Training Application Developer

Programming curriculum

Object Oriented Programming in PHP

Domain A Level 3

Author: Aminah Balfaqih

Date:6-6-2023

Table of contents

Overview

Level:Domain A Level 3
Duration: 4 weeks
Method: Weekly schedule

Prior knowledge

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

Core task and work processes

In this learning line we will work on core task 2.

Learning objectives

After studying you will be able to design and build a simple program with objects.

Materials

The following study materials are used:

- Your laptop with;
- Editor, for example Notepad, Notepad++, Atom (☺ : recommended!)
- Web browser, e.g. Internet Explorer, Firefox, Chrome
- XAMPP web server or usb web server
- Mysql (via phpmyadmin)

Sources

General explanation:

https://www.phphulp.nl

http://php.net/

Additional explanation oop:

http://www.sitemasters.be/tutorials/1/1/567/PHP/OOP_Een_begin_maken_met_OOP

Additional explanation oop on youtube:

https://www.youtube.com/watch?v=8DjzIuu49Rk&list=PL0eyrZgxdwhypQiZnYXM7z7-OTkcMgGPh

Review

This module is concluded with a final assignment. Hand in all assignments (11 in total) together with the final assignmentand (2 in total). The ordinary assignments are a condition for having the final assignment assessed.

Study block 1

Learning objective

In this study block you will learn the principles of object oriented programming .

Study

Object oriented thinking:

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

Wrong thinking:

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

Object oriented programming:

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

Commands

Assignment 1

Describe in your own words what you think is the difference between an object and a class.

Assignment 2

Create the user object from the chapter "object oriented programming" on your own computer. If it works, you're supposed to add a last name and date of birth to the class. Make use of get and set methods. Show on the screen that these new properties work.

Study block 2

Learning objective

In this study block you will learn the difference between private, protected and public variables. You will also learn what the conventions of Zend are and how to apply them.

Study

Private, protected and public:

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

Naming:

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

Exercise

Assignment 3

Explain in your own words what you think the meaning of private, protected and public is. Also, explain why you shouldn't make everything public.

Assignment 4

Make a copy of your more extensive user class from study block 1. Where necessary, apply your new knowledge about private, protected and public. Also apply your knowledge about the naming where necessary. Do not forget the last name and first name. Show on the screen that everything works.

Study block 3

Learning objective

In study block you will learn how to build a constuctor and how to use it. You will also receive an explanation of the how and why of a constructor.

Study

The constructor:

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

Detailed example:

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

Note!

It is possible to create a constructor that also directly declares all the properties of a class. It is very undesirable to do this. The properties then automatically become public. In your class you always want to make sure that your properties are not publicly accessible due to security.

Commands

Assignment 5

Make a copy of your classes from study block 2 assignment 2. Now add a constructor that also fills in the last name and date of birth directly. Show on the screen that it works.

Assignment 6

Try to explain in your own words the chapter "elaborated example" what is happening.

Assignment 7

Explain why a constructor should always be public.

Study block 4

Learning objective

In this study block you will learn how inheritance works and how you can use it. You will also learn what you can use static methods for.

Study

Inheritance

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

Static methods

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

Commands

Opdracht 8

The following correct examples are used for extends in the text:

- MySQL extends Database -> can, MySQL is a type of database- Human extends Animal -> can, every human is an animal- Animal extends Organism -> can, every animal is an organism- Man extends
Organism -> is also possible, after all, every human being is
an organism

Come up with 3 more examples of a correct extends.

Assignment 9

The following incorrect examples are used for extends in the text:

- Query extends MySQL -> can't, a query is not a database! -
Animal extends Rabbit -> can't, not every animal is a rabbit!

Come up with 1 example of an incorrect extends

Assignment 10

Try out all the code in the inheritence chapter .

Assignment 11

Describe in which cases it might be useful to create a class using only static methods.

Rules for OOP

For programming in OOP there are a number of rules:

A class has one task.

This basically means that you shouldn't have too many different things done by one object. This often results in confusing code and is often a sign that you have not thought through your code properly, or have immediately started programming.

A class is responsible for its own data.

Changing data in an object (= instance of a class) is done via a method. An object can then determine whether the offered properties are appropriate, and possibly adjust them or return an error message. Example: a car can have 4 wheels or 3 or maybe 6 but 20 is really too much. That would be more like a trailer for a big truck.

Only for PHP: Class in include files

So far you have seen that the classes are in the same file as the code that calls/creates the objects. In practice, this is never actually done. Always make sure your class is in a separate file. You can then call these classes at the top of your code as an include file. A neat name for a class file always has the following format: onename.class.php. A class is always in a separate file and folder.

Study block 5

Learning objective

In this study block you will learn how to add an autoload function to your program.

Study

If you have already written some more classes, you will notice that it becomes very annoying to include all classes every time at the beginning of your script.

A solution has been devised for this. This solution is called "autoload".

The autoload script contains all your classes. You only have to call this script once at the top of your script and all classes are loaded.

Explanation of the autoload: http://php.net/manual/en/language.oop5.autoload.php

Note the following feature:

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

You can no longer use this function from php 7.2.

Starting with this version, use the following method:

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

Commands

Assignment 1

Use the class you created for assignment 4 from study block 2 (user class).

Now create a new script that uses this class. So create an object of the user class and then show all the data of this user on the screen. Of course, you are supposed to load the class into the script via the autoload function.

Final assignments

Final assignment 1 (4 – 6 hours)

Create a class house.

The features of a house:

- Number of rooms
- Number of toilets
- Heating (only yes or no possible)
- Type of heating (consider what possible heating exists)
- Street name
- House number
- Place
- Square meters of land
- Woz value

If you can think of a few more, please include this one. If you look at funda , you can probably find more.

- Create the class and put the properties in it.
- Make sure a constructor is present. Mandatory properties when creating are: Street name, house number, city
- Create the corresponding getters and setters.

A fictitious government has determined that there will be an extra tax on houses. This tax should ensure that people live as small as possible and more outside the city.

The load is determined by the following:

- Woz value: all houses under 100,000 euros 600 euros tax, all houses between 100,000 euros and 200,0000 euros 2000 euros tax
  and all houses above 200,000 euros 6000 euros tax.

- Number of rooms: for 1 room 100 euros tax, for 2 rooms 300 euros tax,
  and for more than 3 rooms 800
  euros tax

- All houses in Amsterdam, Rotterdam and Groningen are taxed with an additional 1000 euros tax

The load must be calculated in a function.

Show the data of class "house" neatly on the screen.

Final assignment 2 (final assignment 24 – 30 hours)

Make a memory game (of 4x4 "cards") where you process all the knowledge gained. Try to create classes where necessary so that the code becomes clear. Think carefully about which functionality belongs to which data and make it a class. Think in things (objects) and not in program rules. The memory game works with numbers.

Hand in

Hand in the code of all assignments, neatly in different folders, in one zip file.
Please all regular assignments in 1 folder and the final assignments in 1 folder.

Sources

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

Additional explanation oop

- ✓ http://www.sitemasters.be/tutorials/1/1/567/PHP/OOP_Een_begin_maken_met_OOP

Extra explanation oop on youtube

- ✓ https://www.youtube.com/watch?v=8DjzIuu49Rk&list=PL0eyrZgxdwhypQiZnYXM7z7-OTkcMgGPh