

Deuxième Séance d'Exercice

- **Exercice 1** : Les variables et les fonctions static
- **Exercice 2** : Blocs de structures et visibilité des variables
- **Exercice 3** : Obtention de la taille des répertoires et des fichiers. Reperage des plus gros fichiers, des plus gros répertoires
- **Exercice 4** : Un exercice sur la création de fenêtre.
- **Exercice 5** : Une première création d'interface et un test des différents mode d'organisation.
- **Exercice 6** : 1 premier exercice sur les événements
- **Exercice 7** : 1 Inteface de gestion des images

Ex1

Jusqu'à maintenant, on n'a principalement défini et utilisé des variables et des fonctions d'objets. Une fonction d'objet peut être interprétée comme le comportement associé à un objet : pour un objet fichier (de type File), on peut l'ouvrir et le fermer, pour une fenêtre, on peut l'afficher, la fermer, la réduire etc.

Fonction static : fonction qui ne manipule les attributs d'aucun objet (que ceux qui lui seraient passés en paramètre). Par exemple, la fonction `Math.log(double x)` est une fonction static. Il existe aussi des attributs de classe. Ces attributs sont initialisés lors du chargement de classe par la machine virtuelle java (JVM), soit au début du programme. Pour définir une fonction static ou un attribut static :

```
public class Utils{  
  
    public static int nbAppelsMax=0;  
  
    public static int max(int a,int b)  
    {  
        nbAppelsMax++;  
        if(a>b)  
            return a;  
        else  
            return b;  
    }  
}
```

Ici, pour appeler la fonction static : `int n=Utils.max(5,6);`

Pour accéder ou modifier l'attribut de classe `nbAppelsMax` : `Utils.nbAppelsMax=5;`

Soit le code suivant :

```
public class Facture{  
  
int nombreUnitesVendues;  
static int nombreFacturesCreees;
```

```
Facture()  
    {  
        nombreUnitesVendues++;  
        nombreFacturesCreees++;  
    }  
  
}
```

- Ecrire un code qui appelle la construction de trois objets de type Facture.
- Lors de l'exécution du code précédent, combien d'entiers sont stockés en mémoire ? Quelles sont les valeurs de ces trois entiers après la construction des trois objets de type Facture ?

Soit le code :

```
public class Facture{  
  
int nombreUnitesVendues;  
static int nombreFacturesCreees;  
  
Facture()  
    {  
        nombreUnitesVendues++;  
        nombreFacturesCreees++;  
    }  
  
public static void test()  
    {  
        System.out.println(nombreFacturesCreees);  
        System.out.println(nombreUnitesVendues);  
    }  
  
}
```

- Ce code se compile t-il ?
- Pouvez vous envisager de créer une classe telle que toutes les occurrences créées pour cette classe soient collectionnées ?

Ex2

Soit la fonction suivante :

```
public double ecartType(double[] serie)  
    {
```

```

double moyenne=0;
    for(int i=0;i<serie.length;i++)
        moyenne=moyenne+serie[i];
moyenne=moyenne/serie.length;
double ecartType=0;
    for(int j=0;j<serie.length;j++)
    {
        double temp=Math.pow(serie[j]-moyenne,2);
        ecartType+=temp;
    }
ecartType=Math.pow(ecartType/serie.length,0.5);
System.out.println(ecartType+" "+temp);
return ecartType;
}

```

- Quelle(s) erreur(s) renvoie cette fonction à la compilation ? Pourquoi ?

Ex3

L'exercice repose sur l'utilisation de deux classes du package io : File (java.io.File) et FileOutputStream (java.io.FileOutputStream). A partir de ces deux classes, on construit des fonctions d'évaluation de la taille des fichiers et des répertoires.

- 1) On commence par construire une fonction qui renvoie la taille d'un fichier pour lequel on passe le chemin d'accès en entrée.
- 2) On modifie la fonction précédente pour renvoyer non pas la taille, mais un entier qui donne le log base 10 de la taille (en sachant qu'on dispose de la fonction Math.log(double x) qui renvoie le logarithme népérien de x).
- 3) Créez une fonction pour récupérer la taille d'un fichier ou d'un répertoire (sous-répertoires inclus). Affichez tous les fichiers qui sont passés en revue dans cette opération.
- 4) Affichez l'ensemble des fichiers passés en revue dans un fichier .txt
- 5) Modifiez la fonction pour faire en sorte qu'elle renvoie la liste de tous les fichiers d'une taille supérieure à une taille passée en paramètre.
- 6) A partir de la fonction précédente, quelle structure de données vous semble pouvoir contenir la distribution de la taille des fichiers ? Utilisez cette structure et la fonction des points précédents pour calculer et afficher la distribution des tailles de fichiers.

Ex4 Création d'une simple fenêtre

On considère le code suivant :

```
public class TestJFrame{
```

```
public static void main(String[] args)
{
    JFrame fen=new JFrame();
    fen.setSize(300,300);
    fen.setLocation(10,10);
    fen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    fen.setVisible(true);
}
}
```

- Eliminez des instructions de manière à avoir un code minimal pour l'affichage d'une Fenetre à l'écran.