APLIKACJE INTERNETOWE
PHP

TEMAT 4-02: Metody specjalne.

Autor dokumentu: Wojciech Galiński	środa, 16 listopada 2016 r.	351203 Technik informatyk
ŹRÓDŁA WIEDZY:	http://www.php.net/manual/pl, http://webmaster.helion.pl/index.php/kurs-php, http://pl.wikipedia.org/, http://pl.wikibooks.org/wiki/PHP, http://phpkurs.pl/, http://kursphp.com/.	

Zagadnienia obowiązkowe

1. **Konstruktor klasy** – to specjalna publiczna funkcja składowa klasy automatycznie wywoływana bezpośrednio po utworzeniu obiektu. W PHP można zdefiniować tylko 1 konstruktor.

   PRZYKŁAD (przykładowy konstruktor klasy „TLicznik”):

```
class TLicznik
{
   …
   public function __construct($wartosc = 0)          // konstruktor klasy z 1 parametrem
   {
      echo 'Tworzę obiekt klasy "TLicznik"<br />';
      $this->ustaw($wartosc);
   }
   private function ustaw($w = 0)                     // teraz metoda "ustaw" może być prywatna
   {
      $this->wartosc = ($w<=0? rand(2, 9): $w);
   }
   …
};
```

2. **Destruktor klasy** – to specjalna funkcja automatycznie wywoływana bezpośrednio przed usunięciem obiektu. Destruktor nie może mieć parametrów.

   PRZYKŁAD:

```
class TLicznik
{
   …
   public function __destruct()                       // destruktor klasy zawsze bez parametrów
   {
      echo '<p>Usuwam obiekt klasy "TLicznik".</p>';
   }
   …
};
```

ĆWICZENIE 1: *Uwzględniając powyższy kod, zmodyfikuj kod skryptu z definicją klasy „TLicznik” (z poprzedniego tematu) w taki sposób, żeby nie wyświetlały się błędy.*

Gdzie wywoływane są: konstruktor i destruktor?

ĆWICZENIE 2: *Do powyższego skryptu dodaj następujący kod w taki sposób, żeby nie wyświetlały się błędy:*

```
$licznik_2 = new TLicznik(0);   $licznik_2->odliczaj();
$licznik_3 = new TLicznik(3);   $licznik_3->odliczaj();
```

Gdzie są przesyłane liczby w nawiasie podczas tworzenia obiektu klasy „TLicznik” i jak wpływają na wyświetlane wyniki?

3. **Statyczne pole składowe klasy** – to zmienna składowa lub obiekt składowy z dopiskiem „**static**”. Tworzone są na początku skryptu, więc dostępne są nawet przed utworzeniem pierwszego obiektu takiej klasy. Dostęp do nich uzyskujemy za pomocą operatora „**::**”.

   Wartość pola statycznego klasy jest wspólna dla wszystkich obiektów tej klasy.

   PRZYKŁAD:

```
class TLicznik
{
   …
   static public $ile_licznikow = 0;        // definicja pola statycznego
   public function __construct($wartosc = 0) // teraz wyświetla konstruktor nową ilość obiektów
   {
      ++self::$ile_licznikow;
      echo '<p>Tworzę licznik (aktualna ilość liczników: '. self::$ile_licznikow. ').</p>';
      $this->ustaw($wartosc);
   }
   public function __destruct()          // teraz destruktor wyświetla pozostałą ilość obiektów
   {
      --self::$ile_licznikow;
      echo '<p>Usuwam licznik (aktualna ilość liczników: '. self::$ile_licznikow. ').</p>';
   }
   …
};
echo '<p>'. TLicznik::$ile_licznikow. '</p>';       // umieść tę instrukcję na początku skryptu
$licznik_4 = new TLicznik;   echo '<p>'. $licznik_4::$ile_licznikow. '</p>';   // a tę na końcu
```

   ĆWICZENIE 3: *Uwzględniając powyższy kod, zmodyfikuj kod skryptu z definicją klasy „TLicznik” w taki sposób, żeby nie wyświetlały się błędy. Jak to się dzieje, że ilość liczników wyświetla się prawidłowo?*

4. **Statyczne metody klasy** – to funkcje składowe odwołujące się do statycznych pól składowych. Wewnątrz definicji tych funkcji nie można odwoływać się do zmiennej „**$this**”. Metody takie wywołujemy za pomocą operatora „**::**”.

   PRZYKŁAD:

```
class TLicznik
{
   …
   static private $ile_licznikow = 0;                // teraz pole statyczne może być prywatne
   static public function wyswietl_ile_licznikow()   // metoda wyświetlająca wartość w/w pola
   {
      echo '<p>Aktualna liczba liczników: '. self::$ile_licznikow. '.</p>';
   }
   …
};
TLicznik::wyswietl_ile_licznikow();                        // tą instrukcję zamień z pierwszą
$licznik_4 = new Tlicznik(2);   $licznik_4::wyswietl_ile_licznikow();   // tą zamień z ostatnią
```

   ĆWICZENIE 4: *Uwzględniając powyższy kod, zmodyfikuj kod skryptu z definicją klasy „TLicznik” w taki sposób, żeby nie wyświetlały się błędy. Dlaczego powyższe pole statyczne powinno być prywatne?*

Zadania

1. Wykonaj powyższe ćwiczenia.
2. Utwórz klasę implementującą działanie formularza (dla uproszczenia zawartość formularza traktuj jako kod HTML). Klasa ma przechowywać informacje o nazwie, identyfikatorze formularza oraz o akcji i sposobie wysyłania danych do skryptu. Niech wartości te będzie można ustalić podczas tworzenia obiektów tej klasy (wewnątrz konstruktora klasy). Oprócz tego, funkcja ma posiadać metodę zwracającą oraz generującą kod HTML.
3. Wykonaj klasę „Kantor” pozwalającą na zamianę polskiej waluty na 3 inne. Nazwy walut przechowuj w statycznych polach klasy.