

# Informatik 2 für Regenerative Energien

## Klausur vom 30. September 2015: Lösungen

Jörn Loviscach

Versionsstand: 20. März 2016, 00:50



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

### 1. Die Fehler:

Zeile	korrekter Programmtext
25	Adresse adresse;
34	this.adresse = new Adresse(...
35	ereignisse = new List<Ereignis>());
42	foreach (Ereignis e in ereignisse)
50	return zählerstand;
53	public void TrageAblesungEin(...
55	... Ablesung(DateTime.Now, s, zählerstand));
73	if (e.Was == s && e is Zählerwechsel)
83	class Adresse
90	public Adresse(string straße, string hausnr,
100	abstract class Ereignis
110	public Ereignis(DateTime wann, Sparte was)
121	double zählerstand;
124	: base(wann, was)
153	public override double LetzterZählerstand()

### 2. Die Werte sind 2.0, 0.0, 0.0.

### 3. Am Anfang der Methode z. B. dies einfügen:

```
if(zählerstand < LetzterZählerstand(s))
{
    throw new ApplicationException("Zähler läuft zurück?");
}
```

### 4. Zum Beispiel so:

```
class Ablesung : Ereignis
{
    // ...
```

```

bool warSelbstablesung;

public Ablesung(DateTime wann, Sparte was,
                 double zählerstand, bool warSelbstablesung)
    : base(wann, was)
{
    // ...
    this.warSelbstablesung = warSelbstablesung;
}

// ...
}

```

*Das zieht dann in der Methode Kunde.TrageAblesungEin eine Änderung des Aufrufs nach sich!*

#### 5. Zum Beispiel so:

```

public string BaueZeichenkette()
{
    string s = vorname + " " + nachname; //besser: StringBuilder
    foreach (var e in ereignisse)
    {
        s += "\n" + e.Wann.ToString("d. MMM yyyy") + " " + e.Was
            + " " + e.LetzterZählerstand();
    }
    return s;
}

```

#### 6. Wenn keine Ereignisse der gewünschten Sparte vorhanden sind, ist die Liste sortiert **leer und** `sortiert.First()` **wirft eine Exception.**

#### 7. abstract class Bbb

```

{
    int d;
    public virtual double e(int x)
    {
        return 42.0;
    }
}

class Aaa : Bbb
{
    public override double e(int x)
    {
        return 13.0;
    }
}

```

```
class Ccc : Bbb
{
    double f;
}
c1
```

8. Die Werte sind 23, 15, 42.

c1.jl: 42 in 42.0  
geändert, weil  
so sauberer.