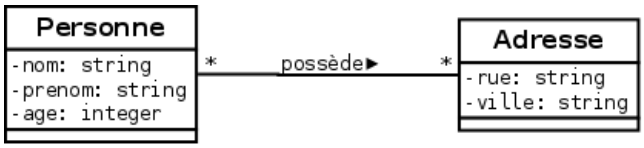
Programmation orientée objet en PHP

Une Personne et plusieurs Adresse(s)

Comment pourrions-nous donner à une classe Personne plusieurs Adresse(s) ? Si une Personne possède plusieurs Adresse(s) cela signifie que la classe Personne possédera plusieurs attributs d'instance Adresse. Ce que nous pourrions représenter par le schéma :



L'association (le lien) entre les deux classes signifie qu'une Personne peut posséder plusieurs Adresse(s) et qu'une Adresse peut appartenir à plusieurs Personne(s). Le caractère "*" signifie "plusieurs".

Il nous suffit donc de créer autant d'attributs d'instance que d'adresses : si la Personne possède 4 adresses alors nous aurons 4 attributs d'instance (adresse1, adresse2, etc...). Mais si le nombre d'Adresse(s) par Personne est indéterminé alors il nous faudra choisir une autre solution, celle du tableau. Notre classe Personne possédera un attribut d'instance de type array qui stockera l'ensemble de ses Adresse(s). Cet attribut d'instance sera nommé adresses (avec un "s" à la fin pour bien montrer qu'il y en a plusieurs).

```php
<?php

// Personne.php
class Personne {

    private $nom;
    private $prenom;
    private $age;
    private $adresses;

    function __construct($nom, $prenom, $age, $adresses = []) {
        $this->nom = $nom;
        $this->prenom = $prenom;
        $this->age = $age;
        $this->adresses = $adresses;
    }

    public function getNom() {
        return $this->nom;
    }

    public function getPrenom() {
        return $this->prenom;
    }

    public function getAge() {
        return $this->age;
    }

    public function getAdresses() {
        return $this->adresses;
    }

    public function setNom($nom) {
        $this->nom = $nom;
```

```
    }

    public function setPrenom($prenom) {
        $this->prenom = $prenom;
    }

    public function setAge($age) {
        $this->age = $age;
    }

    public function setAdresses($adresses) {
        $this->adresses = $adresses;
    }

}
```

Le constructeur de Personne possède un paramètre facultatif qui correspond au tableau de Personne. Nous pourrons, lors de l'instanciation de l'objet, passer en argument au constructeur un tableau d'Adresse(s), dans ce cas une copie de ce tableau deviendra attribut d'instance. Mais nous pourrons aussi laisser ce paramètre vide, dans ce cas l'attribut d'instance sera un tableau vide que nous pourrons remplir plus tard avec une méthode addAdresse() par exemple :

```
<?php

// Personne.php
class Personne {

    private $nom;
    private $prenom;
    private $age;
    private $adresses;

    function __construct($nom, $prenom, $age, $adresses = []) {
        $this->nom = $nom;
        $this->prenom = $prenom;
        $this->age = $age;
        $this->adresses = $adresses;
    }

    public function getNom() {
        return $this->nom;
    }

    public function getPrenom() {
        return $this->prenom;
    }

    public function getAge() {
        return $this->age;
    }

    public function getAdresses() {
        return $this->adresses;
    }

    public function setNom($nom) {
        $this->nom = $nom;
    }

    public function setPrenom($prenom) {
        $this->prenom = $prenom;
    }

    public function setAge($age) {
        $this->age = $age;
```

```
    }

    public function setAdresses($adresses) {
        $this->adresses = $adresses;
    }

    public function addAdresse($adresse) {
        $this->adresses[] = $adresse;
    }

}
```

Et voici notre lanceur :

```
<?php

// lanceur.php
include './Adresse.php';
include './Personne.php';

$adr1 = new Adresse('Lignier', 'Paris');
$adr2 = new Adresse('Turbigo', 'Paris');
$sparrow = new Personne('Sparrow', 'Jack', 42, [$adr1, $adr2]);
$wayne = new Personne('Wayne', 'Bruce', 40);
print_r($sparrow->getAdresses());
$wayne->addAdresse($adr2);
$wayne->addAdresse($adr1);
print_r($wayne->getAdresses());
```

Nous pouvons ajouter à notre classe Adresse une méthode __toString() qui renverra une chaîne de caractères contenant la rue et la ville :

```
<?php

// Adresse.php
class Adresse {

    private $rue;
    private $ville;

    function __construct($rue, $ville) {
        $this->rue = $rue;
        $this->ville = $ville;
    }

    public function getRue() {
        return $this->rue;
    }

    public function getVille() {
        return $this->ville;
    }

    public function setRue($rue) {
        $this->rue = $rue;
    }

    public function setVille($ville) {
        $this->ville = $ville;
    }

    public function __toString() {
        return "rue : $this->rue, ville : $this->ville";
    }

```

```
}
```

Nous ajoutons à Personne une méthode displayAdresses() qui renverra une chaîne de caractères avec toutes les Adresses et qui réutilisera la méthode __toString() d'Adresse. Et nous créons une méthode __toString() qui renverra une chaîne contenant le nom, le prénom, l'age et les adresses de la Personne.

```php
<?php

// Personne.php
class Personne {

    private $nom;
    private $prenom;
    private $age;
    private $adresses;

    function __construct($nom, $prenom, $age, $adresses = []) {
        $this->nom = $nom;
        $this->prenom = $prenom;
        $this->age = $age;
        $this->adresses = $adresses;
    }

    public function getNom() {
        return $this->nom;
    }

    public function getPrenom() {
        return $this->prenom;
    }

    public function getAge() {
        return $this->age;
    }

    public function getAdresses() {
        return $this->adresses;
    }

    public function setNom($nom) {
        $this->nom = $nom;
    }

    public function setPrenom($prenom) {
        $this->prenom = $prenom;
    }

    public function setAge($age) {
        $this->age = $age;
    }

    public function setAdresses($adresses) {
        $this->adresses = $adresses;
    }

    public function addAdresse($adresse) {
        $this->adresses[] = $adresse;
    }

    public function displayAdresses() {
        $result = "";
        foreach ($this->adresses as $adresse) {
            $result .= $adresse . " ";
        }
        return $result;
    }
```

```
61
62      public function __toString() {
63          return "nom : $this->nom prénom : $this->prenom age : "
64                  . "$this->age adresse(s) : {$this->displayAdresses()}";
65      }
66
67  }
```

Notre lanceur devient :

```
1   <?php
2
3   // lanceur.php
4   include './Adresse.php';
5   include './Personne.php';
6
7   $adr1 = new Adresse('Lignier', 'Paris');
8   $adr2 = new Adresse('Turbigo', 'Paris');
9   $sparrow = new Personne('Sparrow', 'Jack', 42, [$adr1, $adr2]);
10  $wayne = new Personne('Wayne', 'Bruce', 40);
11  echo $sparrow;
12  $wayne->addAdresse($adr2);
13  $wayne->addAdresse($adr1);
14  echo $wayne;
```

Fin

Les namespaces
Les exceptions
Les bases de données avec PDO
Les tests avec PHPUnit
Petite application version 2
Petitie application version 3

Mes cours

ADMINISTRATION

Administration du cours

Réglages de mon profil

Connecté sous le nom « Arnaud Lemais » (Déconnexion)
POO PHP