



INSTITUT SUPERIEUR POLYTECHNIQUE DE MADAGASCAR

TP JAVA n°7 – POO et SD : Liste chaînée et Polymorphisme

1) Implémentez les codes sources suivantes :

Personne.java

```
1 package personne;
2 public class Personne {
3     protected String nom;
4     protected int age;
5     // Constructeur
6     public Personne(String nom, int age) {
7         this.nom = nom;
8         this.age = age;
9     }
10    // Redéfinition de equals pour comparer le contenu des objets
11    @Override
12    public boolean equals(Object obj) {
13        if (this == obj)
14            return true;
15        if (obj == null || getClass() != obj.getClass())
16            return false;
17        Personne autre = (Personne) obj;
18        return age == autre.age && (nom != null ? nom.equals(autre.nom) : autre.nom == null);
19    }
20    // Redéfinition de toString pour un affichage lisible
21    @Override
22    public String toString() {
23        return nom+ ", "+ age + " ans.";
24    }
25 }
```

Etudiant.java

```
1 package personne;
2 public class Etudiant extends Personne {
3     private String classe;
4     public Etudiant(String nom, int age, String classe) {
5         super(nom, age);
6         this.classe = classe;
7     }
8    // Redéfinition de equals en tenant compte des attributs hérités et de la classe de l'étudiant
9    @Override
10    public boolean equals(Object obj) {
11        if (!super.equals(obj))
12            return false;
13        if (this == obj)
14            return true;
15        if (obj == null || getClass() != obj.getClass())
16            return false;
17        Etudiant autre = (Etudiant) obj;
18        return classe != null ? classe.equals(autre.classe) : autre.classe == null;
19    }
20    @Override
21    public String toString() {
22        return nom+ ", "+ age + " ans. Etudiant de "+classe;
23    }
24 }
```

Noeud.java

```
1 package personne;
2 public class Noeud {
3     private Personne donnee;
4     private Noeud suivant;
5     public Noeud(Personne donnee, Noeud suivant) {
6         this.donnee = donnee;
7         this.suivant = suivant;
8     }
9     public Personne getDonnee() {
10         return donnee;
11     }
12     public Noeud getSuivant() {
13         return suivant;
14     }
15 }
```

ListeChaine.java

```
1 package personne;
2 public class ListeChaine {
3     private Noeud tete;
4     public ListeChaine() {
5         this.tete = null;
6     }
7     public void empiler(Personne pers) {
8         tete = new Noeud(pers, tete);
9     }
10    public Personne depiler() {
11        if (tete == null) return null;
12        Personne pers = tete.getDonnee();
13        tete = tete.getSuivant();
14        return pers;
15    }
16 }
```

TestListeChaine.java

```
1 package personne;
2 public class TestListeChaine {
3     public static void main(String[] args) {
4         ListeChaine pile = new ListeChaine();
5         Personne p1 = new Personne("Alice", 30);
6         Personne p2 = new Personne("Bob", 25);
7         Etudiant e1 = new Etudiant("Charlie", 20, "L2 Informatique");
8         pile.empiler(p1);
9         pile.empiler(p2);
10        pile.empiler(e1);
11        System.out.println("\nDépilement des éléments :");
12        Personne depile;
13        while ((depile = pile.depiler()) != null) {
14            System.out.println("Élément dépilé : " + depile);
15        }
16    }
17 }
```

- 2) Exécutez le fichier TestListeChaine.java. Quel est l'affichage à l'écran ?
- 3) Dans la classe Personne, mettez les attributs nom et age en private. Que se passe-t-il ? Expliquer.
- 4) Dans la classe Etudiant, enlevez la méthode la redéfinition de toString() (20...23).
Que se passe-t-il ? Expliquer.
- 5) Dans quelle(s) partie(s) de ce travail avez-vous expérimenté le polymorphisme ?