

TP4 Pile, Interface et Généricité

Recopier le répertoire : <http://www.lirmm.fr/~chaumont/download/cours/structuresdedonnees/tp4/>

Ce TP a pour objectif d'aborder la programmation d'une Pile (par tableau de taille fixe), la notion d'interface Java et la notion de généricité.

I. Implémentation de la classe Pile (en respectant l'interface donnée)

L'interface Pile suivante vous est donnée (récupérez le code sur <http://www.lirmm.fr/...>) :

```
public interface Pile {  
    public boolean estVide();  
    public boolean estPleine();  
    public int nbElement();  
    public void empiler (int x);  
    public int depiler();  
    public int sommetPile();  
    public boolean appartient(int e) ;  
    public void inverse ();  
    public String toString();  
} // interface Pile
```

Vous devez « implémenter » l'interface Pile c'est-à-dire écrire une classe (donnez lui par exemple le nom PileArray) tel que l'on ait :

```
public class PileArray implements Pile {  
    // à compléter...  
    public PileArray() { // à compléter...}  
    public PileArray(int taille) { // à compléter...}  
    public PileArray(PileArray p) { // à compléter ...}  
    // à compléter...  
    public static void main (String args[]) { // à compléter... // Test de la classe }  
} //classe PileArray
```

II. La généricité

- Implémentez une classe Pile **générique**. Cette classe ne sera pas une implémentation Java de l'interface Pile ; par contre vous utiliserez les mêmes signatures pour les méthodes. Remarque : Il est impossible de créer en Java un tableau générique ; pour pallier ce problème nous prendrons la notion (dépassée) de « tableau d'Object » et nous utiliserons alors le cast (également appelé forceur ou transtipeur) à bon escient.

```
public class PileGene<T> {  
    //à compléter...  
    private T [] P; // Il n'est pas possible d'allouer un tableau générique  
                    // c'est-à-dire écrire ce genre de chose : P = new T[10];  
    //à compléter...  
    public PileGene() {  
        P = (T []) new Object [1000] ; //Création d'un tableau d'objet  
        // à compléter...  
    }  
  
    public static void main (String args[]) {  
        //Création d'une pile d'entier (attention utilisation du type Integer et non int)  
        PileGene<Integer> p = new PileGene<Integer>();  
        //à compléter...  
    }  
}
```

III. Et s'il reste du temps :

Tester et regarder la documentation des implémentations de l'interface Queue (cf. les Collections sur <http://java.sun.com/docs/books/tutorial>). Les implémentations sont : ArrayBlockingQueue, ArrayDeque, ConcurrentLinkedQueue, DelayQueue, LinkedBlockingDeque, LinkedBlockingQueue, LinkedList, PriorityBlockingQueue, PriorityQueue, SynchronousQueue.