


- [Aucun ami connecté](#)
- [915 Zéros connectés](#)
- [436 256 Zéros inscrits](#)
- 
- [Gya](#)
  - [Avatar](#)
    - [Profil](#)
    - [Identifiants](#)
    - [Études](#)
    - [Réglages](#)
    - [Contacts](#)
    - [Commandes](#)
  -
- [Lisez vos Messages Personnels](#)

## Correction du Q.C.M.

[<= Retour au cours](#)

- 

### Question

Quel mot-clé permet de dire à une classe qu'elle doit hériter d'une autre classe ?

- Bonne réponse

`extends`

- Explications

N'oubliez pas le "s" à la fin. 😊

- 

### Question

De quel type de visibilité d'attributs et méthodes une classe fille n'hérite pas de sa classe mère ?

- Bonne réponse

`private`

- Explications

Seul `private` ne laisse pas ses classes filles hériter de ses attributs et méthodes.

- 

#### Question

Quel mot-clé permet d'interdire toute classe d'hériter d'une certaine classe ?

- Bonne réponse

final

- Explications

C'est le concept de finalisation qui permet d'interdire à toute classe d'hériter d'une telle classe, ou alors d'interdire toute classe héritant d'une certaine classe de réécrire une méthode.

- 

#### Question

Qu'est-ce qui ne va pas dans ce code ?

Code : PHP - [Sélectionner](#)

```
<?php
class abstract Personnage
{
}

class Magicien extends Personnage
{
}
?>
```

- Bonne réponse

Le mot-clé "abstract" doit être placé avant le mot-clé "class"

- Explications

L'erreur dans ce script est situé à la deuxième ligne : le mot-clé `abstract` doit être placé **avant** le mot-clé `class`.

- 

#### Question

Le code suivant est-il fonctionnel ?

Code : PHP - [Sélectionner](#)

```
<?php
abstract class Personnage
{
    abstract public function frapper ($perso);

    final public function recevoirDegats()
    {
    }
}

class Magicien extends Personnage
{
    public function recevoirDegats ($perso)
    {
    }
}
?>
```

- Bonne réponse

Les deux

- Explications

Ce code est doublement bogué : on ne peut pas réécrire de méthode finale et on est obligé d'écrire les méthodes abstraites.

- 

#### Question

Terminons par la résolution statique à la volée et un bon vieux gros code. Si je l'exécute, qu'affichera-t-il à l'écran ?

#### Code : PHP - Sélectionner

```
<?php
class A
{
    public static function foo()
    {
        static::quiEstCe();
    }

    public static function quiEstCe()
    {
        echo 'A';
    }
}

class B extends A
{
    public static function test()
    {
        A::foo();
        parent::foo();
        self::foo();
    }
}
```

```
        public static function quiEstCe ()
        {
            echo 'B';
        }
    }

    class C extends B
    {
        public static function quiEstCe ()
        {
            echo 'C';
        }
    }

    C::test();
?>
```

Bonne réponse

ACC

• Explications

Si vous avez eu faux, je ne vous en voudrai pas, c'est pas facile à comprendre du premier coups. Toutes les explications sont détaillées en troisième partie. 😊



Note : 20 / 20

[<= Retour au cours](#)