

TD/TP 02 : LIFO & FIFO en Java*(13 Septembre 2007)*

Les files d'attente et les piles sont deux structures de données très utilisées dans divers programmes informatiques. Une file d'attente (appelée aussi structure FIFO pour First In First Out) permet de stocker d'objets et de retirer les objets stockés en respectant l'ordre d'arrivée de ces objets. Une pile emplit les objets reçus et renvoie l'objet placé en tête lorsqu'on demande de retirer un objet. Ainsi le dernier objet ajouté dans une pile sera le premier à être lu. On appelle cette structure une structure LIFO pour Last In First Out.

Une file (respectivement une pile) est dite bornée si la capacité de stockage est limitée et fixée à l'avance. On voudrait programmer en Java deux classes qui représentent une file et une pile bornées d'entiers.

Une file offre les services suivants :

`add (int i)` qui permet d'ajouter l'entier `i` à la file si c'est possible. Sinon un message d'erreur sera affiché.

`int get ()` qui renvoie l'entier en tête de la file. L'entier lu sera supprimé de la file. Si la pile est vide on affiche un message d'erreur.

Une pile offre les services suivants :

`empiler (int i)` qui permet d'ajouter l'entier `i` en tête de la pile. Si la pile est pleine on affiche un message d'erreur.

`int depiler ()` qui renvoie l'entier en tête de la pile. Si la pile est vide on affiche un message d'erreur.

1. Dans un premier temps on modélise l'espace de stockage par un tableau d'entiers. Donner le code des deux classes `File` et `Pile`.
2. Développer une classe `TestFilePile` qui permet de tester le fonctionnement d'une pile et d'une file d'entiers.
3. On voudrait rendre la taille de l'espace de stockage dynamique. On propose d'utiliser un objet de la classe `Vector` (du package `java.util`) pour implanter l'espace de stockage d'entiers. Modifier en conséquence le code des deux classes `File` et `Pile` **sans modifier les signatures des méthodes offertes par ces classes.**