



## TP N° 6

### Exercice 1

Recopiez et exécutez les exemples du cours « Gestion des exceptions »

### Exercice 2 (Gestion des Exceptions)

Complétez les classes suivantes en se référant aux exemples du cours

```
class NombreHabitantsException .....  
..... // Constructeur  
..... // toString()  
}
```

```
class Ville {  
    private int nHabitants ; // Nombre Habitants doit être >= 0  
    public Ville ( int nh ) ..... {  
        .....  
        nHabitants = nh ;  
    }  
    public void afficher(){  
        if (nHabitants<10000)  
            System.out.print ("C'est une petite ville ! " ) ;  
        else if (nHabitants<100000)  
            System.out.print ("C'est une ville moyenne ! " ) ;  
        else System.out.print ("C'est une grande ville ! " ) ;  
    }  
}
```

```
public class PP { // Ajoutez les instructions nécessaires à l'intérieur de PP  
    public static void main (String args[]) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print ("Donnez le nombre d'habitants : " ) ;  
        int n = sc.nextInt();  
        Ville ville = new Ville(n);  
        ville.afficher();  
    }  
}
```

Complétez les classes ci-dessus pour que le programme affiche (dans le cas où  $n \leq 0$ ) :

Donnez le nombre d'habitants : -5  
NombreHabitantsException : -5 n'est pas un nombre d'habitants !



**Exercice 3 (Pour révision) :** complétez les classes suivantes en respectant les commentaires

```
..... Personne {  
..... String nom; // nom est un attribut privé sauf pour les classes dérivées  
..... int nbPersonnes = 0; // indique le nombre de Personnes créées  
// Constructeur de Personne  
.....  
// Redéfinir la méthode toString() pour fournir les caractéristiques d'une Personne  
.....  
// afficherPersonne () est une méthode abstraite (à redéfinir dans les classes dérivées)  
.....  
// nbPersonnes () est une méthode qui affiche le nombre total de personnes, de  
//secrétaires et d'enseignants.  
.....  
} // Personne
```

```
class Secretaire .....{ // Une Secretaire est une Personne  
private String numeroBureau;  
..... int nbSecrétaires = 0; // nbSecrétaires indique le nombre de secrétaires  
// Constructeur  
.....  
// Redéfinir la méthode toString() pour qu'elle fournisse en plus des caractéristiques  
// d'une Personne, les caractéristiques d'une Secrétaire  
.....  
// afficherPersonne () affichera la chaine "Secrétaire : " suivie de ses caractéristiques  
.....  
// nbSecrétaires () est une méthode qui retournera le nombre total de secrétaires  
.....  
} // Secretaire
```

```
class Enseignant .....{ // Similaire à la classe Secretaire en faisant attention  
aux //caractéristiques propres à un enseignant  
private String specialite;  
.....  
.....  
} // Enseignant
```

**Classe Principale**

1. Ecrire la classe principale **PPPersonne** où vous créerez un objet **ArrayList** nommé **personnes** qui permettra de contenir que des objets **Secretaire** et **Enseignant**.
2. Créer un objet **Secretaire** et un objet **Enseignant** que vous placerez dans **personnes**.
3. Listez les éléments de l'objet **personnes**, en affichant les caractéristiques de ses éléments (**Secretaire** et **Enseignant**) et en utilisant les boucles "for" vues en cours (for avec index, for-each et for avec Iterator).
4. Afficher le nombre d'objets **Personne**, **Secretaire** et **Enseignant** en appelant **nbPersonnes()**.



## Exercice 2

## Corrigé

```
class NombreHabitantsException extends Exception {  
    int valeur;  
    NombreHabitantsException ( int v ) {  
        valeur=v;  
    }  
  
    String toString() {  
        return super.toString()+ ":"+valeur + " n'est pas un nombre d'habitants"; }  
}
```

```
class Ville {  
    private int nHabitants ; // Nombre Habitants doit être >= 0  
  
    public Ville ( int nh ) throws NombreHabitantsException {  
        if (nh <=0) throw new NombreHabitantsException(nh);  
        nHabitants = nh ;  
    }  
    public void afficher(){  
        if (nHabitants<10000)  
            System.out.print ("C'est une petite ville ! " ) ;  
        else if (nHabitants<100000)  
            System.out.print ("C'est une ville moyenne ! " ) ;  
        else System.out.print ("C'est une grande ville ! " ) ;  
    }  
}
```

```
import java.util.*;  
public class PP{  
    public static void main (String args[]) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print ("donnez le nombre d'habitants : " ) ;  
        int n = sc.nextInt();  
        try {  
            Ville ville = new Ville(n);  
            ville.afficher();  
        }  
        catch (NombreHabitantsException nHE){  
            System.out.println(nHE);  
        }  
    }  
}
```



### Exercice 3

```
abstract class Personne { // classe abstraite, non instanciable

    protected String nom;
    protected static int nbPersonnes = 0; // nombre de Personne

    // constructeur de Personne
    Personne (String nom) {
        this.nom = nom;
        nbPersonnes++; //s'incrémente à chaque instanciation
    }
    // fournir les caractéristiques d'une Personne sous forme
    // d'un objet de la classe String
    public String toString() {
        return nom ;
    }
    // méthode abstraite (à redéfinir dans les classes dérivées)
    abstract void afficherPersonne ();

    static void nbPersonnes () {
        System.out.println ("\nNombre de Personnes : " + nbPersonnes +
            "\nNombre de secrétaires : " + Secretaire.nbSecretaires() +
            "\nNombre d'enseignants : " + Enseignant.nbEnseignants() + ); }

}
```

```
class Secretaire extends Personne { // héritage de classe
    private String numeroBureau;
    private static int nbSecretaires = 0; // nombre de Secretaire

    Secretaire (String nom, String numeroBureau) {
        // le constructeur de la super-classe
        super (nom); // appel du constructeur Personne
        this.numeroBureau = numeroBureau;
        nbSecretaires++;
    }
    public String toString () {
        // super.toString() : toString() de la super-classe
        return super.toString() + "\n N° bureau : " + numeroBureau;
    }

    void afficherPersonne () {
        System.out.println ("Secrétaire : " + toString());
    }

    static int nbSecretaires () { return nbSecretaires; }
} // Secretaire
```



Même chose que la classe Secrétaire

```
class Enseignant .....{  
    // Similaire à la classe Secrétaire en faisant attention aux  
    // caractéristiques propres à un enseignant  
    private String specialite;  
    .....  
    .....  
} // Enseignant
```

```
import java.util.ArrayList;  
class PPPersonne { // Programme Principal Personne  
  
    public static void main(String[] args) {  
        ArrayList<Personne> pers= new ArrayList<Personne>();  
  
        Secrétaire amina = new Secrétaire ("Amina","B021");  
        pers.add(amina);  
        Enseignant reda = new Enseignant ("Reda", "POO");  
        pers.add(reda);  
  
        Personne.nbPersonnes();  
  
        for (Personne p:pers)                // boucle for-each  
            p.afficherPersonne();  
  
    } // main  
} // class PPPersonne
```