

TP numéro 6

Exercice 1

1. Ecrire une méthode « `factorielle()` » qui permet de calculer la factorielle d'un entier.
2. Appelez cette méthode dans la classe **CalcFactorielle** suivante :

```
import java.util.Scanner;

public class CalcFactorielle {
    public static void main(String[] args) {
        int n;
        Scanner clavier = new Scanner(System.in);
        int deb, fin;
        System.out.println("Saisir la taille du tableau");
        n = clavier.nextInt();
        int[] tab = new int[n];
        System.out.println("Saisir deb");
        deb = clavier.nextInt();
        System.out.println("Saisir fin");
        fin = clavier.nextInt();

        for (int i = deb; i < fin; i++)
            tab[i] = factorielle(i);

        for (int i = deb; i < fin; i++)
            System.out.println("Factorielle(" + i + ") = " + tab[i]);
        clavier.close();
    }
}
```

3. Tester le programme précédent avec les valeurs suivantes et analysez le résultat de l'exécution :
 - a - $n = 10$ $deb = 0$ $fin = 10$
 - b - $n = -1$ $deb = 0$ $fin = 10$
 - c - $n = 10$ $deb = 0$ $fin = 11$
 - d - $n = 20$ $deb = 0$ $fin = 20$
 - e - $n = 2,5$ $deb = 0$ $fin = 10$ (saisir 2,5 pour n)
 - f - $n = a$ $deb = 0$ $fin = 10$ (saisir a pour n)
4. Refaire la méthode « `main()` » afin de gérer les différentes exceptions signalés par les exécutions précédentes.

Exercice 2

Reprenez le TP3 (exercice 2) et l'exercice 4 du TP5.

1. Gérez les exceptions suivantes :
 - exception 1 : l'utilisateur saisisse autre chose qu'un nombre ;
 - exception 2 : le compte ne contient pas assez de solde.

Créez le fichier « soldes.txt » contenant les lignes suivantes :

```
Oujdi : Ali : 30 : 1000 Dh
Oujdi : Lina : 27 : 2000 Dh
Berkani : Karim : 35 : 3000 Dh
Berkani : Karima : 40 : 2500 Dh
```

Créez des comptes à partir du fichier « soldes.txt ».

2. Enregistrer les informations concernant les différents comptes créés dans le fichier « soldes_info.txt ».

Exercice 3 (programmation générique)

Dans le chapitre 10, nous avons utilisé la classe **Vector** comme suit :

```
//permet de gerer des tableaux dynamiques(vecteurs)
import java.util.Vector;

public class Erreurs {
    public static void main(String[] args) {
        Vector<Double> liste = new Vector<Double>();
        boolean b = true;
        while (b) {
            liste.add(2.0);
        }
    }
}
```

A la place de **Double** on peut mettre n'importe quelle classe comme suit :

```
Vector<ClasseA> listeClasseA = new Vector<ClasseA>();
```

La classe **Vector** est appelée classe générique.

Pour définir une classe générique, on procède comme suit :

```

class ClasseGenerique <Type> {
    private Type t;

    public void set(Type t) {
        this.t = t;
    }

    public Type get() {
        return t;
    }
}

```

1. Reprenez les classes **Personne**, **Etudiant** et **Professeur** définis dans les TPs précédents et testez les avec la classe suivante :

```

class ClasseObjet {
    private Object o;

    public void set(Object o) {
        this.o = o;
    }

    public Object get() {
        return o;
    }
}

```

Procédez comme suit :

```

Personne p=new Personne("Oujdi", ...);
ClasseObjet obj = new ClasseObjet();
obj.set(p);
//Affichez le contenu de obj

```

2. Refaire les tests avec la classe générique précédente comparez les résultats.