

OOP Objektorientierte Programmierung in PHP - Part 5

Hallo liebe Community!

Dies ist mein erstes Tutorial also seit nicht zu streng mit der Kritik, über Verbesserungsvorschläge würde ich mich dennoch freuen!

Ich setze voraus, dass man weiß wie Funktionen geschrieben werden und dass man mit Variablen umgehen kann. Außerdem sollte man folgende Parts meines Tutorials gelesen haben:

- [\[wiki\]OOP Objektorientierte Programmierung in PHP - Part 1\[/wiki\]](#)
- [\[wiki\]OOP Objektorientierte Programmierung in PHP - Part 2\[/wiki\]](#)
- [\[wiki\]OOP Objektorientierte Programmierung in PHP - Part 3\[/wiki\]](#)
- [\[wiki\]OOP Objektorientierte Programmierung in PHP - Part 4\[/wiki\]](#)

In diesem Part gehe ich genauer auf die Sichtbarkeit von Eigenschaften und Methoden ein. Sichtbarkeit finde ich persönlich ist bei den ganzen Nullen und Einsen ein sehr merkwürdiger Begriff. Ich finde dass man statt visibility eher accessibility (Zugreifbarkeit (1:1 Übersetzung)) sagen sollt, aber Schluss mit meiner Meinung^^

In Part 1 habe ich grob erklärt was public und private bedeutet, und genau den beiden keywords und dem Dritten im Bunde, protected (beschützt), widmet sich dieses Tutorial. Ich hingegen widme es Gambler, der mir fleißig PN's zu meinem Tutorial schickt.

Private

- Es kann nicht von "außerhalb der Klasse" auf private Eigenschaften/Methoden zugegriffen werden.
- In der Klasse selber kann auf die Eigenschaften/Methoden zugegriffen werden. Ein Objekt "A" der Klasse "Buchstabe" kann auf die private properties/methods des Objekts "B" der Klasse "Buchstabe" zugreifen, jedoch nicht auf private properties/methods eines Objekts der Klasse "Zahl". Was damit gemeint ist:

Quellcode

```
1. <?php
2. class Buchstabe
3. {
4.     private $name;
5.     public function __construct($name)
6.     {
7.         $this->name = $name;
8.     }
9.     public function getName()
10.    {
11.        return $this->name;
12.    }
13.     public function getOtherName($object)
14.     {
15.         return $object->name; //Zugriff auf die private property eines Objekts klappt nicht immer!
16.     }
17. }
18. class Zahl
19. {
20.     private $name;
21.     public function __construct($name)
22.     {
23.         $this->name = $name;
24.     }
25. }
```

```

30. }
32. $A = new Buchstabe("A");
33. $B = new Buchstabe("B");
34. $eins = new Zahl(1);
36. # Korrekte Versionen um an "$name" zu kommen
37. echo $A->getName();
38. echo $A->getOtherName($B);
39. echo $B->getName();
40. echo $B->getOtherName($A);
41. echo $B->getOtherName($B);
42. echo $A->getOtherName($A);
43. # Nicht korrekte Versionen
45. echo $A->name;
46. echo $B->name;
47. echo $eins->name;
48. echo $A->getOtherName($eins);
49. echo $B->getOtherName($eins);
50. ?>

```

Alles anzeigen

- Es kann nicht von anderen Klassen auf private properties/methods eines Objekts zugegriffen werden.
- Nicht mal Kindklassen (childclasses) können über parent:: oder \$this-> auf private properties/methods zugreifen!

Protected

- Es kann nicht von "außerhalb der Klasse" auf protected Eigenschaften/Methoden zugegriffen werden.
- In der Klasse selber kann auf die Eigenschaften/Methoden zugegriffen werden. Ein Objekt "z" der Klasse "a" kann auf die protected properties/methods des Objekts "x" der Klasse "a" zugreifen, jedoch nicht auf protected properties/methods eines Objekts der Klasse "b".
- Es kann nicht von anderen Klassen auf protected properties/methods eines Objekts zugegriffen werden.
- Kindklassen (childclasses) können über parent:: oder \$this-> auf protected properties/methods zugreifen!

Public

- Es kann von "außerhalb" auf public properties/methods zugegriffen werden.
- Es kann von allen Objekten der eigenen Klasse auf public properties/methods eines Objekts zugegriffen werden.
- Es kann von anderen Klassen auf public properties/methods eines Objekts zugegriffen werden.
- Kindklassen (childclasses) können über parent:: oder \$this-> auf public properties/methods zugreifen.

Wenn man sich die deutsche bedeutung der Begriffe anschaut kann man es schneller verstehen.

- private also privat: das geht nur mich was an, die anderen können Fragen! Ich und, wenn man eine gespaltene Persönlichkeit hat, alle anderen ich's (alle der Klasse "ich") dürfen das wissen, der Rest (auf meine Kinder) müssen mich Fragen (getter).
- protected also be-/geschützt: es ist ein Geheimniss, und geht nur mich und meine vielen Persönlichkeite etwas an! Außerdem kann ich meinen Kindern das Geheimniss anvertrauen.
- public oder öffentlich: Das dürfen alle sofort wissen, die müssen mich nicht mal fragen! Es ist egal wer das weiß!

Nun auf unsere tollen Raumschiffe bezogen:

Wer den letzten Part gelesen hat, der sollte noch in Erinnerung haben, dass wir nur mit public gearbeitet haben. Das liegt daran, dass wir protected noch nicht kennen gelernt haben! Ich empfehle euch den Umgang mit private und protected zu üben und deren Unterschied voll und ganz zu verstehen, sonst habt ihr später vllt. mal das Problem das ihr vor einem Script sitzt und euch fragt, "wieso erbt der die private Eigenschaft nicht so wie ich das will?"

Nächster Part handelt von erweiterter Vererbung und wie man sie einschränken kann.

n0x-f0x