

## 2.Object-oriented Programming with PHP 4

**1.Каква конструкција се користи за да се дефинира шемата на објектот?**

Вашиот договор е: \_\_\_\_\_

**2.На крај од извршување на следнава скрипта кои вредности ќе се чуваат во \$a->my\_value низата? (Изберете 3)**

```
<?php
class my_class
{
    var $my_value = array();

    function my_class ($value)
    {
        $this->my_value[] = $value;
    }

    function set_value ($value)
    {
        $this->$my_value= $value;
    }
}
$a = new my_class ('a');
$a->my_value[] = 'b';
$a->set_value ('c');
$a->my_class('d');

?>
```

- a) c
- b) b
- c) a
- d) d
- e) e

**3. Како може да се напише класа, така што некои од нејзините својства не може да се пристапат надвор од нејзините методи?**

- a) Со декларирање на класата како приватна
- b) Со декларирање на методите како приватни
- c) Тоа не може да се направи
- d) Со пишување на property overloading метод

**4. Кој објектно -ориентиран модел ќе се користи за имплементирање на класа која мора да се инстанцира само еднаш за целиот животен век на скриптата?**

- a) Model-view-controller
- b) Abstract factory
- c) Singleton
- d) Proxy
- e) State

**5. Класа може да биде изградена како продолжување на други класи со користење на процес познат како наследување. Во PHP, детето класа од колку родителите може да наследи ?**

- a) Еден
- b) Два
- c) Зависи од системските ресурси
- d) Три
- e) Колку што е потребно

**6. Кој ООР валиден конструктор во PHP 5 го прави приближно истото како и следнава скрипта ?**

```
<?php  
  
class my_class  
{  
    function my_func($my_param)  
    {  
        user_error ("Please define me", E_ERROR);  
    }  
    function b()  
    {  
        return 10;  
    }  
}  
  
?>
```

- a) Повеќекратното наследување
- b) Интерфејсите
- c) Апстрактните методи
- d) Приватните методи
- e) Преоптоварувањето на функција

**7. Претпоставуваме дека класата наречена testclass е дефинирана. Кое мора да биде името на овој конструктор метод?**

- a) \_\_construct
- b) initialize
- c) testclass
- d) \_\_testclass
- e) Само PHP 5 поддржуваат конструктори

**8. Како една класа може да го замени стандардниот продолжен механизам за неговите објекти?**

- a) Со имплементирање на \_\_shutdown и \_\_startup методите
- b) Со повикување на register\_shutdown\_function()
- c) Со имплементирање на \_\_sleep() и \_\_wakeup()
- d) Стандардниот продолжен механизам не може да се замени
- e) Со додавање на класа за излезниот визуелен механизам користејќи ob\_start()

**9. Во PHP 5, кои објектно – ориентирани конструктори се достапни?**

- Abstract класите
- Final класите
- Public, private, protected (PPP) методите
- Интерфејси

- a) Abstract classes
- b) PPP методите
- c) Ниту PPP методите ниту пак интерфејсите
- d) Никој од горенаведените не се достапни
- e) Сите од горенаведените се достапни

**10. Како ќе го повикате mymethod методот на класата во рамките на самата класа?**

- a) \$self=>mymethod();
- b) \$this->mymethod();
- c) \$current->mymethod();
- d) \$this::mymethod();
- e) Никое од горенаведените не е точно

**11. Што е излезот од следнава скрипта?**

```
<?php
class my_class
{
    var $my_var;
    function _my_class ($value)
    {
        $this->my_var = $value;
    }
}
$a = new my_class (10);
echo $a->my_var;
?>
```

- a) 10
- b) Нула
- c) Празно
- d) Ништо
- e) Грешка

**12. Што е излезот од следнава скрипта?**

```
<?php
class my_class
{
    var $value;
}
$a = new my_class;
$a->my_value = 5;
$b = $a;
$b->my_value = 10;
echo $a->my_value;
?>
```

- a) 10
- b) 5
- c) 2
- d) Нула
- e) ништо

**13. Разгледајте ја следнава скрипта. Што ќе биде нејзиниот излез?**

```
<?php
$global_obj = null;
class my_class
{
    var $value;
    function my_class()
    {
        global $global_obj;
        $global_obj = &$amp;this;
    }
}
$a = new my_class;
$a->my_value = 5;
$global_obj->my_value = 10;
echo $a->my_value;
?>
```

- a) 5
- b) 10
- c) Ништо
- d) Конструкторот ќе најде грешка
- e) 510

**14. Разгледајте го следниов сегмент од PHP кодот. Кога тој ќе се изврши, која вредност ќе ја има стрингот вратен од \$eight\_tenths->to\_string методот?**

```
<?php
class fraction {
var $numerator;
var $denominator;
function fraction($n, $d) {
$this->set_numerator($n);
$this->set_denominator($d);
}
function set_numerator($num){
$this->numerator= (int)$num;
}
function set_denominator($num) {
$this->denominator = (int)$num;
}
function to_string() {
return "{ $this->numerator}
/ { $this->denominator}";
}
}
function gcd($a, $b) {
return ($b > 0) ? gcd($b, $a % $b) : $a;
}
function reduce_fraction($fraction) {
$gcd= gcd($fraction->numerator,
$fraction->denominator);
$fraction->numerator /= $gcd;
$fraction->denominator /= $gcd;
}
$eight_tenths = new fraction(8,10);
/* Reduce the fraction */
reduce_fraction($eight_tenths);
var_dump($eight_tenths->to_string());
?>
```

- a) reduce\_fraction функцијата мора да врати вредност
- b) reduce\_fraction функцијата треба да прифати целобројни вредности
- c) gcd функцијата е неистинита
- d) Мора да го поминете \$eight\_tenths објектот со-референца
- e) Не можете да ги поминете инстанците на објектите со ништо освен со методи

**15. Што прави следниов PHP дел од код?**

```
<?php
require_once("myclass.php");
myclass::mymethod();
?>
```

- a) Го повикува mymethod методот во класата статички.
- b) Креира инстанци на myclass и го повикува mymethod методот.
- c) Генерира синтаксичка грешка
- d) Ги зема вредностите на последно создадената инстанца на myclass и го повикува mymethod()
- e) Ја повикува функцијата myclass::mymethod()

**16. Дали статични променливи на класа постојат во PHP?**

- a) Да
- b) Не

**17. Што е излезот од следнава скрипта?**

```
<?php
class a
{
function a ($x = 1)
{
$this->myvar= $x;
}
}
class b extends a
{
var $myvar;
function b ($x = 2)
{
$this->myvar= $x;
parent::a();
}
}
$obj = new b;
echo $obj->myvar;
?>
```

- a) 1
- b) 2
- c) Грешка, бидејќи a::\$myvar не е дефинирана
- d) Предупредување, бидејќи a::\$myvar не е дефинирана
- c) Ништо

**18. Како може да се вчитаат класи по барање како што тие се бараат од страна на интерпретерот?**

- a) Со користење на \_\_autoload магичната функција
- б) Со нивно дефинирање како напредни класи
- в) Со воведување на посебен управувач за грешка
- г) Не е можно да се вчитаат класи по барање
- д) Со нивно вклучување во условни искази

**19. \_\_\_\_\_ се користи за да се обезбеди висок квалитет на решенијата за проблемот на повторлив дизајн со користење на објектно - ориентираното програмирање.**  
Вашиот одговор е: \_\_\_\_\_

**20. Што е излезот од следнава скрипта?**

```
<?php
class a
{
function a()
{
echo 'Parent called';
}
}
class b
{
function b()
{
}
}
$c = new b();
?>
```

- a) Parent called
- b) Грешка
- c) Предупредување
- d) Ништо