

Rapport de Travaux Dirigés/Pratiques : Les exceptions

Ingénierie des Systèmes Informatiques et Logiciels (ISIL)

Module : Java Avancée & JEE

Réalisé par : LAKHL Rachid

Encadré par : Mr. KARAMI Fahd

Année Universitaire : 2018-2019

EX1 : (Déclenchement et traitement d'une exception)

- La classe EntNat et ErrConst :

```
2  package jee01;
3
4  class EntNat {
5      private int nbr;
6      public EntNat(int nbr) throws ErrConst {
7          if(nbr<0) throw new ErrConst();
8          this.nbr =nbr;
9      }
10     public int getN(){
11         return this.nbr;
12     }
13 }
14 class ErrConst extends Exception{
15 }
16 class Test {
17     public static void main (String [] args) {
18         try {
19             EntNat nbr1= new EntNat(7);
20             EntNat nbr2= new EntNat(-1);
21         } catch (ErrConst e) {
22             System.err.println("La valeur est negative");
23         }
24     }
25 }
```

Output - JEE01 (run)

```
> run:
La valeur est negative
BUILD SUCCESSFUL (total time: 0 seconds)
```

EX2: (Transmission d'information au gestionnaire)

```
3  class EntNat {
4      private int nbr;
5  [- public EntNat(int nbr) throws ErrConst {
6      |     if(nbr<0) throw new ErrConst(nbr);
7      |     this.nbr =nbr;
8      | }
9  [- public int getN(){ return this.nbr; }
10 }
11 class ErrConst extends Exception{
12     int a;
13 [- public ErrConst(int a){
14     |     this.a=a;
15     | }
16 [- public int getNum(){ return this.a; }
17 }
18 class Test {
19 [- public static void main (String [] args) {
20     |     try {
21     |         EntNat nbr1= new EntNat(7);
22     |         EntNat nbr2= new EntNat(-1);
23     |     } catch (ErrConst e) {
24     |         System.err.println("La valeur"+e.getNum()+" est negative");
25     |     }
26     | }
27 }
```

Output - JEE01 (run)

```
run:
La valeur-1 est negative
BUILD SUCCESSFUL (total time: 0 seconds)
```

Problème : (Synthèse du chapitre)

- La classe ErrConst, ErrSom, ErrDiff and ErrProd

```
9      class ErrConst extends ErrNat {
10          int nbr;
11      ErrConst(int n3) {
12          this.nbr = nbr;
13      }
14      public int getNum(){return this.nbr;}
15  }
16  class ErrSom extends ErrNat {
17      int a,b;
18      ErrSom(int a, int b){
19          this.a=a; this.b=b;
20      }
21  }
22  class ErrDiff extends ErrNat {
23
24      int a,b;
25      ErrDiff(int a, int b){
26          this.a=a; this.b=b;
27      }
28  }
29  class ErrProd extends ErrNat{
30      int a,b;
31      ErrProd(int a, int b){
32          this.a=a; this.b=b;
33      }
34  }
```

- La classe EntNat

```
5 public class EntNat {
6     private int nbr;
7     public EntNat(int nbr) throws ErrConst{
8         if (nbr<0) throw new ErrConst(nbr);
9         this.nbr=nbr;
10    }
11    public static EntNat somme(EntNat N1, EntNat N2) throws ErrConst,ErrSom{
12        int n1=N1.nbr;
13        int n2=N2.nbr;
14        long som=n1+n2;
15        System.out.println(som);
16        if (som >Integer.MAX_VALUE) throw new ErrSom(n1,n2);
17        return new EntNat((int)som);
18    }
19    public static EntNat diff(EntNat N1, EntNat N2) throws ErrDiff, ErrConst{
20        int n1=N1.nbr;
21        int n2=N2.nbr;
22        int dif=n1-n2;
23        if (dif <0) throw new ErrDiff(n1,n2);
24        return new EntNat(dif);
25    }
26    public static EntNat produit(EntNat N1, EntNat N2) throws ErrProd, ErrConst{
27        int n1=N1.nbr;
28        int n2=N2.nbr;
29        long pro=(long)n1*(long)n2;
30        if (pro >Integer.MAX_VALUE) throw new ErrProd(n1,n2);
31        return new EntNat((int)pro);
32    }
33    public int getN() {return this.nbr;}
34 }
```

▪ L'implémentation :

```

36 class Test {
37     public static void main (String[] args){
38         System.out.println("Le nombre naturel le plus grand est:  "+Integer.MAX_VA
39         try{
40             EntNat n1,n2,d;
41             n1=new EntNat(13);
42             n2=new EntNat(22);
43             d= EntNat.diff(n2, n1);
44         }catch (ErrNat e){
45             System.err.println("Erreur d'entier");
46         }
47         try{
48             EntNat n1,n2,res;
49             n1=new EntNat(992111999);
50             n2=new EntNat(941888799);
51             System.out.println("Nombre 1= "+n1.getN());
52             System.out.println("Nombre 2= "+n2.getN());
53
54             res= EntNat.somme(n1,n2);
55             System.out.println("La somme = "+res.getN());
56             res= EntNat.diff(n1,n2);
57             System.out.println("La diff = "+res.getN());
58             res= EntNat.produit(n1,n2);
59             System.out.println("Le produit = "+res.getN());
60         }
61         catch (ErrConst e){
62             System.err.println("Erreur : le construction du nombre"+e.getNum());
63         }
64
65         catch (ErrSom e){
66             System.err.println("Erreur : la somme des valeurs  "+e.a+" et "+e.b);
67         }
68         catch (ErrDiff e){
69             System.err.println("Erreur : la difference des valeurs  "+e.a+" et "+e.b);
70         }
71         catch (ErrProd e){
72             System.err.println("Erreur : le produit des valeurs  "+e.a+" et "+e.b);
73         }
74     }
75 }

```

Output - JEE01 (run)

```

run:
Le nombre naturel le plus grand est:  2147483647
Nombre 1= 992111999
Nombre 2= 941888799
1934000798
La somme = 1934000798
La diff = 50223200
Erreur : le produit des valeurs  992111999 et 941888799
BUILD SUCCESSFUL (total time: 0 seconds)

```