



LES EXCEPTIONS

En Java, les **Exceptions** représentent le mécanisme de gestion des erreurs.

Les erreurs concernées proviennent de **l'exécution du code**.



LES EXCEPTIONS

- Exemple d'exception

```
...  
int [ ] tab = new int[10];  
tab[12] = 50;
```

...

L'exécution de ce code entraîne la levée d'une exception et l'arrêt du programme.

Affichage de la console :

```
Exception in thread "main"  
java.lang.ArrayIndexOutOfBoundsException:..
```



LES EXCEPTIONS

- Autre Exemple

```
...  
StringBuilder sb;
```

```
System.out.println(sb.charAt( 0 ));
```

L'exécution de ce code entraine la levée d'une exception et **l'arrêt du programme.**

Affichage de la console :

Exception in thread "main" Java.lang.NullPointerException



LES EXCEPTIONS

- Autre Exemple

```
...  
String s = In.readString();  
double resultat = Double.valueOf(s);
```

Si le contenu de s ne permet pas la conversion en double (Ex: 25,10), l'exécution du code entraîne la levée d'une exception et **l'arrêt du programme**.

Affichage de la console :

Exception in thread "main" Java.lang.NumberFormatException



LES EXCEPTIONS

Pour les opérations considérées comme risquées on peut mettre en place une gestion de l'Exception (**capture**) afin d'éviter l'arrêt brutal du programme.



LES EXCEPTIONS

-Capture d'une exception:

```
String s = In.readString();  
double resultat;
```

dans le bloc try :
Code susceptible de lever
une exception

```
try {  
    resultat = Double.valueOf(s);  
} catch (NumberFormatException ex) {
```

```
    System.err.println("Erreur" + ex.getMessage());
```

```
}
```

dans le bloc catch :
Code exécuté si l'exception
est capturée

.....

Code exécuté après le
traitement par try-catch
le programme n'est pas stoppé



LES EXCEPTIONS

Il est nécessaire de connaître le type d'exception pouvant être levée. (`ArrayIndexOutOfBoundsException` , `NullPointerException`, `ArithmeticException`..)

Toutes les classes d'exception dérivent par héritage de la classe mère `Exception`.

On peut également utiliser cette classe.



LES EXCEPTIONS

-Capture d'une exception de type **Exception**

```
try {  
    val1 = 100;  
    double resultat = val1 / In.readInteger();  
  
} catch (Exception ex) {  
    System.out.println("Erreur " + ex.getMessage());  
}
```




LES EXCEPTIONS

Il est possible de contrôler plusieurs exceptions pour le même code

```
try {  
    //code à risque  
} catch (UneException ex) {  
    System.err.println("Erreur");  
} catch (AutreException ex){  
    System.err.println("Autre Erreur");  
} catch (EncoreUneAutreException ex){  
    System.err.println("Autre Erreur");  
}
```



LES EXCEPTIONS

l'ordre est important les Exceptions "filles" doivent être avant les Exceptions "mères".

```
try {  
    //code à risque  
} catch (ArithmeticException ex) {  
    System.err.println("Erreur" +ex.getMessage());  
} catch (Exception ex){  
    System.err.println("Autre Erreur" +ex.getMessage());  
}
```



LES EXCEPTIONS

Exceptions Contrôlées

Certaines méthodes sont considérées par java comme risquées et rendent **obligatoire** la gestion des Exceptions. Ce sont des **Exceptions contrôlées**

Ce sont en général celles qui entraînent des interactions avec le Système (Fichiers, Matériel, Temporalisation..)



LES EXCEPTIONS

Exceptions Contrôlées

Exemple : Temporisation

Thread.sleep(2000);

Cette méthode risque de lever l'Exception

InterruptedException

C'est une Exception contrôlée, son traitement est obligatoire



LES EXCEPTIONS

Exceptions Contrôlées

Exemple : Temporisation

-L'EDI (Intellij) propose les deux possibilités pour gérer l'exception:

- **la capture (try – catch)**
- **La propagation (throws)**

...



Add exception to method signature



Surround with try/catch



LES EXCEPTIONS

Exceptions Contrôlées Exemple : Temporisation

Capture :

```
try {  
    Thread.sleep(2000);  
} catch (InterruptedException ex) {  
    System.err.println("création erreur tempo");  
    .....  
}
```

Le contenu du bloc catch est à gérer par le développeur



LES EXCEPTIONS

Exceptions Contrôlées Exemple : Temporisation

Propagation

```
public static void temporiser() throws InterruptedException {
```

```
...
```

```
Thread.sleep(2000);
```

L'exception est propagée à la méthode appelante

Si l'exception est levée, l'exécution du code en cours est stoppée, Java renvoie l'exception à la méthode appelante qui doit la gérer.



LES EXCEPTIONS

Exceptions Contrôlées Exemple : Temporisation

Propagation :

```
public static void main (String [ ] args) {  
    try {  
        temporiser();  
    } catch (....  
        .....  
    }
```

L'exception est propagée
à la méthode appelante ,
c'est elle qui doit gérer
l'exception

```
public static void temporiser() throws InterruptedException {  
    ...  
    Thread.sleep(2000);  
    ....  
}
```




LES EXCEPTIONS

