

Java DUT 1 Feuille TD4
Université Paris-Est Marne-la-Vallée

Exercice 1.—

- a) On considère une interface qui permet de manipuler une pile d'entiers positifs (des `int`). Une pile est une structure dans laquelle on peut empiler et dépiler des éléments. On peut également consulter le sommet de la pile, tester si la pile est vide ou pleine (lorsque sa capacité est limitée). On ne traitera pas les cas où les opérations sont impossibles (voir plus tard avec la gestion des exceptions). On n'empilera rien lorsque la pile est pleine. On ne dépilera rien lorsque la pile est vide. On retournera par exemple `-1` lorsqu'on demande le sommet d'une pile vide.

```
interface Stack {
    boolean isEmpty ();
    boolean isFull();
    void push(int item);
    int peek();
    int pop();
}
```

Écrire une classe `ArrayStack` qui implémente l'interface `Stack`. La classe `ArrayStack` contiendra un tableau `table` d'entiers dont la taille maximale est une constante `MAX` comme ci-dessous. Un champ `height` désigne la hauteur de la pile, c'est-à-dire le nombre d'éléments empilés.

```
public class ArrayStack implements Stack {
    static final int MAX=4;
    private int height = 0;
    private final int[] table = new int[MAX];
    ...
}
```

- b) Écrire une classe `ListStack` qui implémente l'interface `Stack`. La classe `ListStack` contiendra cette fois une `ArrayList<Integer>`. Dans ce cas la pile ne sera jamais pleine.

```
public class ListStack implements Stack {
    private final List<Integer> table;
    ....
}
```