

## Examen : Programmation Orientée Objet JAVA

### Session 1: 2022-2023

#### Exercice 1 : (répondre sur la feuille)

Nom :

Prénom :

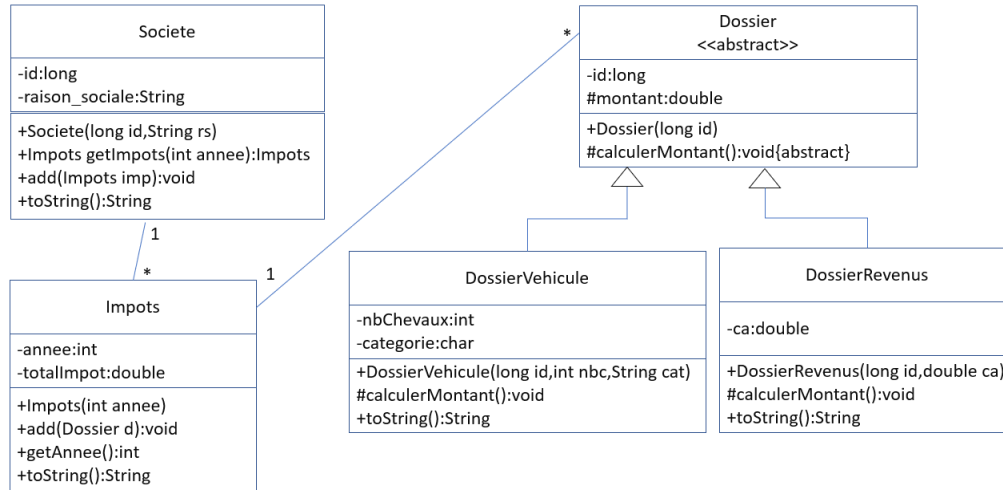
<pre> 1 package com.exam.session2.exercice1; 2 3 class Point{ 4     private int x=0; private int y=0; 5     public Point(int x,int y){ this.x=x;this.y=y;} 6     public void deplacer(int x,int y){ this.x+=x;this.y+=y;} 7     @Override 8     public String toString() {return "("+x+","+y+"");} 9 } 10 public class Question1 { 11     public static void main(String[] args) { 12         Point[] points=new Point[3]; 13         points[0].deplacer(2, 2); points[1].deplacer(3, 3); 14         for(int i=0;i&lt;points.length;i++) 15             System.out.println(points[i]); 16     } 17 } </pre>	<p><b>Si le programme ne contient pas d'erreur d'exécution</b> Résultat à afficher par le programme est :</p> <p><b>Si le programme contient des erreurs</b> Lignes des erreurs sont :</p> <p>Corrections à faire :</p> <p>Résultat à afficher après correction est :</p>
<pre> 1 import java.util.ArrayList; 2 import java.util.List; 3 abstract class Didi{ 4     public String getDetail(){return "droite -";} } 5 class Fifi extends Didi{ 6     @Override 7     public String getDetail() {return "gauche -";} } 8 class Nini extends Didi{ 9     @Override 10    public String getDetail() {return "milieu -";} } 11 public class Question2 { 12     public static void main(String[] args) { 13         List&lt;Didi&gt; list=new List(); 14         list.add(new Didi());list.add(new Fifi());list.add(new Nini()); 15         for(Didi d:list) System.out.print(d.getDetail()); 16     } } </pre>	<p><b>Réponse :</b> <b>Si le programme ne contient pas d'erreur d'exécution</b> Résultat à afficher par le programme est :</p> <p><b>Si le programme contient des erreurs</b> Lignes des erreurs sont :</p> <p>Corrections à faire :</p> <p>Résultat à afficher après correction est :</p>
<pre> 1 abstract class Fa{ 2     protected int value=5; 3     abstract void f(); 4     void g(){System.out.println("Traitement 1 de g:"+value);} 5     @Override 6     public String toString() {return "("+value+"");} } 7 class Ma extends Fa{ 8     void f(){System.out.println("Traitement de f");this.value++;} 9     @Override 10    void g(){System.out.println("Traitement 2 de g:"+value);this.value++;} 11    void h(){ 12        System.out.println("je suis un traitement de h");this.value++;} } 13 class Ta extends Ma{} 14 public class Question3 { 15     public static void main(String[] args) { 16         Fa ma=new Ma(); ma.f(); ma.g(); 17         ma.h(); 18         System.out.print(ma); 19         Fa ta=new Ta(); ta.f(); ta.g(); 20         ta.h(); 21         System.out.print(ta); } } </pre>	<p><b>Réponse :</b> <b>Si le programme ne contient pas d'erreur d'exécution</b> Résultat à afficher par le programme est :</p> <p><b>Si le programme contient des erreurs</b> Lignes des erreurs sont :</p> <p>Corrections à faire :</p> <p>Résultat à afficher après correction est :</p> <p>Résultat à afficher après correction est :</p>

**Exercice 2 :**

1. Quelles sont les différences entre une méthode surchargée (overriding) et une méthode redéfinie « overloading)
2. Quelles sont les différences entre une classe abstraite et une interface ?
3. Étant donné une classe abstraite A qui implémente une interface F, est ce que A doit redéfinir toutes les méthodes de F ? justifier la réponse.
4. Quelles sont les différences entre le mot clé this et le mot clé super ?
5. Pourquoi utiliser les interfaces ?
6. Est-ce qu'une classe abstraite peut contenir un constructeur ? si oui, à quoi peut servir ? si non, pourquoi ?
7. Quand peut-on utiliser l'héritage ?
8. Qu'est-ce que le polymorphisme ? une méthode polymorphe ? donner un exemple
9. Dans quels cas utiliser LinkedList à la place de ArrayList ? (faire une petite recherche)
10. Dans quels cas utiliser ArrayList à la place de LinkedList? (faire une petite recherche)
11. Pourquoi le concept de généricité est utilisé ?

**Exercice 3 :**

Soit le diagramme de classes suivant :



Implémenter les classes suivantes :

- Classe Societe:
  - La méthode getImpots(int annee) retourne un enregistrement Impots qui correspond à annee à partir de la liste des impôts
  - La méthode add(Impots imp) ajoute un enregistrement Impots à sa liste des impôts
  - La méthode toString() retourne l'id et la raison sociale de la société
- Classe Impots :
  - La méthode add(Dossier d) ajoute un dossier à la liste des dossiers d'un enregistrement Impots
  - La méthode toString() retourne le détail de la liste des dossiers (voir exécution)
- Classe Dossier :
  - La méthode calculerMontant() est abstraite
- Classe DossierVehicule :
  - La méthode toString retourne le détail d'un dossier sous la forme : Vehicule E 6 CH, montant:350.0
  - La méthode calculerMontant() calcule le montant selon le tableau ci-dessous
- Classe DossierRevenus :
  - La méthode toString retourne le détail d'un dossier sous la forme : 15% de revenus 1000000.0: 150000.0
  - La méthode calculerMontant() calcule le montant des impôts qui représente 15% du chiffre d'affaire ca

Le montant des impôts des véhicules dépend du nombre de chevaux et de la catégorie du véhicule selon le tableau suivant :

Catégorie/ nb Chevaux	Inférieur à 8 CH	De 8 à 10 CH	De 11 à 14 CH	Sup ou égal à 15 CH
Véhicule à moteur Essence	350 DH	650 DH	3000 DH	8000 DH
Véhicule à moteur Gasoil	700 DH	1500 DH	6000 DH	20.000 DH

**Classe Program et son exécution**

```

public class Program { //à ne pas modifier
    public static void main(String[] args) {
        Societe societe=new Societe(1, "SDCC");
        Impots imp1=new Impots(2022);
        societe.addImpots(imp1);
        imp1.add(new DossierRevenus(111, 1000000));
        imp1.add(new DossierVehicule(222, 6, 'E'));
        imp1.add(new DossierVehicule(222, 8, 'G'));
        imp1.add(new DossierVehicule(222, 11, 'E'));
        imp1.add(new DossierVehicule(222, 15, 'G'));
        System.out.println(societe);
        System.out.println(societe.getImpots(2022));
    }
}
  
```

```

Société:1 - SDCC
Liste des impôts 2022
15% de revenus 1000000.0, Montant:150000.0
Vehicule E 6 CH, montant:350.0
Vehicule G 8 CH, montant:1300.0
Vehicule E 11 CH, montant:3000.0
Vehicule G 15 CH, montant:20000.0
Total:174650.0
  
```

Enset de Mohammedia

Master- II-BDCC, Formation Continue

POO java