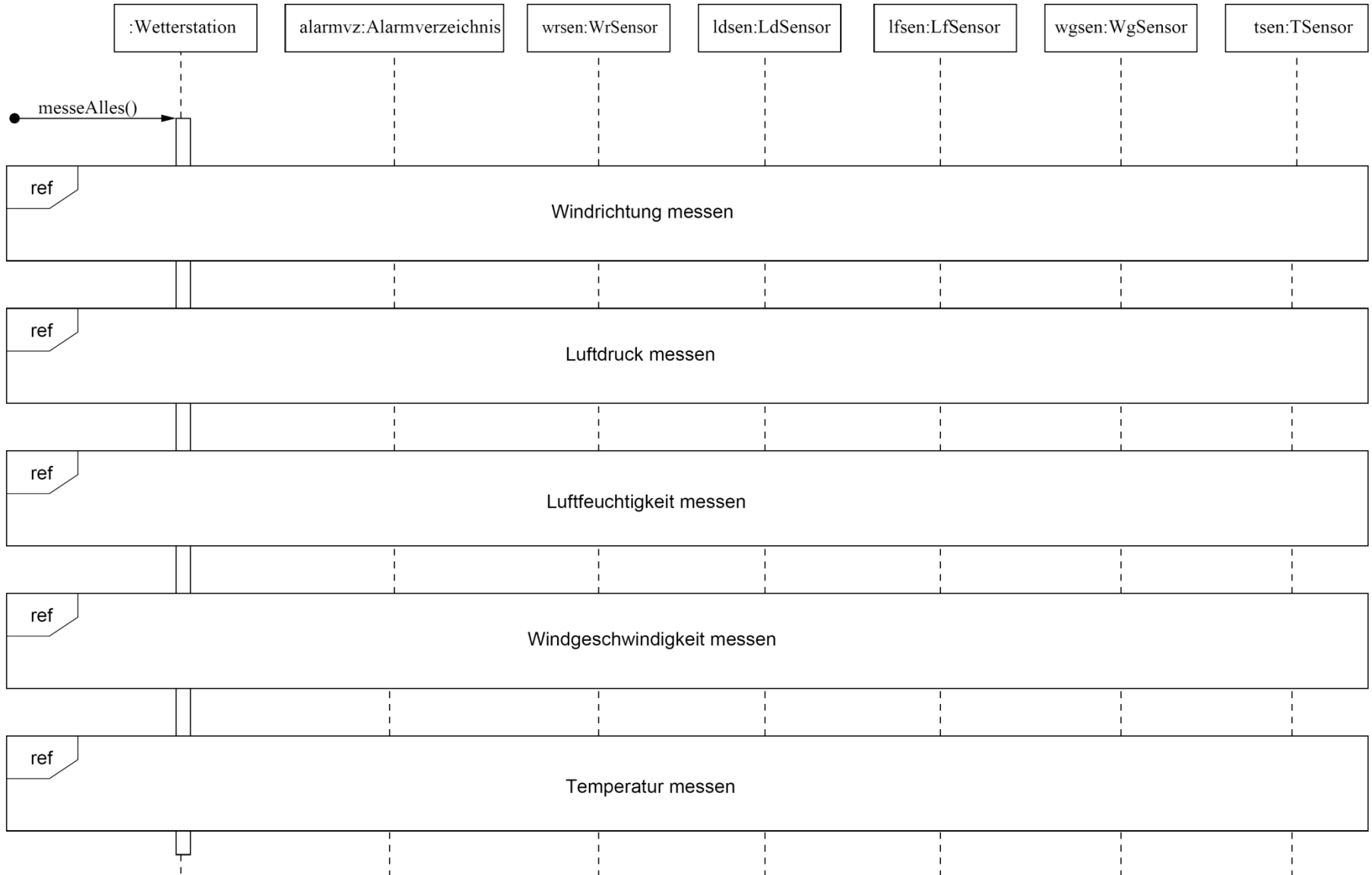


- Führe Schritt 1 für „einfache“ Klassen durch:
 - **Wetterstation**
 - **WrSensor** (bzw. **Sensor**)
- **Hands-On (am Klassendiagramm)**

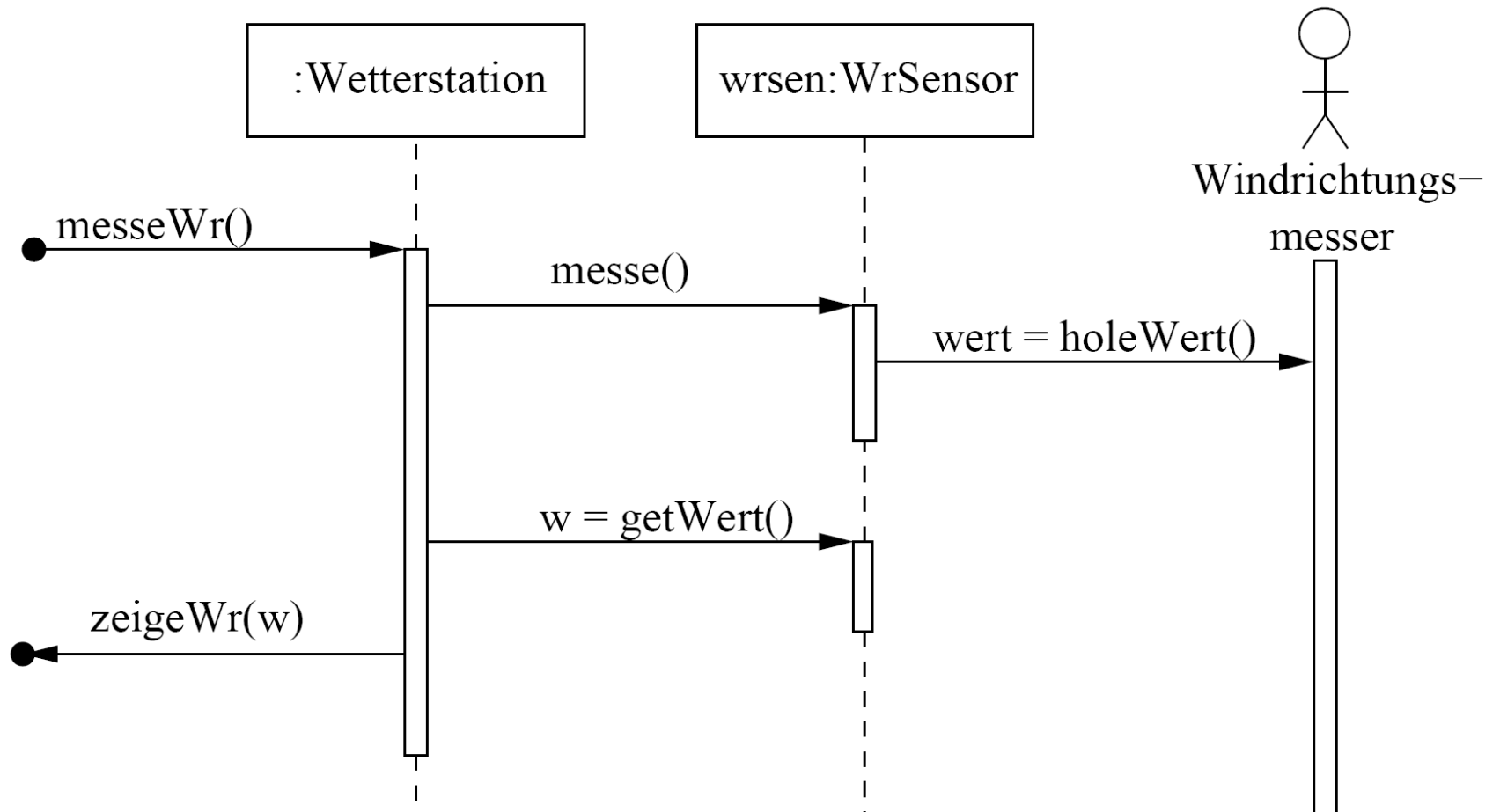
Sequenzdiagramm „Wetterstation starten“



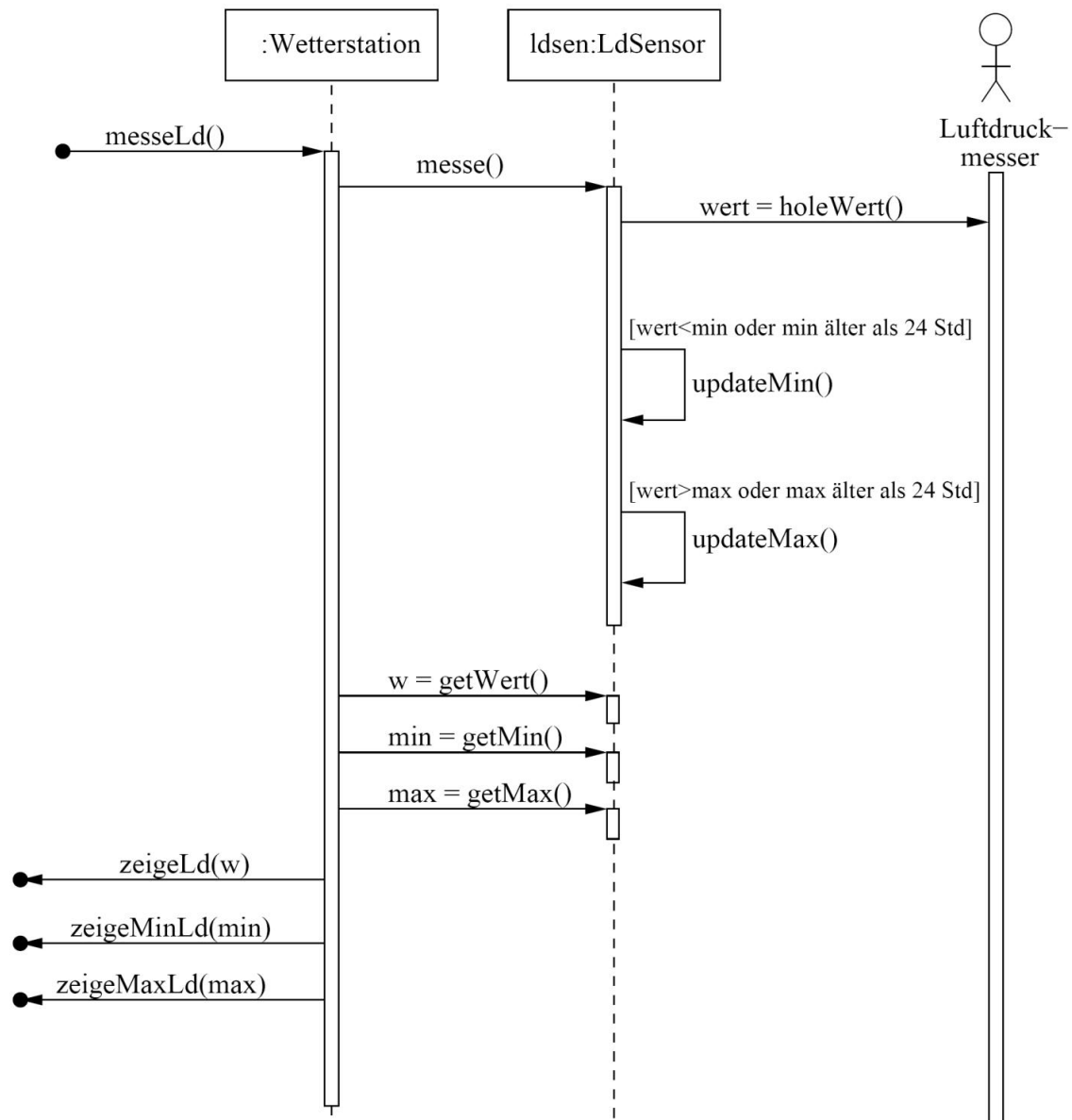
Sequenzdiagramm „Werte messen“



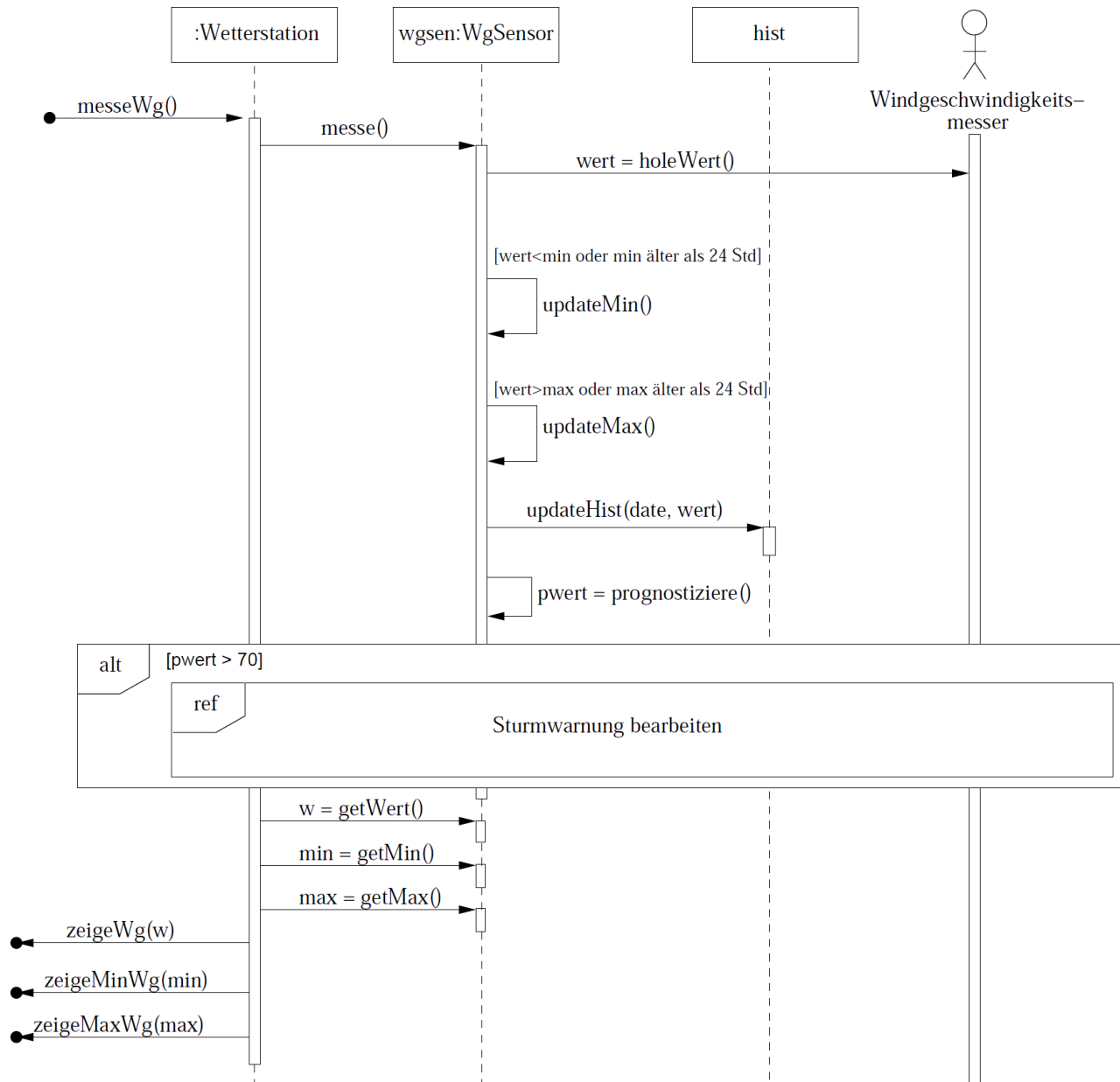
Sequenzdiagramm „Windrichtung messen“



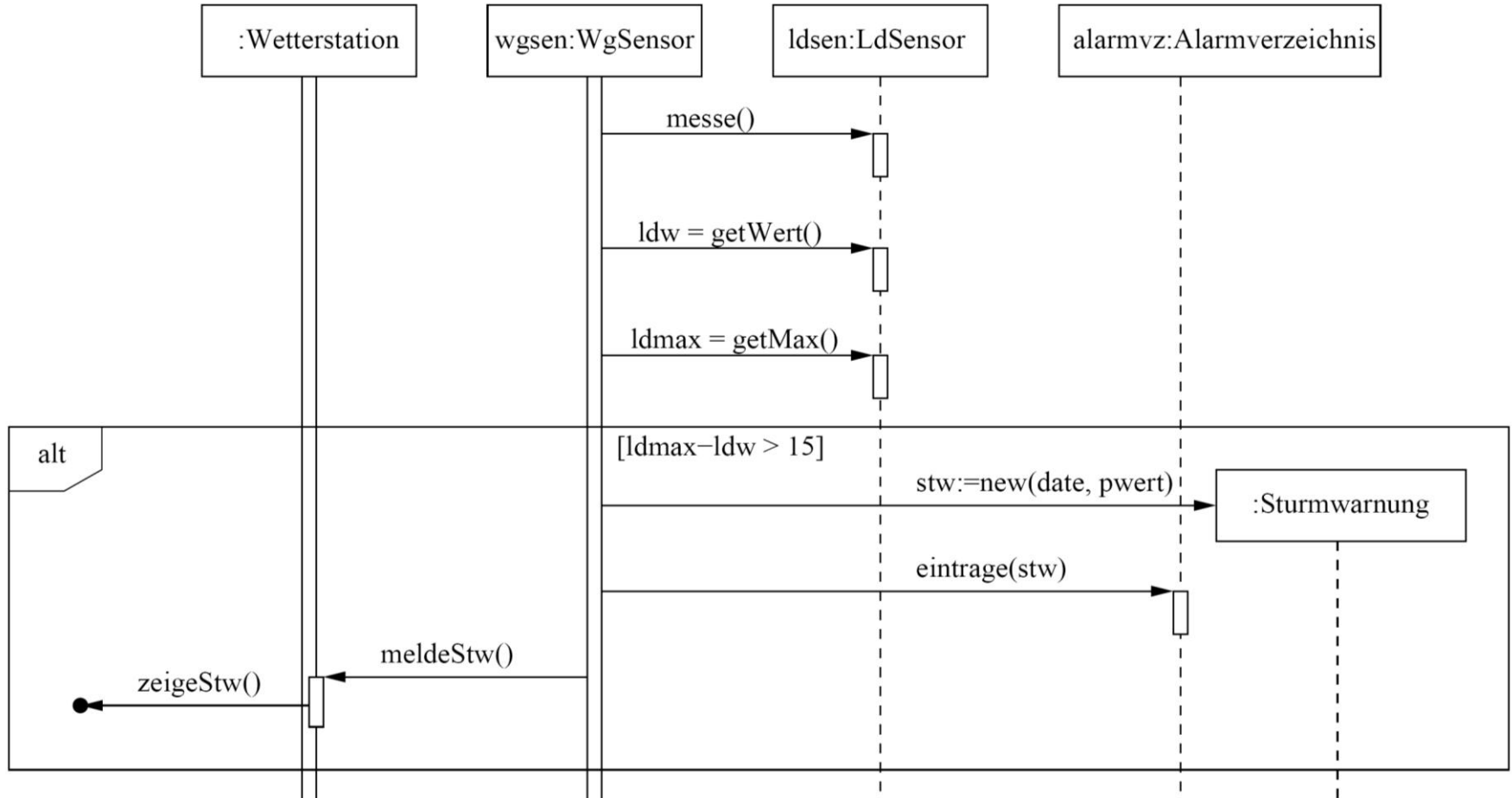
Sequenzdiagramm „Luftdruck messen“



Sequenzdiagramm „Windgeschwindigkeit messen“

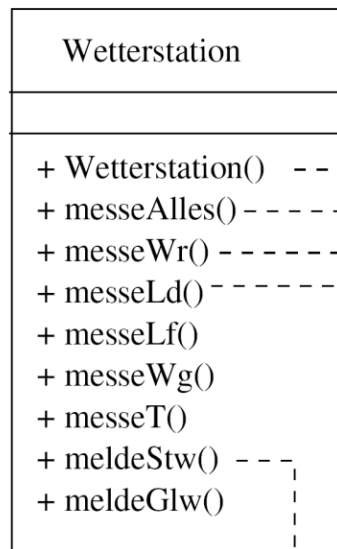


Sequenzdiagramm „Sturmwarnung bearbeiten“





- Nach Durchführung ist ein neues Klassendiagramm entstanden.



```

alarmvz = new Alarmverzeichnis();
wrsen = new WrSensor();
ldsen = new LdSensor();
lfsen = new LfSensor();
wgsen = new WgSensor(ldsen, alarmvz, this);
tsen = new TSensor(lfsen, alarmvz, this);

```

```

messeWr();
messeLd();
messeLf();
messeWg();
messeT();

```

```

wrsen.messe();
Real w = wrsen.getWert();
gui.zeigeWr(w);

```

```

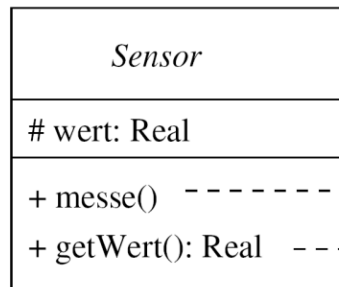
ldsen.messe();
Real w = ldsen.getWert();
Real min = ldsen.getMin();
Real max = ldsen.getMax();
gui.zeigeLd(w);
gui.zeigeMinLd(min);
gui.zeigeMaxLd(max);

```

```

gui.zeigeStw();

```



```

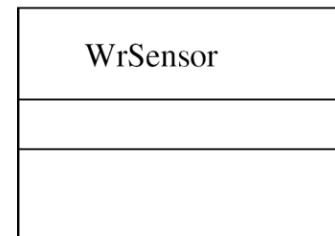
wert = ioPort.holeWert();

```

```

return wert;

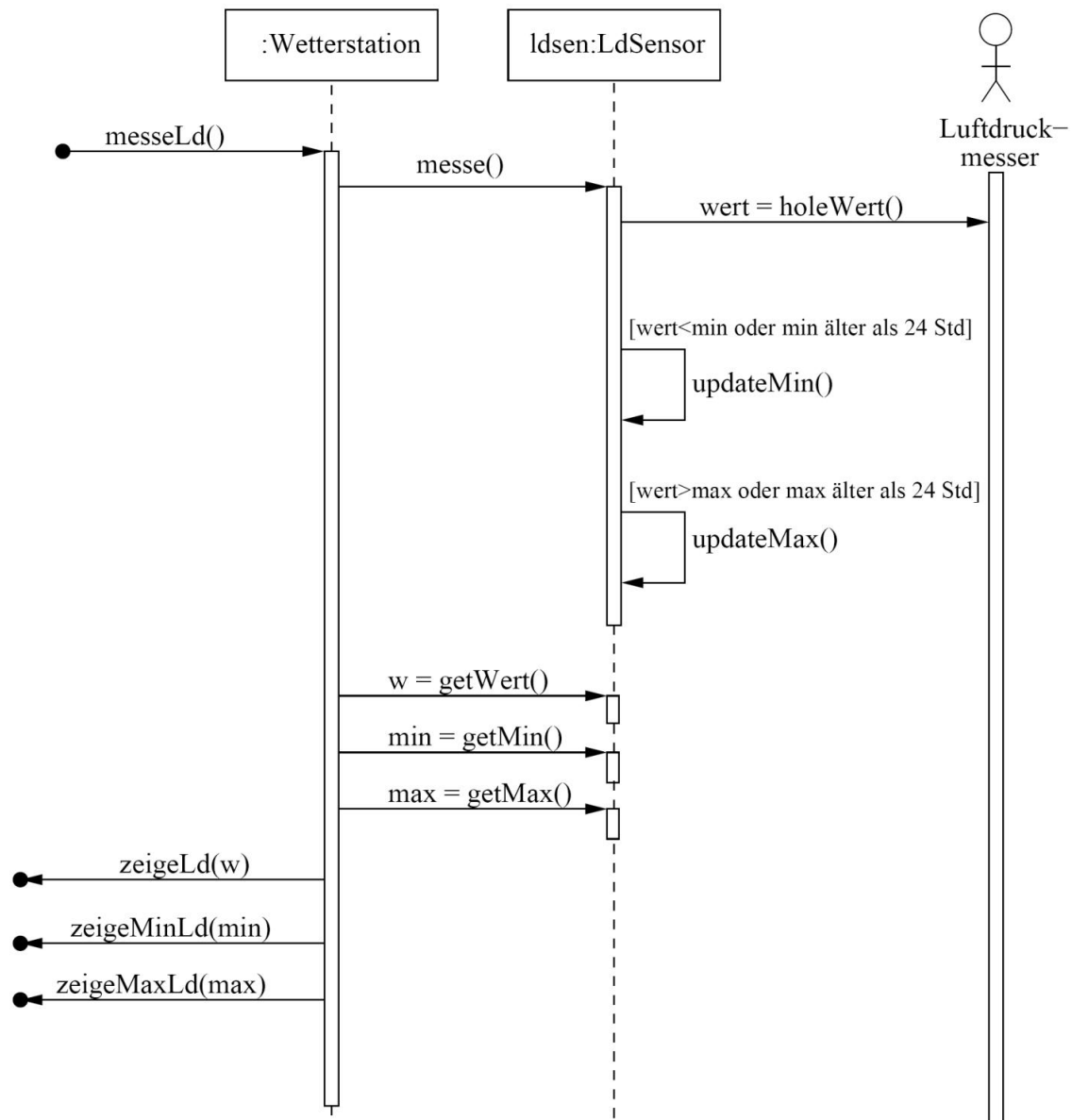
```



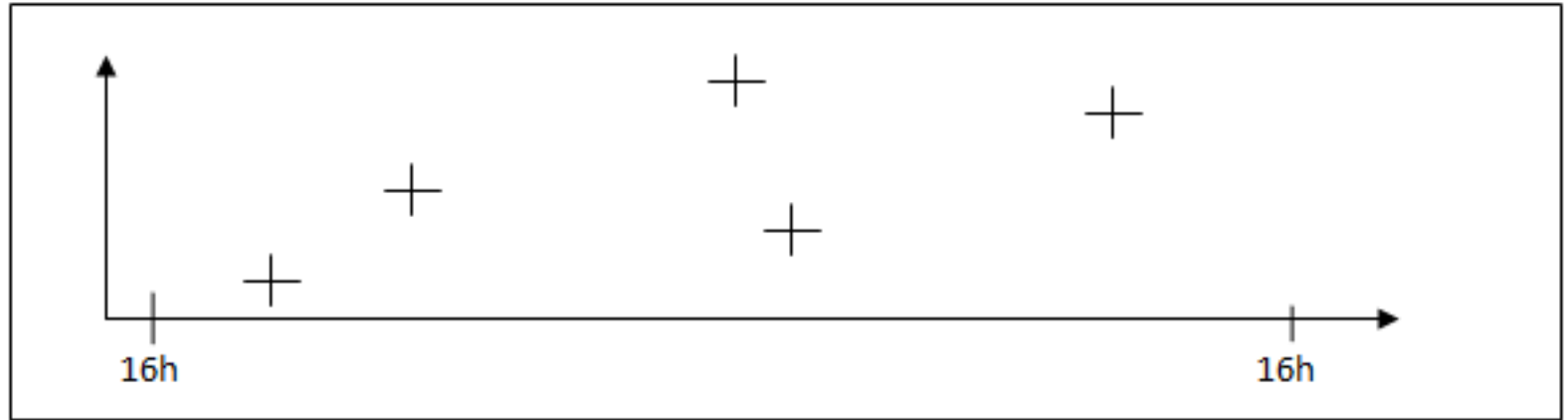


- Führe Schritt 1 für „komplexere“ Klassen durch:
- **Zunächst: LdSensor/LfSensor (bzw. MinMaxSensor)**
- **Hands-On (am Klassendiagramm)**

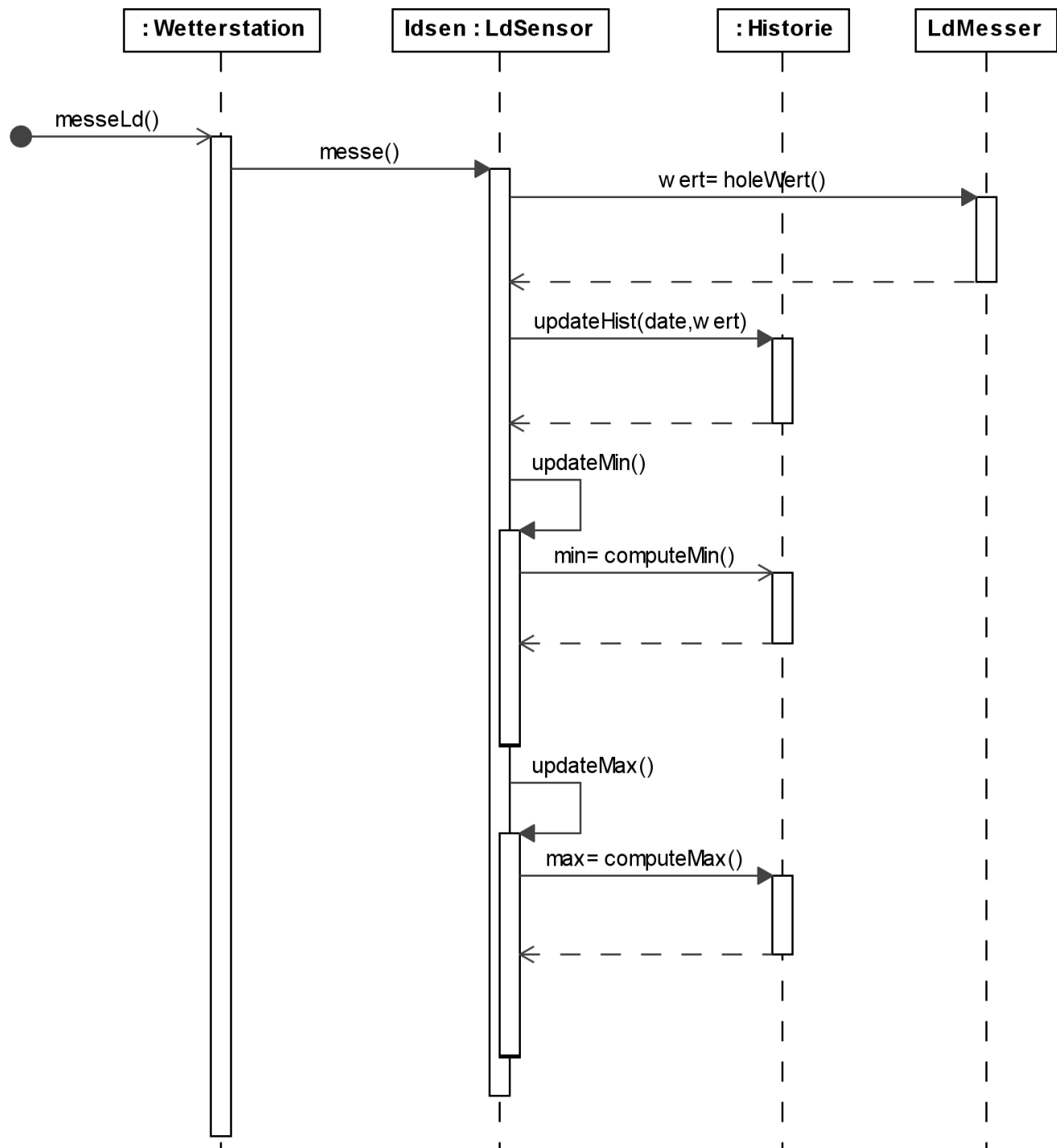
Sequenzdiagramm „Luftdruck messen“



- Interessant: **updateMin()** / updateMax()
- Was passiert, wenn die 24-Stunden-Marke überschritten wird und das aktuelle Minimum herausfällt?

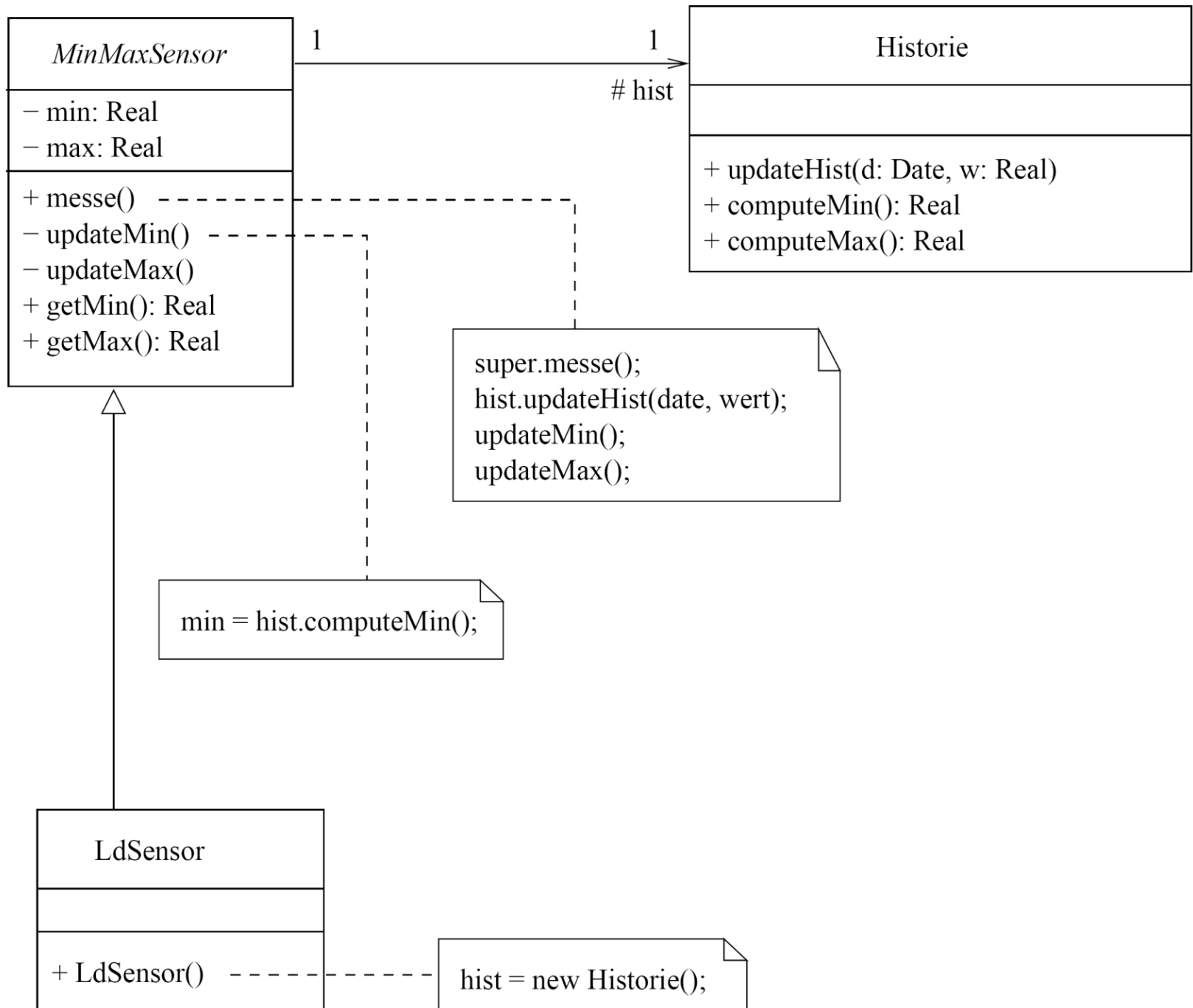


- Implementierung von `messe()` / `updateMin()` in `LdSensor` erfordert:
 - Speicherung aller Werte der letzten 24h
 - D.h. Verbindung von `LdSensor` mit Historie
 - D.h. Aktualisierung der Historie nach jeder Messung
 - D.h. Methoden zur Berechnung des aktuellen Minimums/Maximums erforderlich (z.B. in Historie).
 - Aktualisierung des Minimums bei jeder Messung
 - Frage: Wie wird „wert < min oder min älter als 24h“ implementiert?
 - Lösung: Gleiche Implementierung für beide Fälle
 - Einfach neu berechnen.





- Nach Durchführung ist ein neues Klassendiagramm entstanden.



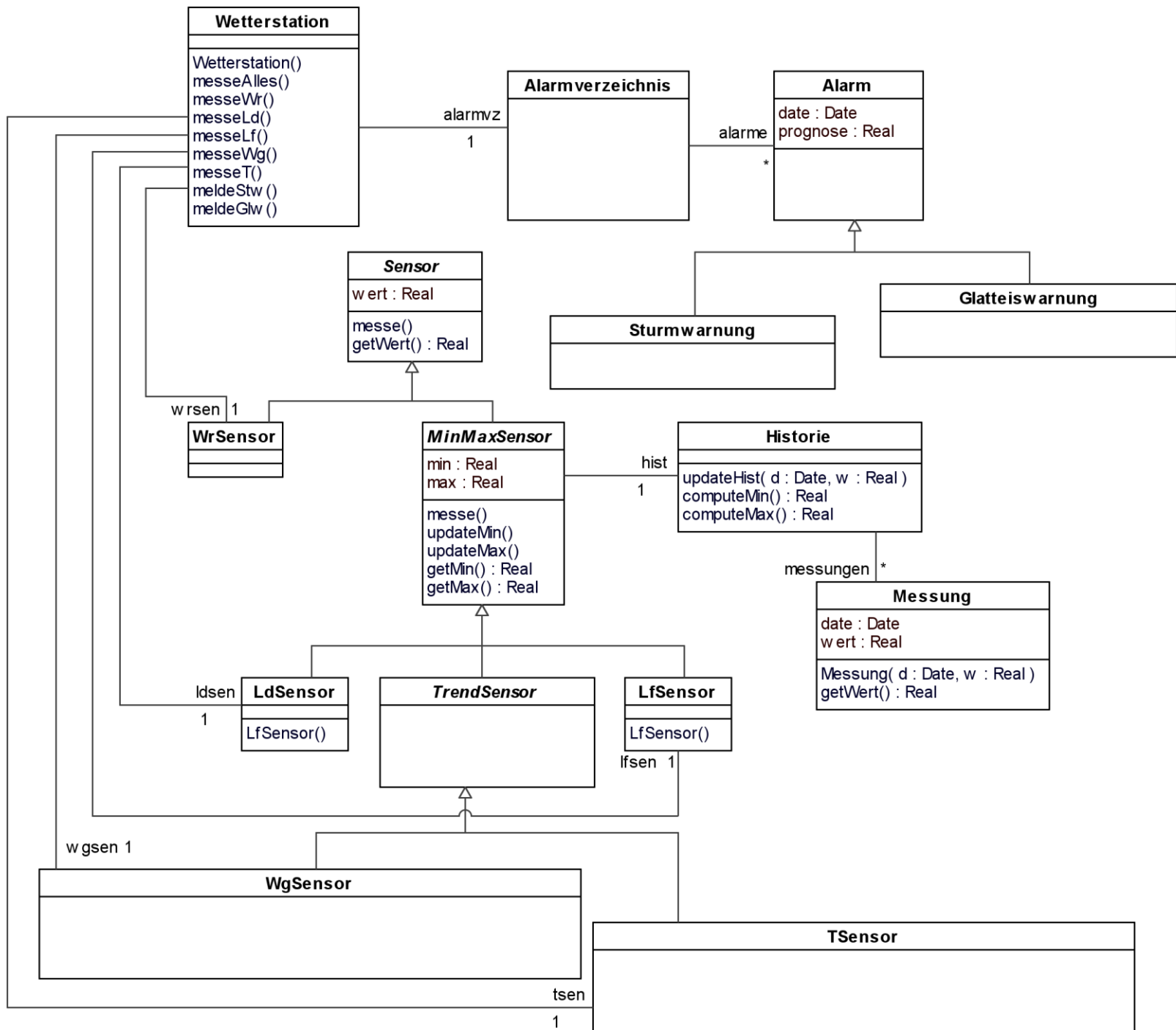
Softwaretechnik

WiSe 09/10

Übungsblatt 9 (Objektentwurf) – **Zweiter Teil** –

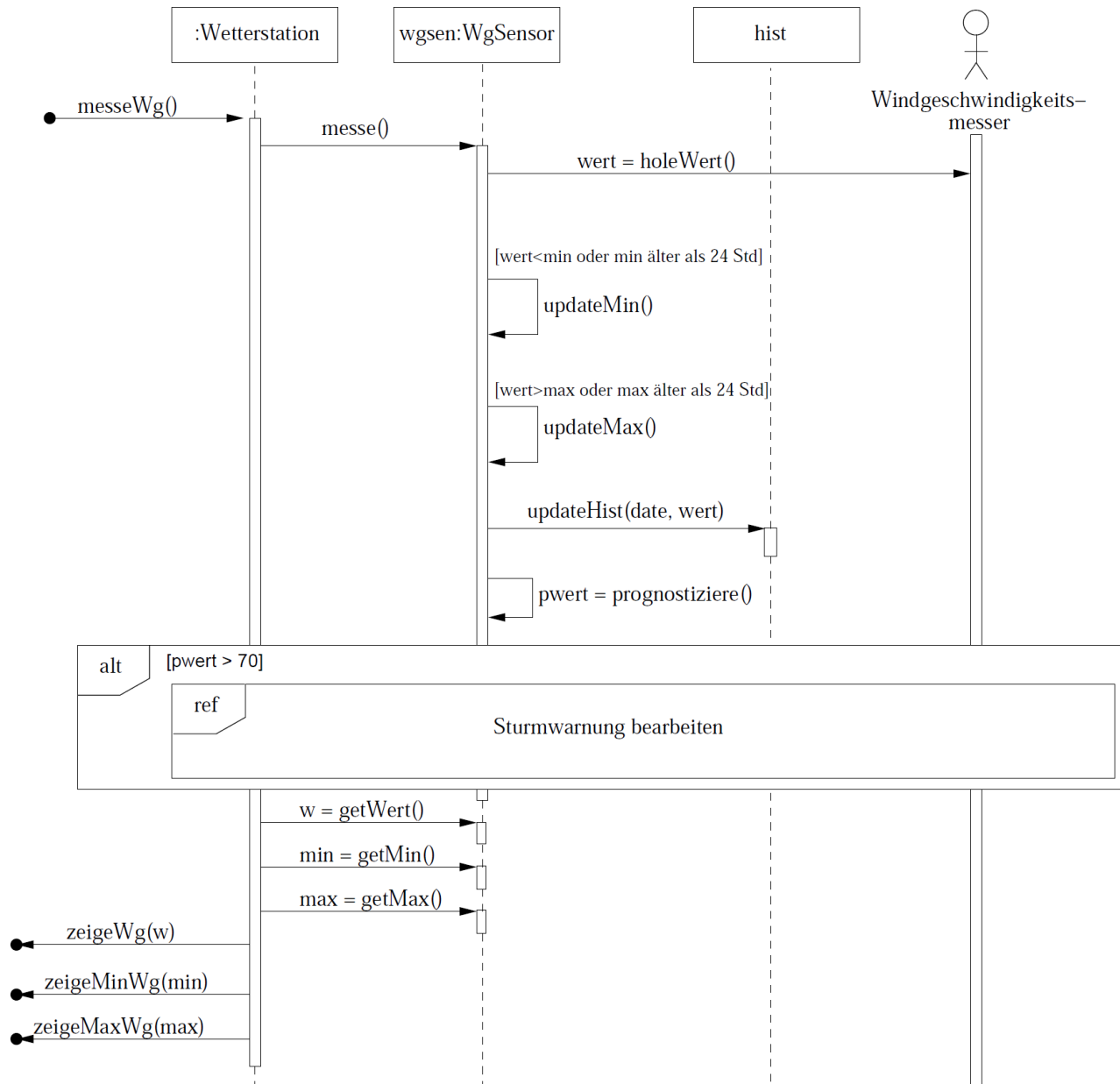
Algorithmus der Vorlesung in 5 Schritten:

1. Operationen
 1. Operationen aus Sequenz-, Zustand-, und Aktivitätsdiagrammen übernehmen
 2. Algorithmen definieren (=> führt ggf. zu weiteren Operationen)
2. Assoziationen ausrichten
3. Zugriffsrechte bestimmen
4. Mehrfachvererbung auflösen
5. Wiederverwendung von Klassen

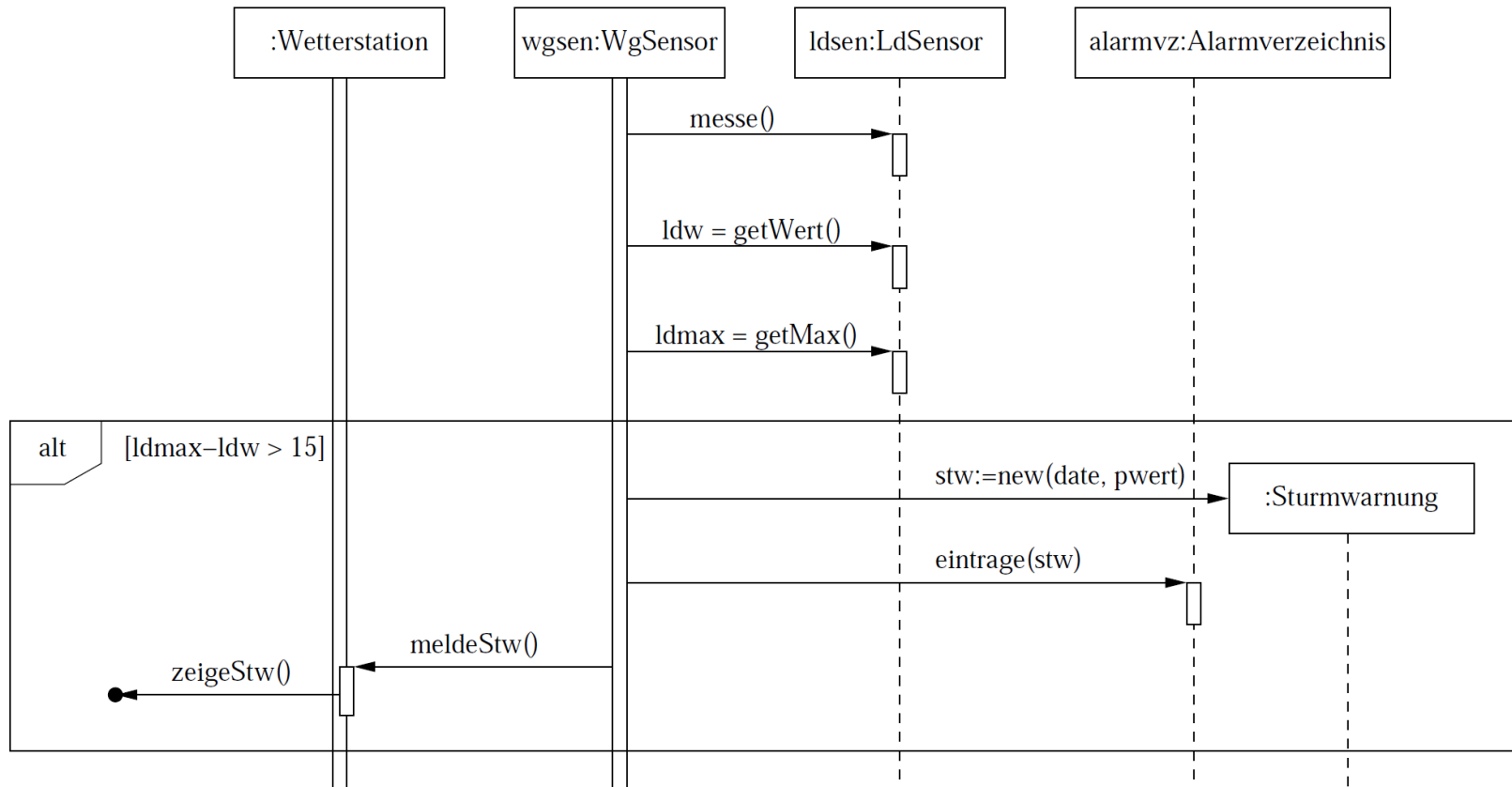


- Es fehlen noch die Operationen für **WgSensor** und **TSensor** (d.h. **TrendSensor**)
- **Hands-On (am Klassendiagramm)**

Sequenzdiagramm „Windgeschwindigkeit messen“

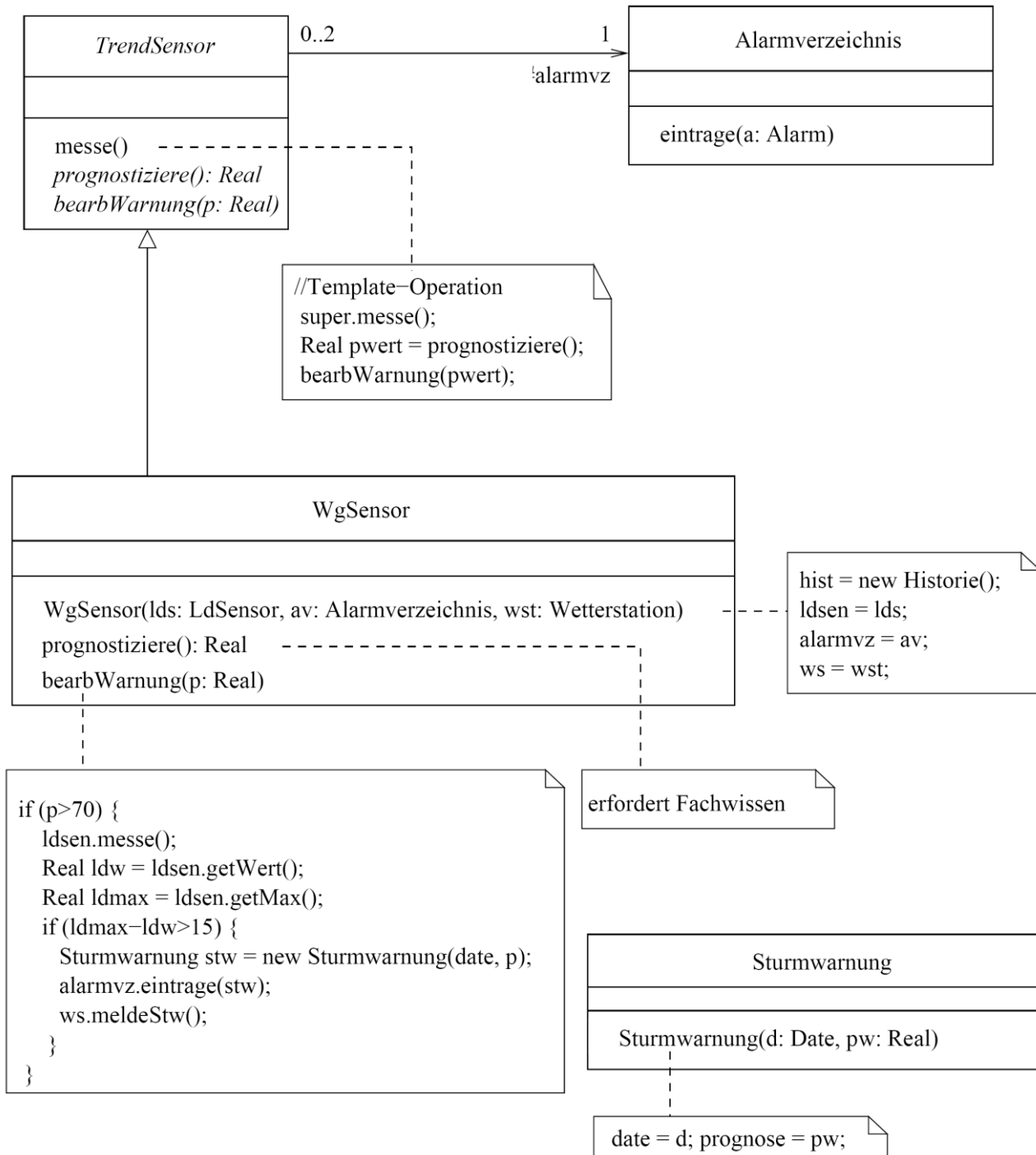


Sequenzdiagramm „Sturmwarnung bearbeiten“

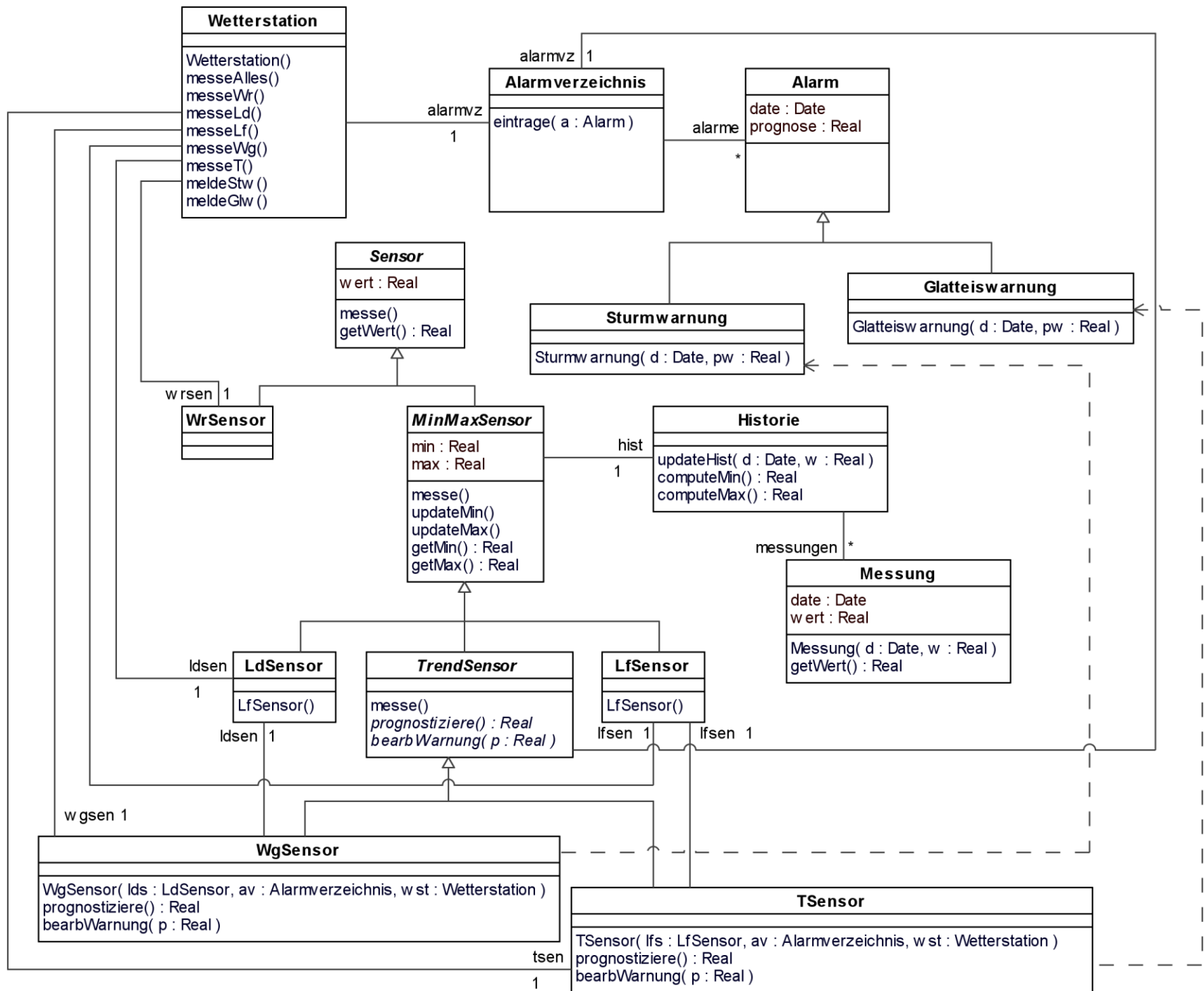




- Nach Durchführung ist ein neues Klassendiagramm entstanden.



- Nach Durchführung ist das Klassendiagramm gemäß Schritt 1 des Algorithmus der Vorlesung komplett.



- **Algorithmus der Vorlesung in 5 Schritten:**

1. Operationen

1. Operationen aus Sequenz-, Zustand-, und Aktivitätsdiagrammen übernehmen
2. Algorithmen definieren (=> führt ggf. zu weiteren Operationen)

2. Assoziationen ausrichten
3. Zugriffsrechte bestimmen
4. Mehrfachvererbung auflösen
5. Wiederverwendung von Klassen

- Analysiere in welcher/welchen Richtung(en) eine Assoziation (beim Senden von Nachrichten bzw. Operationsaufrufen) durchlaufen wird.
- Falls eine Assoziation nur in einer Richtung durchlaufen wird, dann richte sie entsprechend aus.
- **Wetterstation:**
 - Überprüfe Nachrichten!



- Bestimme die Zugriffsrechte für Attribute, Rollennamen und Operationen
- Beachte insbesondere: Attribute und Rollennamen sollten nicht öffentlich zugreifbar sein!
- **Wetterstation:**
 - Überprüfe Attribute & Rollennamen
 - Überprüfe Operationen



- Ist notwendig, wenn die Zielsprache keine Mehrfachvererbung für Klassen unterstützt (z.B. Java).
- Die Auflösung der Mehrfachvererbung ist möglich durch Einführung einer Schnittstelle.
- **Wetterstation:** Keine Mehrfachvererbung genutzt.



- Häufig ist es günstig, schon vorhandene (wohlerprobte und qualitativ hochwertige) Klassen im Entwurf wiederzuverwenden
- Die beiden gängigen Mechanismen hierfür sind:
 - Subclassing
 - Delegation
- **Wetterstation:**
 - Diskussion der Kandidaten

Übung 9

ENDE