

# 7. Les exceptions

# 1. Présentation

Les exceptions représentent le mécanisme de **gestion** des **erreurs** intégré au langage Java.

Il se compose d'objets représentant les erreurs et d'un ensemble de **trois mots clés** qui permettent de détecter et de traiter ces erreurs (**try**, **catch** et **finally** ) mais aussi de les lever ou les propager (throw et throws).

Lors de la **détection** d'une erreur, un objet qui **hérite** de la classe **Exception** est créé et **propagé** à travers la pile d'exécution jusqu'à ce qu'il soit **traité**.

Ces mécanismes permettent de **renforcer** la **sécurité** du code Java.

```
public class TestException {  
    public static void main(java.lang.String[] args) {  
        int i = 3;  
        int j = 0;  
        System.out.println("résultat = " + (i / j));  
    }  
}
```

```
C:>java TestException
```

```
Exception in thread "main" java.lang.ArithmeticException: /  
by zero
```

```
    at tests.TestException.main(TestException.java:23)
```

## 2. Les mots clés **try**, **catch** et **finally**

- Le bloc **try** rassemble les appels de méthodes **susceptibles** de **produire** des **erreurs** ou des **exceptions**.
- L'instruction **try** est suivie d'instructions entre des accolades.

---

```
try {  
    operation_risquée1;  
    opération_risquée2;  
} catch (ExceptionInteressante e) {  
    traitements  
} catch (ExceptionParticulière e) {  
    traitements  
} catch (Exception e) {  
    traitements  
} finally {  
    traitement_pour_terminer_proprement;  
}
```

---

Il faut faire attention à **l'ordre** des **clauses catch** pour **traiter** en premier les exceptions les **plus précises** (sous-classes) avant les exceptions plus **générales**. Un message d'erreur est émis par le compilateur dans le cas contraire.



```
public class TestException {  
    public static void main(java.lang.String[] args) {  
        // Insert code to start the application here.  
        int i = 3;  
        int j = 0;  
        try {  
            System.out.println("résultat = " + (i / j));  
        } catch (Exception e) {  
        } catch (ArithmeticException e) {  
        }  
    }  
}
```

- Erreur à la **compil** car **Exception** est traité en premier alors que **ArithmeticException** est une **sous-classe** de **Exception**

---

```
C:\tests>javac TestException.java
```

```
TestException.java:11: catch not reached.
```

```
    catch (ArithmeticException e) {  
        ^
```

```
1 error
```

```
public class Main {  
    public static void main(String[ ] args)  
    {  
        int[] Num = {1, 2, 3};  
        System.out.println(Num[4]);  
    }  
}
```

```
public class Main {  
    public static void main(String[ ] args)  
    {  
        try {  
            int[] Num = {1, 2, 3};  
            System.out.println(Num [10]);  
        }  
        catch (Exception e) { System.out.println("Something went wrong.");  
        }  
    }  
}
```

```
public class Main {  
    public static void main(String[] args)  
    {  
        try {  
            int[] Num = {1, 2, 3};  
            System.out.println(Num [10]);  
        }  
        catch (Exception e) {  
            System.out.println("Something went wrong.");  
        }  
        finally {  
            System.out.println("The 'try catch' is finished.");  
        }  
    }  
}
```