

<b>Etablissement</b> : ISET-Charguia	<b>Département</b> : Technologies de l'Informatique
<b>Matière</b> : Programmation Objet Avancée	<b>Année</b> : 2 <sup>ème</sup> année
<b>Année Universitaire</b> : 2023- 2024	

## TP n° 1 : Les exceptions sous JAVA

### **Objectifs du TP :**

- **Comprendre le mécanisme d'exception sous JAVA**
- **Utiliser les exceptions sous Java : Traiter, propager et personnaliser les exceptions**

### **Exercice :**

- 1) Ecrire une classe ville renfermant :
  - Les attributs de type entier : nbrHabitants, nbrJeunes
  - Les accesseurs et les mutateurs relatifs aux attributs de la classe
  - Un constructeur paramétré permettant d'initialiser les deux attributs
  - Une méthode publique pourcentageJeunes renvoyant le pourcentage de jeunes par rapport à l'ensemble d'habitants dont la valeur est la suivante :  $(nbrJeunes/nbrHabitants)*100$ .
- 2) Ecrire une classe Program contenant la méthode main et permettant de tester la méthode pourcentageJeunes successivement avec les valeurs :
  - nbrHabitant= 300 et nbrJeunes=100
  - nbrHabitant= 0 et nbrJeunes=100
 Que constatez vous ?
- 3) **Capter l'exception :**  
 Il s'agit maintenant de capturer l'exception de notre division par zéro, et d'afficher un message personnalisé. Pour ce faire, modifier le code de la méthode pourcentageJeunes comme suit :

```

public int pourcentageJeunes(){
    int res=0;
    try {
        res=( getNombreJeunes()*100/getNombreHabitants());
        return res;
    }
    catch (ArithmeticException e) {
        System.out.println("Nombre Habitants doit être non nul : "+e.getMessage());
        return res;
    }
}

```

- 4) Utiliser le bloc finally : Ajouter le bloc finally comme suit :

```

finally
{
    System.out.println("Exécuté dans tous les cas ");
}

```

5) Exception de type checked Exception

- Créer un fichier texte nommé fichier.txt dans le répertoire de votre projet
- Ecrire dans ce fichier le nom de votre ville.
- Ajouter une méthode lireFichier dans la classe ville dont le code est le suivant :

```
public void lireFichier() {  
    BufferedReader br = new BufferedReader(new FileReader("fichier.txt"));  
    String line = br.readLine();  
    System.out.println(line);  
    br.close();  
}
```

Compilez.

Traitez les exceptions en ajoutant le code suivant :

```
public static void lireFichier() {  
    try {  
        BufferedReader br = new BufferedReader(new FileReader("fichier.txt"));  
        String line = br.readLine();  
        System.out.println(line);  
        br.close();  
    } catch (FileNotFoundException e) {  
        System.out.println("Fichier non trouvé");  
    } catch (IOException e) {  
        System.out.println("Erreur de lecture");  
    }  
}
```

Appelez la méthode dans la méthode main et testez la.

6) Propager l'exception :

On souhaite ne pas traiter l'exception immédiatement dans la méthode **lireFichier**. Dans ce cas on doit déclarer celle-ci en utilisant le mot-clé **throws** dans l'en-tête de la méthode. On propage alors l'exception à la méthode appelante, c'est à dire à la méthode **main** qui fait appel à la méthode à partir de laquelle l'exception est générée.

Modifier la méthode lireFichier comme suit :

```
public void lireFichier() throws FileNotFoundException,IOException{  
    BufferedReader br = new BufferedReader(new FileReader("fichier.txt"));  
    String line = br.readLine();  
    System.out.println(line);  
    br.close();  
}
```

Modifier le code d'appel dans la méthode main pour prendre en charge les exceptions ainsi passées à l'appelant.

## 7) Personnalisation de l'exception

Pour ce faire :

1. Nous devons créer une classe héritée de la classe `Exception` : appelons la **`NombreHabitantsException`**. Par convention, les exceptions ont un nom se terminant par **`Exception`**.

```
public class NombreHabitantsException extends Exception {  
    public NombreHabitantsException() {  
        System.out.println("Vous essayez d'instancier une classe Ville avec un nombre  
d'habitants négatif !");  
    }  
}
```

2. Nous devons renvoyer l'exception levée à notre classe **`NombreHabitantsException`**. Nous allons le faire au niveau du constructeur avec un paramètre permettant d'initialiser le nombre d'habitants de la ville comme suit :

```
public Ville(int nombreHabitants) throws NombreHabitantsException  
{  
    if (nombreHabitants<0)  
        throw new NombreHabitantsException();  
    else  
  
        this.nombreHabitants=nombreHabitants;  
}
```

3. Ensuite, gérer celle-ci dans notre classe **`NombreHabitantsException`**.

Avant de le faire Ecrire la méthode main comme suit :

```
public static void main(String[] args) {  
  
    Ville v=new Ville(-500);  
}
```

Que Remarquez vous ?

Modifier ceci pour traiter l'exception

```
try {  
    Ville v=new Ville(-500);  
}  
catch (NombreHabitantsException e) {}
```