

Exercice 1: (10 points)

1. Choisissez la (ou les) réponse(s) correctes:(Tout est juste 0,5 ptt, une réponse juste et une fausse 0,25 ptt, une réponse juste deux fausses 0,25 ptt) **(3,5points)**

1.1 . (a),(c),(d) **0,5 point**

1.2.(c) **(0,5 point)**

1.3. (b) **(0,5 point)**

1.4. (c) **(0,5 point)**

1.5 (b) **(0,5 point)**

1.6. (a),(b) **(0,5 point)**

1.7 (d) **(0,5 point)**

2. 1. Qu'affiche le programme suivant ? : **(3,25 points)**

```
public class Exercice {  
    static private String msg = null;  
    static private int n;  
    Exercice(){  
        n = 1;  
        if (msg == null)  
            msg = "Rouge";  
    affiche();  
    }  
    private void affiche(){  
        System.out.println(n + msg);  
        if (msg!="Vert"){  
            msg = "Vert";  
            new Exercice();  
        }  
    }  
    public static void main(String[] args){  
        Exercice x = new Exercice();  
        n++;  
        x.affiche();  
        Exercice y = new Exercice();  
        n++;  
        x.affiche();  
        y.affiche();  
    }  
}
```

(0,25 ptt sur la valeur et 0,25 sur l'affichage rouge ou vert)

1Rouge **(0,5 point)**

1Vert **(0,5 point)**

2Vert **(0,5 point)**

1Vert **(0,5 point)**

2Vert **(0,5 point)**

2Vert **(0,5 point)**

2.2 Erreur de compilation **(0,25 point)**

3. Trouvez et corrigez les erreurs dans le code suivant : **(3,25 point)**

Première solution :

```
package p1;  
  
class A{  
    protected int i;  
    private int j;  
    int getJ() {(0,5 point)  
    return j;}  
    public void m(){System.out.println("Somme : "+(i+j));}
```

```
package p1;  
  
(1 point) public class B extends A{  
    public void m(){System.out.println("Produit : "+(i*getJ())); (0,5  
    point)}  
}
```

```
package p2;  
  
import p1.B;(1 point)  
  
class C extends B {  
(0,25ptt) public void m(){System.out.println("Carré : "+(i*i));}  
}  
}
```

Deuxième solution :

```
package p1;  
  
class A {  
    protected int i;  
    (1 point) protected ou public int j;  
    public void m(){System.out.println("Somme :  
    "+(i+j));}  
}
```

```
package p1;

(1 point) public class B extends A{
    public void m(){System.out.println("Produit :
    "+(i*j));} }
```

```
package p2;

import p1.B; (1 point)

class C extends B {
(0,25 ptt) public void m(){System.out.println("Carré : "+(i*i));}
}
}
```

Exercice 2 (10 points):

1. La classe sallesdecinema (1,75 point)

```
public class sallesdecinema {
    private String nomSalle; (0,25 sur l'attribut)
    private int capaciteMax;(0,25 sur l'attribut)
    private double tarifNormal;(0,25 sur l'attribut)
    private double tarifReduit;(0,25 sur l'attribut)
    private int netNormal;(0,25 sur l'attribut)
    private int netReduit;(0,25 sur l'attribut)
    public static double tarifStandard=250; (0,25 sur l'attribut)
    //sallesdecinema autresalle; (attribut déclarée pour la méthode
    accedercinema (2ème solution)
}
```

2. Les constructeurs (2,25 ptt)

```
1. (0,75 ptt)
public sallesdecinema(String nomSalle, int capaciteMax,
double tarifNormal, double tarifReduit) (0,25 sur les
paramètres){
    this.nomSalle = nomSalle; (0,5 sur les
attributs)

    this.capaciteMax = capaciteMax;
    this.tarifNormal = tarifNormal;
    this.tarifReduit = tarifReduit;
}
```

2. (0,75 ptt)

```
public sallesdecinema(sallesdecinema salle) (0,25 sur les
```

```
paramètres){
    this.capaciteMax= salle.capaciteMax; (0,5 sur les attributs)
    this.tarifNormal = salle.tarifNormal;
    this.tarifReduit=salle.tarifReduit;
}
```

3. (0,75 ptt)

```
public sallesdecinema(String nomSalle, int capaciteMax)
(0,25 sur les paramètres) {
    this.nomSalle=nomSalle; (0,5 sur les attributs)
    this.capaciteMax=capaciteMax;
    tarifNormal=tarifStandard;
    tarifReduit=tarifStandard/2;
}
```

3. Méthode nombre places disponibles (0,75 point)

```
public int NbPlacesDisponibles() {
(0,75 point) return capaciteMax - (netNormal
+ netReduit);
}
```

4. (1 point)

```
public void vendrePlaces(int nombre, boolean tarifReduit)
(0,5 sur les paramètres) {
    if (tarifReduit) {
        netReduit += nombre;
    } else {
        netNormal += nombre;(0,5 sur le reste)
    }
}
```

5. (0,5 point)

```
public void reinitialiserSalle() {
    netNormal = 0;(0,25 )
    netReduit = 0;(0,25 )
}
```

6. (1 point)

```
public double SommeVentes() {
(1 ptt) return (netNormal * tarifNormal) +
(netReduit * tarifReduit);
}
}
```

II. 1. (1 point)

```
public personne(String nom, boolean tarifReduit)
(0,5 sur les paramètres){
    this.nom = nom;   (0,5 sur les attributs)
    this.tarifReduit = tarifReduit;
}
```

4. (0,75 ptt)

Première solution :

```
public void accederSalleDeCinema(salledecinema salle) (0,25
sur les paramètres) {
    if (salle.NbPlacesDisponibles() > 0) {
        salle.vendrePlaces(1, tarifReduit);   (0,25 ptt)
        System.out.println("La personne " + nom
+ " a accédé à la salle de cinéma ");
    }
    else System.out.println("vous pouvez acceder a une autre
salle"); (0,25 ptt)
}
```

deuxième solution :

```
public void accederSalleDeCinema(salledecinema salle) (0,25
sur les paramètres) {
    if (salle.NbPlacesDisponibles() > 0) {
        salle.vendrePlaces(1, true); (0,25 ptt)
        System.out.println("La
personne  a accédé à la salle de cinéma ");}
    else
        //autresalle est déclarée dans ce cas au début de la classe
        if (autresalle != null && autresalle.NbPlacesDisponibles()>
0) {   (0,25 ptt)
            autresalle.vendrePlaces(1, false);
            System.out.println("Accès accordé à une autre salle");
        }
}
```

III. (1point)

```
public class testsalledecinema {
0,5 points sur la création des objets et 0,5 points sur l'appel des
méthodes
    public static void main(String[] args) {
```

```
salledecinema salle1 = new
salledecinema("Salle 1", 100, 10.0, 8.0);
salledecinema salle2 = new
salledecinema("Salle 2", 150);
personne personne1 = new
personne("personne1", false);
personne1.accederSalleDeCinema(salle1);
personne1.accederSalleDeCinema(salle2);
salle1.vendrePlaces(5, true);
salle2.vendrePlaces(10, false);
System.out.println("Somme des ventes pour la salle 1 : " +
salle1.SommeVentes());
salle1.reinitialiserSalle();
System.out.println("Somme des ventes pour la salle 2 : " +
salle2.NbPlacesDisponibles()); }
}
```