

# Exceptions

## Exercice 1 : Comprendre les exceptions

Dans cet exercice, on considère le programme donné au listing 1 qui compile sans erreur. Les identifiants sont volontairement non significatifs.

Listing 1 – Les exceptions en Java

```
1  class ExempleException {
2      private void m2(String p) {
3          System.out.print("<");
4          if (p == null) {
5              throw new NullPointerException();
6          }
7          if (p.length() == 0) {
8              throw new IllegalArgumentException("Chaîne vide");
9          }
10         System.out.print(p.charAt(0));
11         System.out.print(p.charAt(1));
12         System.out.print(">");
13     }
14
15     public void m1(String p) {
16         System.out.print("[");
17         try {
18             System.out.print("(");
19             m2(p);
20             System.out.print(")");
21         } catch (NullPointerException e) {
22             System.out.print("N");
23         } catch (IllegalArgumentException e) {
24             System.out.print("I");
25         } finally {
26             System.out.print("F");
27         }
28         System.out.print("]");
29     }
30
31     class ClassePrincipale {
32         public static void main(String[] args) {
33             String argument = (args.length == 0) ? null : args[0];
34             new ExempleException().m1(argument);
35             System.out.println(".");
36         }
37     }
38 }
```

**1.1.** Indiquer ce qu'affiche l'exécution des commandes suivantes :

```
1  java ClassePrincipale un
2  java ClassePrincipale ""
3  java ClassePrincipale x
```

**1.2.** L'ordre des **catch** est-il important ?

**1.3.** Les exceptions `IllegalArgumentException` et `NullPointerException` sont-elles vérifiées ?  
En particulier, on indiquera ce qui fait qu'une exception est vérifiée en Java, qui vérifie et l'intérêt de cette notion.

## Exercice 2 : Somme des arguments de la ligne de commande

La classe Somme (listing 2) affiche la somme des nombres réels donnés en argument de la ligne de commande. Par exemple, « java Somme 10 15.5 4 » affiche 29.5.

Listing 2 – La classe Somme

```
1  /** Calculer la somme des paramètres de la ligne de commande. */
2  public class Somme {
3
4      /* Afficher la somme des arguments de la ligne de commande */
5      public static void main(String[] args) {
6          double somme = 0;
7          for (int i = 0; i < args.length; i++) {
8              somme += Double.parseDouble(args[i]);
9          }
10         System.out.println(somme);
11     }
12
13 }
```

### 2.1. L'exécution de java Somme 10 x 3 affiche ce qui suit dans le terminal :

```
Exception in thread "main" java.lang.NumberFormatException: For input string: "x"
    at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2054)
    at java.base/jdk.internal.math.FloatingDecimal.parseDouble(FloatingDecimal.java:110)
    at java.base/java.lang.Double.parseDouble(Double.java:543)
    at Somme.main(Somme.java:8)
```

Expliquer comment interpréter cet affichage pour comprendre ce qu'il s'est passé.

### 2.2. Modifier la classe Somme pour afficher la somme de tous les arguments de la ligne de commande en ignorant ceux qui ne sont pas réels. On indiquera le nombre de données ignorées.

```
> java Somme 10 x 3 y
13.0
Nombre de données ignorées : 2
> java Somme 10 3
13.0
```

## Exercice 3 : Livret A

Un Livret A est un produit bancaire dont le titulaire peut faire des opérations de dépôt dans la limite d'un plafond et des retraits dont le montant ne peut pas dépasser le solde du livret. Les sommes versées sur un Livret A donnent lieu à rémunération. Le taux d'intérêt ainsi que le plafond du Livret A sont identiques pour tous les Livrets A. Au 12 mars 2015, le taux d'intérêt est fixé à 1% et le plafond à 22 950 euros.

Tout comme pour les comptes courants, un historique des opérations est géré par la banque. Il est ainsi possible d'accéder aux dernières opérations de crédit ou de débit et d'éditer un relevé.

### 3.1. Compléter le diagramme de classes UML du système pour faire apparaître les Livrets A.

### 3.2. Écrire la classe LivretA.

### 3.3. Proposer un programme de test de la classe LivretA.