

# Informatik 2 für Regenerative Energien

## Klausur vom 27. Juli 2017: Lösungen

Jörn Loviscach

Versionsstand: 28. Juli 2017, 22:17



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
49	<code>public void Verbinde(Verbindung v)</code>
60	<code>public List&lt;Anschluss&gt; FindeVerbundeneAnschlüsse()</code>
62	<code>List&lt;Anschluss&gt; ergebnis = new List&lt;Anschluss&gt;();</code>
66	<code>for (int i = 0; i &lt; alle.Count; i++)</code>
70	<code>ergebnis.Add(alle[i]);</code>
74	<code>return ergebnis;</code>
96	<code>protected Anschluss[] anschlüsse;</code>
106	<code>public Anschluss GibAnschluss(string name)</code>
110	<code>if(anschlüsse[i].Name == name)</code>
115	<code>return null;</code>
118	<code>public bool PrüfeObAlleAnschlüsseVerbunden()</code>
132	<code>List&lt;Bauelement&gt; b = new List&lt;Bauelement&gt;();</code>
138	<code>Bauelement be = a[j].Element;</code>
166	<code>public Transistor(bool istPNP)</code>
167	<code>: base(istPNP ? "BC 177A" : "BC 107A")</code>

2. Die Werte sind `true`, `true`, `"R 1000 Ω"`.

3. Zum Beispiel am Anfang der Methode `Verbinde` der Klasse `Anschluss`:

```
if(verbundenMit != null)
{
    throw new ApplicationException(
        "Nur eine Verbindung pro Anschluss!");
}
```

4. Zum Beispiel so:

```
class Blockbatterie : Batterie
```

```

{
    public Blockbatterie()
        : base(9.0)
    {
    }
}

```

**5. Zum Beispiel in der Klasse Anschluss:**

```

public void Löse(Verbindung v)
{
    if(verbundenMit != null)
    {
        v.Entferne(this);
        verbundenMit = null;
    }
}

```

**Und in der Klasse Verbindung:**

```

public void Entferne(Anschluss a)
{
    verbundeneAnschlüsse.Remove(a);
}

```

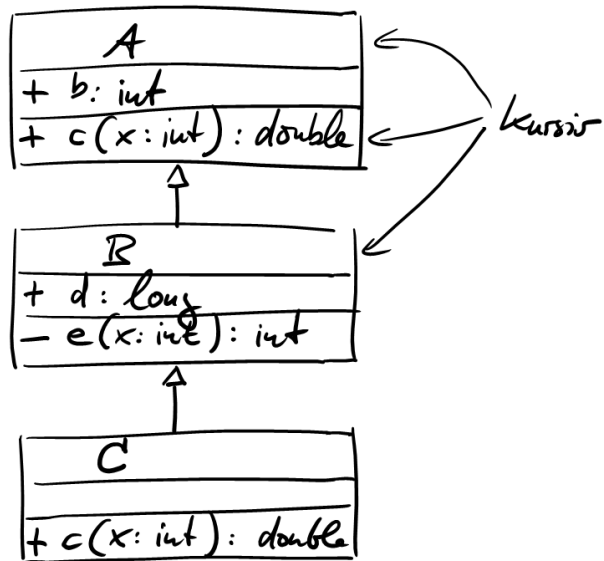
**6. Zum Beispiel so:**

```

public bool PrüfeObVerbunden(Bauelement b)
{
    for (int i = 0; i < anschlüsse.Length; i++)
    {
        List<Anschluss> verbundene
            = anschlüsse[i].FindeVerbundeneAnschlüsse();
        for (int j = 0; j < verbundene.Count; j++)
        {
            if (verbundene[j].Element == b)
            {
                return true;
            }
        }
    }
    return false;
}

```

7. Das UML-Diagramm:



8. Die Werte sind 2, 1, 2.