

# PROGRAMMATION ORIENTÉE OBJET (POO)

## Application au langage Java

### Travaux pratiques

POO

### Série N° 4

## Exceptions en Java

Java

Youssef EL ALLIOUI

[y.elalloui@usms.ma](mailto:y.elalloui@usms.ma)

### Exercice 1.

Ecrire une classe *Pile* qui implémente une pile d'objets avec un tableau de taille fixe. On définira pour cela deux exceptions *PilePleine* et *PileVide*. On utilisera pour écrire les méthodes, l'exception *ArrayOutOfBoundsException* qui indique qu'on a tenté d'accéder à une case non définie du tableau. Les champs de la classe seront :

```
private final static int taille = 10;
private Object [] pile;
private int pos;
```

Ecrire la méthode *main ()* qui permet d'empiler des nombres aléatoires compris entre  $-n$  et  $n$  et qui les réécrit dans l'ordre inverse. On aura par exemple :

```
Pile 1 2 3 4 5 6 7 8 9 10 11 12 13 10 9 8 7 6 5 4 3 2 1
```

### Exercice 2.

Ecrire une classe *Enfant* qui contient un attribut *âge*. L'Age doit être compris entre 1 et 10 sinon une exception doit être levée.

Ecrire une exception *AgeEnfantException* afin qu'elle renvoie un message du type :

```
[age] ans n'est pas un âge valide !
```

### Exercice 3.

Ecrire une classe *Ville* qui contiendra les attributs : *nom*, *population* et *pays*. La population doit être un nombre positif sinon une exception (à créer) de type *PopulationException* doit être lancée.

Ecrire un programme pour tester l'exercice.

### Exercice 4.

Soit la classe *AdresseIP* :

```
public class AdresseIP {
    private int[] octets;
```

```
private String classe;  
  
public AdresseIP (int o1,int o2, int o3, int o4) {  
    octets = new int[] {o1,o2,o3,o4};  
}  
//....  
  
public String toString() {  
    return octets[0] + "." + octets[1] + "." + octets[2] + "."+  
    octets[3] ;  
}  
}
```

Créer une exception *AdresseIPException* pour éviter que l'on fournisse des nombres non valides  
Ecrire la méthode *main ()* pour tester l'exception.