

## Les Classes et Objets en Java

### Qu'est-ce qu'une classe ?

Une classe est comme un plan de construction (ou une recette) pour créer des objets. Elle décrit ce qu'un objet sait (ses attributs) et ce qu'il sait faire (ses méthodes). Chaque objet créé à partir d'une classe est une instance de cette classe.

 Analogie : Une classe, c'est la fiche technique d'un objet. Par exemple, la classe Voiture décrit toutes les voitures (marque, couleur, vitesse...), mais chaque voiture réelle est un objet de cette classe.

### Exemple simple

```
public class Voiture {  
    // Attributs = les caractéristiques de la voiture  
    String marque;  
    int vitesse;  
  
    // Méthode = ce que la voiture sait faire  
    public void accelerer() {  
        vitesse += 10;  
        System.out.println("La voiture accélère à " + vitesse + " km/h");  
    }  
}
```

### Créer et utiliser un objet

```
public class Main {  
    public static void main(String[] args) {  
        Voiture maVoiture = new Voiture(); // création d'un objet  
        maVoiture.marque = "Toyota";  
        maVoiture.vitesse = 0;  
  
        maVoiture.accelerer();  
    }  
}
```

## Les attributs

Les attributs représentent les données propres à chaque objet. Chaque objet a ses propres valeurs pour ces attributs.

```
public class Personne {  
    String nom;  
    int age;  
}
```

## Les méthodes d'instance

```
public class Personne {  
    String nom;  
    int age;  
  
    public void sePresenter() {  
        System.out.println("Je m'appelle " + nom + " et j'ai " + age + " ans.");  
    }  
}
```

## Le constructeur

Un constructeur est une méthode spéciale appelée automatiquement quand on crée un objet. Il sert à initialiser les attributs.

```
public class Personne {  
    String nom;  
    int age;  
  
    // Constructeur  
    public Personne(String nom, int age) {  
        this.nom = nom;  
        this.age = age;  
    }  
}
```

## Le mot-clé this

Le mot-clé this fait référence à l'objet courant (celui sur lequel on agit). Il est utilisé pour différencier les attributs de la classe des paramètres de la méthode ou du constructeur.

```
public class Personne {  
    String nom;  
    int age;  
  
    public Personne(String nom, int age) {  
        this.nom = nom; // this.nom = attribut de l'objet, nom = paramètre  
        this.age = age;  
    }  
}
```

## Les modificateurs d'accès

Les modificateurs d'accès contrôlent la visibilité des attributs et méthodes d'une classe.  
Voici les plus courants :

Modificateur	Description
public	Accessible depuis n'importe quelle autre classe.
private	Accessible uniquement à l'intérieur de la même classe.
protected	Accessible dans la même classe et par les classes filles (héritage).

 En général, on déclare les attributs en private et on y accède via des méthodes publiques appelées getters et setters (vu plus tard).

## Exemple complet

```
public class Etudiant {  
    private String nom;  
    private double moyenne;  
  
    public Etudiant(String nom, double moyenne) {  
        this.nom = nom;  
        this.moyenne = moyenne;  
    }  
  
    public void afficherInfos() {  
        System.out.println("Étudiant : " + nom);  
        System.out.println("Moyenne : " + moyenne);  
        if (moyenne >= 10) {  
            System.out.println("Statut : Admis");  
        } else {  
            System.out.println("Statut : Recalé");  
        }  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Etudiant e1 = new Etudiant("Marie", 16.5);  
        e1.afficherInfos();  
    }  
}
```

## Bonnes pratiques

- Nomme ta classe avec une majuscule (Personne, Animal)
- Nomme tes attributs en minuscule (nom, age)
- Mets une seule responsabilité par classe
- Utilise des constructeurs pour initialiser proprement les objets
- Utilise les modificateurs d'accès (souvent private) pour protéger tes données