



 resources (/uploads/document/document/120/c373669fdad85e48a06443f20c4ab198-j07.tgz)

## Sections

### Exercice 00

#### Exercice 00

Utilisez les fichiers fournis en annexe pour tester l'exercice.

La classe Tyrion DOIT hériter de Lannister;  
Son constructeur DOIT utiliser celui de son parent.

Si tout est OK, donnez les points.

 Yes

No

### Exercice 01

#### Exercice 01

Utilisez les fichiers fournis en annexe pour tester l'exercice.

test1.php doit fonctionner;  
test2.php doit produire une erreur fatale.

Le rendu devrait être une simple classe content un attribut protected.

Si tout est OK, donnez les points.

 Yes

No

### Exercice 02

#### Exercice 02

Utilisez les fichiers fournis en annexe pour tester l'exercice.

Le rendu devrait contenir une methode getBurned qui appelle static::resistsFire. Cependant, il y a d'autres solutions possibles, discutez-en.

Si tout est OK, donnez les points.

 Yes

No

### Exercice 03

#### Exercice 03

Utilisez les fichiers fournis en annexe pour tester l'exercice.  
test1.php doit fonctionner;

test2.php doit produire une erreur fatale.

Le rendu devrait être une classe comportant 3 methodes abstract (getHouse{Name,Motto,Seat}) et une methode introduit appelant les 3 methodes en question via linkage dynamique.

Si tout est OK, donnez les points

 Yes

No

## Exercice 04

### Exercice 04

Utilisez les fichiers fournis en annexe pour tester l'exercice.

Le rendu devrait jouer avec l'opérateur instanceof pour vérifier de quel type est l'argument de sleepWith.

Si tout est OK, donnez les points

 Yes

No

## Exercice 05

### Exercice 05

Utilisez les fichiers fournis en annexe pour tester l'exercice.

test1.php doit fonctionner;

test2.php doit produire une erreur fatale.

Le rendu devrait être une interface IFighter proposant la méthode fight(), et une classe NightsWatch qui maintient un Array interne, rempli via recruit(), et donc la méthode fight() vérifie que chaque membre de l'array implémente bien l'interface IFighter avant d'en faire quelque chose.

Si tout est OK, donnez les points

 Yes

No

## Exercice 06

### Exercice 06

Utilisez les fichiers fournis en annexe pour tester l'exercice.

test1.php doit fonctionner;

test2.php doit produire une erreur fatale.

Il y a pas mal de méthodes pour répondre à la problématique de la Factory, la plus simple est certainement que la classe Fighter propose une méthode du style copy/clone, qui renvoie une nouvelle instance du type concret de l'objet.

Si tout est OK, donnez les points.

 Yes

No

# Conclusion

Leave a comment on this correction

**\* (required)** Comment

Finish correction