



Sujets d'examens



Cette partie regroupe un ensemble de sujets d'examen avec correction pour permettre à l'apprenant de tester ses connaissances.



EMD POO 2016-2017 _____

EMD POO 2017-2018 _____

EMD POO 2018-2019 _____



Exercice 1 : Considérons la classe Point suivante (14 Points)

```
public class Point {
    // Les coordonnées du point
    private double abs,
    private double ord;
    //constructeur par défaut initialiser les coordonnées du point à 0.0
    public Point () {
        abs = 0; ord = 0;
    }
    // Initialiser le point courant à partir des coordonnées d'un autre point
    public Point (Point p) {
        abs = p.abs; ord = p.ord;
    }
    // Initialiser le point courant avec les coordonnées passées en paramètres
    public Point (double abs, double ord) {
        this.abs = abs; this.ord = ord;
    }
    // Déplacer le point avec les valeurs données en paramètres
    public void déplacer(double dx, double dy) {
        abs += dx; ord += dy;
    }
    public void afficher() {
        System.out.println("abscisse : " + abs + " ordonnée : " + ord);
    }
    // Retourner une copie du point courant
    public Point clone() {
        Point p = new Point(); p.abs = this.abs; p.ord = this.ord; return p;
    }
}
```

A) Soit une classe Rectangle qui utilise la classe Point :

```
public class Rectangle {
    private Point pig; //le point inférieur gauche du rectangle
    private double largeur;
    private double longueur ;
    public void monIdentite() {
        System.out.println("Je suis un rectangle");
    }
}
```

Complétez les méthodes suivantes de la classe Rectangle:

1. Initialiser les coordonnées du rectangle, sa largeur et sa longueur à 0.0 (1 Point)

```
public Rectangle() {
    .....
}
```

2. Initialiser les coordonnées du rectangle avec les coordonnées du point passé en paramètre. La largeur et la longueur sont également données en paramètres. (01 Point)

```
public Rectangle (Point p, double largeur, double longueur) {
    .....
```

}

3. Les coordonnées du rectangle, sa largeur, sa longueur sont données en paramètres (01Point)

```
public Rectangle(double abs, double ord, double largeur, double longueur) {
```

.....

}

4. Déplacer un rectangle avec les données passées en paramètres. (1.5 Points)

```
public void deplacer(double dx, double dy) {
```

.....

}

5. Afficher l'identité et toutes les informations d'un rectangle (1.5 Points)

```
public void afficher() {
```

.....

}

6. Retourner une copie du rectangle courant (1 Point)

```
public Rectangle clone() {
```

.....

}

B) Considérons maintenant une classe Carre qui est un rectangle particulier.

```
public class Carre extends Rectangle {
```

}

1. Complétez la définition de la classe Carre

```
public Carre () { (01 Point)
```

.....

}

```
public Carre(Point p, double cote) { (01 Point)
```

.....

}

```
public Carre(double abs, double ord, double cote) { (01 Point)
```

.....

}

```
public void deplacer(double dx, double dy) { (01 Point)
```

.....

}

```
public void monIdentite() { (0.5 Point)
```

.....

}

```
public void afficher() { (0.5 Point)
```

.....

}

2. Quel est le résultat de l'exécution des instructions suivantes :

```
Rectangle a = new Carre(new Point(2,5),10); (01 Point)
```

```
a.afficher();
```

.....

.....

```
Carre b = new Rectangle(new Point(3,4),10,15); (01 Point)  
b.afficher();
```

.....

Exercice 2 : (06 Points)

Considérons deux interfaces Imprimable et Joignable ainsi qu'une classe nommée Humain

```
public interface Joignable {  
    public String getPhone(); //permet de retourner le numéro de téléphone  
    public String getMail(); //permet de retourner l'adresse mail  
}  
  
public interface Imprimable {  
    public void afficher(); //permet d'afficher le nom, prénom, age, nss, phone, mail  
}  
  
public class Humain {  
    private String nom, prenom;  
    private int age;  
    Humain(String nom, String prenom, int age) {  
        this.nom = nom;  
        this.prenom = prenom;  
        this.age = age;  
    }  
    public void imprime() {  
  
        System.out.println("nom : " + nom + "prenom : " + prenom + "age : " + age);  
    }  
}
```

Sachant qu'un enseignant est un humain qui possède un grade (grade) et un numéro de la sécurité sociale (nss) et qui doit être à la fois imprimable et joignable, définissez la classe Enseignant.



Exercice 1 : (2.5 pts)

Considérons la classe Compteur

<pre>class Compteur { private int compteur = 0; Compteur() { compteur += 1;} public static void main(String[] args) { Compteur c1 = new Compteur(); Compteur c2 = new Compteur(); Compteur c3 = c1; Compteur c4 = c2; System.out.println(" compteur est : " + compteur); } }</pre>	<p>Est ce qu'il y a une erreur de compilation dans ce programme ? OUI / NON.</p> <p>Si c'est oui, justifiez :</p>
--	---

Exercice 2 : (2.5 pts)

Considérons les classes suivantes :	Mettez une croix dans la case correspondant à la bonne réponse, si l'instruction proposée est insérée à l'endroit proposé		
<pre>class A { void fairececi() { }; } class B extends A { void fairececi() { }} class C { void fairececi() { }} class Principale { public static void main(String[] args) { A a = new A(); B b = new B(); /*insérer ici*/ } }</pre>		Erreur de compilation	Pas d'erreur
	A x = a;		
	A x = b;		
	B x = a;		
	B x = b;		
	A x = (A) b ;		

Exercice 3 : (3 pts)

Entourez l'unique bonne réponse.

1) class <i>Forme</i> {	
2) private <i>String</i> <i>color</i> ;	Erreur de compilation dans la ligne 9
3) public <i>Forme</i> (<i>String</i> <i>color</i>) {	
4) System.out.print ("Forme");	Erreur de compilation ligne 14
5) this.color = <i>color</i> ; }	
6) }	Pas d'erreur, le programme affiche Forme
7) class <i>Rectangle</i> extends <i>Forme</i> {	
8) public <i>Rectangle</i> () {	Pas d'erreur, le programme affiche Rectangle
9) super ("bleu");	
10) System.out.print ("Rectangle"); }	
11) }	Pas d'erreur, le programme affiche FormeRectangle
12) public class <i>Exo3</i> {	
13) public static void <i>main</i> (<i>String</i> [] <i>args</i>) {	
14) new <i>Rectangle</i> (); }	Pas d'erreur, le programme affiche RectangleForme
15) }	

Exercice 4 : (2 pts)

Sachant qu'un cheval peut se comporter comme un Animal et il peut se comporter comme un moyen de transport. Parfois il est de la classe Animal et parfois de la classe MoyenDeTransport. Comme l'héritage multiple n'est pas supporté en java, proposez un mécanisme permettant de remédier à cette lacune du langage JAVA. Expliquez votre démarche.

Exercice 5 : (10 pts)

Considérons la classe de base EtreVivant dont le comportement crie() et marche() ne peuvent pas être définis à ce niveau d'abstraction. En revanche, tous les êtres vivants mangent de la même façon.

```
abstract class EtreVivant {  
    private String nom;  
    private int age;  
    public EtreVivant (String nom, int age) {  
        this.nom = nom;  
        this.age = age;  
    }  
    abstract void crie();  
    abstract void marche();  
    void mange() {  
        System.out.println("Je mange comme tout être vivant");  
    }  
    void affiche() {
```

```
        System.out.println("(" + nom + ", " + age + ")");
    }
}
```

Considérons deux classes : les bipèdes et les quadrupèdes, tels qu'un bipède est un être vivant marchant sur deux pattes et un quadrupède est un être vivant qui marche sur 4 pattes. Les bipèdes ne crient pas tous de la même façon et même chose pour les quadrupèdes. Le comportement marche() pour un bipède consiste à afficher le message « **je marche sur deux pattes** », quant au quadrupède c'est le message « je marche sur 4 pattes »

1) Définir les deux classes Bipede et Quadrupede

Considérons maintenant deux classes Homme et Chien de sorte qu'un homme est un bipède et un chien est un quadrupède. Le comportement crie() pour un homme consiste à afficher le message : « **je crie comme un homme** » et quant au chien ça sera le message : « **je crie comme un chien** »

2) Définir les deux classes Homme et Chien

3) Que fournit le programme suivant : (2 pts)

Programme	Résultats
<pre>class Exo5 { public static void main(String[] args) { EtreVivant etrevivant = new Homme("mohand",30); etrevivant.marche(); etrevivant.crie(); etrevivant = new Chien("felix",5); etrevivant.marche(); etrevivant.crie(); } }</pre>	



EMD 2018-2019

Exercice 1 : Pour chacun des morceaux de programme suivants, donnez le résultat d'exécution s'il existe (expliquez le résultat obtenu). En cas d'erreur, justifier. (4 Pts)

<p style="text-align: center;"><i>(a) (2 pts)</i></p> <pre>class Parent { Parent() { System.out.print("A"); } } class Child extends Parent { public Child(int x) { System.out.print("B"); } public Child() { this(123); System.out.print("C"); } } class Test1{ public static void main(String[] args) { new Child(); } }</pre>	
---	--

<p style="text-align: center;"><i>(b) (2 pts)</i></p> <pre>abstract class Oeuf { protected abstract void getNomVolaille() ; public double getPrixVolaille(int x) { return x*2 ; } } class Poule extends Oeuf { void getNomVolaille() { System.out.println("Je suis une Poule "); } public static void main (String args[]) { Oeuf p=new Poule(); p.getNomVolaille(); p.getPrixVolaille(2); } }</pre>	
--	--

Exercice 2 : (2.5 pts)

Considérons les classes suivantes :	Mettez une croix dans la case correspondant à la bonne réponse, si l'instruction proposée est insérée à l'endroit proposé.
-------------------------------------	--

```

class Arme {
void attaquer() {.....}
}
class Pistolet extends Arme {
void tirer(){}
}
class Arsenal {
public void main (String [] args) {
Arme a1 = new Arme();
Arme a2 = new Pistolet();
Pistolet p = new Pistolet();
/** insérer ici ****/
}}

```

	Erreur de compilation/ exécution	Pas d'erreur	
a2.tirer()			
(Pistolet) a2.tirer()			
p.attaquer()			
(Pistolet) a1.tirer()			
a1=p ; a2.tirer() ;			

Exercice 2 : (2.5 pts)

On vous donne le morceau de programme suivant :	Dites si ce programme est correct, si oui donnez le résultat (justifier votre réponse).
<pre> class Test { static int i=0; static void f(){System.out.println("i="+i);} static void g(){ this.f();} public static void main (String args[]) { Test a = new Test(); a.g(); } } </pre>	

Exercice 3 : (3 pts)

On vous donne le morceau de programme suivant :	Prédire l'exécution (expliquez le résultat obtenu)
<pre> class A{ String s="A"; void f(String s1) { System.out.println(s+" "+s1); } } class B extends A { String s="B"; void f(String s1) { super.f(s+" "+s1); } } class C extends B { String s="C"; CC(String s) { this.s= s; } } </pre>	

<pre> } void f(String s1) { super.f(s+" "+s1); } public static void main (String args[]) { A a=new C("D"); a.f("X"); } } </pre>	
---	--

Exercice 4 : (3pts)

<p>Soit la classe Compte</p>	<p>Compléter les deux méthodes suivantes :</p> <p>virerVers : permet de transférer le montant donné du compte courant vers le compte donné en paramètre</p> <p>virerDe : permet de transférer le montant donné du compte donné vers le compte courant.</p>
<pre> class Compte { private int solde ; Compte(int solde) { this.solde = solde; } void deposer(int montant) { solde += montant; } void retirer(int montant) { solde -= montant; } } </pre>	<pre> void virerVers(int montant, Compte destination) { } void VirerDe(int montant, Compte source) { } </pre>

Exercice 5 : (5pts)

Dans le programme correct suivant, que sera affiché dans le programme principal ?

```

class A {
    public String f(B objet) { return "A et B"; }
    public String f(A objet) { return "A et A"; }
    public String f() { return "f() de A"; }
}
class B extends A {
    public String f(B objet) { return "B et B"; }
    public String f(A objet) { return "B et A"; }
    public String f() { return "f() de B"; }
}
class Test {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new B();
        B b = new B();
        System.out.println(a1.f(a1) + "," +
a1.f()); .....
        System.out.println(a1.f(a2) + "," +
a2.f()); .....
        System.out.println(a1.f(b) + "," + a1.f()); .....
        System.out.println(b.f(a1) + "," + a1.f()); .....
        System.out.println(b.f(a1) + "," + a1.f()); .....
        System.out.println(a2.f(a1) + "," + a1.f()); .....
        System.out.println(a2.f(a2) + "," + a2.f()); .....
        System.out.println(b.f(a2) + "," + a2.f()); .....
        System.out.println(a2.f(b) + "," + b.f()); .....
        System.out.println(b.f(b) + "," + b.f()); .....
    }
}

```