

Exercises - Object Oriented Programming

1. Klasser, egenskaper och metoder

1 . Skriv en klass med namnet `Book` . Den ska ha två (*publika*) egenskaper: `title` och `author` .

2 . Skapa ett objekt av klassen `Book` och spara det i en variabel med namnet `$book` . Sätt värdet på egenskaperna `title` till "*Främlingen*" och `author` till "*Albert Camus*". Testa att du har gjort rätt genom att skriva ut värdet på egenskaperna `title` och `author` så här:

```
echo "Titel: $book->title <br>Författare: $book->author <br>";
```

3 . Skapa ett till objekt av klassen `Book` med `title` : "*Harry Potter and the Philosopher's Stone*" och `author` : "*J K Rowling*".

4 . Ändra det senaste objektet så att `title` blir "*Harry Potter and the Order of the Phoenix*".

5 . Skapa en metod i `Book` med namnet `printTitle` . Metoden ska inte ta några parametrar. När den anropas ska den skriva ut bokens titel med `echo` .

6 . Skapa en klass med namnet `Car` . Den ska ha flera egenskaper: `model` , `color` och `price` . Skapa ett objekt av klassen `Car` och ge det lämpliga värden på egenskaperna.

7 . Lägg till en metod i `Car` med namnet `printInfo` . Metoden ska inte ha några parametrar. När metoden anropas ska den skriva ut information om `Car` -objektet. Till exempel, om `model="Volvo"` , `color="red"` och `price=25000` så ska funktionen skriva:

```
"Det här är en röd Volvo som kostar 25000 kr".
```

8 . Skapa en metod i `Car` med namnet `halfPrice` . När metoden anropas ska den ändra värdet på egenskapen `price` till hälften.

9 . Skapa en klass med namnet `Product` . Klassen representerar något som man kan sälja i en webbshop. Den ska ha egenskaperna: `name` , `description` , `price` , `imageUrl` , `quantity` .

10 . Skriv en klass med namnet `SingleBookLibrary` . Den ska ha egenskaper med namnet `book` och `isBorrowed` . Egenskapen `book` ska vara ett objekt av klassen `Book` . Lägg till en metod med namnet `borrow` , som ändrar värdet på `isBorrowed` till `true` . Metoden ska också skriva ut med `echo` om det gick att låna, eller om boken redan var utlånad.

11 . Skapa en metod i `Product` med namnet `printInfo` , som fungerar på samma sätt som `printInfo` i uppgift 7.

2. Konstruktörer

1 . Lägg till en konstruktor till klassen `Book` från förra stycket. Konstruktorn ska ta två argument och använda dem för att sätta värdet på egenskaperna `title` och `author` .

2 . Lägg till en konstruktor till klassen `Car` . Konstruktorn ska ta tre argument och sätta värdet på egenskaperna `model` , `color` och `price` .

3 . Lägg till en egenskap till klassen `Car` , `sellDate` , som motsvarar när bilen såldes. Konstruktorn ska sätta `sellDate` till dagens datum. Exempel: "2017-03-27". Tips: använd `PHP.net: Date`

4 . Skapa en konstruktor till klassen `Product` som tar argumenten `$name` och `$price` . Konstruktorn ska hitta på lämpliga värden på de andra parametrarna.

5 . Ändra konstruktorn för klassen `Product` så att den förutom `name` och `price` även tar `description` , `imageUrl` och `quantity` . Men de ska ha lämpliga default-värden.

6. Ändra konstruktorn för klassen `Car` så att den tar en parameter för varje egenskap som `Car` har. Varje parameter ska ha ett lämpligt default-värde.

Lösningsförslag

```
<?php

/*=====
=           Book Class           =
=====*/

class Book {
    private $title;
    public $author;

    public function __construct($title, $author) {
        $this->title = $title;
        $this->author = $author;
    }
    public function printTitle() {
        return "Titel: $this->title <br>";
    }
    public function getTitle() {
        return $this->title;
    }
    public function setTitle($titleName) {
        $this->title = $titleName;
    }
}
$book1 = new Book('The Dark Forest', 'Liu Cixin');

$book1->setTitle('Harry Potter');
echo "Titel: " . $book1->getTitle() . "<br>Författare: $book1->author <br>";

echo $book1->printTitle();
```

```
/*=====
=           Car Class           =
=====*/

class Car {
    private $model;
    private $color;
    private $price;
    private $sellDate;

    public function __construct($model, $color, $price, $sellDate = "2017-04-18") {
        $this->model = $model;
        $this->color = $color;
        $this->price = $price;
        $this->sellDate = "2017-03-27";
        // TODO: använd date(..)
    }
    public function printInfo() {
        echo "Det här är en $this->color $this->model "
            . "som kostar $this->price kr.<br>";
    }
    public function halfPrice() {
        $this->price = $this->price / 2;
    }
    public function changeCar($model, $color, $price) {
        $this->model = $model;
        $this->color = $color;
        $this->price = $price;
    }
}

$car1 = new Car('Tesla', 'silverfärgad', 1000000);
$car1->halfPrice();
$car1->changeCar('Fiat', 'blå', 80000);
$car1->printInfo();
```

```
/*=====
=          SingleBookLibrary Class          =
=====*/

class SingleBookLibrary {
    public $book;
    public $isBorrowed;

    public function borrow() {
        if( $this->isBorrowed ) {
            echo "Boken är redan utlånad.<br>";
        } else {
            echo "Boken var inte utlånad, det går bra att låna.<br>";
            $this->isBorrowed = true;
        }
    }
}

$library = new SingleBookLibrary();
$library->isBorrowed = false;
$library->book = new Book('Pesten', 'Albert Camus');
    // Book har egenskaperna: title, author
$library->borrow();
$library->borrow();
$library->borrow();
```