

TP2

<p>Lectures préalables :</p> <ul style="list-style-type: none">• Java_I• tutorial<ul style="list-style-type: none">◦ Getting Started◦ Learning the Java Language	<p>Thèmes du TP :</p> <ul style="list-style-type: none">• Une Pile d'entiers• Plusieurs classes• Tableaux• Variables d'instance
--	--



Développer la traditionnelle structure de données **Pile** (premier entré, dernier sorti).

Une Pile est ici une structure de données homogènes, de taille fixe, sur laquelle seul un nombre limité d'opérations peut être effectué :

- constructeur = créer une pile vide d'une certaine taille ou d'une taille par défaut
- procédure empiler = mettre une valeur au sommet de la pile
- fonction dépiler = retirer la valeur du sommet de la pile et la retourner
- fonction estVide = retourne vrai ou faux selon que la pile est vide ou non
- fonction estPleine = retourne vrai ou faux selon que la pile est pleine ou non

Contraintes :

- La pile sera implantée par un tableau.
- Les éléments sont de type **int**.
- Les méthodes dont les signatures figurent ci-dessous doivent être implantées dans la classe Pile, pour l'instant sans utiliser d'exceptions en cas de débordement.

```
public Pile()  
public Pile( int taille )  
public void empiler( int i )  
public int dépiler()  
public boolean estVide()  
public boolean estPleine()
```

Pour pouvoir visionner le contenu de la pile, ajoutez à la classe Pile la méthode **public String toString()**;

Cette méthode retourne le contenu de la pile sous la forme "[i_1 , i_2 , i_3]" pour une pile de 5 cases avec 3 éléments où i_1 est l'entier au sommet de la pile.

```
public String toString()
```

Lors de l'appel du constructeur avec **une taille négative** (étourderie ou un test malicieux ...), c'est la taille par défaut `CAPACITE_PAR_DEFAULT` qui est retenue

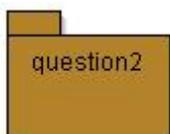
Une utilisation, répondez aux questions (en commentaire, après la question) :

*Exemple la pile résultante des actions `p1.empiler(2)` suivi de `p1.empiler(5)` est en "String" la valeur **[5,2]***

```
public class UneUtilisation{ // fichier TestPile.java
    public static void main(String[] args){
        Pile p1 = new Pile(5);
        Pile p2 = new Pile(10);
        // déjà complété par vos tests...
        System.out.println(" la pile p1 : " + p1.toString());
        // ou bien
        System.out.println(" la pile p1 : " + p1);

        Object o = new Object();
        o = p1;
        System.out.println(" o = " + o); // Quel est l'affichage résultant ? pou

        Pile p3 = new Pile(4); p3.empiler(5); p3.empiler(6);
        Pile p4 = new Pile(6); p4.empiler(8);
        p3 = p4;
        int res = p3.dépiler();
        System.out.println(p4.estVide()); // true ou false ? pourquoi ?
    }
}
```



Ajouter à la classe Pile la gestion des exceptions :

- l'exception "**Pile Vide**" lorsque l'on tente de "dépiler" une pile qui est vide
- l'exception "**Pile Pleine**" lorsque l'on tente d'"empiler" sur une pile déjà pleine.
- Ces **2** exceptions sont implantées par les **2** classes Java :


```
public class PileVideException extends Exception{}
// fichier PileVideException.java
public class PilePleineException extends Exception{}
// fichier PilePleineException.java
```
- Il n'est pas nécessaire de rattraper les exceptions dans cette question (voir question suivante).



Complétez l'applette Pile:

Modifiez la classe **ApplettePile** afin de prendre en compte les exceptions susceptibles d'être levées et d'en informer l'utilisateur. Exceptions comme `NumberFormatException`, `PilVideException`, `PilePleineException`

Vous devriez obtenir une applette avec le comportement ci-dessous :



ApplettePile avec exceptions

Cette IHM est perfectible : proposer une extension dans la classe `ApplettePile2` en invalidant le bouton "empiler" lorsque la pile est pleine (appel de `boutonEmpiler.setEnabled(false)`); de même pour le bouton "depiler", comme le suggère cette mise en oeuvre.



ApplettePile2 de taille 3

AIDE sous Linux : Si les applettes n'apparaissent pas ci-dessus avec Konqueror, menu Outils/Configuration_HTML/Java, puis recharger la page.

N'oubliez pas de faire un submit avant de quitter la salle !