



# DEVOIR SURVEILLE

Filière : L2 Informatique

Nombre de pages : 2

Matière : Programmation Java

Durée : 1 heure

Enseignant : Akil ELKAMEL

Date : 17/11/2025

## Documents non autorisés

### Exercice 1 : (8 pts)

Question 1. (4 pts) Expliquer les termes suivants :

- Encapsulation
- Instanciation

Question 2. (2 pts) Indiquer si l'affirmation est correcte ou non.

(Sur la feuille de réponse il suffit de mentionner le numéro de l'affirmation)

Vrai      Faux

1. Un champ est marqué **static** lorsque sa valeur ne doit pas changer au cours de l'exécution du programme.
2. Lors de la déclaration d'une méthode, le mot clé **final** indique que la méthode ne doit pas être redéfinie dans les sous-classes.
3. Une classe dont tous les champs sont déclarés **private** ne peut pas être déclarée **public**.
4. La classe **Object** est une sous-classe de toutes les classes que l'on peut définir.

Question 3. (2 pts) Considérer les classes dans le programme suivant et indiquer s'il compile ou non ?

- S'il compile, déterminer l'affichage produit lors de l'exécution.
- S'il ne compile pas, préciser clairement la cause.

```
class Shape {  
    private String color;  
    public Shape(String color) {  
        System.out.println("Shape");  
        this.color = color;  
    }  
  
    class Rectangle extends Shape {  
        public Rectangle() {  
            System.out.println("Rectangle");  
        }  
    }  
  
    public class Question2 {  
        public static void main(String[] args) {  
            Rectangle r1 = new Rectangle();  
        }  
    }  
}
```

---

## Exercice 2 : (12 pts)

- ✓ . Ecrire la classe **SalleCinema** correspondante aux informations suivantes :
- **id** : un entier qui représente un identifiant unique de la salle.
  - **titreFilm** : une chaîne de caractères qui contient le titre du film joué.
  - **maxPlaces** : un entier qui représente le nombre maximum de places dans la salle.
  - **prixTicket** : un réel (double) qui représente le prix unitaire d'un ticket.
  - **nbrPlacesVendues** : un entier qui contient le nombre de places qui ont été vendues.
- ✓ . Les valeurs des quatre premières caractéristiques (**id**, **titreFilm**, **maxPlaces** et **prixTicket**) sont fixées lors de la création d'un nouvel objet de type **SalleCinema**, et l'attribut **nbrPlacesVendues** est initialisé à zéro.
- ✓ . Ecrire les méthodes d'accès aux attributs pour **titreFilm** et **prixTicket**. Pour les autres attributs supposons que nous avons défini les méthodes d'accès.
- La classe **SalleCinema** doit posséder les méthodes suivantes :
- **public int nbPlacesDisponibles()** : qui calcule et renvoie le nombre de places encore disponibles dans la salle.
  - **public void vendreUnePlace()** : qui permet de vendre un ticket.
  - **public void vendreDesPlaces(int nbre)** : qui permet de vendre des tickets. **nbre** indique le nombre de places demandées.
- Remarque : pour les deux méthodes de vente de tickets, si le nombre de places demandées est supérieur au nombre de places disponibles la vente n'est pas effectuée et la méthode affiche un message indiquant que la vente n'est pas possible. Sinon, le nombre de places vendues est mis à jour et le montant à payer est affiché.
- **public double chiffreAffaires()** : qui calcule et retourne le chiffre d'affaires produit par la salle pour la séance en cours (total des ventes depuis la création de l'objet salle).
- Créer une classe **TestSalleCinema** qui représente notre programme principal dans lequel on va créer une salle de cinéma avec un **id = 1**, le film en cours est « *The Amazing Spider-Man* », le nombre maximum de places est initialisé à 60 et le prix d'un ticket est 5.9 DT.
- Vendre 20 places dans la salle 1 en une seule fois.
- Vendre une seule place dans la salle 1.
- Afficher le chiffre d'affaires de la salle 1.
- Vendre 30 places dans la salle 1 en une seule fois.
- Afficher le chiffre d'affaires de la salle 1.
- Vendre 15 places dans la salle 1 en une seule fois.

Bon travail.