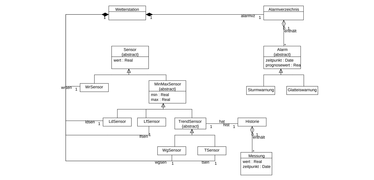
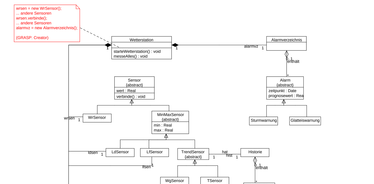
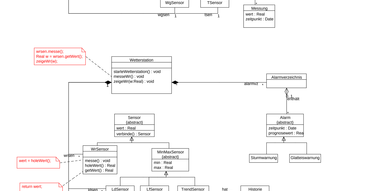
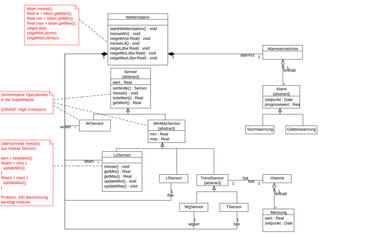
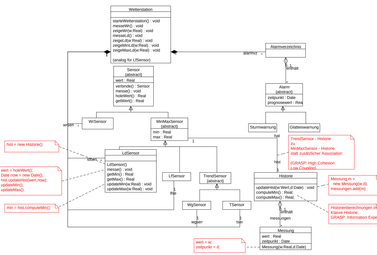
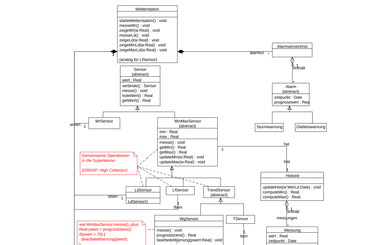

Wetterstation
Alarmverzeichnis
alarmist
1
1
Sensor
[abstract]
wert : Real
Alarm
[abstract]
zeitpunkt : Date
prognosewert : Real
WrSensor
MinMaxSensor
[abstract]
min : Real
max : Real
Sturmwarnung
Gleitwarnung
LtSensor
LtSensor
TrendSensor
[abstract]
Historie
WgSensor
TSensor
Messung
wert : Real
zeitpunkt : Date

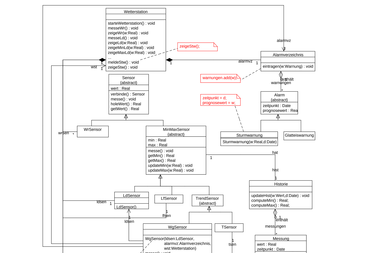

sensor = new WrSensor();
// ordnet Sensoren
sensor.verbinde();
// ordnet Sensoren
alarmist = new Alarmverzeichnis();
//GRASP: Creator
Wetterstation
startWetterstation() : void
messeWerte() : void
Alarmverzeichnis
Sensor
[abstract]
wert : Real
verbinde() : void
Alarm
[abstract]
zeitpunkt : Date
prognosewert : Real
WrSensor
MinMaxSensor
[abstract]
min : Real
max : Real
Sturmwarnung
Gleitwarnung
LtSensor
LtSensor
TrendSensor
[abstract]
Historie
WgSensor
TSensor
Messung
wert : Real
zeitpunkt : Date

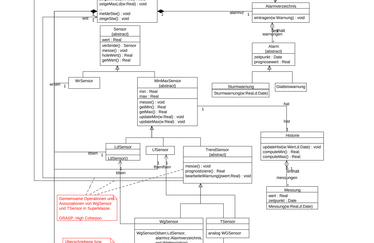

sensor.messe();
Real w = sensor.getWert();
zeigeWert(w);
Wetterstation
startWetterstation() : void
messeWerte() : void
zeigeWert(w Real) : void
Alarmverzeichnis
Sensor
[abstract]
wert : Real
verbinde() : Sensor
Alarm
[abstract]
zeitpunkt : Date
prognosewert : Real
WrSensor
MinMaxSensor
[abstract]
min : Real
max : Real
Sturmwarnung
Gleitwarnung
wert = holeWert();
messe() : void
holeWert() : Real
getWert() : Real
return wert;
LtSensor
LtSensor
TrendSensor
[abstract]
Historie
WgSensor
TSensor
Messung
wert : Real
zeitpunkt : Date

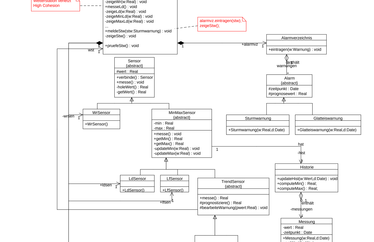

sensor.messe();
Real w = sensor.getWert();
zeigeWert(w);
Wetterstation
startWetterstation() : void
messeWerte() : void
zeigeWert(w Real) : void
messeLt() : void
zeigeLtAw Real) : void
zeigeLtBw Real) : void
zeigeMittelLtAw Real) : void
zeigeMittelLtBw Real) : void
Alarmverzeichnis
Sensor
[abstract]
wert : Real
verbinde() : Sensor
messe() : void
holeWert() : Real
getWert() : Real
Alarm
[abstract]
zeitpunkt : Date
prognosewert : Real
WrSensor
MinMaxSensor
[abstract]
min : Real
max : Real
Sturmwarnung
Gleitwarnung
Gemeinsame Operationen
in der Superklasse
//GRASP: High Cohesion
LtSensor
messe() : void
getWert() : Real
updateWert() : void
updateMax() : void
updateMin() : void
TrendSensor
[abstract]
Historie
WgSensor
TSensor
Messung
wert : Real
zeitpunkt : Date
//berechnet messe()
aus Klasse Sensor
wert = holeWert();
if(wert > max) {
updateMax();
}
if(wert < min) {
updateMin();
}
Problem: Zeit-Berechnung
benötigt Historie


Wetterstation
startWetterstation() : void
messeWerte() : void
zeigeWert(w Real) : void
messeLt() : void
zeigeLtAw Real) : void
zeigeLtBw Real) : void
zeigeMittelLtAw Real) : void
zeigeMittelLtBw Real) : void
Leading for LtSensor()
Alarmverzeichnis
Sensor
[abstract]
wert : Real
verbinde() : Sensor
messe() : void
holeWert() : Real
getWert() : Real
Alarm
[abstract]
zeitpunkt : Date
prognosewert : Real
WrSensor
MinMaxSensor
[abstract]
min : Real
max : Real
Sturmwarnung
Gleitwarnung
TrendSensor - Historie
MinMaxSensor - Historie
statt zustätzlicher Association
//GRASP: High Cohesion
//GRASP: Indirection
hist = new Historie();
LtSensor
LtSensor()
getWert() : Real
updateWert() : void
updateMax() : void
updateMin() : void
LtSensor
TrendSensor
[abstract]
Historie
updateWert(w Wert,d Date) : void
computeMx() : Real
computeMn() : Real
WgSensor
TSensor
Messung
wert : Real
zeitpunkt : Date
messungen
Messung in =
messungen.add(d);
messungen.add(m);
wert = holeWert();
Date tzpx = new Date();
hist.updateWert(wert,tzpx);
updateMax();
updateMin();
mw = hist.computeMx();
Historienberechnungen als
Operationen in Historie
//GRASP: Information Expert
wert = w;
zeitpunkt = d;
Messung
wert : Real
zeitpunkt : Date
Messungw.Real d.Date()


Wetterstation
startWetterstation() : void
messeWerte() : void
zeigeWert(w Real) : void
messeLt() : void
zeigeLtAw Real) : void
zeigeLtBw Real) : void
zeigeMittelLtAw Real) : void
zeigeMittelLtBw Real) : void
(analog for LtSensor)
Alarmverzeichnis
Sensor
[abstract]
wert : Real
verbinde() : Sensor
messe() : void
holeWert() : Real
getWert() : Real
Alarm
[abstract]
zeitpunkt : Date
prognosewert : Real
WrSensor
MinMaxSensor
[abstract]
min : Real
max : Real
Sturmwarnung
Gleitwarnung
Gemeinsame Operationen
in der Superklasse
//GRASP: High Cohesion
LtSensor
LtSensor
TrendSensor
[abstract]
Historie
updateWert(w Wert,d Date) : void
computeMx() : Real
computeMn() : Real
WgSensor
TSensor
Messung
wert : Real
zeitpunkt : Date
messungen
sie MinMaxSensor messe(); plus:
Real gwert = prognosewert();
if(gwert > max) {
beachteWarnungAuswert();
}
Messung in =
messungen.add(d);
messungen.add(m);
Historienberechnungen als
Operationen in Historie
//GRASP: Information Expert
Messungw.Real d.Date()
//kann messe() :void
Real tw = kein getWert();
Real lWert = kein getWert();
if(tw > lWert) {
Date tzwx = new Date();
Sturmwarnung sw =
new Sturmwarnung(gwert, now);
sw.alarmist.entragen(this);
sw.messe();
}
//GRASP: Low Coupling


Wetterstation
startWetterstation() : void
messeWerte() : void
zeigeWert(w Real) : void
messeLt() : void
zeigeLtAw Real) : void
zeigeLtBw Real) : void
zeigeMittelLtAw Real) : void
zeigeMittelLtBw Real) : void
zeigeSw();
Alarmverzeichnis
entragenw Warnung) : void
warnungen
warnungen.add(w);
Sensor
[abstract]
wert : Real
verbinde() : Sensor
messe() : void
holeWert() : Real
getWert() : Real
Alarm
[abstract]
zeitpunkt : Date
prognosewert : Real
WrSensor
MinMaxSensor
[abstract]
min : Real
max : Real
Sturmwarnung
Gleitwarnung
messe() : void
getWert() : Real
updateWert() : void
updateMax() : void
updateMin() : void
Sturmwarnung
Sturmwarnungw.Real d.Date()
Gleitwarnung
LtSensor
LtSensor
TrendSensor
[abstract]
Historie
updateWert(w Wert,d Date) : void
computeMx() : Real
computeMn() : Real
WgSensor
TSensor
Messung
wert : Real
zeitpunkt : Date
messungen
WgSensor(sWetter:LtSensor,
sAlarm:Alarmverzeichnis,
sw:Wetterstation)
messe() : void
prognosewert() : Real
beachteWarnungAuswert() : void
this.sWetter = sWetter;
this.sAlarm = sAlarm;
this.sw = sw;
//Code für startWetterstation
in Klasse Wetterstation
muss angepasst werden!!
Messungw.Real d.Date()


Wetterstation
startWetterstation() : void
messeWerte() : void
zeigeWert(w Real) : void
messeLt() : void
zeigeLtAw Real) : void
zeigeLtBw Real) : void
zeigeMittelLtAw Real) : void
zeigeMittelLtBw Real) : void
zeigeSw();
Alarmverzeichnis
entragenw Warnung) : void
warnungen
warnungen.add(w);
Sensor
[abstract]
wert : Real
verbinde() : Sensor
messe() : void
holeWert() : Real
getWert() : Real
Alarm
[abstract]
zeitpunkt : Date
prognosewert : Real
WrSensor
MinMaxSensor
[abstract]
min : Real
max : Real
Sturmwarnung
Sturmwarnungw.Real d.Date()
Gleitwarnung
messe() : void
getWert() : Real
updateWert() : void
updateMax() : void
updateMin() : void
LtSensor
LtSensor
TrendSensor
[abstract]
Historie
updateWert(w Wert,d Date) : void
computeMx() : Real
computeMn() : Real
WgSensor
TSensor
Messung
wert : Real
zeitpunkt : Date
messungen
Messungw.Real d.Date()
Gemeinsame Operationen und
Attribute in Superklasse
und TSensor in Superklasse
//GRASP: High Cohesion
WgSensor(sWetter:LtSensor,
sAlarm:Alarmverzeichnis,
sw:Wetterstation)
messe() : void
prognosewert() : Real
beachteWarnungAuswert() : void
TSensor
analog WGSensor
Überschreibende Sw
implementierte Methoden
WgSensor


Wetterstation
startWetterstation() : void
messeWerte() : void
zeigeWert(w Real) : void
messeLt() : void
zeigeLtAw Real) : void
zeigeLtBw Real) : void
zeigeMittelLtAw Real) : void
zeigeMittelLtBw Real) : void
zeigeSw();
Wetterstation verteilt
High Cohesion
alarmist.entragen(this);
//zeigeSw();
Alarmverzeichnis
entragenw Warnung) : void
warnungen
warnungen.add(w);
Sensor
[abstract]
wert : Real
verbinde() : Sensor
messe() : void
holeWert() : Real
getWert() : Real
Alarm
[abstract]
zeitpunkt : Date
prognosewert : Real
WrSensor
MinMaxSensor
[abstract]
min : Real
max : Real
Sturmwarnung
Sturmwarnung Real d.Date()
Gleitwarnung
Gleitwarnung Real d.Date()
messe() : void
getWert() : Real
updateWert() : void
updateMax() : void
updateMin() : void
LtSensor
LtSensor
TrendSensor
[abstract]
Historie
updateWert(w Wert,d Date) : void
computeMx() : Real
computeMn() : Real
WgSensor
TSensor
Messung
wert : Real
zeitpunkt : Date
messungen
Messung(w Real,d Date)
Messungw.Real d.Date()
WgSensor(sWetter:LtSensor,
sAlarm:Alarmverzeichnis,
sw:Wetterstation)
messe() : void
prognosewert() : Real
beachteWarnungAuswert() : void
TSensor
analog WGSensor
this.sWetter = sWetter;
this.sAlarm = sAlarm;
this.sw = sw;
//Code für startWetterstation
in Klasse Wetterstation
muss angepasst werden!!
WgSensor