



Opleiding Applicatie Ontwikkelaar

Leerlijn Programmeren Object Oriented Programming in PHP

Domein A Level 3

Auteur: Aminah Balfaqih

Datum: 16-12-2020

Inhoudsopgave

| | |
|--|----|
| Overzicht | 3 |
| Voorkennis..... | 3 |
| Kerntaak en werkprocessen | 3 |
| Leerdoelen..... | 3 |
| Materialen | 3 |
| Bronnen | 3 |
| Toetsing | 3 |
| Studieblok 1..... | 4 |
| Bestuderen | 4 |
| Opdrachten..... | 4 |
| Studieblok 2..... | 5 |
| Bestuderen | 5 |
| Oefening | 5 |
| Studieblok 3..... | 6 |
| Bestuderen | 6 |
| Opdrachten..... | 6 |
| Studieblok 4..... | 7 |
| Bestuderen | 7 |
| Opdrachten..... | 7 |
| Regels voor OOP..... | 9 |
| Voor het programmeren in OOP zijn er een aantal regels:..... | 9 |
| Eindopdrachten | 10 |
| Eindopdracht 1 | 11 |
| Eindopdracht 2 | 12 |
| Inleveren..... | 12 |
| Bronnen | 13 |
| Extra uitleg oop | 13 |
| Extra uitleg oop op youtube..... | 13 |

Overzicht

Level: Domein A Level 3
Duur: 4 weken
Methode: Weekplanning

Voorkennis

Module A Level 2, Module B Level 2, Module C1

Kerntaak en werkprocessen

In deze leerlijn zullen we werken aan kerntaak 2 .

Leerdoelen

Na het bestuderen zal je een simpel programma kunnen ontwerpen en bouwen met objecten.

Materialen

Er wordt gebruik gemaakt van de volgende studiematerialen:

- Je laptop met;
- Editor, bijvoorbeeld Kladblok, Notepad++, Atom (☺ : aanrader!)
- Webbrowser, bijvoorbeeld Internet Explorer, Firefox, Chrome
- XAMPP webserver of usb webserver
- Mysql (via phpmyadmin)

Bronnen

Algemene uitleg:

<https://www.phphulp.nl>

<http://php.net/>

Extra uitleg oop:

[http://www.sitemasters.be/tutorials/1/1/567/PHP/OOP Een begin maken met OOP](http://www.sitemasters.be/tutorials/1/1/567/PHP/OOP_Een_begin_maken_met_OOP)

Extra uitleg oop op youtube:

<https://www.youtube.com/watch?v=8Djzluu49Rk&list=PL0eyrZgxdwhypQiZnYXM7z7-OTkcMgGPh>

Toetsing

Deze module wordt afgesloten met een eindopdracht. Lever alle opdrachten (in totaal 11) in samen met de eindopdrachten (in totaal 2). De gewone opdrachten zijn een voorwaarde om de eindopdracht te laten beoordelen.

Studieblok 1

Leerdoel

In dit studieblok ga je de beginselen van het object oriented programming leren.

Bestuderen

Object georienteerd denken:

<https://www.phphulp.nl/php/tutorial/overig/oop-beginnershandleiding-php5/701/object-gerienteerd-denken/1840/>

Foute denkwijze:

<https://www.phphulp.nl/php/tutorial/overig/oop-beginnershandleiding-php5/701/foute-denkwijze/1839/>

Object georienteerd programmeren:

<https://www.phphulp.nl/php/tutorial/overig/oop-beginnershandleiding-php5/701/object-georienteerd-programmeren/1838/>

Opdrachten

Opdracht 1

Beschrijf in je eigen woorden wat volgens jou het verschil is tussen een object en een klasse.

Opdracht 2

Maak de userobject uit het hoofdstuk “object georienteerd programmeren” op jouw eigen computer. Als het werkt dan is het de bedoeling dat je een achternaam en geboortedatum toevoegt aan de klasse. Maak gebruik van get en set methoden. Laat op het scherm zien dat deze nieuwe eigenschappen werken.

Studieblok 2

Leerdoel

In dit studieblok ga je leren wat het verschil is tussen private, protected en public variabelen. Tevens ga je leren wat de conventies van Zend zijn en hoe je deze toepast.

Bestuderen

Private, protected en public:

<https://www.phphulp.nl/php/tutorial/overig/oop-beginnershandleiding-php5/701/visibility/1842/>

Naamgeving:

<https://www.phphulp.nl/php/tutorial/overig/oop-beginnershandleiding-php5/701/naamgeving/1843/>

Oefening

Opdracht 3

Leg in eigen woorden uit wat volgens jou de betekenis is van private, protected en public. Leg ook uit waarom je niet alles public zou moeten maken.

Opdracht 4

Maak een kopie van jouw uitgebreidere user klasse uit studieblok 1. Pas waar nodig jouw nieuwe kennis over private, protected en public toe. Pas ook waar nodig is jouw kennis toe over de naamgeving. Vergeet de achternaam en voornaam niet. Laat op het scherm zien dat alles werkt.

Studieblok 3

Leerdoel

In studieblok ga je leren hoe je een constructor op kunt bouwen en hoe je deze kunt gebruiken. Tevens krijg je uitleg over het hoe en waarom van een constructor.

Bestuderen

De constructor:

<https://www.phphulp.nl/php/tutorial/overig/oop-beginnershandleiding-php5/701/constructor-construct/1844/>

Uitgewerkt voorbeeld:

<https://www.phphulp.nl/php/tutorial/overig/oop-beginnershandleiding-php5/701/voorbeeld-html-tabel/1845/>

Let op!

Het is mogelijk om een constructor te maken die direct ook alle eigenschappen van een klasse declareert. Het is zeer onwenselijk om dit te doen. De eigenschappen worden dan automatisch public. In jouw class wil je er altijd voor zorgen dat je properties niet publiekelijk toegankelijk zijn ivm beveiliging.

Opdrachten

Opdracht 5

Maak een kopie van jouw klassen uit studieblok 2 opdracht 2. Voeg nu een constructor toe die ook de achternaam en geboortedatum direct invult. Laat op het scherm zien dat het werkt.

Opdracht 6

Probeer van het hoofdstuk “uitgewerkt voorbeeld” in eigen woorden een uitleg te geven van wat er gebeurt.

Opdracht 7

Leg uit waarom een constructor altijd public moet zijn.

Studieblok 4

Leerdoel

In dit studieblok ga je leren hoe overerving werkt en hoe je dit kunt gebruiken. Ook zal je leren waarvoor je static methoden kunt gebruiken.

Bestuderen

Inheritance/ overerving

<https://www.phphulp.nl/php/tutorial/overig/oop-beginnershandleiding-php5/701/inheritance/1846/>

Static methoden

<https://www.phphulp.nl/php/tutorial/overig/oop-beginnershandleiding-php5/701/static-methods-en-properties/1847/>

Opdrachten

Opdracht 8

In de tekst worden de volgende correcte voorbeelden gebruikt voor extends:

- MySQL extends Database -> kan, MySQL is een type database
- Mens extends Dier -> kan, ieder mens is een dier
- Dier extends Organisme -> kan, ieder dier is een organisme
- Mens extends Organisme -> kan ook, ieder mens is immers een organisme

Bedenk zelf nog 3 voorbeelden van een correcte extends.

Opdracht 9

In de tekst worden de volgende incorrecte voorbeelden gebruikt voor extends:

- Query extends MySQL -> kan niet, een query is geen database!
- Dier extends Konijn -> kan niet, niet ieder dier is een konijn!

Bedenk zelf nog 1 voorbeeld van een incorrecte extends

Opdracht 10

Probeer alle code in het hoofdstuk inheritance uit.



Opdracht 11

Beschrijf in welke gevallen het handig zou kunnen zijn om een klasse te maken met alleen maar static methoden.

Regels voor OOP

Voor het programmeren in OOP zijn er een aantal regels:

Een klasse heeft één taak.

Dit betekent eigenlijk dat je niet teveel verschillende dingen door één object moet laten doen. Dit levert vaak verwarrende code op en is vaak een teken dat je je code niet goed overdacht hebt, of direct bent begonnen met programmeren.

Een klasse is zelf verantwoordelijk voor zijn data.

Wijzigen van data in een object (=instantie van een klasse) gaat via een methode. Een object kan zelf dan bepalen of de aangeboden eigenschappen wel passend zijn, en ze eventueel aanpassen of een foutmelding teruggeven. Voorbeeld: een auto kan best 4 wielen hebben of 3 of misschien wel 6 maar 20 is echt teveel. Dat zou meer een aanhanger voor een grote truck zijn.

Alleen voor PHP: Klasse in include files

Tot nu toe heb je gezien dat de klassen in dezelfde file staan als de code die de objecten aanroep/aanmaakt. In de praktijk wordt dit eigenlijk nooit gedaan. Zorg er altijd voor dat je klasse in een aparte file staat. Deze klassen kan je dan bovenin jouw code aanroepen als een include file. Een nette naam voor een klasse file heeft altijd het volgende format: `eennaam.class.php`. Een klasse staat altijd in een aparte file en in een aparte map.

Studieblok 5

Leerdoel

In dit studieblok ga je leren hoe je een autoload functie kan toevoegen aan jouw programma.

Bestuderen

Als je al wat meer klassen geschreven hebt dan zal je merken dat het heel vervelend wordt om elke keer aan het begin van je script alle klassen te includen.

Er is hier een oplossing voor bedacht. Deze oplossing heet “autoload”.

De autoload script bevat al jouw klassen. Deze script hoeft je dan maar één keer aan te roepen bovenin jouw script en alle klassen zijn geladen.

Uitleg van de autoload: <http://php.net/manual/en/language.oop5 autoload.php>

Let op de volgende functie:

```
void __autoload ( string $class )
```

Deze functie kan je niet meer gebruiken vanaf php 7.2.

Gebruik vanaf deze versie de volgende methode:

<http://php.net/manual/en/function.spl-autoload-register.php>

Opdrachten

Opdracht 1

Gebruik de klasse die je gemaakt hebt bij opdracht 4 uit studieblok 2 (user klasse).

Maak nu een nieuw script die gebruik maakt van deze klasse. Dus maak een object aan van de user klasse en laat dan alle gegevens van deze user zien op het scherm. Natuurlijk is het de bedoeling dat je de klasse in het script laad via de autoload functie.

Eindopdrachten

Eindopdracht 1 (4 – 6 uur)

Maak een klasse huis.

De eigenschappen van een huis:

- Aantal kamers
- Aantal toiletten
- Verwarming (alleen ja of nee mogelijk)
- Soort verwarming (bedenk welke mogelijke verwarming er bestaat)
- Straatnaam
- Huisnummer
- Plaats
- Vierkant meter grond
- Woz waarde

Als je nog een paar kunt bedenken zet deze er dan ook bij. Als je bij funda kijkt dan kan je er vast nog meer vinden.

- Maak de klasse en zet er de eigenschappen in.
- Zorg dat er een constructor aanwezig is. Verplichte eigenschappen bij het aanmaken zijn: Straatnaam, huisnummer, plaats
- Maak de bijbehorende getters en setters.

Een fictieve overheid heeft bepaald dat er een extra belasting komt op huizen. Deze belasting moet ervoor gaan zorgen dat mensen zo klein mogelijk en meer buiten de stad gaan wonen.

De belasting wordt door het volgende bepaald:

- Woz waarde:
alle huizen onder de 100.000 euro 600 euro belasting,
alle huizen tussen de 100.000 euro en 200.0000 euro 2000 euro belasting
en alle huizen boven de 200.000 euro 6000 euro belasting.
- Aantal kamers:
bij 1 kamer 100 euro belasting,
bij 2 kamers 300 euro belasting,
en bij meer dan 3 kamers 800 euro belasting
- Alle huizen in Amsterdam, Rotterdam en Groningen worden belast met een extra 1000 euro belasting

De belasting moet berekent worden in een functie.

Laat de gegevens van klasse "huis" netjes op het scherm zien.



Eindopdracht 2 (eindopdracht 24 – 30 uur)

Maak een memory spel (van 4x4 “kaartjes”) waarbij je alle opgedane kennis verwerkt. Probeer waar nodig klassen te maken zodat de code overzichtelijk wordt. Denk goed na over welke functionaliteit bij welke data hoort en maak daar een klasse van. Denk in dingen (objecten) en niet in programmaregels. Het memory spel werkt met cijfers.

Inleveren

Lever de code van alle opdrachten in, keurig in verschillende mapjes, in één zip file.
Graag alle gewone opdrachten in 1 map en de eindopdrachten in 1 map.

Bronnen

- ✓ <https://www.phphulp.nl>
- ✓ <http://php.net/>

Extra uitleg oop

- ✓ [http://www.sitemasters.be/tutorials/1/1/567/PHP/OOP Een begin maken met OOP](http://www.sitemasters.be/tutorials/1/1/567/PHP/OOP_Een_begin_maken_met_OOP)

Extra uitleg oop op youtube

- ✓ <https://www.youtube.com/watch?v=8Djzluu49Rk&list=PL0eyrZgxdwhypQiZnYXM7z7-OTkcMgGPh>