Etablissement : ISET-Charguia	**Département :** Technologies de l'Informatique
Matière : Atelier POO Avancée	**Année :** 2ème année
Année Universitaire : 2023- 2024	

TP n° 2 : Les collections sous JAVA

Objectifs du TP :
- Comprendre la généricité sous Java
- Utiliser un premier type de collection : ArrayList

Partie 1 : Généricité

Exercice 1 :

Soit la classe **Paire** définie comme suit :

```
public class Paire {

	private int x, y;

	public Paire(int premier, int second) {
		this.x = premier;
		this.y = second;
	}
	public void afficher()
	{
		System.out.println("Premier = " + x + " Second = "+ y);
	}
}
```

1) Récrire une classe **Paire** générique où les attributs *x* et *y* peuvent être de n'importe quel type (le même pour les 2 attributs).
2) Ecrire une classe **Personne** définie par son nom, son age, un constructeur paramétré et la méthode toString qui renvoie une chaîne représentant le nom et l'âge de la personne.
3) Ecrire une classe de **Test** contenant un programme principal permettant de créer et d'afficher une paire d'entiers, une paire de chaînes de caractères et une paire de Personnes.
4) Ecrire la classe **PaireBis** qui est une classe générique identique à la classe *Paire* mais où les attributs *x* et *y* peuvent être de types différents.
5) Créer et afficher les 3 instances suivantes de la classe *PaireBis* : ("RSI", "DSI"), (15.5, 2.4f) et (10, (Nom= Omar et Age = 25))
6) Ecrire la classe **Triplet** qui hérite de la classe *PaireBis* et qui offre un attribut z pouvant avoir un type différent de x et y.
7) Créer et afficher les 2 instances suivantes de la classe *Triplet* : (10, "Ali", 15.3) et (23, 15, "RSI")
8) Apporter les modifications nécessaires aux *Triplet* et *Test* de sorte que l'attribut *z* ait le même type que *x*.

Exercice 2 :

Soit la classe **Pile** implémentant une pile d'entiers utilisant un tableau contenant les différents éléments de la pile et un indice sommet représentant l'indice du sommet de la pile.

```
public class Pile {
	private int T[]= new int [100];
	private int sommet;
	public Pile() {sommet=-1;}
	Pile(int t) {sommet=-1;}
	public void empiler(int elt) {sommet++;T[sommet]=elt;}
	public void depiler() {sommet--;}
	public int som() {return T[sommet];}
	public void vider() {sommet=-1;}
	public void afficher(){
		for (int i=0;i<=sommet; i++)
			System.out.println(T[i]+" ");}
	public boolean pileVide() {return (sommet==-1);}
	public boolean pilePleine() {return (sommet==T.length-1);}
}
```

1) En vous inspirant de la classe précédente récrire une classe Pile générique où les éléments peuvent être de n'importe quel type.

2) Ecrire une classe de Test contenant un programme principal permettant de créer une pile de chaînes de caractères et de la tester à travers le menu suivant:

1- Empiler
2- Dépiler
3- Afficher la valeur au sommet
4- Afficher la pile
5- Vider la pile
6- Quitter

Partie 2 : Utilisation de la collection ArrayList

Exercice 3 :

1) Implémenter une nouvelle classe Pile d'entiers en utilisant cette fois une ArrayList contenant les différents éléments de la pile.

2) Réécrire la classe de Test.

Exercice 4 :

1) Implémenter une nouvelle classe Pile générique ou les éléments peuvent être de n'importe quel type en utilisant cette fois une ArrayList contenant les différents éléments de la pile.

2) Réécrire la classe de Test en créant une liste d'entiers.