

# BPOO - Support TD 2 - Eléments de Solution

Dut/Info-S2/M2103

## Table des matières

- 1. Utilisation de classe
  - 1.1. Exercice Pile
  - 1.2. Exercice Devinette



### Version corrigée

Cette version comporte des indications pour les réponses aux exercices.

PreReq	Cours 1 : approche objet. S1.
ObjTD	<b>Utiliser des objets en Java.</b>
Durée	<b>1 séance</b> de 1,5h

## 1. Utilisation de classe

### 1.1. Exercice Pile

Soit la classe Pile suivante dont on ne donne que les constructeurs et les méthodes (sans les corps) :

```

class Pile {
    private String[] elements;
    private int indiceSommet;

    public Pile() { ... }
    public Pile(int pfTaille) {
        this.elements = new String [pfTaille];
        this.indiceSommet = -1
    }

    public boolean estVide() { ... }

    public void empiler(String pfElement) throws Exception { ... }

    public void depiler() throws Exception { ... }

    public String sommet() throws Exception { ... }
}

```

A faire :

1. Étudier le code de la classe Pile.
2. Ecrire (feuille jointe) un programme main() utilisant **obligatoirement des objets Pile** permettant : a) de saisir 10 chaînes de caractères, b) qui sépare d'un côté celles plus petites que "moto" et de l'autre celles plus grandes, et c) qui affiche ces deux "sous-listes" dans l'ordre inverse de leur saisie.

Exemple : saisie de "a" "b" "n" "o" "p" "c" "m" "d" "e" "q" affichera :

⇒ e d m c b a

⇒ q p o n

NB : pour comparer deux chaînes, on dispose de la méthode suivante dans la classe String :

- public int compareTo (String otherString) : compare la chaîne avec otherString et renvoie 0 si elles sont égales, une valeur <0 si la chaîne est plus petite que otherString, une valeur >0 si la chaîne est plus grande que otherString.



## Solution

```
public class Test {  
    public static void main (String argv[]) throws Exception {  
        Scanner sc = new Scanner(System.in);  
        String ch;  
  
        Pile moins, plus;  
        moins = new Pile(10);  
        plus = new Pile(10);  
  
        for (int i=0; i<10; i++) {  
            ch = sc.next();  
            if (ch.compareTo("moto") >= 0)  
                plus.empiler(ch);  
            else  
                moins.empiler(ch);  
        }  
  
        System.out.println();  
        while (! moins.estVide()) {  
            ch = moins.sommet();  
            System.out.println(ch);  
            moins.depiler();  
        }  
  
        System.out.println();  
        while (! plus.estVide()) {  
            System.out.println(plus.sommet());  
            plus.depiler();  
        }  
    }  
}
```

## 1.2. Exercice Devinette

Nous allons utiliser une classe Devinette pour jouer.

Au jeu de la "devinette" :

- On suppose une valeur entière à trouver, tirée au sort par le programme, entre une borne basse ( $10 < x < 20$ ) et haute ( $40 < x < 50$ ).
- L'utilisateur cherche la valeur par essais successifs, la machine répondant à chaque coup si le nombre recherché est supérieur ou inférieur au nombre soumis.
- Le jeu s'arrête quand la valeur est trouvée.

Exemple d'exécution :