Indkøbsvogn i PHP

I denne artikel vil jeg prøve at lave en indkøbskurv som let kan udvides, og som ikke er svær at forstå. Det er bedst med viden om OOP(klasser), helst PHP5, men PHP4 kan også gøre det. Man kan med en god hjerne også gætte sig til lidt OOP :)

Skrevet den **06. Feb 2009** af **mysitesolution** I kategorien **Programmering / PHP** | ★★★☆☆

Som sagt kræves der helst lidt viden om OOP hvis man virkelig vil lærer noget, og ikke bare lave copy-paste kode :)

Edit:
$_SESSION["cart"] == $cart->GetProducts();

skulle være

$_SESSION["cart"] = $cart->GetProducts();

Af forskelle fra PHP4 og PHP5 er kan nævnes:

Variabler
En variabel i en PHP4 klasse defines med "var $var1;" og den kan bruges både indenfor og udenfor klassen. I PHP5 kan man selv vælge om den skal kunne ses udenfor klassen. Ved "private $var1;" kan den kun ses indenfor klassen, og ved "public $var1;" kan den ses både udenfor og indenfor klassen.

Funktioner
PHP4´s funktioner ses ud som dette "function FunktionsNavn() {}" og kan som "var" ses både udenfor og indenfor klassen. Ved PHP5 kan man også gøre som PHP4, men man kan også sætte private eller public foran.

Klassen
Først laver vi selvfølgelig klassen, som jeg ikke regner med skal forklares...

```
<?php

    class ShoppingCart
    {
    }

?>
```

Produkt variabel
Vi skal så have en variabel til at gemme vores produkter i. Her bruger jeg en privat variabel, da jeg ikke synes det er hensigtmæssigt at man skal kunne ændre den på alle leder og kanter udefra. Min kodestil, er at bruge små bokstaver til private variabler og stort bokstav til offentlige variabler. Derfor bliver navnet $products.

```
<?php

    class ShoppingCart
    {
        private $products;
    }

?>
```

Til produkterne vil jeg gemme 2 ting, varernummer (pnum) og antal (count), man kunne også vælge at gemme navne på produkterne osv., men jeg synes det er en klart bedre metode at konstant hente navne osv. direkte fra database. Man kunne evt. lave en cache til dette.

Funktion: GetProducts
Vi skal have en funktion som henter vores produkter ud som et array, og det gøres meget let, ved bare at retunere vores product array.

```
private function GetProducts()
{
    return $this->products;
}
```

Funktion: GetProductCount
Vi laver også en funktion til at hente antal produkter af en slags. Vi sender varenummeret med funktionen, og hvis varen ikke findes, så skal den retunere false.

Det gør vi med if (isset($this->products[$pnum])) hvor isset retunere true hvis produktet findes, og false hvis det ikke findes. Hvis det findes retunere vi så nummeret, og hvis ikke så retunere vi false.

```
public function GetProductCount($pnum)
{
    if (isset($this->products["$pnum"]))
        return $this->products["$pnum"];
    else
        return false;
}
```

Funktion: AddProduct
Vi skal selvfølgelig også en funktion til at tilføje et produkt. Det produkt skal ikke tilføjes hvis det allerede findes, men hvis det findes, skal antal af produktet forhøjes med antallet.

Vi tjekker også om varenummeret og antallet virkelig er et tal. Dette gøres med is_numeric(). Der er den ulempe ved is_numeric at et decimal tal også bliver accepteret. Det kan løses på flere måder, men det er stortset underordnet.

Derefter tjekker vi om produktet findes, som vi også gjorde før. Hvis det gør, så skal antallet bare forhøjes, og ellers skal produktet tilføjes.

```
public function AddProduct($pnum, $count)
{
    if (is_numeric($pnum) && is_numeric($count))
    {
        if (isset($this->products["$pnum"]))
            $this->products["$pnum"] += $count;
        else
            $this->products["$pnum"] = $count;
    }
    else
        return false;
}
```

Funktion: SetProducts
Denne funktion forklares yderligere senere, men den skal bare kunne tjekke om et array sendt med, er numre, og derefter udskifte vores product array med det der er sendt med. Der er ikke så meget nyt i denne funktion.

```
public function SetProducts($array)
{
    $this->products = array();

    foreach ($array as $key => $value)
        if (is_numeric($key) && is_numeric($value))
            $this->products["$key"] = $value;
}
```

Endelig klasse
Jeg har en masse andre funktioner at ville fyrer ind i den. Men da jeg mener det er bedst med forholdsvis overskuelige artikler så stopper jeg her. Til dem der vil have mere, kan jeg evt. lave endnu en artikel med en udvidet klasse.

```
<?php

    class ShoppingCart
    {
        private $products;

        public function AddProduct($pnum, $count)
        {
            if (is_numeric($pnum) && is_numeric($count))
            {
                if (isset($this->products["$pnum"]))
                    $this->products["$pnum"] += $count;
```

```
                else
                    $this->products["$pnum"] = $count;
            }
            else
                return false;
        }

        public function GetProductCount($pnum)
        {
            if (isset($this->products["$pnum"]))
                return $this->products["$pnum"];
            else
                return false;
        }

        public function GetProducts()
        {
            return $this->products;
        }

        public function SetProducts($array)
        {
            $this->products = array();

            foreach ($array as $key => $value)
                if (is_numeric($key) && is_numeric($value))
                    $this->products["$key"] = $value;
        }
    }

?>
```

Afprøvning
jeg har lavet en lille fil som kan teste din kurv. Den ses her:

```
<?php

    require("cart.class.php");           //Loader vores klasse

    $cart = new ShoppingCart;            //Opretter vores klasse

    $products["1212"] = 2;
    $products["1214"] = 2;

    $cart->SetProducts($products);       //Sætter vores nye produkter

    $cart->AddProduct("1212", 2);        //Tilføjer produkt
    $cart->AddProduct("1212", 4);        //Tilføjer produkt

    echo $cart->GetProductCount("1212"); //Udskriver vare antal for 1212
```

```
    print_r($cart->GetProducts());       //Udskrive produkterne

    if ($cart->AddProduct("jkh", false)) //Er dette add gyldigt?
        echo "Produktet blev tilføjet.";
    else
        echo "Produktet var ugyldigt.";  //Nej er det ikke!

?>
```

Hvis den virker skulle den gerne retunere noget lignede dette:

```
8Array
(
    [1212] => 8
    [1214] => 2
)
Produktet var ugyldigt.
```

Brug i det "virkelige" liv
Nu har vi jo set hvordan vi kan holde på produkterne osv. men hvordan overfører vi det side til side?

Det er ret let :)

Vi kan gøre det på forskellige måder. Den ene er at gemme indholdet i en session. Det kan være en dårlig måde, da det gemmes i rammene, så man kan også gemme et id og hente det fra en database.

Vi tager den lette her i første omgang (Den anden kan jeg tage til en artikel 2).

Sidst i et PHP dokument tilføjer vi bare dette:

```
session_start();

$_SESSION["cart"] = $cart->GetProducts();
```

Vi kan så hente indholdet i starten af en side ved at bruge:

```
session_start();

$cart = new ShoppingCart;

$cart->SetProducts($_SESSION["products"]);
```

Dette burde give sig selv. Hvis du kan noget OOP, så kan du i hvert fald også sessions :)

Da dette er min første artikel, så må i også gerne vurdere min måde at skrive artikler på, så jeg ved hvad

jeg skal gøre anderledes en anden gang. :)

Ellers, så skriv hvis jeg har lavet et par fejl.

Kommentar af olebole d. 29. Jul 2007 | 1

Du skøjter voldsomt hurtigt hen over forskellene på OOP under PHP4 og PHP5

Kommentar af cf560 d. 25. Nov 2006 | 2

Super

Kommentar af plazm d. 28. Nov 2005 | 3

Kanon artikkel. Har ikke testet den, men den giver en meget god gennemgang og beskriver koden godt.

Kommentar af nyhuusdk d. 02. Jun 2008 | 4

Jeg synes den er rigtig god, med lidt udbygning virker den geneialt :)

Lære mig os lidt mere om formålet med oop og klasser generalt

Kommentar af wicez (nedlagt brugerprofil) d. 02. Dec 2005 | 5

God beskrivelse af OOP

Kommentar af mr-kill d. 28. Nov 2005 | 6

super nice :D

Kommentar af there-is-only-xul d. 30. Dec 2005 | 7

fair artikel, men 1) brug singleton 2) $this->products["$pnum"] er lidt forkert, da der ikke skal " " omkring $variable 3) undgå at ligne php.net/oop så meget..

Kommentar af pvtsommer d. 07. Apr 2006 | 8

Virkelig flot og overskueligt sat op!

Kommentar af sommer89 d. 29. Nov 2005 | 9

Fin artikel. Kan bestemt anbefales.

Kommentar af windcape d. 29. Jul 2007 | 10

rimelig god, men konceptet kunne være designet meget bedre med automatisk serialization via. en singleton.