

## 9.1 Vragen over codevoorbeeld 'lottoKlasse.php'.

9.1.1 In C# kan je de constructor herkennen zodra je de klassenaam weet. In PHP ziet deconstructor er anders uit. Hoe?

`__construct()`

9.1.2 Op lijn 30 start een while-lus. Wat is de bedoeling van die lus?

Deze overloopt de lottoArray en checkt of het nieuwe gegenereerde getal niet reeds aanwezig is, indien dit wel het geval is, wordt een nieuw getal gegenereerd.

9.1.3 In de method ToonLottoGetallen(), wordt er gebruik gemaakt van `&nbsp;`. Wat stelt zoiets voor?

Een non-breaking space, wanneer je bijvoorbeeld een prijs en valuta weergeeft wil je niet dat deze laatstgenoemde op een nieuwe lijn zou verschijnen.

## 9.2 Vragen over codevoorbeeld 'oopSimpleObjects.php'.

9.2.1 Op lijn 17 wordt een property beleg gekoppeld met een nieuw array met ingrediënten. Waar wordt de property beleg gedefinieerd?

Nergens, deze property wordt 'on the fly' bij gecreëerd.

9.2.2 Op lijn 47 start een foreach-lus, maar die lus heeft in zich ook nog 2 foreachlussen. Waarom precies?

De foreach-lus op lijn 47 dient om al de pizza's in de 'pizzas'-array te overlopen. De foreach-lus die daarbinnen staat dient om alle properties van deze pizza te overlopen. Wanneer 1 van deze properties zelf ook een object of array bijhoudt, wordt dit opgevangen in de binnenste foreach-lus, die op zijn beurt ook al deze key-values zal overlopen.

## 9.3 Vragen over codevoorbeeld 'oopClassObjects.php'.

9.3.1 Waarom staat er op lijn 18 twee keer "NA"?

Dit is een standaardwaarde, indien de constructor zou worden opgeroepen zonder parameters mee te geven.

9.3.2 Wat is de reden dat er op lijn 34 `this` gebruikt wordt in het statement `$this->soort = $pSoort;` ?

Met `this` kan een object naar zichzelf refereren, dit kan dus gebruikt worden om via een methode een property van zichzelf aan te passen.