

Classe(s) :	Nom :	Prénom :	N° Ins. :
Date : 07/06/2022 Durée : 1h30	Session : <input checked="" type="checkbox"/> Principale <input type="checkbox"/> Rattrapage	Documents : <input checked="" type="checkbox"/> non autorisés <input type="checkbox"/> autorisés Calculatrice : <input checked="" type="checkbox"/> non autorisée <input type="checkbox"/> autorisée	Nbr. de pages : 7
Matière : Programmation Orientée Objet Enseignant(es) : I. Msadaa & T. Bchini & H. Gharsallaoui			Code :

Exercice 1 : (6 points)

1. Cocher Vrai ou Faux pour chacune de ces affirmations :

	Vrai	Faux
a. la classe qui contient une ou des méthodes abstraites est nécessairement abstraite		
b. Une classe déclarée <code>abstract</code> peut être instanciée		
c. Une classe qui hérite d'une classe abstraite est obligatoirement concrète		
d. Une classe qui hérite d'une classe abstraite est nécessairement abstraite		
e. Une classe peut implémenter plusieurs interfaces mais doit étendre une seule classe		
f. Une classe peut implémenter plusieurs classes mais doit étendre une seule interface		
g. Une classe peut implémenter plusieurs classes et peut étendre plusieurs interfaces		
h. Un attribut privé d'une classe mère A n'est pas visible par sa classe fille B		
i. Un attribut protégé d'une classe mère A est visible par sa classe fille B		
j. Une interface n'implémente aucune méthode		
k. Une variable static d'une classe est aussi appelée « variable d'instance »		
l. Une variable static d'une classe est aussi appelée « variable de classe »		

2. Considérons les classes suivantes

```
public class Vehicule {
    void start(){System.out.println("Le vehicule démarre");}
}
class Voiture extends Vehicule {
    void start(){System.out.println("La voiture démarre");}
    void start(int n){System.out.println("C'est le démarrage numero: "+n);}
}
```

Donner le résultat d'exécution de chaque bloc d'instructions au niveau du programme principal :

```
Vehicule vh1 = new Vehicule();
vh1.start();

.....

Voiture v2 = new Voiture();
v2.start();
v2.start(5);

.....

Vehicule vh3 = new Voiture();
vh3.start();

.....

vh3.start(6);

.....

((Voiture)vh3).start(6);

.....

((Voiture)vh1).start(3);

.....
```

Ne Rien Écrire Ici



3. Donner le résultat d'exécution de chacun de ces blocs d'instructions

```
class C {  
    public static int i;  
    public int j;  
    public C() {i++; j=i; }  
}
```

qu'affichera le code suivant?

```
C x=new C(); C y=new C(); C z= x;  
System.out.println(z.i + " et " + z.j);
```

3.1 Entourer la bonne réponse :

- a. 2 et 2
- b. 1 et 1
- c. 2 et 1
- d. 1 et 3

```
class B {  
    public B(){System.out.print(" Ciao");};  
    public B(int i) {this(); System.out.println(" Bonjour "+i);};  
}
```

qu'affichera l'instruction suivante?

```
B monB=new B(2003);
```

3.2 Entourer la bonne réponse :

- a. erreur de compilation
- b. Ciao
- c. CiaoBonjour 2003
- d. Bonjour 2003

Exercice 2 : (6 points)

1. Définir une classe Animal en Java avec ses éléments nécessaires (attributs, constructeur et méthodes).

.....

.....

.....

.....

.....

.....

Ne Rien Écrire Ici

✂ - - - - -

.....

.....

.....

.....

.....

.....

.....

.....

2. Définir trois classes `AnimalMer` (vit dans la mer), `AnimalTerre` et `AnimalAir` (qui vole) qui héritent de la classe `Animal` et qui ont leurs propres particularités (préciser ces particularités). Prévoir un constructeur et une fonction d’affichage pour chacune de ces 3 classes.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ne Rien Écrire Ici

✂

.....

.....

.....

.....

3. Définir une classe compilable et exécutable qui teste les classes `AnimalMer`, `AnimalTerre` et `AnimalAir`, avec des fonctions d'affichage.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Exercice 3 : (8 points)

On se propose d'écrire un programme en Java permettant de gérer des formes géométriques.

Pour cela, définir la classe abstraite **Forme** définie à travers les coordonnées de son centre `cx` et `cy` (protégés de type `int`).

1. Déclarer la classe `Forme` ; déclarer et définir au sein de la classe `Forme` :
 - a. Un constructeur paramétrique.
 - b. Un accesseur et un mutateur pour chaque attribut.
 - c. 2 méthodes abstraites `perimetre()` et `aire()`
 - d. Une méthode `deplace()` qui reçoit en argument deux entiers `dx` et `dy` et déplace le centre de `dx` et `dy` ; la méthode `deplace()` affiche les nouvelles coordonnées du centre.

.....

.....

Ne Rien Écrire Ici



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Considérons maintenant les trois classes concrètes Cercle, Carre et Rectangle qui héritent de la classe Forme.

- La classe Cercle est caractérisée par l'attribut rayon de type int.
- La classe Carre est caractérisée par l'attribut cote correspondant au côté d'un carré.
- La classe Rectangle est caractérisée par les attributs longueur et largeur de type int.
 - a. définir les classes Cercle, Carre et Rectangle. Prévoir dans chaque classe un constructeur paramétrique
 - b. définir au sein de chaque classe les fonctions perimetre() et aire() qui retournent la bonne valeur du périmètre, respectivement aire de la forme en question.
 - c. Définir au niveau des classes Carre et Rectangle une méthode dessine() permettant d'afficher la forme en question. Deux exemples d'affichage sont fournis ci-dessous.

* * * * *

* * * * *

* * * * *

* * * * *

* * * * *

* * * * *

* * * * * * *

* * * * * * *

* * * * * * *

* * * * * * *

Un carré de côté 7

Un rectangle de longueur 8 et de largeur 4

Ne Rien Écrire Ici



A series of horizontal dotted lines for writing practice, consisting of 20 lines.

Ne Rien Écrire Ici



3. Déclarer au niveau d'une classe MainClass 4 objets tels que définis dans le tableau ci-dessous.
- a. Afficher pour le cercle c1 les valeurs du perimetre et de l'aire.

Type de l'objet	Nom de l'objet	cx	cy	rayon	longueur	largeur	Cote
Cercle	c1	2	2	5	--	--	--
Rectangle	r1	0	-1	--	8	3	
Carre	ca1	5	5	--	--	--	4

.....

.....

.....

.....

.....

.....

- b. Déplacer le rectangle r1 de dx=4 et dy=5 et dessiner le carré ca1.

.....

.....

- c. Afficher les coordonnées du centre du carre ca1

.....

.....

- d. Peut-on créer un objet f1 de cette manière ? Expliquer. Si oui, comment peut-on appeler la méthode dessiner() ?

```
Forme f1= new Carre(1, 1, 4);
```

.....

.....

.....

.....