

EXCEPTIONS LOGICIELLES

Walter Rudametkin

Maître de Conférences

Bureau F011

Walter.Rudametkin@polytech-lille.fr

Exceptions

- Evènement qui suspend l'exécution normale (interruption soft)
- Récupérable par programme, à l'exécution
- Objectifs
 - Fiabilité et «tolérance aux pannes»
 - Séparer l'essentiel de l'exceptionnel dans les traitements
- Exceptions pré-programmées : exceptions du langage et de ses bibliothèques
- Programmer ses propres exceptions : exceptions utilisateur

Exemples : langage et bibliothèques

Exception

```
RuntimeException // erreurs du noyau
  ArithmeticException // division par zero
  ClassCastException // echec de downcast
  IndexOutOfBoundsException // i<0 | i>=taille
    ArrayIndexOutOfBoundsException
    StringIndexOutOfBoundsException
  NullPointerException
  SecurityException
IOException // bibliothèque d'e/s
  EOFException
  FileNotFoundException
AWTException // interface graphique
```

Exemple: NullPointerException (NPE)

Exception in thread "main" java.lang.IllegalStateException: A book has a null property

at com.example.myproject.Author.getBookIds(Author.java:38)

at com.example.myproject.Bootstrap.main(Bootstrap.java:14)

Caused by: java.lang.NullPointerException

at com.example.myproject.Book.getId(Book.java:22)

at com.example.myproject.Author.getBookIds(Author.java:35)

... 1 more

Capture d'exception

```
try {
    // code susceptible de générer des exceptions
} catch (ExceptionType1 id) {
    // traitant
} catch (ExceptionType2 id) {
    // traitant
} ...
```

• Mécanisme

Quand une exception survient dans un bloc :

- l'exécution normale est arrêtée
- l'exception est propagée jusqu'au premier bloc capable de la capturer (catch)
- selon l'imbrication lexicale des blocs puis de la pile des messages et éventuellement jusqu'au système si aucune capture n'est assurée (arrêt du programme).

Représentation

- En Java les exceptions sont représentées par des objets
 - décrits par des classes telles que celles ci-dessus
 - Instanciées lorsque l'exception survient.
- L'une des principales méthodes :


```
printStackTrace()
```

 qui affiche la pile des messages qui a conduit à l'exception.
- Par défaut cette méthode est appliquée si l'exception n'est pas capturée (remontée jusqu'au système).

Exception du langage : exemple

```
class Pile {
    int espace[];
    int sommet=-1;
    Pile(int taille) {
        espace = new int[taille];
    }
    void printEtat() {...}
    void empiler(int x) {
        try {
            sommet +=1;
            espace[sommet]=x;
        } catch (ArrayIndexOutOfBoundsException ex) {
            printEtat();
        }
    }
}
```

Exceptions utilisateur

- Sous-classer la classe `Exception`
- Provoquer explicitement l'exception par l'instruction:


```
throw <objet exception>
```
- Spécifier l'exception dans la déclaration de la méthode provocante :


```
<methode> throws
    <ClasseException>[, <ClasseException>]* {...}
```
- Attention : cette déclaration fait partie du profil de la méthode et doit être respectée en cas de redéfinition.

Exception utilisateur : exemple

```

class PilePleineException extends Exception {}

class Pile {
    boolean pleine() {return (sommet==espace.length-1);}
    void vider() {...}
    void empiler(int x) throws PilePleineException {
        if (pleine()) throw new PilePleineException();
        else {sommet +=1;espace[sommet]=x;}}
}

class Client {
    Pile p = new Pile(N);
    void appli(int x) {
        //throws PilePleineException si non traitee
        try {
            p.empiler(x);
        } catch (PilePleineException ex) {p.vider();}}
}

```

Programmation « orientée exceptions »

// Essayer d'abord vs. tester d'abord.

```

class PileVideException extends Exception {}

class Pile {
    int top() throws PileVideException {
        //ESSAYER
        try {return espace[sommet];}
        catch (ArrayIndexOutOfBoundsException ex) {
            throw new PileVideException(); //PROPAGER
        }
    }

    void empiler(int x) throws PilePleineException {
        //ESSAYER
        try {sommet +=1; espace[sommet]=x;}
        catch (ArrayIndexOutOfBoundsException ex) {
            sommet -=1; //REPARER LOCALEMENT
            throw new PilePleineException();//PROPAGER
        }
    }
}

```