

OOP Objektorientierte Programmierung in PHP - Part 7

Hallo liebe Community!

Dies ist mein erstes Tutorial also seid 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\[/wiki\]](#)
- [\[wiki\]OOP Objektorientierte Programmierung in PHP - Part 5\[/wiki\]](#)
- [\[wiki\]OOP Objektorientierte Programmierung in PHP - Part 6\[/wiki\]](#)

Dieser Part behandelt die Vererbung nochmal genauer.

Was ist wenn nun mal etwas NICHT von der Kindklasse geändert werden soll?? Das keyword ist final. Eine Methode die mit final definiert wurde kann nicht nach der vererbung überschrieben werden. Man darf also in der Kindklasse keine Methode mit dem gleichen Namen definieren. Es können auch Klassen als final definiert werden, dann darf keine Klasse von dieser Klasse erben. Ich glaube das Beispiel aus dem [PHP Manual](#) sagt den Rest:

Quellcode

```

1. <?php
2. class BaseClass {
3. public function test() {
4. echo "BaseClass::test() aufgerufen\n";
5. }
6. final public function moreTesting() {
8. echo "BaseClass::moreTesting() aufgerufen\n";
9. }
10. }
12. class ChildClass extends BaseClass {
13. public function moreTesting() {
14. echo "ChildClass::moreTesting() aufgerufen\n";
15. }
16. }
17. // Erzeugt einen fatalen Fehler: Cannot override final method BaseClass::moreTesting()
18. ?>
20. <?php
21. final class BaseClass {
22. public function test() {
23. echo "BaseClass::test() aufgerufen\n";
24. }
26. // Es macht hier keinen Unterschied, ob die Methode final ist oder nicht
27. final public function moreTesting() {
28. echo "BaseClass::moreTesting() aufgerufen\n";
29. }
30. }
32. class ChildClass extends BaseClass {
33. }
34. // Erzeugt einen fatalen Fehler: Class ChildClass may not inherit from final class (BaseClass)
35. ?>

```

Alles anzeigen

Das final keyword wird mit PHP 5 eingeführt und ist vorher nicht verfügbar!

n0x-f0x