Rapport

Tp1

Exceptions

Réalisé par : Meriem AFTIMI

Encadré par : Fahd KARAMI

Année :2019 /2020

EX1 : (Déclenchement et traitement d'une exception)

```
package ex1;

public class EntNat {
int entier;
public EntNat(int n )throws ErrConst{
    if (n<0) throw new ErrConst();
    this.entier=n;
}
public int getN() {
    return this.entier;
}
}
```

Figure 1:Classe EntNat

La classe EntNat permettant de manipuler des entiers naturels (positifs ou nuls), cette classe disposera d'un constructeur à un argument de type int qui générera une exception de type ErrConst lorsque la valeur est négative, est une méthode getN qui retourne la valeur entrée.

```
package ex1;
public class ErrConst extends Exception{     }
```

Figure 2: classe ErrConst

La classe Main qui permet de tester les deux classes, on utilisons un bloc try catch de la classe ErrConst .

```
package ex1;
public class Main {
public static void main (String [] args) {
    try {
        EntNat n1= new EntNat(19);

    System.err.println("La valeur d'entier est :  "+n1.getN());
    EntNat n2= new EntNat(-8);
    System.err.println("La valeur d'entier est : "+n2.getN());

} catch (ErrConst ex) {
    System.err.println("La valeur entree est negative");
    }
    }
}
```

Figure 3: La classe Main

La resultat du code

```
<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (5 févr. 2020 à 21:05:12)
La valeur d'entier est :  19
La valeur entree est negative
```

Figure 4:le test

EX2 (Transmission d'information au gestionnaire) :

La classe EntNat de l'exercice 1 est adaptée de manière à disposer dans le gestionnaire d'exception du type ErrConst de la valeur fournie à tort au constructeur.

```java
package ex2;
public class EntNat {
int N;
  public EntNat(int a) throws ErrConst{
    if(a<0)throw new ErrConst(a);
    this.N=a;
 }
   public int getN(){
    return this.N;
   }
}
```

Figure 5 :la classe EntNat

```java
package ex2;

public class ErrConst extends Exception{
    int n;
    public ErrConst(int n){
    this.n=n;
    }
public int getNum(){return n;
}
}
```

Figure 6 la classe ErrConst

```java
public class main {
    public static void main (String []  args) {
        try {
            EntNat n1= new EntNat(19);
            System.err.println("la valeur est : "+n1.getN());
            EntNat n2= new EntNat(-5);
            System.err.println("La valeur est : "+n2.getN());
            } catch (ErrConst ex) {
                System.err.println("La valeur entree est negative "+ex.getNum());
            }
        }
    }
```

Figure 7: la classe Main

```
<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (5 févr. 2020 à 21:42:46)
la valeur est : 19
La valeur entree est negative -5
```

Figure 8: test

Problème:(Synthèse du chapitre)

La classe EntNat permettant de manipuler des entiers naturels (positifs ou nul) disposant de :

- Un constructeur à un argument de type int;il générera une exception ErrConst si la valeur de son argument est négative.
- Méthodes statiquesde somme, de différence et de produit de deux naturels ; elles généreront respectivement des exceptions ErrSom, ErrDiff et ErrProd lorsque le résultat ne sera pas représentable ; la limite des valeurs des naturels sera fixée à la plus grande valeur du type int.
- Une méthode d'accès getN fournissant sous forme d'un int la valeur de l'entier naturel.

```java
package ex3;
public class EntNat {
    private int n;
    public EntNat(int nbr) throws ErrConst{
        if (nbr<0) throw new ErrConst(nbr);
        this.n=nbr;
        }
    public static EntNat somme(EntNat N1, EntNat N2) throws ErrConst,ErrSom{
        int n1=N1.n;
        int n2=N2.n;
        long  som=n1+n2;
        if (som >Integer.MAX_VALUE) throw new ErrSom(n1,n2);
        return new EntNat((int)som);
        }
    public static EntNat diff(EntNat N1, EntNat N2) throws ErrDiff, ErrConst{
        int n1=N1.n;
        int n2=N2.n;
        int dif=n1-n2;
        if (dif <0) throw new ErrDiff(n1,n2);
        return new EntNat(dif);
}
public static EntNat produit(EntNat N1, EntNat N2) throws ErrProd, ErrConst{
    int n1=N1.n;
    int n2=N2.n;
    long pro=(long)n1*(long)n2;
    if (pro >Integer.MAX_VALUE) throw new ErrProd(n1,n2);
return new EntNat((int)pro);
}
public int getN()   {
    return n;
}
}
```

Figure 9: classe entNat

```java
package ex3;
public class ErrConst extends ErrNat {
    int n;
    ErrConst(int n3) {
    this.n=n;
    }
public int getNum(){return n;}
}
```

Figure 10 : classe Errconst

```java
package ex3;
class ErrProd extends ErrNat{
    int n1,n2;
ErrProd(int a, int b){
    this.n1=a;
    this.n2=b;
    }
}
```

Figure 11 : classe errProd

```java
package ex3;

public class ErrDiff extends ErrNat {
    int n1,n2;
      public ErrDiff(int a, int b){
    this.n1=a;
    this.n2=b;
    }
```

Figure 12 :classe Errdif

```java
package ex3;

public class ErrSom extends ErrNat {
    int n1,n2;
    ErrSom(int a, int b){
        this.n1=a;
        this.n2=b;
        }
```

Figure 13 :classe ErrSom

```java
package ex3;

public class ErrNat extends Exception{  }
```

Figure 14 : classe ErrNat

```
public class main {
    public static void main (String[] args){
         System.out.println("Le plus grand nombre naturel=  "+Integer.MAX_VALUE);
        try{
            EntNat nbr;
            EntNat nbr2;
            nbr=new EntNat(9);
            nbr2=new EntNat(2);
            nbr2= EntNat.diff(nbr2, nbr);
            }catch(ErrNat e){
                 System.out.println("");
                System.err.println(" * Erreur d'entier");
                }
        try{
             System.out.println("**********************************");
                    EntNat nbr;
                    EntNat nbr2;
                    nbr=new EntNat(994567899);
                    nbr2=new EntNat(956578599);
                    EntNat res;
                     System.out.println(" ==> Nombre 1= "+nbr.getN());
                    System.out.println(" ==> Nombre 2= "+nbr2.getN());
                    res= EntNat.somme(nbr,nbr2);
                    System.out.println(" ==> La somme = "+res.getN());
                    res= EntNat.diff(nbr,nbr2);
                    System.out.println(" ==> La diff = "+res.getN());
                    res= EntNat.produit(nbr,nbr2);
                    System.out.println(" ==> Le produit = "+res.getN());

                    }
                catch (ErrConst e){
                        System.err.println("Erreur de construction du nombre"+e.getNum());
                        }
                catch (ErrSom e){
    System.err.println("** Erreur de somme des valeurs  "+e.n1+" et "+e.n2);
    }
catch (ErrDiff e){
    System.err.println("** Erreur de difference des valeurs  "+e.n1+" et "+e.n2);
    }
catch (ErrProd e){
    System.err.println("** Erreur de produit des valeurs   "+e.n1+" et "+e.n2);
    }}}
```

Figure 15 :Classe Main

```
<terminated> main (1) [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.e
Le plus grand nombre naturel=  2147483647

************************************
 ==> Nombre 1= 994567899
 * Erreur d'entier
 ==> Nombre 2= 956578599
 ==> La somme = 1951146498
 ==> La diff = 37989300
** Erreur de produit des valeurs   994567899 et 956578599
```

Figure 16: resultat