Zusammenfassung: Jahr 2

Inhaltsverzeichnis

1 Lernfeld 6: Programmieren - C-Sharp

1.1 Objektorientierung

Ein Objekt ist eine Instanz einer Klasse. Ein Objekt hat Attribute und beherrscht Methoden. Attribute sind Konstanten oder Variablen mit einem festen Datentyp. In der Regel sind Attribute private, dass bedeutet, dass sie von außen nicht verändert werden können (Datenkapselung). Daher gibt es zum Ändern von Attributen meist public Methoden, die entsprechende Abfragen enthalten, damit Attribute nur korrekte Daten erhalten. Methoden sind entweder Funktionen oder Prozeduren. Im Gegensatz zu Funktionen liefern Prozeduren keinen Zückgabewert.

1.2 Übung: Kleines 1x1

Bei der Übung „Kleines 1x1“ soll der unten gezeigte Output auf der Konsole ausgegeben werden. Anhand dieser Übung sollen erste Unterschiede zwischen PHP und C# aufgezeigt werden.

```
01 02 03 04 05 06 07 08 09 10
02 04 06 08 10 12 14 16 18 20
03 06 09 12 15 18 21 24 27 30
04 08 12 16 20 24 28 32 36 40
05 10 15 20 25 30 35 40 45 50
06 12 18 24 30 36 42 48 54 60
07 14 21 28 35 42 49 56 63 70
08 16 24 32 40 48 56 64 72 80
09 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```

```
1  <?php
2   for ($zeile=1;$zeile<=10;$zeile++)
3   {
4    for ($spalte=1;$spalte<=10;$spalte++)
5    {
6     $ergebnis=$zeile*$spalte;
7     if ($ergebnis<10)
8     {
9      echo "0";
10    }
11    echo "$ergebnis ";
12   }
13   echo "</br>";
14  }
15 ?>
```

Listing 1: Lösung der Übung mit PHP

```
1  using System;
2
3  namespace kleines1x1_v1
4  {
5   class kleines1x1
6   {
7    public static void Main (string[] args)
8    {
9     int zeile;
10    int spalte;
11    int ergebnis;
12
13    for (zeile = 1; zeile <= 10; zeile++)
14    {
15     for (spalte = 1; spalte <= 10; spalte++)
16     {
17      ergebnis = zeile * spalte;
18      if (ergebnis < 10)
19      {
20       Console.Write(0);
21      }
22      Console.Write(ergebnis+" ");
23     }
24     Console.WriteLine();
25    }
26   }
27  }
28 }
```

Listing 2: Lösung der Übung mit C#

1.3 Konstruktoren

Konstruktoren sind Methoden von Klassen, die bei der Erzeugung eines Objektes Werte übergeben. Ein Konstruktor hat keinen Rückgabetyp, auch nicht void. Eine Klasse kann mehrere Konstruktoren besitze, die jeweils eine andere Zahl an Parametern erwarten. Eine Klasse kann mehrere Konstruktoren besitzen, die sich bezüglich der übergebenen Parameter unterscheiden. Konstuktoren, d.h. die Methoden, werden ebenso wie die Klasse benannt.

```
1  using System;
2
3  public class Person
4  {
5          //Attribute
6          private string name;
7
8          //Konstruktor
9          public Person()
10         {
11                 this.name = "";
12         }
13
14         public Person(string name)
15         {
16                 this.name = name;
17         }
18
```

```
19          //Methoden
20          public string GetName()
21          {
22                  Return this.name;
23          }
24
25          public void SetName(string name)
26          {
27                  this.nachname = name;
28          }
29
30          //Ausgabe
31          public void PrintInfo()
32          {
33                  Console.WriteLine (this.name);
34          }
35  }
```

Listing 3: Lösung der Übung mit C#

1.4 Eigenschaften

Eigenschaften sind nicht mit Attributen zu verwechseln. Eigenschaften sind eher wie Methoden. Statt je einer Set- und Get-Methode, wird eine Eigenschaft definiert, die wie im Beispiel zu sehen ist, je ein set und ein get definiert.

```
1   public Int16 Kosten
2   {
3           set
4           {
5                   if (value < 10)
6                   {
7                           hersteller = value;
8                   }
9           }
10          get
11          {
12                  return hersteller;
13          }
14  }
```

Listing 4: Beispiel: Eigenschaften