

## Exercice 1 : classe générique (implémentant une interface générique)

Voici une interface définissant un type abstrait "**Pile de <A>**" avec les fonctionnalités

classiques d'une pile :

```
public interface IPile<A> {  
  
    boolean estVide(); void empile(A a); A depile(); // retourne  
    l'élément en sommet de pile et dépile int nbElements(); A sommet(); //  
    retourne le sommet de pile mais ne le dépile pas  
  
}
```

1. Ecrivez une classe générique **CPile** qui implémente l'interface **IPile**. Vous stockerez les éléments de la pile dans une liste chaînée (instance de **java.util.LinkedList**, voir en *annexe* quelques méthodes publiques de cette classe).

2. Ecrivez un petit programme qui crée et manipule des piles en instanciant la classe générique de différentes façons (par exemple pile de String, pile de Integer, ...).

## Exercice 2 : une autre classe générique

La classe suivante **Tableau** encapsule un tableau d'entiers et comporte deux méthodes permettant de trier le tableau et d'afficher son contenu. La méthode main de cette classe fournit un exemple d'utilisation. On voudrait disposer d'une méthode de tri *pour n'importe quel type de tableau*. Proposez une solution basée sur la généricité.

```
public class Tableau {  
  
    private int T [];  
    public Tableau (int T []) {this.T = T;} // on fait ici une  
    copie "superficielle"
```

```

public void trieBulles () {

    int i = T.length -2;
    boolean ech = true;
    while (i >=2 && ech)
    {
        ech = false;
        for (int j = 0; j <= i; j ++){
            if (T[j] > T[j+1])
            {
                int aux = T[j];
                T[j] = T[j+1];
                T[j+1] = aux;
                ech = true;
            }
        }
    }

    public void affiche (){
        for(int i = 0; i < T.length; i++){
            System.out.print(T[i]+" ");
        }
        System.out.println();
    }

    public static void main(String[] args)
    {
        int T[] = {10,2,6,11, 7, 2, -1, 0, 9};
        Tableau obj = new Tableau(T);
        obj.trieBulles();
        obj.affiche();
    }
}

```