Structures de données

L3MI

Université de Nîmes

TD6/TP6
File, Interface et Généricité

Ce TD/TP a pour objectif d'aborder la programmation d'une File (par tableau de taille fixe et **gestion circulaire des indices**), la notion d'interface Java et la notion de généricité.

Partie TD

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

L'interface File (ci-dessous) vous est donnée.

```
interface File {
  public boolean fileVide();
  public boolean filePleine();
  public void ajoutFile (int o) ;
  public int oteFile();
  int nbElement();
  public String toString();
} //Interface File
```

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

```
class FileArray implements File {

  private int [] mem;
  private int tete ;
  private int queue ;

  private int  suivant(int i) ; //retourne l'élément suivant dans le tableau mem

  public FileArray () ;
  public FileArray (int n) ;
  public FileArray (FileArray f) ;

  // à compléter...

  public static void main (String args[]) {// à compléter... // Test de la classe }

} //classe FileArray
```

Consignes : A l'initialisation, la *tête* égale la *queue*. On ajoute les éléments en *queue*. On retire les éléments en *tête*.

II. Manipulation de la classe File

Ecrire un programme qui :

- crée une file d'entiers f0,
- crée une file d'entiers f1 de contenance maximum 3 éléments,
- donne le nombre d'éléments de la file f1,
- affiche les files f0 et f1,
- ajoute les valeurs 1, 2, 3 à la file f1,
- créer une file f2 qui est la copie de f1,
- ôte un élément à f1,
- affiche f1 et f2.

III. Implémentation de la classe File (en respectant la nouvelle interface)

Les deux méthodes suivantes sont ajoutées à l'interface File :

```
public int ithelement(int i);
public boolean egal(File f);
```

- Ajoutez l'implémentation de ces deux méthodes dans la classe FileArray.
- Ajoutez le constructeur suivant dans la classe FileArray :

```
public FileArray(File f);
```

Partie TP

Recopier le répertoire : http://www.lirmm.fr/~chaumont/download/cours/structuresdedonnees/tp6/

IV. Implémentation de la classe File (en respectant l'interface donnée)

L'interface File (ci-dessous) vous est donnée.

```
public interface File {
  public boolean fileVide();
  public boolean filePleine();
  public void ajoutFile (int o);
  public int oteFile();
  public int nbElement();
  public String toString();
  public int ithelement(int i);
  public boolean egal(File f);
} // interface File
```

Vous devez implémenter **l'interface File**.

V. La généricité

- Implémentez une classe File **générique**. Cette classe ne sera pas une implémentation Java de l'interface File ; par contre vous utiliserez les mêmes méthodes (qui seront génériques). 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 transtypeur) à bon escient.

```
public class FileGene<T> {

 private T[ ] mem; // Il n'est pas possible d'allouer un tableau générique
                   // c'est-à-dire écrire ce genre de chose : mem = new T[10];
 //à compléter...


 public FileGene() {
  mem = (T[ ]) new Object[1000] ; //Création d'un tableau d'objet qui est ensuite « casté »
  // à compléter
 }

 //à compléter...

 public static void main (String args[]) {
  //Création d'une file d'entiers (attention utilisation du type Integer et non int)
  FileGene<Integer> l = new FileGene<Integer>();
  //à compléter...
 }
}
```

VI. Implémentations Java de l'interface Queue :

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.