

Informatik 2 für Regenerative Energien

Klausur vom 3. April 2017: Lösungen

Jörn Loviscach

Versionsstand: 3. April 2017, 17:06



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
9	<code>int x = s4.Länge;</code>
29	<code>protected string name = "Nullsignal";</code>
35	<code>public virtual double GibAbtastwert(int nummer)</code>
45	<code>public double BestimmeEffektivwert()</code>
47	<code>double summe = 0.0;</code>
50	<code>double d = GibAbtastwert(i);</code>
57	<code>class KonstantesSignal : Signal</code>
76	<code>double frequenz;</code>
87	<code>return Math.Sin(2.0 * ...</code>
100	<code>originalsignal = s;</code>
107	<code>return verstärkungsfaktor * originalsignal.Gib...</code>
122	<code>return abtastwerte[nummer];</code>
143	<code>class SignalAusDatei : SignalMitDatenspeicher</code>
164	<code>: base(s1.Abtastfrequenz)</code>
173	<code>public override double GibAbtastwert(int nummer)</code>

Das Programm ist ein Beispiel für das Design-Pattern „Decorator“.

2. Die Werte sind 3, 48.0, "Summe von Konstante 23, verstärkt um Faktor 2 und Signal aus Datei test.txt".

3. Zum Beispiel in der for-Schleife im Konstruktor:

```
double d = double.Parse(zeilen[i]);
if(d > 1.0 || d < -1.0)
{
    throw new ApplicationException("Außerhalb Wertebereich!");
}
abtastwerte[i] = d;
```

4. Zum Beispiel könnte man `BestimmeEffektivwert` in der Klasse `Signal` virtuell machen. Dann könnte man diese Methode in den Klassen `KonstantesSignal` und `Sinusschwingung` durch eine spezifischere Methode überschreiben. Diese könnte einfach den jeweils sofort bekannten Effektivwert zurückliefern, nämlich die Konstante beziehungsweise `Math.Sqrt(0.5)`.
5. In der Klasse `SignalAusDatei` muss man ein Feld `dateiname` anlegen und im Konstruktor initialisieren. Die Methode `ListeVerwendeteDateien` wird in dieser Klasse so überschrieben, dass sie eine Liste zurückgibt, die nur den einen Eintrag aus `dateiname` enthält. In der Klasse `SummeVonSignalen` wird sie so überschrieben, dass sie eine Gesamtliste der Ergebnisse von `s1.ListeVerwendeteDateien()` und von `s2.ListeVerwendeteDateien()` zurückgibt.
6. Zum Beispiel so:

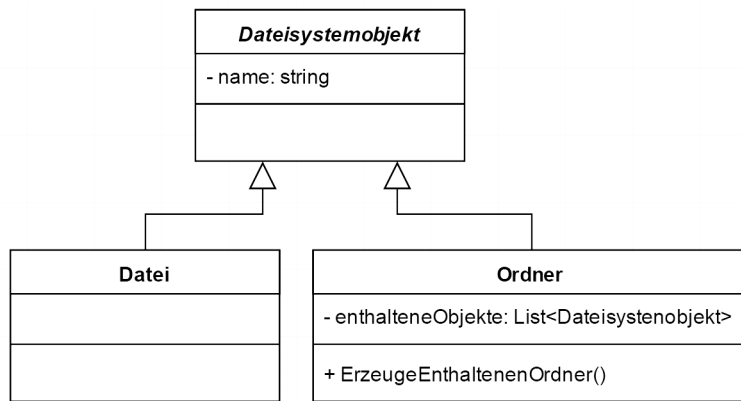
```
class SignalRückwärts : Signal
{
    Signal originalsignal;

    public SignalRückwärts(Signal s)
    : base(s.Abtastfrequenz)
    {
        länge = s.Länge;
        originalsignal = s;
        name = s.Name + " rückwärts";
    }

    public override double GibAbtastwert(int nummer)
    {
        return originalsignal.GibAbtastwert(Länge - 1 - nummer);
    }
}
```

In der Klasse `Ausleihvorgang` ist dazu das Feld `endeLeihfrist` `protected` und die Methode `Verlängere` virtuell zu machen.

7. Zum Beispiel so:



Dies ist ein Beispiel für das Design-Pattern „Composite“.

8. Die Werte sind 1, 2, 3.