

Master M1
Programmation Scientifique Orientée Objet
Partiel (1h30)

Exercice 1

Vous devez répondre à chacune des questions ci-dessous.

- 1) Qu'indique le mot clé "final" ?
- 2) Qu'est ce qu'une classe ?
- 3) Que signifie l'instanciation ?
- 4) Quelle "méthode" particulière est appelée lorsqu'on instancie un objet?
- 5) Quel mot clé permet d'appeler cette "méthode"?
- 6) Quelle méthode est appelée à la destruction d'un objet?
- 7) Quel mot clé désigne l'objet courant?
- 8) Que signifie l'utilisation du mot clé "static" lorsqu'on définit une méthode?
- 9) Quelle est la différence entre "double" et "Double" ?
- 10) Soit :

```
1 int a, b;  
2 a = 10;  
3 b = a;  
4 a = a + 1;
```

Que vaut b ?

- 11) Soit :

```
1 Point a, b;  
2 a = new Point(10, 50);  
3 b = a;  
4 System.out.println("x = " + a.x + "; y = " + a.y + ";");  
5 a.x = 11;  
6 System.out.println("x = " + b.x + "; y = " + b.y);
```

Que va afficher ce code ?

- 12) Que signifie cette notation :

```
1 int[] entiers;
```

- 13) Expliquez à quoi servent les trois blocs "try", "catch", et "finally" qui sont utilisés pour gérer des exceptions?
- 14) Que permet l'héritage? Et comment utilise-t-on ce mécanisme?
- 15) Que se passe-t-il si on définit une classe comme "final"? Qu'en est-il d'une méthode "final"?
- 16) Qu'est ce qu'une classe abstraite?

17) Quelles sont les différences entre une classe abstraite et une interface?

18) Soit une classe "ClasseA" qui possède la méthode "uneMethode()", et une classe "ClasseB" qui hérite de "ClasseA". "uneMethode()" a été redéfinie dans "ClasseB".

```
1 ClasseA a = new ClasseB();  
2 a.uneMethode();
```

A-t-on le droit d'écrire ce code? Et si oui, quelle méthode sera appelée?

19) Quelles sont les trois grandes étapes dans la gestion d'un flux d'entrée ou de sortie?

20) Que désigne "System.in" ?

Exercice 2

Dans cet exercice vous allez devoir développer une classe Vecteur contenant des réels. Vous devrez notamment développer les méthodes suivantes :

- public static Vecteur litVecteur() : cette première méthode demande à un utilisateur de saisir la taille du vecteur qu'il souhaite créer, ainsi que la valeur de chacun de ses éléments.
- public Vecteur getOpposite() : cette méthode crée un nouveau vecteur correspondant au vecteur opposé du vecteur courant tel que : si \vec{v} est le vecteur opposé de \vec{u} alors : $\vec{u} + \vec{v} = \vec{0}$
- public Vecteur subtraction(Vecteur v) : cette dernière méthode calcul la soustraction entre deux vecteurs tel que : $\vec{u} - \vec{v} = \vec{u} + (-\vec{v})$ (où $(-\vec{v})$ est le vecteur opposé de \vec{v})

Vous pourrez développer toute méthode ou tout constructeur dont vous pourriez avoir besoin.

Une attention particulière devra être portée sur l'indentation de votre code que vous devrez également commenter.

Vous pourrez vous aider d'une méthode "main" afin de tester votre code.

Enfin, vous enverrez par mail (à l'adresse thibaut.demare@univ-lehavre.fr) une archive ZIP du package contenant votre classe Vecteur. Le nom du package devra être de la forme "nom.prenom". Le sujet du mail devra comporter l'intitulé "[PartielPSOO]".