



INSTITUT SUPERIEUR POLYTECHNIQUE DE MADAGASCAR
TP JAVA n°6 – POO : Initiation à l'héritage

Partie A – Redéfinition de la méthode equals

```
1  package personne;
2  public class Personne {
3      private String nom;
4      private int age;
5      // Constructeur
6      public Personne(String nom, int age) {
7          this.nom = nom;
8          this.age = age;
9      }
10     // Méthodes d'accès (getters)
11     public String getNom() {
12         return nom;
13     }
14     public int getAge() {
15         return age;
16     }
17     // Méthode utile : Vérifier si la personne est majeure
18     public boolean estMajeure() {
19         return age >= 18;
20     }
21     // Méthode utile : Incrémenter l'âge (simuler un anniversaire)
22     public void anniversaire() {
23         age++;
24     }
25     // Redéfinition de equals pour comparer le contenu des objets
26     @Override
27     public boolean equals(Object obj) {
28         if (this == obj)
29             return true;
30         if (obj == null || getClass() != obj.getClass())
31             return false;
32         Personne autre = (Personne) obj;
33         return age == autre.age && (nom != null ? nom.equals(autre.nom) : autre.nom == null);
34     }
35     // Redéfinition de toString pour un affichage lisible
36     @Override
37     public String toString() {
38         return nom+ ", "+ age + " ans.";
39     }
40     public static void main(String[] args) {
41         Personne p1 = new Personne("Alice", 20);
42         Personne p2 = new Personne("Alice", 20);
43         System.out.println("p1.equals(p2) : " + p1.equals(p2));
44         System.out.println("p1 : " + p1);
45     }
46 }
```

- 1) Copier le code ci-dessus et exécuter.
- 2) Enlever *temporairement* la méthode boolean equals(Object obj) (28...36). Quel est l'impact de ce changement ?
- 3) Expliquer, selon vos connaissances, la ligne 32.
- 4) Remettre la méthode boolean equals(Object obj) (28...36). Enlever juste les lignes 29...32. Exécuter. Quels sont les changements ? Est-ce que ces lignes sont utiles ? Expliquer.

Partie B – Premier Héritage

```
1 package personne;
2 public class Etudiant extends Personne {
3     // Attribut représentant la classe de l'étudiant (par exemple "L2 Informatique")
4     private String classe;
5     // Constructeur
6     public Etudiant(String nom, int age, String classe) {
7         super(nom, age);
8         this.classe = classe;
9     }
10    // Redéfinition de equals en tenant compte des attributs hérités et de la classe de l'étudiant
11    @Override
12    public boolean equals(Object obj) {
13        // Vérifier l'égalité des attributs hérités
14        if (!super.equals(obj))
15            return false;
16        // Vérifier si les deux références sont identiques
17        if (this == obj)
18            return true;
19        // Vérifier que l'objet n'est pas nul et de la même classe
20        if (obj == null || getClass() != obj.getClass())
21            return false;
22        Etudiant autre = (Etudiant) obj;
23        return classe != null ? classe.equals(autre.classe) : autre.classe == null;
24    }
25
26    public static void main(String[] args) {
27        Etudiant e1 = new Etudiant("Bob", 22, "L2 Informatique");
28        Etudiant e2 = new Etudiant("Bob", 22, "L2 Informatique");
29        Etudiant e3 = new Etudiant("Bob", 22, "M1 Informatique");
30
31        System.out.println("e1.equals(e2) : " + e1.equals(e2)); // Devrait renvoyer true
32        System.out.println("e1.equals(e3) : " + e1.equals(e3)); // Devrait renvoyer false
33        System.out.println("e1 : " + e1);
34    }
35 }
```

- 5) Ecrire la méthode `public String getClasse()` (getter), qui permet d'accéder à l'attribut `classe` de l'extérieur.
- 6) Expliquer le constructeur dans la classe `Etudiant`.
- 7) Enlever *temporairement* la méthode `boolean equals(Object obj)` (11...24). Quel est l'impact de ce changement ?
- 8) Redéfinir la méthode `toString()`. Quel est l'impact de cette redéfinition ?