

## UF10.- EXCEPCIONS

### - *Teoria* -

PROGRAMACIÓ  
CFGs DAW

Joan V. Cassany

[jv.cassanycoscolla@edu.gva.es](mailto:jv.cassanycoscolla@edu.gva.es)

2022/2023 1

# 1. INTRODUCCIÓ

## TIPUS D'ERRORS



GENERALITAT  
VALENCIANA



En programar es poden produir diferents errors alguns dels quals es produeixen en **temps de compilació** i generalment passa quan hi ha alguna sentència que no està ben escrita, per exemple, quan falta el punt i coma al final de la sentència.

Si un programa dóna errors en temps de compilació, no es crea el fitxer de *bytecode* amb l'extensió *.class*.

```
System.out.println("Hola món!");
```

Hi ha altres errors que es produeixen en **temps d'execució** que anomenarem excepcions: dividir per zero, accedir a una posició d'un array fora de límits, esperar la lectura per consola d'un enter i que s'introdueixin caràcters, etc.

# 1. INTRODUCCIÓ

## CLASSE EXCEPTION



GENERALITAT  
VALENCIANA

Quan ocorre un error en temps d'execució la màquina virtual Java crea un objecte de la classe **Exception**. En llançar una excepció (**Throwing Exception**) s'està notificant aquesta anomalia al sistema d'execució.

```
12 public class Ejemplos_excepciones {
13
14     public static void main(String[] args) {
15         int div, x, y;
16
17         x = 3;
18         y = 0;
19
20         div = x / y;
21
22         System.out.println("El resultado es " + div);
23     }
24
25 }
```

$div = 3 / 0$


ArithmeticException

$num = Integer.parseInt("56s")$

NumberFormatException

$v[-5]$

ArrayIndexOutOfBoundsException



```
run:
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at ejemplos_excepciones.Ejemplos_excepciones.main(Ejemplos_excepciones.java:20)
    C:\Users\user\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

**Nota:** Veure Exemples 1, 2 i 3

# 1. INTRODUCCIÓ

## CLASSE EXCEPTION



GENERALITAT  
VALENCIANA

ceedcv  
CENTRE ESPECÍFIC  
D'EDUCACIÓ A DISTÀNCIA DE  
LA COMUNITAT VALENCIANA

java.awt.print  
java.beans  
java.beans.beancontext  
java.io  
java.lang  
java.lang.annotation  
java.lang.instrument  
java.lang.invoke  
java.lang.management  
java.lang.ref  
java.lang.reflect  
java.math  
java.net  
java.nio  
java.security  
java.sql  
java.time  
java.util  
java.util.concurrent  
java.util.concurrent.atomic  
java.util.concurrent.locks  
java.util.jar  
java.util.logging  
java.util.regex  
java.util.zip  
System  
Thread  
ThreadGroup  
ThreadLocal  
Throwable  
Void  
Enums  
Character.UnicodeScript  
ProcessBuilder.Redirect.Type  
Thread.State  
Exceptions  
ArithmeticException  
ArrayIndexOutOfBoundsException  
ArrayStoreException  
ClassCastException  
ClassNotFoundException  
CloneNotSupportedException  
EnumConstantNotPresentException  
Exception  
IllegalAccessException  
IllegalArgumentException  
IllegalMonitorStateException

Overview Package **Class** Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

java.lang

**Class ArithmeticException**

java.lang.Object  
java.lang.Throwable  
java.lang.Exception  
java.lang.RuntimeException  
java.lang.ArithmeticException

**All Implemented Interfaces:**

Serializable

**public class ArithmeticException**  
extends RuntimeException

Thrown when an exceptional arithmetic condition has occurred. For example, the virtual machine as if suppression were disabled and/or the stack trace was

**Since:**

JDK1.0

**See Also:**

Serialized Form

**Constructor Summary**

API de Java

<https://docs.oracle.com/javase/7/docs/api/>

## 2. LLANÇAR EXCEPCIONS

### *throw*



GENERALITAT  
VALENCIANA

ceedcv  
CENTRE ESPECÍFIC  
D'EDUCACIÓ A DISTÀNCIA DE  
LA COMUNITAT VALENCIANA

Les excepcions eviten que el programa es pare al temps que es pot especificar el tractament a realitzar.

La sentència **throw** permet crear una excepció de forma artificial. Com les excepcions són objectes s'ha d'instanciar.

```
throw new Exception();  
throw new ArithmeticException(); ...
```

```
Exception e = new Exception();  
throw e;
```

Opcionalment es pot incloure un missatge de tipus String.

```
throw new Exception ("El denominador no pot ser 0.");
```

El més recomanable és llançar la excepció genèrica **Exception**.

## 2. LLANÇAR EXCEPCIONS

### CAPÇALERA DEL MÈTODE



GENERALITAT  
VALENCIANA



Es obligatori indicar en la capçalera del mètode que pot llançar excepcions

```
// Setter de l'atribut edat. Ha de ser >= 0
public void setEdat(int edat) throws Exception {
    if (edat < 0)
        throw new Exception("Edat negativa no permesa");
    else
        this.edat = edat;
}
```

Un mètode pot llançar diferents tipus d'excepcions. En tal cas cal especificar tots els tipus possibles en la capçalera, separats per comes.

```
public Persona(String dni, int edat) throws InvalidDniException, InvalidEdatException {
    if (!dni.matches("[0-9]{8}[A-Z]")) {
        throw new InvalidDniException("DNI no vàlid: " + dni);
    }
    if (edat < 0) {
        throw new InvalidEdatException("Edat no vàlida: " + edat);
    }

    this.dni = dni;
    this.edat = edat;
}
```

## 2. LLANÇAR EXCEPCIONS

### *PILA DE CRIDADES A MÈTODES*



GENERALITAT  
VALENCIANA



Cal tindre en compte que en llançar una excepció es parará l'execució d'aquest mètode (no s'executarà la resta del codi del mètode) i es llançarà l'excepció al mètode que el va cridar.

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$  (seqüència de crides de mètodes)

$A \leftarrow B \leftarrow C \leftarrow D \leftarrow E$  (seqüència de llançament d'excepcions)

Si el mètode E llança una excepció, li arribarà a D, i així fins a arribar al A.

Per tant, com tots aquests mètodes poden acabar llançant una excepció, en les seues capçaleres caldrà incloure el **throws Exception**.

### 3. MANEJADORS D' EXCEPCIONS

*try, catch, final*



GENERALITAT  
VALENCIANA

ceedcv  
CENTRE ESPECÍFIC  
D'EDUCACIÓ A DISTÀNCIA DE  
LA COMUNITAT VALENCIANA

Bloc **try**: conté el codi que podria llançar una excepció.

Bloc **catch**: captura les excepcions i determina les accions a prendre.

Bloc **finally**: opcional, s'executarà tant si hi ha excepció com si no hi ha

```
try {  
    // instruccions que poden produir diferents tipus d'Excepcions  
}  
catch (TipusExcepcio1 e1) {  
    // instruccions per a manejar un TipusExcepcio1  
}  
catch (TipusExcepcio2 e2) {  
    // instruccions per a manejar un TipusExcepcio2  
...  
catch (TipusExcepcioN eN) {  
    // instruccions per a manejar un TipusExcepcioN  
}  
finally { // opcional  
    // instruccions que s'executaran tant si hi ha excepció com si no  
}
```

El bloc catch sols capturarà excepcions del tipus indicat. No obstant això capturarà **excepcions heretades** del tipus indicat. El cas més general és *catch (Exception e)* perquè a Java totes les excepcions hereten de Exception.

**Nota:** Veure Exemples 4 i 5



### 3. MANEJADORS D' EXCEPCIONS

#### L'OBJECTE EXCEPTION



GENERALITAT  
VALENCIANA



Tota excepció genera un objecte de la classe **Exception** i per tant pot utilitzar els seus mètodes. Entre els més útils estan:

**getMessage()** → Retorna un String amb un text simple sobre l'error.

**printStackTrace()** → És el que més informació proporciona.

**toString()** → Els objectes de tipus Exception el tenen precarregat

```
...
catch (Exception e){

    // Mostrem el missatge de l'excepció
    System.err.println("Error: " + e.getMessage());

    // Anem mostrar tota la informació, missatge i pila de crides
    e.printStackTrace();

    // Mostrem el missatge de l'excepció
    System.out.println(e);

}
...
```

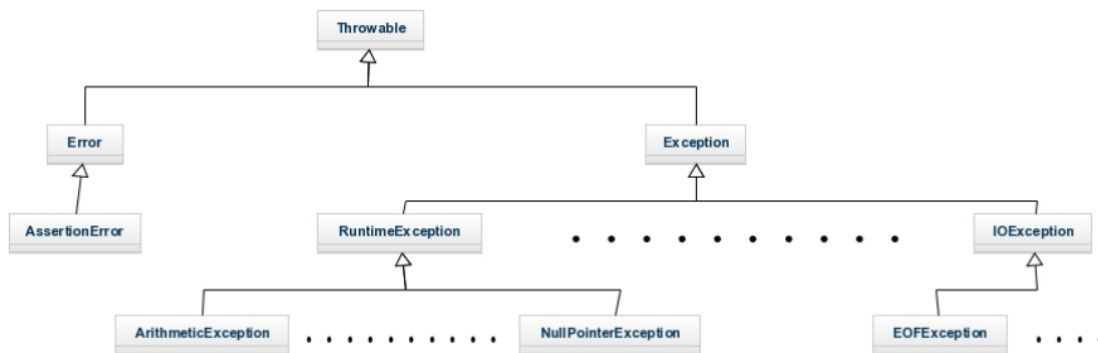
## 4. JERARQUIA DE TIPUS D'EXCEPCIONS



GENERALITAT  
VALENCIANA

La classe **Exception** hereta de **Throwable**, i ahora, totes las excepcions hereten de **Exception**.

Com **java.lang** és importat de forma implícita en tots els programes, la major part de les excepcions derivades de **RuntimeException** estan disponibles de manera automàtica. A més no és necessari incloure-les en cap capçalera de mètode mitjançant **throws**.



Overview Package **Class** Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

java.lang

### Class ArithmeticException

java.lang.Object  
java.lang.Throwable  
java.lang.Exception  
java.lang.RuntimeException  
java.lang.ArithmeticException

## 4. JERARQUIA DE TIPUS D'EXCEPCIONS



GENERALITAT  
VALENCIANA



Les Excepcions poden ser comprovades i no comprovades:

- **Excepcions comprovades:** comprovades durant la compilació.

Excepció	Significat
<i>ClassNotFoundException</i>	No s'ha trobat la classe.
<i>CloneNotSupportedException</i>	Intent de duplicat d'un objecte que no implementa la interfície clonable.
<i>IllegalAccessException</i>	S'ha denegat l'accés a una classe.
<i>InstantiationException</i>	Intent de crear un objecte d'una classe abstracta o interfície.
<i>InterruptedException</i>	Fil interromput per un altre fil.
<i>NoSuchFieldException</i>	El camp sol·licitat no existeix.
<i>NoSuchMethodException</i>	El mètode sol·licitat no existeix.

## 4. JERARQUIA DE TIPUS D'EXCEPCIONS



GENERALITAT  
VALENCIANA

ceedcv  
CENTRE ESPECÍFIC  
D'EDUCACIÓ A DISTÀNCIA DE  
LA COMUNITAT VALENCIANA

- **Excepcions no comprovades:** es produiran durant l'execució del programa.

Excepció	Significat
<i>ArithmeticException</i>	Error aritmètic com a divisió entre zero.
<i>ArrayIndexOutOfBoundsException</i>	Índex de la matriu fora del seu límit.
<i>ArrayStoreException</i>	Assignació a una matriu de tipus incompatible.
<i>ClassCastException</i>	Conversió invàlida.
<i>IllegalArgumentException</i>	Ús invàlid d'un argument en cridar a un mètode.
<i>IllegalMonitorStateException</i>	Operació de monitor invàlida, com esperar un fil no bloquejat.
<i>IllegalStateException</i>	L'entorn o aplicació estan en un estat incorrecte.
<i>IllegalThreadStateException</i>	L'operació sol·licitada és incompatible amb l'estat actual del fil.
<i>IndexOutOfBoundsException</i>	Algun tipus d'índex està fora del seu rang o del seu límit.
<i>NegativeArraySizeException</i>	La matriu té una grandària negativa.
<i>NullPointerException</i>	Ús incorrecte d'una referència NULL.
<i>NumberFormatException</i>	Conversió incorrecta d'una cadena a un format numèric.
<i>SecurityException</i>	Intent de violació de seguretat.
<i>StringIndexOutOfBounds</i>	Intent de sobrepassar el límit d'una cadena.
<i>TypeNotPresentException</i>	Tipus no trobat.
<i>UnsupportedOperationException</i>	Operació no admesa.

## 5. DEFINIR EXCEPCIONS PRÒPIES



GENERALITAT  
VALENCIANA

ceedcv  
CENTRE ESPECÍFIC  
D'EDUCACIÓ A DISTÀNCIA DE  
LA COMUNITAT VALENCIANA

Per a crear excepcions pròpies només cal utilitzar una de les característiques més importants de la programació orientada a objectes: l'herència. Crear una nova excepció serà tan senzill com implementar una **subclasse de l'Exception**.

```
10 public class UF10_Exemple6 {
11     public static void main (String[] args){
12
13         try {
14             metodo (n:1);
15             metodo (n:20);
16         } catch (ExcepcioPropia e) {
17             System.out.println ("Capturada: " + e);
18         }
19     }
20
21     static void metodo (int n) throws ExcepcioPropia {
22
23         System.out.println ("Parametre rebut (" + n + ")");
24
25         if (n>10)
26             throw new ExcepcioPropia(n);
27
28         System.out.println (":Finalitza sense excepcions.");
29     }
30 }
```

```
10 public class ExcepcioPropia extends Exception {
11
12     private int num;
13
14     ExcepcioPropia (int n) {
15         this.num=n;
16     }
17
18     public String toString(){
19         return "Excepció Propia [" + this.num + "];"
20     }
21 }
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ UF10Exercicis ---
Parametre rebut (1)
Finalitza sense excepcions.
Parametre rebut (20)
Capturada: Excepció Propia [20]
```

## RECOMANACIÓ FINAL



GENERALITAT  
VALENCIANA



En termes generals, no s'ha de deixar en mans del control d'excepcions allò que es pot fer mitjançant senzilles comprovacions prèvies.

# EXERCICIS PROPOSATS

