



المدرسة العليا
للتكنولوجيا - الصويرة
L'ÉCOLE SUPÉRIEURE DE
TECHNOLOGIE - ESSAQUIRA



Travaux pratique 1 :

Gestion des exceptions

Préparer par :

Elhamidi Maroua

Encadrer par :

Karami Fahd

Année universitaire 2019/2020

Introduction :

Les exceptions servent à gérer les erreurs qui peuvent arriver dans un programme.

- Connexion à une base de données échouée.
- Erreur de programmation : une méthode appliquée sur une référence nulle.
- Pas de gestion automatique d'exceptions : obligation de débrouiller avec code d'erreur.

EX1 : (Déclenchement et traitement d'une exception) :

```
1 package s6;
2
3 public class EntNat {
4
5     private int n;
6     public EntNat(int n) throws Errconst {
7         if(n<0) throw new Errconst();
8         this.n=n;
9     }
10    public int getN() {
11        return n;
12    }
13    public static void main(String[] args) {
14        // TODO Auto-generated method stub
15        try {
16            EntNat e= new EntNat(-4);
17            System.out.println("votre entier ==>" +e.getN());
18        } catch (Errconst e) {}
19        // TODO: handle exception
20        System.out.println("erreur de construction");
21    }
22 }
23
24 class Errconst extends Exception{}
```

Problems @ Javadoc Declaration Console

<terminated> EntNat [Java Application] C:\Program Files\Java\jdk1.8.0_231\bin\javaw.exe (3 févr
erreur de construction

EX2: (Transmission d'information au gestionnaire):

Les exceptions de type dérivés ErrEnt

Programme de test :

```
public static void main(String[] args){
    // TODO Auto-generated method stub

    try {

        test();

    } catch (ErrEnt e) {
        // TODO: handle exception
    }
}
```

La définition de la classe ErrNat

```
package s6;

public class EntNat {

    private int n;
    public EntNat(int n) throws Errconst {
        if(n<0) throw new Errconst();
        this.n=n;
    }

    public int getN() {
        return n;
    }
    public int Diff(int dn) throws ErrDiff{
        if((this.n-dn)<0) throw new ErrDiff();
        return this.n-dn;
    }
    public int Som(int sn) throws ErrSom{
        if((this.n+sn)<0) throw new ErrSom();
        return this.n+sn;
    }
    public int Prod(int pn) throws ErrProd{
        if((this.n*pn)<0) throw new ErrProd();
        return this.n*pn;
    }
    public static void test() throws Errconst,ErrDiff,ErrSom,ErrProd{

        EntNat e= new EntNat(4);
        System.out.println(" construction==>" +e.getN());
        System.out.println("produit ==>" +e.Prod(6));
        System.out.println("somme ==>" +e.Som(6));
        System.out.println("soustraction ==>" +e.Diff(6));
    }
}
```

La création des classes dérivés :

```
class ErrEnt extends Exception{}
class Errconst extends ErrEnt{
    public Errconst() {
        System.err.println("erreur de construction");
    }
}
class ErrSom extends ErrEnt{
    public ErrSom() {
        System.err.println("erreur de la somme");
    }
}
class ErrDiff extends ErrEnt{
    public ErrDiff() {
        System.err.println("erreur de soustraction");
    }
}
class ErrProd extends ErrEnt{
    public ErrProd() {
        System.err.println("erreur de multiplication");
    }
}
```

```
construction==>4
erreur de soustraction
produit ==>24
somme ==>10
```

La nature de l'exception en affichant leurs informations

```
public static void main(String[] args) throws Errconst {  
    // TODO Auto-generated method stub  
    EntNat e= new EntNat(6);  
    System.out.println("le construction "+e.getN());  
  
    try {  
        System.out.println("produit"+e.Prod(-3));    }  
    catch (ErrProd p) {  
        // TODO: handle exception  
        System.err.println("erreur de produit");  
    }  
  
    try {  
        System.out.println("somme"+e.Som(-8));    }  
    catch (ErrSom s) {  
        // TODO: handle exception  
        System.err.println("erreur de somme");  
    }  
    try {  
        System.out.println("soustraction"+e.Diff(9));  
    }  
    catch (ErrDiff d) {  
        // TODO: handle exception  
        System.err.println("erreur de soustraction");  
    }  
}
```

```
le construction 6  
erreur de produit  
erreur de somme  
erreur de soustraction
```

Conclusion :

Les exceptions servent à gérer les erreurs qui peuvent arriver dans un programme, pour un code à la fois plus élégant, plus propre et plus maintenable.

