Informatik 2 für Regenerative Energien

Klausur vom 8. Juli 2022: Lösungen

Jörn Loviscach

Versionsstand: 13. Juli 2022, 20:34

1. Die Fehler:

Zeile	korrekter Programmtext
15	`Messstation[] stationen ...`
24	`...ZahlDerWarnungen;`
34	`public int ...`
42	`...(Sensor sensor)`
53	`...GibAktuellenWert();`
80	`&& Math.Abs(...`
86	`abstract class Sensor`
96	`protected static ...`
106	`class CO2Sensor : Sensor`
109	`static int ...`
117	`public override ...`
123	`double wert ...`
125	`... Station != null)`
130	`return new ...`
160	`public bool ...`

2. `x` ist `2`, weil an zwei Sensoren jeweils einmal gemessen worden ist. `y` ist `2`, weil beide Messungen an der angegebenen Position waren. `z` ist `2`, weil beide Messwerte ≥ 500 sind.

3. *Zum Beispiel:* Am Anfang der besagten Methode dies ergänzen:

```
if(hatPositionsangabe)
{
    throw new ApplicationException("Hat schon eine Position!");
}
```

4. *Zum Beispiel:* In der Klasse `Messwert` ergänzen:

```
public double Wert { get { return wert; } }
public Sensortyp Sensortyp { get { return sensortyp; } }
```

Und in der Klasse `Luftbelastungsdatenbank` die gefragte Methode so implementieren:

```
public double BestimmeMaximumVonCO2()
{
    bool etwasGefunden = false;
    double max = double.MinValue;

    foreach (Messwert messwert in messwerte)
    {
        if (messwert.Sensortyp == Sensortyp.CO2
            && messwert.Wert > max)
        {
            max = messwert.Wert;
            etwasGefunden = true;
        }
    }

    if (etwasGefunden)
    {
        return max;
    }
    else
    {
        return double.NaN;
    }
}
```

Oder eleganter dies als Körper der obigen Funktion:

```
var co2Messungen = messwerte.FindAll(
                        mw => mw.Sensortyp == Sensortyp.CO2);
if (co2Messungen.Count == 0)
{
    return double.NaN;
}
return co2Messungen.Max(mw => mw.Wert);
```

5. *Zum Beispiel:*

```
class MobileMessstation : Messstation
{
    public MobileMessstation(string name,
                             Geokoordinaten anfangsposition)
        : base(name, anfangsposition)
    { }

    public void ÄnderePosition(Geokoordinaten neuePosition)
    {
        position = neuePosition;
    }
}
```

   Und in der Klasse `Messstation` das Feld `Geokoordinaten position` `protected` machen.

6. *Zum Beispiel:* In der Klasse `CO2Sensor` dieses Feld hinzufügen:

```
List<DateTime> zeitpunkteDerÜberschreitungen =
                                          new List<DateTime>();
```

   Und in derselben Klasse diese Methode hinzufügen (alternativ mit einer entsprechenden `for`-Schleife):

```
public int ZähleÜberschreitungenDerVergangenen24h()
{
    return zeitpunkteDerÜberschreitungen.Count(
            z => DateTime.Now - z <= TimeSpan.FromHours(24.0));
}
```

   Und am Ende der Methode `GibAktuellenWert` der Klasse `CO2Sensor` dies ergänzen bzw. ändern:

```
DateTime now = DateTime.Now;
if (wert >= limitFürWarnungen)
{
    zeitpunkteDerÜberschreitungen.Add(now);
}
return new Messwert(wert, Sensortyp.CO2, now);
```

7. *Zum Beispiel:*

```
class B
{
    public virtual double f(int u)
    {
        return 7.0;
```

```
    }
}

abstract class D : B
{
    int x;

    public override double f(int u)
    {
        return 23.0;
    }

    abstract public bool g(float v);
}

class A : D
{
    double z;

    public override bool g(float v)
    {
        return true;
    }
}
```

8. Die Werte sind 11, 13, 12.