


- [Aucun ami connecté](#)
- [681 Zéros connectés](#)
- [436 267 Zéros inscrits](#)
- 
- [Gya](#)
 - [Avatar](#)
 - [Profil](#)
 - [Identifiants](#)
 - [Études](#)
 - [Réglages](#)
 - [Contacts](#)
 - [Commandes](#)
 -
- [Lisez vos Messages Personnels](#)

Correction du Q.C.M.

[<= Retour au cours](#)

-

Question

Qu'est-ce qu'une interface ?

- Bonne réponse

Une classe complètement abstraite

- Explications

Une interface est un squelette. Elle peut être implémentée par une ou plusieurs classe(s) et celle(s)-ci devra(ont) écrire toutes les méthodes de l'interface.

-

Question

Ce code est-il fonctionnel ?

Code : PHP - [Sélectionner](#)

```
<?php
interface A
{
    public function test();
}

class A implements A
{
```

```
public function test()  
{  
    }  
}  
?>
```

Bonne réponse

Non

• Explications

Le code ne peut fonctionner : une classe ne peut avoir le même nom qu'une interface.

•

Question

Et celui-ci, fonctionnera-t-il ?

Code : PHP - [Sélectionner](#)

```
<?php  
interface iA  
{  
    protected function test1();  
    abstract public function test2();  
}  
  
class A implements iA  
{  
    protected function test1()  
    {  
    }  
  
    abstract public function test2();  
}  
?>
```

Bonne réponse

Non, pour les deux raisons ci-dessus

• Explications

Dans une interface, toutes les méthodes doivent être publiques, et aucune ne peut être abstraite ou finale. 😊

•

Question

Une interface héritant d'une autre peut-elle écraser ses méthodes ? Et ses constantes ?

- Bonne réponse

Elle ne peut écraser ni les méthodes, ni les constantes

- Explications

Une interface ne peut réécrire une méthode d'une interface parente. Pour les constantes, le principe est le même qu'avec les classes : on ne peut écraser de constante.

-

Question

Quelle interface hérite de *Iterator* ?

- Bonne réponse

SeekableIterator

- Explications

Il s'agit de *SeekableIterator*. Les deux autres n'héritent d'aucune interface.



Note : 20 / 20

[<= Retour au cours](#)