



INSTITUT SUPERIEUR POLYTECHNIQUE DE MADAGASCAR  
TP JAVA n°5 – Initiation à la POO : Classe Personne

```
1 package personne;
2
3 public class Personne {
4     private String nom;
5     private int age;
6
7     // Constructeur
8     public Personne(String nom, int age) {
9         this.nom = nom;
10        this.age = age;
11    }
12
13    // Méthodes d'accès (getters)
14    public String getNom() {
15        return nom;
16    }
17
18    public int getAge() {
19        return age;
20    }
21
22    // Méthode utile 1 : Vérifier si la personne est majeure
23    public boolean estMajeure() {
24        return age >= 18;
25    }
26
27    // Méthode utile 2 : Incrémenter l'âge (simuler un anniversaire)
28    public void anniversaire() {
29        age++;
30    }
31
32    // Représentation sous forme de chaîne de caractères
33    @Override
34    public String toString() {
35        return "Personne{" +
36            "nom='" + nom + '\'' +
37            ", age=" + age +
38            '}';
39    }
40
41    // Méthode main pour tester le comportement de la classe
42    public static void main(String[] args) {
43        // Création de deux objets Personne avec le même nom "Alice"
44        Personne p1 = new Personne("Alice", 20);
45        Personne p2 = new Personne(new String("Alice"), 20);
46        Personne p3 = p1; // p3 référence le même objet que p1
47
48        // Affichage des objets
49        System.out.println("p1 : " + p1);
50        System.out.println("p2 : " + p2);
51        System.out.println("p3 : " + p3);
52
53        // Comparaison par référence des objets
54        System.out.println("\nComparaison des objets avec '==' :");
55        System.out.println("p1 == p2 : " + (p1 == p2));
56        System.out.println("p1 == p3 : " + (p1 == p3));
57    }
```

```

58      // Comparaison des attributs nom
59      System.out.println("\nComparaison des noms :");
60      System.out.println("p1.getNom() == p2.getNom() : " + (p1.getNom() == p2.getNom()));
61      System.out.println("p1.getNom().equals(p2.getNom()) : " +
    p1.getNom().equals(p2.getNom()));
62
63     }
64 }

```

- 1) Copier le code ci-dessus et exécuter.
- 2) Observer et analyser les résultats. Selon vous, quelle est la différence entre `equals()` et `==` ?
- 3) A la ligne 62, évaluer et afficher `p1.equals(p2)` ensuite `p1.equals(p3)`. Que se passe-t-il ? Expliquer.
- 4) Supprimer la méthode `toString()`, quels sont les changements ? Expliquer.
- 5) Ecrire trois autres méthodes pertinentes.