

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	Utiliser des objets en Java.
Durée	1 séance 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 = -1;

    public Pile() { ... }

    public Pile(int pfTaille) { ... }

    public boolean estVide() { ... }

    public void empiler(String element) 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();
            moins.depiler();
            System.out.println(ch);
        }

        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<>20) et haute (40<>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 :

```
Coup : (13-49) : 33 // ❶
Plus Haut ... Coup : (13-49) : 40 // ❷
Plus Haut ... Coup : (13-49) : 49 // ❸
Plus Bas ... Coup : (13-49) : 48
Plus Bas ... Coup : (13-49) : 41
Plus Haut ... Coup : (13-49) : 47
Plus Bas ... Coup : (13-49) : 44
Plus Haut ... Coup : (13-49) : 46
Bravo, en 8 coups.
```

- ❶ Saisie utilisateur de 33
- ❷ Saisie utilisateur de 40
- ❸ ...

A faire :

1. Étudier la documentation de la classe Devinette (constructeurs/méthodes).

2. Un objet Devinette, instance de la classe Devinette, est un objet qui :
 - lors de sa création, détermine des bornes min et max au jeu et choisit la valeur à chercher,
 - tirer au sort la borne supérieure ($40 <= 50$) et la borne inférieure ($10 <= 20$) du jeu.
 - tirer au sort la valeur à rechercher entre les bornes choisies ci-dessus.
 Exemple : borne inf 12, borne sup 47, valeur à chercher 22.

 - dispose de méthodes permettant de jouer et d'interroger l'état de la partie (nombre de coups joués, résultat d'un coup, ...).
3. Écrire un programme main permettant de faire une partie de devinette en utilisant la classe Devinette : tirage au sort des valeurs, demander une valeur comprise entre les deux bornes et répondre à l'utilisateur sur la validité de son coup, afficher « partie gagnée » à la fin ainsi que le nombre de coups joués (essais).
4. Écrire un autre programme main permettant de jouer successivement 3 parties de devinette. Écrire d'abord un sous-programme `void jouer (Devinette d)` permettant de "dérouler" une partie de devinette avec l'objet d en paramètre. Puis écrire le programme demandé (quelques lignes).



Solution

```
import java.util.Scanner;
import classesteps.*;

public class TestDevinette
{
    private static void jouer (Devinette d) throws ErreurExecutionDevinette
    {
        int val;
        Scanner in;

        in = new Scanner(System.in);

        System.out.print("Coup : (" + d.getBas() + "-" + d.getHaut() + ") : ");
        val = in.nextInt();
        d.soumettreCoup (val);
        while (!d.isDernierCoupGagnant())
        {
            if (d.isDernierCoupTropBas())
                System.out.print("Plus Haut ... ");
            else
                System.out.print("Plus Bas ... ");

            System.out.print("Coup : (" + d.getBas() + "-" + d.getHaut() + ") : ");
            val = in.nextInt();
            d.soumettreCoup (val);
        }
        System.out.println("Bravo, en " + d.getNbCoupsJoues() + " coups.");
    }

    public static void main2 () throws ErreurExecutionDevinette
    {
        Devinette d;
        int i;

        for (i=0; i<3; i++)
        {
            d = new Devinette() ;
            TestDevinette.jouer (d);
        }
    }

    public static void main1 ()
    {
        Devinette d;
        int val;
        Scanner in;

        in = new Scanner(System.in);

        d = new Devinette() ;

        System.out.print("Coup : (" + d.getBas() + "-" + d.getHaut() + ") : ");
        val = in.nextInt();
        try {
            d.soumettreCoup (val);
        } catch (ErreurExecutionDevinette e) {
            System.out.println(e);
        }
    }
}
```

```

        System.out.println("Bug !!");
    }
    while (!d.isDernierCoupGagnant())
    {
        if (d.isDernierCoupTropBas())
            System.out.print("Plus Haut ... ");
        else
            System.out.print("Plus Bas ... ");

        System.out.print("Coup : (" + d.getBas() + "-" + d.getHaut() + ") : ");
        val = in.nextInt();
        try {
            d.soumettreCoup (val);
        } catch (ErreurExecutionDevinette e) {
            System.out.println(e);
            System.out.println("Bug !!");
        }
    }
    System.out.println("Bravo, en " + d.getNbCoupsJoues() + " coups.");

    // d.soumettreCoup (20); // Plantage car partie finie ...
}
public static void main (String argv[]) throws ErreurExecutionDevinette
{
    System.out.println("Exo 1");
    TestDevinette.main1();

    System.out.println("Exo 1");
    TestDevinette.main2();
}
}

```