

# Guide Complet : POO et Java pour Vrais Débutants

## Table des Matières

1. [Comprendre les Objets](#)
2. [Les Classes : Le Moule à Gâteaux](#)
3. [Programmer en Java](#)
4. [Exercices Corrigés](#)

## 1. COMPRENDRE LES OBJETS

### C'est quoi un objet ?

Imagine que tu regardes autour de toi. Tu vois des **choses** : un téléphone, une voiture, une personne, un stylo...

En programmation, on appelle ces "choses" des **OBJETS**.

 **Définition simple :** Un objet, c'est quelque chose qui a :

- Des **caractéristiques** (comment il est)
- Des **capacités** (ce qu'il peut faire)

### Les ATTRIBUTS (Caractéristiques)

Les attributs, ce sont les **informations** qui décrivent l'objet.

#### Exemple concret : Une Voiture

Caractéristique	Valeur
Couleur	Rouge
Marque	Toyota
Modèle	Corolla
Nombre de portes	4

En langage informatique, on écrit :



Voiture → couleur = "Rouge"

Voiture → marque = "Toyota"

Voiture → modèle = "Corolla"

 **Définition :** Un **attribut d'instance**, c'est une variable qui stocke une information propre à UN objet spécifique.

### Mémo pour retenir

**Attribut = Adjectif** → Les attributs **décrivent** l'objet

Pense à une carte d'identité : nom, prénom, âge, taille... Ce sont des ATTRIBUTS !

---

## ⚡ Les MÉTHODES (Actions/Comportements)

Les méthodes, ce sont les **actions** que l'objet peut faire.

### 🚗 Exemple : Voiture

- démarrer()
- accélérer()
- freiner()
- klaxonner()

### 👤 Exemple : Personne

- marcher()
- parler()
- manger()
- sePresenter()

📘 Définition : Une **méthode d'instance**, c'est une action ou un service qu'un objet peut réaliser.

### 🎯 Mémo pour retenir

**Méthode = Verbe** → Les méthodes sont des **ACTIONS**

Pense à ce que TU peux faire : marcher, parler, manger... Ce sont des **MÉTHODES** !

---

## ✉ MESSAGE vs MÉTHODE

**Situation :** Tu veux que ta voiture démarre.

- La **MÉTHODE** : C'est la fonction `démarrer()` programmée dans la voiture
- Le **MESSAGE** : C'est le fait de **demander** à la voiture de démarrer

### 📘 Définition :

- **Message** = La demande que tu envoies à un objet
- **Méthode** = L'action qui s'exécute quand l'objet reçoit le message

### 💡 Analogie

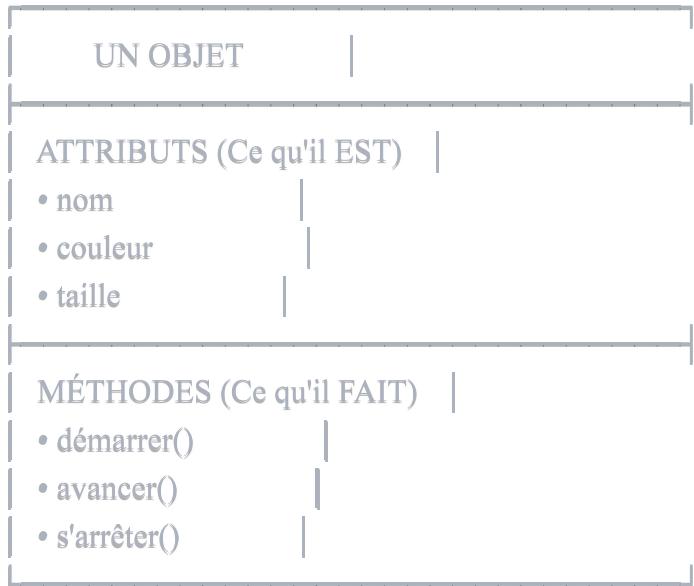


TOI : "Alexa, joue de la musique !" → MESSAGE

ALEXA : [exécute la fonction `jouerMusique()`] → MÉTHODE

---

## RÉSUMÉ : Un Objet



## 2. LES CLASSES : Le Moule à Gâteaux

### C'est quoi une classe ?

Imagine une **usine de voitures** :

- Il y a un **plan/modèle** unique pour fabriquer des Toyota Corolla
- Avec ce plan, on fabrique **plein de voitures** différentes (rouge, bleue, verte...)

En programmation :

- Le **PLAN** = La **CLASSE**
- Les **VOITURES fabriquées** = Les **OBJETS** (ou **INSTANCES**)

 **Définition** : Une **classe**, c'est un modèle/plan qui définit :

- Quels attributs auront les objets
- Quelles méthodes auront les objets

### Classe vs Objets (Instances)

**Exemple : Classe Voiture**



## CLASSE Voiture (le plan)

- └ Attributs : marque, modèle, couleur
- └ Méthodes : demarrer(), accelerer()

## OBJETS créés à partir de cette classe :

- └ voiture1 : Toyota, Rav4, Rouge
- └ voiture2 : Nissan, Kicks, Verte
- └ voiture3 : Mazda, CX-9, Bleue

📘 **Définition :** Une **instance**, c'est un objet créé à partir d'une classe.

🎯 **Analogie pour retenir**



Moule à gâteau = CLASSE

Gâteaux fabriqués = INSTANCES (objets)

Avec UN SEUL moule, tu fais plein de gâteaux différents !

---

## VS **Attribut d'INSTANCE vs Attribut de CLASSE**

📋 **Attribut d'INSTANCE**

Chaque objet a SA PROPRE valeur.



voiture1.couleur = "Rouge" ← Propre à voiture1

voiture2.couleur = "Bleue" ← Propre à voiture2

🌐 **Attribut de CLASSE**

**Partagé** par TOUS les objets de la classe.



Voiture.nombreVoitures = 3 ← Commun à TOUTES les voitures

📘 **Définition :**

- **Attribut d'instance** : Chaque objet a sa propre copie
- **Attribut de classe** : Une seule copie partagée par tous les objets

### 💡 Exemple concret



Classe : Etudiant

- └ Attribut d'instance : nom (chaque étudiant a SON nom)
- └ Attribut de classe : nombreEtudiants (total pour TOUTE la classe)

## 3. PROGRAMMER EN JAVA ☕

### ✍ Structure d'une Classe Java



java

```
public class Etudiant {  
    // 1 ATTRIBUTS (variables)  
    private String nom;  
    private String prenom;  
    private int age;  
  
    // 2 CONSTRUCTEUR (pour créer l'objet)  
    public Etudiant(String nom, String prenom, int age) {  
        this.nom = nom;  
        this.prenom = prenom;  
        this.age = age;  
    }  
  
    // 3 MÉTHODES (actions)  
    public void sePresenter() {  
        System.out.println("Je suis " + this.nom + " " + this.prenom);  
        System.out.println("J'ai " + this.age + " ans.");  
    }  
}
```

## 🔑 Les Mots-Clés Importants

Mot-clé	Signification	Exemple
public	Accessible partout	public class Etudiant
private	Accessible seulement dans la classe	private String nom
class	Définit une classe	class Voiture
new	Crée un nouvel objet	new Etudiant(...)
this	Fait référence à l'objet actuel	this.nom
void	La méthode ne retourne rien	void sePresenter()

## 🏗 Le CONSTRUCTEUR

C'est une méthode spéciale qui **crée** l'objet.



// CONSTRUCTEUR

```
public Etudiant(String nom, String prenom, int age) {  
    this.nom = nom;      // ← "this" = cet objet  
    this.prenom = prenom;  
    this.age = age;  
}
```

📘 Définition : Le **constructeur** initialise les attributs quand on crée un objet avec new.

💡 Pourquoi "this" ?



```
public Etudiant(String nom, String prenom, int age) {  
    this.nom = nom;  
    // ↑ ↑  
    // | |  
    // attribut paramètre  
}
```

this.nom = l'attribut de l'objet

nom = le paramètre reçu

## 🎮 Créer et Utiliser des Objets



java

```
public class Main {  
    public static void main(String[] args) {  
        // 1 CRÉER un objet  
        Etudiant etudiant1 = new Etudiant("AMAH", "Kwatcha", 18);  
  
        // 2 UTILISER l'objet (envoyer un message)  
        etudiant1.sePresenter();  
  
        // Résultat affiché :  
        // Je suis AMAH Kwatcha  
        // J'ai 18 ans.  
    }  
}
```

## 💡 Getters et Setters

Ce sont des méthodes pour **lire** et **modifier** les attributs privés.



java

```
// GETTER (lire)  
public String getNom() {  
    return nom;  
}  
  
// SETTER (modifiez)  
public void setNom(String nom) {  
    this.nom = nom;  
}
```

**Pourquoi** ? Les attributs sont **private** (protégés), on passe par des méthodes publiques.

## 💡 Utilisation



java

// Lire

```
String monNom = etudiant1.getNom();
```

// Modifier

```
etudiant1.setAge(22);
```

---

## 4. EXERCICES CORRIGÉS 💪

### 📝 Exercice 1 : Identifier les attributs et méthodes

Objets à analyser :

1. Un téléphone
2. Un livre
3. Un chien

#### 📋 CORRECTION :

##### 📱 Téléphone

Attributs :

- marque (String)
- modèle (String)
- couleur (String)
- batterie (int) ← en %

Méthodes :

- allumer()
- éteindre()
- appeler(String numéro)
- envoyerSMS(String message)

##### 📖 Livre

Attributs :

- titre (String)
- auteur (String)
- nombrePages (int)
- isbn (String)

Méthodes :

- ouvrir()
- fermer()
- lirePage(int numéro)
- afficherInfos()

##### 🐕 Chien

## Attributs :

- nom (String)
- race (String)
- age (int)
- couleur (String)

## Méthodes :

- aboyer()
- courir()
- manger()
- sePresenter()

---

## 📝 Exercice 2 : Compléter le tableau

**Consigne :** Classe Personne avec 3 instances

Nom	Prénom	Age	Ville
KOFFI	Ama	25	Lomé
MENSAH	Koku	30	Accra
ADDO	Edem	22	Cotonou

## 📋 CORRECTION :



java

```

public class Personne {
    private String nom;
    private String prenom;
    private int age;
    private String ville;

    public Personne(String nom, String prenom, int age, String ville) {
        this.nom = nom;
        this.prenom = prenom;
        this.age = age;
        this.ville = ville;
    }

    public void sePresenter() {
        System.out.println("Je m'appelle " + prenom + " " + nom);
        System.out.println("J'ai " + age + " ans et j'habite à " + ville);
    }
}

// Dans le main :
Personne p1 = new Personne("KOFFI", "Ama", 25, "Lomé");
Personne p2 = new Personne("MENSAH", "Koku", 30, "Accra");
Personne p3 = new Personne("ADDO", "Edem", 22, "Cotonou");

p1.sePresenter();
p2.sePresenter();
p3.sePresenter();

```

---

## Exercice 3 : Créer la classe Enseignant

**Consigne :** Crée une classe Enseignant avec :

- Attributs : nom, prenom, matière, expérience (en années)
- Méthode : sePresenter() qui affiche "Je suis [nom] [prenom], j'enseigne [matière] depuis [expérience] ans."

## CORRECTION :



java

```
public class Enseignant {  
    // Attributs  
    private String nom;  
    private String prenom;  
    private String matiere;  
    private int experience;  
  
    // Constructeur  
    public Enseignant(String nom, String prenom, String matiere, int experience) {  
        this.nom = nom;  
        this.prenom = prenom;  
        this.matiere = matiere;  
        this.experience = experience;  
    }  
  
    // Méthode de présentation  
    public void sePresenter() {  
        System.out.println("Je suis " + nom + " " + prenom);  
        System.out.println("J'enseigne " + matiere + " depuis " + experience + " ans.");  
    }  
  
    // Getters  
    public String getNom() {  
        return nom;  
    }  
  
    public String getPrenom() {  
        return prenom;  
    }  
  
    public String getMatiere() {  
        return matiere;  
    }  
  
    public int getExperience() {  
        return experience;  
    }  
  
    // Setters  
    public void setNom(String nom) {  
        this.nom = nom;  
    }  
  
    public void setPrenom(String prenom) {
```

```

this.prenom = prenom;
}

public void setMatiere(String matiere) {
    this.matiere = matiere;
}

public void setExperience(int experience) {
    this.experience = experience;
}
}

// ===== UTILISATION =====
public class Main {
    public static void main(String[] args) {
        // Créer 3 enseignants
        Enseignant prof1 = new Enseignant("HOETOWOU", "Yaovi", "POO", 5);
        Enseignant prof2 = new Enseignant("ADAME", "Koffi", "Mathématiques", 10);
        Enseignant prof3 = new Enseignant("AKAKPO", "Sena", "Physique", 7);

        // Les faire se présenter
        prof1.sePresenter();
        System.out.println(); // Ligne vide

        prof2.sePresenter();
        System.out.println();

        prof3.sePresenter();
    }
}

```

 Résultat affiché :



Je suis HOETOWOU Yaovi  
J'enseigne POO depuis 5 ans.

Je suis ADAME Koffi  
J'enseigne Mathématiques depuis 10 ans.

Je suis AKAKPO Sena  
J'enseigne Physique depuis 7 ans.

### Exercice 4 : Modifier un attribut

**Consigne :** Modifie l'âge de `etudiant1` de 18 à 22, puis fais-le se présenter à nouveau.

#### CORRECTION :



```
public class Main {  
    public static void main(String[] args) {  
        // Créer un étudiant  
        Etudiant etudiant1 = new Etudiant("AMAH", "Kwatcha", 18);  
  
        // Première présentation  
        System.out.println("== AVANT ==");  
        etudiant1.sePresenter();  
  
        //Modifier l'âge  
        etudiant1.setAge(22);  
  
        // Deuxième présentation  
        System.out.println("\n== APRÈS ==");  
        etudiant1.sePresenter();  
    }  
}
```

#### Résultat :



==== AVANT ====

Je suis AMAH Kwatcha

J'ai 18 ans.

==== APRÈS ====

Je suis AMAH Kwatcha

J'ai 22 ans.

---

### Exercice 5 : Attribut de classe (DIFFICILE)

**Consigne :** Ajoute un attribut `nombreEtudiants` (attribut de classe) qui compte le nombre total d'étudiants créés.

#### CORRECTION :



java

```

public class Etudiant {
    // Attributs d'instance
    private String nom;
    private String prenom;
    private int age;

    // Attribut de classe (static)
    private static int nombreEtudiants = 0;

    // Constructeur
    public Etudiant(String nom, String prenom, int age) {
        this.nom = nom;
        this.prenom = prenom;
        this.age = age;
        nombreEtudiants++; // Incrémente le compteur
    }

    // Méthode de classe pour obtenir le nombre d'étudiants
    public static int getNombreEtudiants() {
        return nombreEtudiants;
    }

    public void sePresenter() {
        System.out.println("Je suis " + nom + " " + prenom);
        System.out.println("J'ai " + age + " ans.");
    }
}

// ===== UTILISATION =====
public class Main {
    public static void main(String[] args) {
        System.out.println("Nombre d'étudiants : " + Etudiant.getNombreEtudiants());
        // Affiche : 0

        Etudiant e1 = new Etudiant("AMAH", "Kwatcha", 18);
        System.out.println("Nombre d'étudiants : " + Etudiant.getNombreEtudiants());
        // Affiche : 1

        Etudiant e2 = new Etudiant("ABALO", "Afi", 21);
        Etudiant e3 = new Etudiant("AKAKPO", "Tcha-Esso", 21);

        System.out.println("Nombre total d'étudiants : " + Etudiant.getNombreEtudiants());
        // Affiche : 3
    }
}

```

}

### 💡 Explication :

- static → L'attribut appartient à la CLASSE, pas aux objets
- nombreEtudiants++ → À chaque création d'objet, on ajoute 1
- Etudiant.getNombreEtudiants() → On appelle avec le NOM de la classe, pas un objet

## 🎯 MÉMO FINAL : Les Points Clés

### 📌 À retenir absolument



OBJET = ATTRIBUTS + MÉTHODES

↓      ↓      ↓

Chose   Ce qu'il EST   Ce qu'il FAIT

### 📌 En Java



java

```
// CRÉER une classe
public class NomClasse {
    private Type attribut; // private = protégé

    public NomClasse(params) {} // Constructeur

    public void methode() {} // Méthode
}
```

```
// UTILISER la classe
NomClasse objet = new NomClasse(valeurs);
objet.methode();
```

## Le vocabulaire

Français	Java	Exemple
Attribut	variable	private String nom;
Méthode	fonction	public void sePresenter()
Instance	objet	new Etudiant(...)
Créer un objet instancier	Etudiant e1 = new Etudiant(...)	

## BRAVO !

Tu maîtrises maintenant :  Ce qu'est un objet, un attribut, une méthode

-  La différence entre classe et instance
-  Comment créer une classe en Java
-  Comment créer et utiliser des objets
-  Les getters et setters
-  La différence entre attribut d'instance et de classe

 Continue à pratiquer ! Plus tu codes, plus ça devient facile.