



Programmation orientée objet en PHP

[Accueil](#) ▶ [Mes cours](#) ▶ [Développement logiciel](#) ▶ [POO PHP](#) ▶ [Les interfaces](#) ▶ [Introduction](#)

Introduction

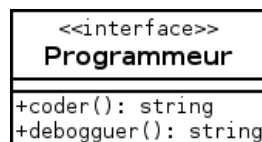
Les [interfaces](#) sont ce que l'on appelle des contrats d'implémentation. Elles servent à indiquer les "compétences" que doit posséder une classe.

Nous trouverons dans une interface une ou plusieurs méthodes abstraites, c'est-à-dire des méthodes sans implémentation. Ces méthodes peuvent être de classe ou d'instance.

Une interface ne déclarera aucun attribut d'instance ou de classe, ni de méthode "non abstraite". Par contre elle pourra déclarer des constantes. Toute méthode déclarée dans une interface est publique et abstraite.

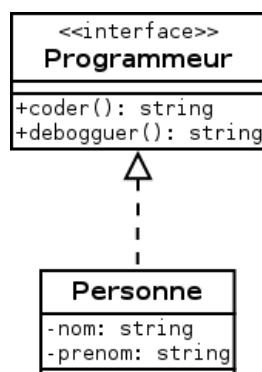
Les classes peuvent implémenter une ou plusieurs interfaces. Pour être "concrètes", ces classes doivent fournir une implémentation aux méthodes définies dans les interfaces. Une classe qui implémente une ou plusieurs interfaces mais qui ne fournit pas d'implémentation aux méthodes abstraites sera forcément une classe abstraite.

Dans un diagramme de classe une interface porte le stéréotype <<interface>> :



```
1 <?php
2
3 // Programmeur.php
4 interface Programmeur {
5
6     public function coder();
7
8     public function debogguer();
9 }
```

Créons une classe *Personne* qui implémente l'interface *Programmeur*. Dans un diagramme la flèche en pointillés signifie "implémente".



En PHP, une classe implémente une ou plusieurs interfaces en utilisant le mot clé "implements". Les noms des interfaces sont séparés par des virgules s'il y en a plusieurs.

```
1 <?php
2
3 // Personne.php
4 class Personne implements Programmeur {
5 }
```

```

6     protected $nom;
7     protected $prenom;
8
9     function __construct($nom, $prenom) {
10         $this->nom = $nom;
11         $this->prenom = $prenom;
12     }
13
14     public function getNom() {
15         return $this->nom;
16     }
17
18     public function getPrenom() {
19         return $this->prenom;
20     }
21
22     public function setNom($nom) {
23         $this->nom = $nom;
24     }
25
26     public function setPrenom($prenom) {
27         $this->prenom = $prenom;
28     }
29
30     public function coder() {
31         echo 'je code...';
32     }
33
34     public function debogguer() {
35         echo 'je déboggue...';
36     }
37
38 }

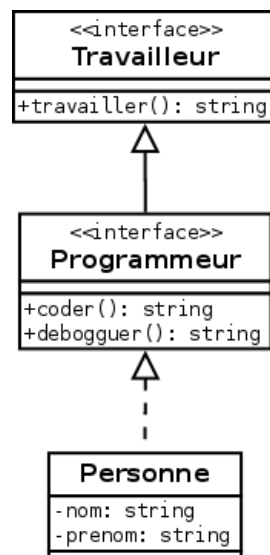
```

```

1 <?php
2
3 // lanceur.php
4 include './Programmeur.php';
5 include './Personne.php';
6
7 $pers = new Personne('Sparrow', 'Jack');
8 $pers->coder();
9 $pers->debogguer();

```

Les interfaces peuvent aussi hériter d'autres interfaces avec le mot clé extends :



```

1 <?php

```

```
2
3 // Travailleur.php
4
5 interface Travailleur {
6
7     public function travailler();
8 }
```

```
1 <?php
2
3 // Programmeur.php
4 interface Programmeur extends Travailleur {
5
6     public function coder();
7
8     public function debogguer();
9 }
```

```
1 <?php
2
3 // Personne.php
4 class Personne implements Programmeur {
5
6     protected $nom;
7     protected $prenom;
8
9     function __construct($nom, $prenom) {
10         $this->nom = $nom;
11         $this->prenom = $prenom;
12     }
13
14     public function getNom() {
15         return $this->nom;
16     }
17
18     public function getPrenom() {
19         return $this->prenom;
20     }
21
22     public function setNom($nom) {
23         $this->nom = $nom;
24     }
25
26     public function setPrenom($prenom) {
27         $this->prenom = $prenom;
28     }
29
30     public function coder() {
31         echo 'je code...';
32     }
33
34     public function debogguer() {
35         echo 'je déboggue...';
36     }
37
38     public function travailler() {
39         echo 'je travaille...';
40     }
41
42 }
```

```
1 <?php
2
```

```
3 // lanceur.php
4 include './Travailleur.php';
5 include './Programmeur.php';
6 include './Personne.php';
7
8 $pers = new Personne('Sparrow', 'Jack');
9 $pers->coder();
10 $pers->debogguer();
11 $pers->travailler();
```

[Fin](#)

NAVIGATION



Accueil

■ Ma page

[Pages du site](#)[Mon profil](#)[Cours actuel](#)

POO PHP

[Participants](#)[Généralités](#)[La programmation orientée objet : premiers pas](#)[L'héritage](#)[Les interfaces](#)[Introduction](#)[T.P. employés v2](#)[Le typage](#)[Les namespaces](#)[Les exceptions](#)[Les bases de données avec PDO](#)[Les tests avec PHPUnit](#)[Petite application version 2](#)[Petite application version 3](#)[Mes cours](#)

ADMINISTRATION

[Administration du cours](#)[Réglages de mon profil](#)

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