**Les exceptions en Java**

**Exercices**

# But

Mettre en pratique l’utilisation et la création d’exceptions en Java.

Intégrer les exceptions dans les tests unitaires.

# Travail à affectuer

## Les entiers naturels

L’ensemble des entiers naturels, noté ℕ, contient tous les nombres entiers positifs. Pour cet exercice, le nombre zéro fait partie de cet ensemble.

Vous devez créer une classe, nommée EntierNaturel, qui mémorise un nombre entier de l’ensemble ℕ.

Cette classe doit disposer :

* D’un constructeur à un argument de type int, qui générera une exception de type EntierNaturelValeurException si la valeur de son argument est négative.
* D’un accesseur à l’entier naturel.
* D’une implémentation de la méthode toString().
* De méthodes statiques permettant de calculer la somme, la différence et le produit de deux entiers naturels. Ces méthodes générerons des exceptions EntierNaturelSommeException (respectivement EntierNaturelDiffException et EntierNaturelProduitException) lorsque le résultat ne sera pas représentable (la limite des valeurs des naturels est fixée à la plus grande valeur représentable par le type int).

Il faudra réaliser une hiérarchie cohérente d’exceptions, afin que les problèmes liés à la somme, à la différence et au produit contiennent les deux opérandes à la source du problème.

Écrire également une suite de tests pour la classe EntierNaturel, afin de garantir un comportement correct en toutes situations.

Enfin, écrire une petite application utilisant la classe EntierNaturel dans les deux cas suivants :

* En premier lieu, en interceptant sans discernement les exceptions concernant un entier naturel ;
* En deuxième lieu, en explicitant la nature de l’exception et en affichant les informations disponibles.

Pour cet exercice, toutes les exceptions générées par EntierNaturel sont explicites.

La hiérarchie suivante d’exceptions est mise en place :



Package application :

**Application.java**

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package application;

import entiers\_naturels.EntierNaturel;

import entiers\_naturels.EntierNaturelDiffException;

import entiers\_naturels.EntierNaturelException;

import entiers\_naturels.EntierNaturelOperationException;

import entiers\_naturels.EntierNaturelValeurException;

import java.util.InputMismatchException;

import java.util.Scanner;

/\*\*

 \* Classe qui représente l'application principale.

 \* @author conujer

 \*/

public class Application {

 /\*\*

 \* Point d'entrée du programme.

 \* @param args Arguments de la ligne de commande.

 \*/

 public static void main(String[] args) {

 // Teste la classe EntierNaturel : l'utilisateur doit saisir deux nombres.

 int opérande1 = saisirEntier("Saisir le premier entier naturel : ");

 int opérande2 = saisirEntier("Saisie le deuxième entier naturel : ");

 try {

 EntierNaturel entierN1 = new EntierNaturel(opérande1);

 EntierNaturel entierN2 = new EntierNaturel(opérande2);

 System.out.println("Somme : " + EntierNaturel.somme(entierN1, entierN2));

 System.out.println("Différence : " + EntierNaturel.diff(entierN1, entierN2));

 System.out.println("Produit : " + EntierNaturel.produit(entierN1, entierN2));

 } catch (EntierNaturelException ex) {

 ex.printStackTrace(System.out);

 }

 }

 /\*\*

 \* Saisi un nombre entier au clavier.

 \* La saisie est répétée autant que nécessaire.

 \* @return le nombre entier saisi.

 \*/

 static int saisirEntier(String message) {

 Scanner scanner = new Scanner(System.in);

 int nombreSaisi = 0;

 while (true) {

 System.out.print(message);

 try {

 nombreSaisi = scanner.nextInt();

 break;

 } catch (InputMismatchException e) {

 System.out.println("Saisie invalide !");

 scanner.next(); // Il faut ignorer la saisie invalide

 }

 }

 return nombreSaisi;

 }

}

Package entiers\_naturels :

**EntierNaturel.java**

package entiers\_naturels;

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

/\*\*

 \* Classe qui représente un entier de l'ensemble des nombres naturels.

 \* @author conujer

 \*/

public class EntierNaturel {

 private int nombre;

 /\*\*

 \* Construit un entier naturel.

 \* @param nombre Entier à considérer comme entier naturel.

 \* @throws EntierNaturelValeurException

 \*/

 public EntierNaturel(int nombre) throws EntierNaturelValeurException {

 if (nombre < 0) {

 throw new EntierNaturelValeurException(nombre);

 } else {

 this.nombre = nombre;

 }

 }

 /\*\*

 \* @return l'entier naturel sous forme d'une chaîne.

 \*/

 @Override

 public String toString() {

 return String.valueOf(nombre);

 }

 /\*\*

 \* Méthode qui calcule la somme des deux entiers naturels passés en paramètre.

 \* @param opérande1 Opérande 1.

 \* @param opérande2 Opérande 2.

 \* @return la somme sous forme d'un entier naturel.

 \* @throws EntierNaturelValeurException

 \* @throws EntierNaturelSommeException

 \*/

 public static EntierNaturel somme(EntierNaturel opérande1, EntierNaturel opérande2) throws EntierNaturelValeurException, EntierNaturelSommeException {

 long somme = opérande1.nombre + opérande2.nombre;

 if (somme > Integer.MAX\_VALUE)

 throw new EntierNaturelSommeException(opérande1, opérande2);

 return new EntierNaturel((int)somme);

 }

 /\*\*

 \* Méthode qui calcule la différence des deux entiers naturels passés en paramètre.

 \* @param opérande1 Opérande 1.

 \* @param opérande2 Opérande 2.

 \* @return la différence sous forme d'un entier naturel.

 \* @throws EntierNaturelValeurException

 \* @throws EntierNaturelDiffException

 \*/

 public static EntierNaturel diff(EntierNaturel opérande1, EntierNaturel opérande2) throws EntierNaturelValeurException, EntierNaturelDiffException {

 int diff = opérande1.nombre - opérande2.nombre;

 if (diff < 0)

 throw new EntierNaturelDiffException(opérande1, opérande2);

 return new EntierNaturel(diff);

 }

 /\*\*

 \* Méthode qui calcule le produit des deux entiers naturels passés en paramètre.

 \* @param opérande1 Opérande 1.

 \* @param opérande2 Opérande 2.

 \* @return le produit sous forme d'un entier naturel.

 \* @throws EntierNaturelValeurException

 \* @throws EntierNaturelProduitException

 \*/

 public static EntierNaturel produit(EntierNaturel opérande1, EntierNaturel opérande2) throws EntierNaturelValeurException, EntierNaturelProduitException {

 long produit = opérande1.nombre \* opérande2.nombre;

 if (produit > Integer.MAX\_VALUE)

 throw new EntierNaturelProduitException(opérande1, opérande2);

 return new EntierNaturel((int)produit);

 }

}

**EntierNaturelException.java**

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package entiers\_naturels;

/\*\*

 \* Classe représentant un problème concernant un entier naturel.

 \* @author conujer

 \*/

public class EntierNaturelException extends Exception {

 /\*\*

 \* Constructeur vide.

 \*/

 public EntierNaturelException() {

 }

 /\*\*

 \* Constructeur avec message.

 \* @param message Message d'erreur.

 \*/

 public EntierNaturelException(String message) {

 super(message);

 }

}

**EntierNaturelValeurException.java**

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package entiers\_naturels;

/\*\*

 \* Classe représentant un problème d'appartenance au domaine des entiers naturels.

 \* @author conujer

 \*/

public class EntierNaturelValeurException extends EntierNaturelException {

 private int nombre;

 /\*\*

 \* Constructeur.

 \* @param nombre Nombre censé être un entier naturel.

 \*/

 public EntierNaturelValeurException(int nombre) {

 super("Number " + nombre + " invalid in N domain.");

 this.nombre = nombre;

 }

}

**EntierNaturelOperationException.java**

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package entiers\_naturels;

/\*\*

 \* Classe représentant un problème d'opération entre deux entiers naturels.

 \* @author conujer

 \*/

public class EntierNaturelOperationException extends EntierNaturelException {

 // En pratique, on pourrait admettre que ces membres soient publics afin

 // d'éviter de passer par un getter.

 private final EntierNaturel opérande1;

 private final EntierNaturel opérande2;

 /\*\*

 \* Constructeur.

 \* @param opérande1 Opérande 1.

 \* @param opérande2 Opérande 2.

 \*/

 public EntierNaturelOperationException(EntierNaturel opérande1, EntierNaturel opérande2) {

 super(String.format("Operation error with operands %s and %s.", opérande1, opérande2));

 this.opérande1 = opérande1;

 this.opérande2 = opérande2;

 }

 /\*\*

 \* @return le premier opérande.

 \*/

 public EntierNaturel getOpérande1() {

 return opérande1;

 }

 /\*\*

 \* @return le deuxième opérande.

 \*/

 public EntierNaturel getOpérande2() {

 return opérande2;

 }

}

**EntierNaturelSommeException.java**

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package entiers\_naturels;

/\*\*

 \* Classe représentant un problème d'addition de deux entiers naturels.

 \* @author conujer

 \*/

public class EntierNaturelSommeException extends EntierNaturelOperationException {

 /\*\*

 \* Constructeur.

 \* @param opérande1 Opérande 1.

 \* @param opérande2 Opérande 2.

 \*/

 public EntierNaturelSommeException(EntierNaturel opérande1, EntierNaturel opérande2) {

 super(opérande1, opérande2);

 }

}

**EntierNaturelDiffException.java**

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package entiers\_naturels;

/\*\*

 \* Classe représentant un problème de soustraction entre deux entiers naturels.

 \* @author conujer

 \*/

public class EntierNaturelDiffException extends EntierNaturelOperationException {

 /\*\*

 \* Constructeur.

 \* @param opérande1 Opérande 1.

 \* @param opérande2 Opérande 2.

 \*/

 public EntierNaturelDiffException(EntierNaturel opérande1, EntierNaturel opérande2) {

 super(opérande1, opérande2);

 }

}

**EntierNaturelProduitException.java**

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package entiers\_naturels;

/\*\*

 \* Classe représentant un problème de multiplication de deux entiers naturels.

 \* @author conujer

 \*/

public class EntierNaturelProduitException extends EntierNaturelOperationException {

 /\*\*

 \* Constructeur.

 \* @param opérande1 Opérande 1.

 \* @param opérande2 Opérande 2.

 \*/

 public EntierNaturelProduitException(EntierNaturel opérande1, EntierNaturel opérande2) {

 super(opérande1, opérande2);

 }

}