Classes vs 2D arrays

Asked 16 years, 4 months ago Modified 12 years, 1 month ago Viewed 2k times

Part of PHP Collective

6

Which is better to use in PHP, a 2D array or a class? I've included an example of what I mean by this.

```
// Using a class
class someClass
{
    public  $name;
    public  $height;
    public  $weight;

    function __construct($name, $height, $weight)
    {
        $this -> name       = $name;
        $this -> height = $height;
        $this -> weight = $weight;
    }
}

$classArray[1] = new someClass('Bob', 10, 20);
$classArray[2] = new someClass('Fred', 15, 10);
$classArray[3] = new someClass('Ned', 25, 30);


// Using a 2D array
$normalArray[1]['name'] = 'Bob';
$normalArray[1]['height']   = 10;
$normalArray[1]['weight']   = 20;

$normalArray[2]['name'] = 'Fred';
$normalArray[2]['height']   = 15;
$normalArray[2]['weight']   = 10;

$normalArray[3]['name'] = 'Ned';
$normalArray[3]['height']   = 25;
$normalArray[3]['weight']   = 30;
```

Assuming that somebody doesn't come out and show that classes are too slow, it looks like class wins.

I've not idea which answer I should accept to I've just upvoted all of them.

And I have now written two near identical pages, one using the 2D array (written before this question was posted) and now one using a class and I must say that the class produces much nicer code. I have no idea how much overhead is going to be generated but I doubt it will rival the improvement to the code itself.

Thank you for helping to make me a better programmer.

PHP php arrays class

Share
Improve this question
Follow

edited Nov 21, 2011 at 3:53
user50049

asked Aug 23, 2008 at 10:41
Teifion
111k ● 76 ● 164 ● 196

9 Answers

Sorted by: Highest score (default)

9

The "class" that you've constructed above is what most people would use a *struct* for in other languages. I'm not sure what the performance implications are in PHP, though I suspect instantiating the objects is probably more costly here, if only by a little bit.

That being said, if the cost is relatively low, it IS a bit easier to manage the objects, in my opinion.

I'm only saying the following based on the title and your question, but: Bear in mind that classes provide the advantage of methods and access control, as well. So if you wanted to ensure that people weren't changing weights to negative numbers, you could make the `weight` field private and provide some accessor methods, like `getWeight()` and `setWeight()`. Inside `setWeight()`, you could do some value checking, like so:

```
public function setWeight($weight)
{
    if($weight >= 0)
    {
        $this->weight = $weight;
    }
    else
    {
        // Handle this scenario however you like
    }
}
```

Share Improve this answer Follow

answered Aug 23, 2008 at 10:52
Brian Warshaw
23k ● 9 ● 54 ● 72

4

It depends exactly what you mean by 'better'. I'd go for the object oriented way (using classes) because I find it makes for cleaner code (at least in my opinion). However, I'm not sure what the speed penalties might be for that option.

Share Improve this answer Follow

answered Aug 23, 2008 at 10:42
robintw
28.5k ● 51 ● 143 ● 206

4

Generally, I follow this rule:

1) Make it a class if multiple parts of your application use the data structure.

2) Make it a 2D array if you're using it for quick processing of data in one part of your application.

Share Improve this answer Follow

answered Aug 23, 2008 at 10:56
Steve M
10.6k ● 12 ● 53 ● 63

3

> > It's the speed that I am thinking of mostly, for anything more complex than what I have here I'd probably go with classes but the question is, what is the cost of a class?

This would seem to be premature optimisation. Your application isn't going to take any real-world performance hit either way, but using a class lets you use getter and setter methods and is generally going to be better for code encapsulation and code reuse.

With the arrays you're incurring cost in harder to read and maintain code, you can't unit test the code as easily and with a good class structure other developers should find it easier to understand if they need to take it on.

And when later on you need to add other methods to manipulate these, you won't have an architecture to extend.

Share Improve this answer Follow

answered Aug 23, 2008 at 11:10
Polsonby
22.9k ● 19 ● 60 ● 74

2

The class that you have is not a real class in OO terms - its just been contructed to take the space of the instance variables.

That said - there propably isnt much issue with speed - its just a style thing in your example.

The intresting bit - is if you contsrtucted the object to be a real "person" class - and thinkng about the other attributes and actions that you may want of the person class - then you would notice not only a style performance - writting code - but also speed performance.

Share Improve this answer Follow

answered Aug 23, 2008 at 12:12

littlegeek

0

If your code uses lot of functions that operate on those attributes (name/height/weight), then using class could be a good option.

Share Improve this answer Follow

answered Aug 23, 2008 at 15:42

Imran

90.8k ● 26 ● 100 ● 131

0

Teifion, if you use classes as a mere replacement for arrays, you are nowhere near OOP. The essence of OOP is that objects have knowledge and responsibility, can actually do things and cooperate with other classes. Your objects have knowledge only and can't do anything else than idly exist, however they seem to be good candidates for persistence providers (objects that know how to store/retrieve themselves into/from database).

Don't worry about performance, too. Objects in PHP are fast and lightweight and performance in general is much overrated. It's cheaper to save your time as a programmer using the right approach than to save microseconds in your program with some obscure, hard to debug and fix piece of code.

Share Improve this answer Follow

answered Aug 29, 2008 at 8:37

Michał Niedźwiedzki

12.9k ● 7 ● 46 ● 47

Most tests that time arrays vs classes only test instancing them. Once you actually start to do something with them.

0

I was a "purist" that used only arrays because the performance was SO much better. I wrote the following code to justify to myself to justify the extra hassle of not using classes (even though they are easier on the programmer)

Let's just say I was VERY surprised at the results!

```
    <?php
$rx = "";
$rt = "";
$rf = "";

$ta = 0; // total array time
$tc = 0; // total class time

// flip these to test different attributes
$test_globals = true;
$test_functions = true;
$test_assignments = true;
$test_reads = true;


// define class


class TestObject
{
  public $a;
  public $b;
  public $c;
  public $d;
  public $e;
  public $f;

  public function __construct($a,$b,$c,$d,$e,$f)
  {
    $this->a = $a;
    $this->b = $b;
    $this->c = $c;
    $this->d = $d;
    $this->e = $e;
    $this->f = $f;
  }

  public function setAtoB()
  {
      $this->a = $this->b;
  }
}

// begin test

echo "<br>test reads: " . $test_reads;
echo "<br>test assignments: " . $test_assignments;
echo "<br>test globals: " . $test_globals;
echo "<br>test functions: " . $test_functions;
echo "<br>";

for ($z=0;$z<10;$z++)
{
    $starta = microtime(true);

    for ($x=0;$x<100000;$x++)
    {
        $xr =
getArray('aaa','bbb','ccccccccc','ddddddddd','eeeeeeee','fffffffffff');

        if ($test_assignments)
        {
            $xr['e'] = "e";
            $xr['c'] = "sea biscut";
        }

        if ($test_reads)
        {
            $rt = $x['b'];
            $rx  = $x['f'];
        }

        if ($test_functions) { setArrAtoB($xr); }
        if ($test_globals) { $rf = glb_arr(); }
    }
    $ta = $ta + (microtime(true)-$starta);
    echo "<br/>Array time = " . (microtime(true)-$starta) . "\n\n";


    $startc = microtime(true);

    for ($x=0;$x<100000;$x++)
    {
        $xo = new
TestObject('aaa','bbb','ccccccccc','ddddddddd','eeeeeeee','fffffffffff');

        if ($test_assignments)
        {
            $xo->e = "e";
            $xo->c = "sea biscut";
        }

        if ($test_reads)
        {
            $rt = $xo->b;
            $rx = $xo->f;
        }

        if ($test_functions) { $xo->setAtoB(); }
        if ($test_globals) { $xf = glb_cls(); }
    }

    $tc = $tc + (microtime(true)-$startc);
    echo "<br>Class time = " . (microtime(true)-$startc) . "\n\n";

    echo "<br>";
    echo "<br>Total Array time (so far) = " . $ta . "(100,000 iterations)
\n\n";
    echo "<br>Total Class time (so far) = " . $tc . "(100,000 iterations)
\n\n";
    echo "<br>";

}
echo "TOTAL TIMES:";
echo "<br>";
echo "<br>Total Array time = " . $ta . "(1,000,000 iterations) \n\n";
echo "<br>Total Class time = " . $tc . "(1,000,000 iterations)\n\n";


// test functions

function getArray($a,$b,$c,$d,$e,$f)
{
    $arr = array();
    $arr['a'] = $a;
    $arr['b'] = $b;
    $arr['c'] = $c;
    $arr['d'] = $d;
    $arr['d'] = $e;
    $arr['d'] = $f;
    return($arr);
}

//------------------------------------

function setArrAtoB($r)
{
    $r['a'] = $r['b'];
}

//------------------------------------

function glb_cls()
{
    global $xo;

    $xo->d = "ddxxdd";
    return ($xo->f);
}

//------------------------------------

function glb_arr()
{
    global $xr;

    $xr['d'] = "ddxxdd";
    return ($xr['f']);
}

//------------------------------------

?>
```

test reads: 1 test assignments: 1 test globals: 1 test functions: 1

Array time = 1.58905816078 Class time = 1.11980104446 Total Array time (so far) = 1.58903813362(100,000 iterations) Total Class time (so far) = 1.11979603767(100,000 iterations)

Array time = 1.02581000328 Class time = 1.22492313385 Total Array time (so far) = 2.61484408379(100,000 iterations) Total Class time (so far) =

```
2.34471416473(100,000 iterations)

Array time = 1.29942297935 Class time = 1.18844485283 Total Array time (so far) =
3.91425895691(100,000 iterations) Total Class time (so far) = 3.5331492424(100,000
iterations)

Array time = 1.28776097298 Class time = 1.02383089066 Total Array time (so far) =
5.2020149231(100,000 iterations) Total Class time (so far) = 4.55697512627(100,000
iterations)

Array time = 1.31235599518 Class time = 1.38880181313 Total Array time (so far) =
6.51436591148(100,000 iterations) Total Class time (so far) =
5.94577097893(100,000 iterations)

Array time = 1.3007349968 Class time = 1.07644081116 Total Array time (so far) =
7.81509685516(100,000 iterations) Total Class time (so far) =
7.02220678329(100,000 iterations)

Array time = 1.12752890587 Class time = 1.07106018066 Total Array time (so far) =
8.94262075424(100,000 iterations) Total Class time (so far) =
8.09326195717(100,000 iterations)

Array time = 1.08890199661 Class time = 1.09139609337 Total Array time (so far) =
10.0315177441(100,000 iterations) Total Class time (so far) =
9.18465089798(100,000 iterations)

Array time = 1.6172170639 Class time = 1.14714384079 Total Array time (so far) =
11.6487307549(100,000 iterations) Total Class time (so far) =
10.3317887783(100,000 iterations)

Array time = 1.53738498688 Class time = 1.28127002716 Total Array time (so far) =
13.1861097813(100,000 iterations) Total Class time (so far) =
11.6130547523(100,000 iterations)

TOTAL TIMES: Total Array time = 13.1861097813(1,000,000 iterations) Total Class
time = 11.6130547523(1,000,000 iterations)
```

So, either way the difference is pretty negligible. I was very suprized to find that once you start accessing things globally, classes actually become a little faster.

But don't trust me, run it for your self. I personally now feel completely guilt free about using classes in my high performance applications. :D

Share Improve this answer Follow

answered Dec 31, 2010 at 18:03

Richard Varno
547 ● 6 ● 9

0

@Richard Varno

I Ran your exact code (after fixed the small bugs), and got much different results than you. Classes ran much on my PHP 5.3.17 install.

Array time = 0.69054913520813 Class time = 1.1762700080872

Total Array time (so far) = 0.69054508209229(100,000 iterations) Total Class time (so far) = 1.1762590408325(100,000 iterations)

Array time = 0.99001502990723 Class time = 1.22034907341

Total Array time (so far) = 1.6805560588837(100,000 iterations) Total Class time (so far) = 2.3966031074524(100,000 iterations)

Array time = 0.99191808700562 Class time = 1.2245700359344

Total Array time (so far) = 2.6724660396576(100,000 iterations) Total Class time (so far) = 3.6211669445038(100,000 iterations)

Array time = 0.9890251159668 Class time = 1.2246470451355

Total Array time (so far) = 3.661484003067(100,000 iterations) Total Class time (so far) = 4.8458080291748(100,000 iterations)

Array time = 0.99573588371277 Class time = 1.1242771148682

Total Array time (so far) = 4.6572148799896(100,000 iterations) Total Class time (so far) = 5.9700801372528(100,000 iterations)

Array time = 0.88518786430359 Class time = 1.1427340507507

Total Array time (so far) = 5.5423986911774(100,000 iterations) Total Class time (so far) = 7.1128082275391(100,000 iterations)

Array time = 0.87605404853821 Class time = 0.95899105072021

Total Array time (so far) = 6.4184486865997(100,000 iterations) Total Class time (so far) = 8.0717933177948(100,000 iterations)

Array time = 0.73414516448975 Class time = 1.0223190784454

Total Array time (so far) = 7.1525888442993(100,000 iterations) Total Class time (so far) = 9.0941033363342(100,000 iterations)

Array time = 0.95230412483215 Class time = 1.059828042984

Total Array time (so far) = 8.1048839092255(100,000 iterations) Total Class time (so far) = 10.153927326202(100,000 iterations)

Array time = 0.75814390182495 Class time = 0.84455919265747

Total Array time (so far) = 8.8630249500275(100,000 iterations) Total Class time (so far) = 10.998482465744(100,000 iterations) TOTAL TIMES:

Total Array time = 8.8630249500275(1,000,000 iterations) Total Class time = 10.998482465744(1,000,000 iterations)

answered Nov 3, 2012 at 20:46

rtconner

1,238 ● 12 ● 21