Les piles

I Description

Les piles sont des conteneurs très utilisés en informatique (variables locales ou arguments de fonction sont stockés dans la pile, calculs des expressions arithmétiques, ...), ainsi que dans la conception de certains processeurs (processeurs dits à pile, par opposition aux registres). Une pile est une liste ordonnée sans fin (néanmoins limitée par la mémoire disponible) d'éléments, dans laquelle on peut enlever ou introduire un élément à une extrémité (tête ou sommet de la pile). On distingue deux types de pile :

- Pile LIFO, acronyme de Last In First Out. On compare cette pile généralement avec une pile d'assiettes (la dernière assiette, posée sur le sommet de la pile d'assiettes, est la première à être retirée).
- Pile FIFO, acronyme de First In First Out. On compare cette pile généralement avec une file d'attente à un guichet : la première personne arrivée est la première à sortir de la file.

II Opérations

Les opérations sur les piles sont

- l'empilage (en anglais, push) : on ajoute un élément dans la pile,
- le dépilage (en anglais, pop) : on retire un élément de la pile.

On ne considèrera dans ce TD que les piles FIFO.
On ne pourra ajouter qu'un nombre maximal d'éléments, et on ne peut pas enlever un élément d'une une liste vide. La contrainte du nombre maximal d'éléments peut être enlevée avec une liste chaînée (*cf* Section V).
On considèrera aussi les opérations suivantes :

- initialisation de la pile,
- vider la pile,
- test si la pile est vide,
- récupérer le sommet de la pile.

III Implémentation

On se propose d'écrire une classe gérant une pile FIFO. Cette classe sera générique et le type des éléments sera donc un template.
Un diagramme de classe est proposé Figure 1. La description des attributs et des méthodes est la suivante :

- **max_size_** : la taille (maximale) du tableau.
- **top_** : l'indice de la première position libre dans le tableau.
- **array_** : le tableau de taille **max_size_**.
- Constructeur : il initialise les membres de données. **size** est la taille maximale de la pile.
- Destructeur et constructeur de copie : à implémenter.
- **is_empty()** : teste si la pile est vide ou non.
- **empty()** : vide la pile.

pop ← 1 2 3 ← push.
0 1 2 indices

— **push()** : ajoute **item** sur la pile.
— **pop()** : retire le dernier élément de la pile et le retourne.
— **peek()** : renvoie le dernier élément de la pile.

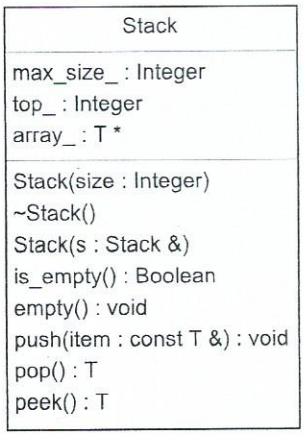

FIGURE 1 – Pile.

IV Utilisation

On testera le code avec un programme utilisant une pile de 5 entiers. Une fois remplie, on l'affichera et on la videra.

V Amélioration : liste chaînée

On pourra améliorer le code en utilisant une liste chaînée à la place d'un tableau.