



Métodos Avanzados de Programación Científica y Computación

M^a Luisa Díez Platas

Tema 4. Excepciones

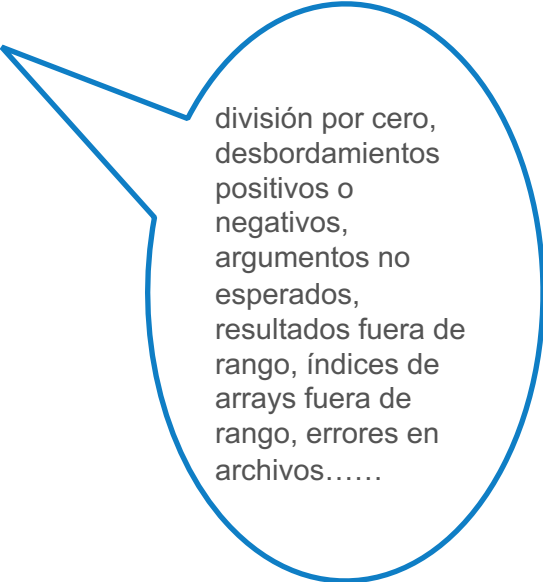
¿Cómo estudiar este tema?

IDEAS CLAVE	LO + RECOMENDADO	+ INFORMACIÓN	TEST
<p>¿Cómo estudiar este tema?</p> <p>Excepciones</p> <p>Captura y gestión de excepciones</p> <p>Lanzamiento de excepciones</p> <p>La clase <code>Throwable</code></p> <p>Creación de excepciones</p>	<p>No dejes de leer...</p> <p>Gestionar y crear excepciones</p> <p>No dejes de ver...</p> <p> Excepciones en Java</p>	<p>A fondo</p> <p>Excepciones de Oracle</p> <p>Excepciones desde un punto de vista menos técnico</p> <p>Programación en Java: Desarrolla Aplicaciones Java</p>	

- Conceptos
- Definición y lanzamiento de excepciones
- Captura y gestión

Excepción

Anomalía o condición de error no esperada que se produce durante la ejecución de un programa. Normalmente las excepciones suspenden la ejecución del programa, emitiendo el sistema un mensaje de error.



división por cero,
desbordamientos
positivos o
negativos,
argumentos no
esperados,
resultados fuera de
rango, índices de
arrays fuera de
rango, errores en
archivos.....

Se tratan excepciones síncronas, es decir, que suceden en un momento predecible.

Conceptos básicos del sistema de excepciones

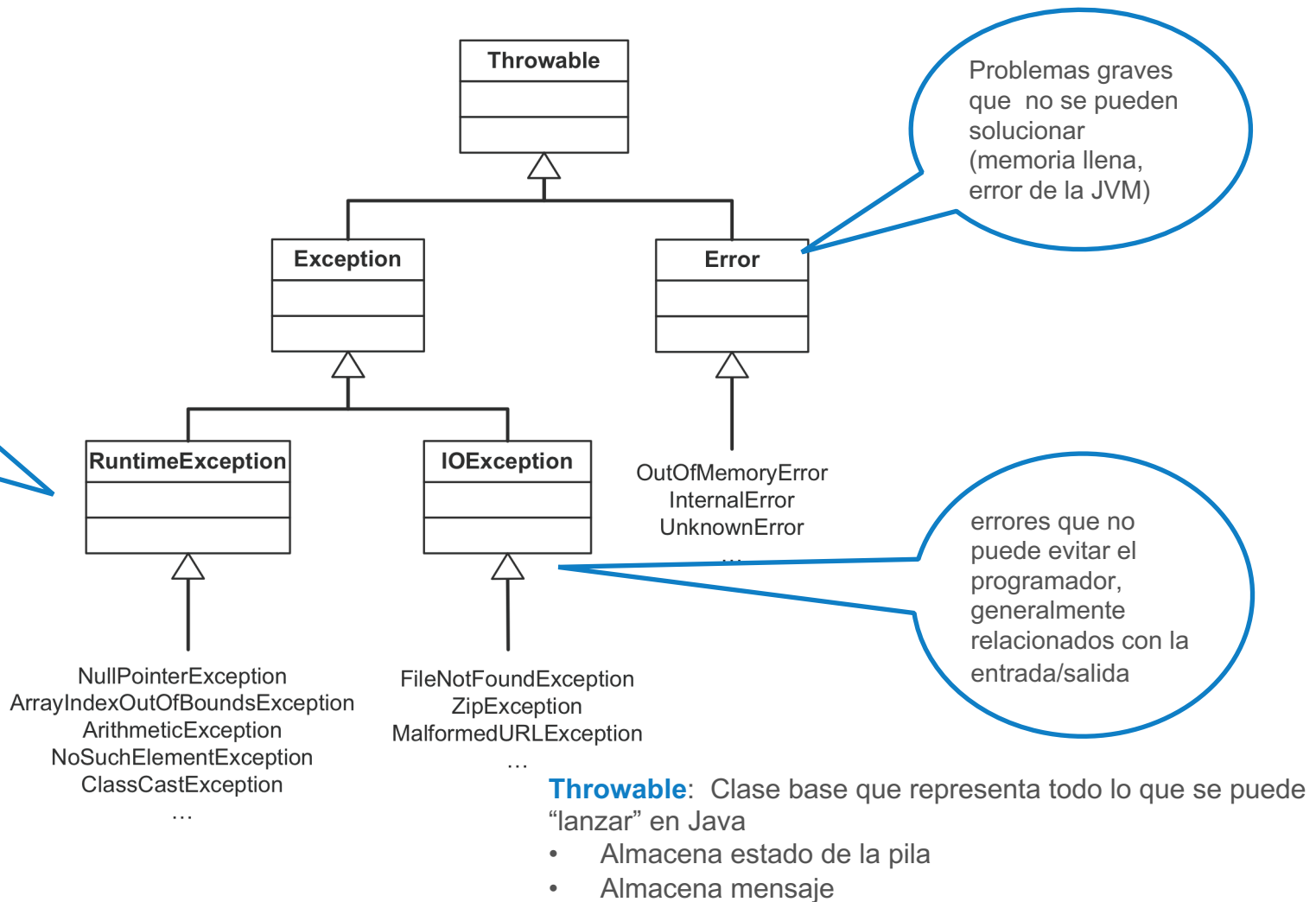
El modelo de excepciones de Java se basa en el concepto de salto no local, implementado con las palabras reservadas **throw**, **catch** y **try**, y se basa en los siguientes conceptos:

- Las instrucciones que pueden dar lugar a posibles excepciones son aisladas en bloques especiales (**try**) para controlar la propagación de las mismas.
- Cuando se detecta una excepción, el programa lanza (**throw**) una excepción que será tratada en otro punto del mismo.
- La excepción se trata mediante manejadores de excepciones que capturan (**catch**) la excepción

Funcionamiento

- ▶ Generación de una excepción dentro de un método.
 - Construcción del objeto Exception.
 - Lanzamiento de la excepción.
- ▶ Captura de la excepción:
 - Dentro del propio método.
 - Fuera del método a través de la pila de llamadas.
- ▶ Ejecución del manejador de la excepción.

Jerarquía de clases parra el manejo de excepciones



Lanzamiento de una excepción

Permite que los métodos respondan frente a circunstancias inesperadas.

Sentencia **throw**

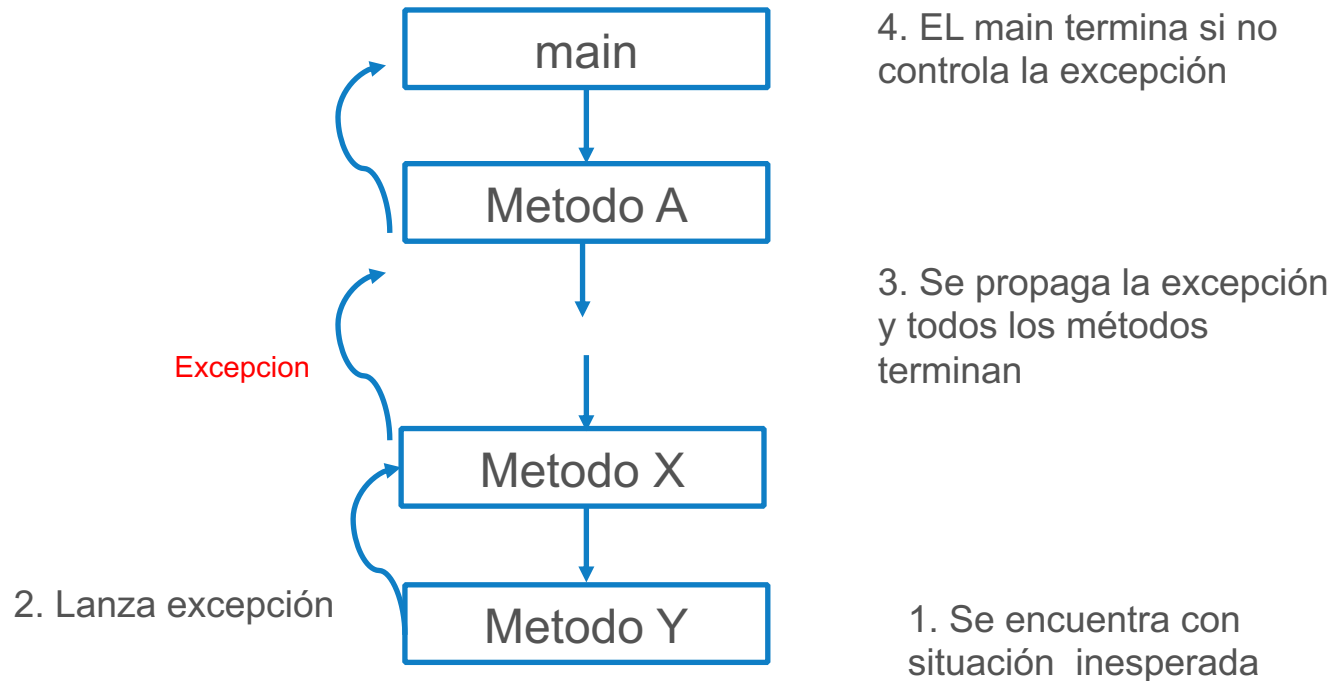
- **throw** referencia (objeto de una subclase de **Exception**)
- **throw** new clase_excepción(argumento_String) (una subclase de Exception)

El lanzamiento de una excepción hace que el método que se está ejecutando finalice de forma inmediata

La excepción lanzada se propaga por todas las invocaciones habidas hasta llegar al método, a no ser que encuentre un capturador

- a. Asegura la transmisión de cualquier cantidad de información desde el punto en que se eleva la excepción hasta el controlador que la trata.
- b. Suministra una manera de agrupar excepciones de modo que se puedan escribir controladores para capturar y tratar tanto excepciones individuales como grupos.
- c. Fácil de entender y usar.
- d. No impone coste adicional en cuanto a tiempo de ejecución a un código que no eleve ninguna excepción.
- e. Fácil de implementar.

Lanzamiento de una excepción




El lanzamiento de una excepción hace que el método que se está ejecutando finalice de forma inmediata.

La excepción lanzada se propaga por todas las invocaciones habidas hasta llegar al método, a no ser que encuentre un controlador.

Clausula Throws

Clausula que se incluye en la cabecera de los métodos. Es obligatorio si el método puede lanzar potencialmente una excepción.

```
public static void main(String[] argumentos) throws Exception
```



Puede ser una lista de excepciones

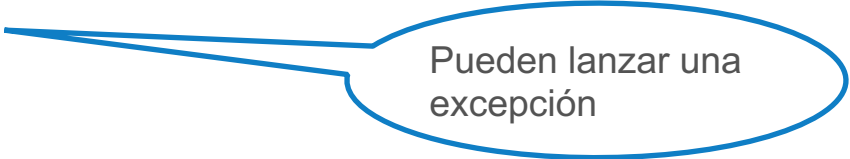
Bloque try

El controlador de excepciones es llamado a través de una expresión de elevación (**throw**) invocada por el código que se ejecuta dentro de un bloque de intento o **bloque try**, o en funciones invocadas desde el bloque **try**.

- **bloque try**:

try sentencias *lista_de_controladores*

- **controlador**:



Pueden lanzar una excepción

catch (declaración_de_excepciones) proposiciones

Bloque try

```
try{  
    objeto.invocación_método_lanza_excepción();  
}  
catch (Exception e){  
    sentencias que se ejecutan si se lanza una excepción en el bloque try  
}
```

```
try{  
    throw excepción;  
}  
catch (Exception e){  
    sentencias que se ejecutan si se lanza una excepción en el bloque try  
}
```

Bloque try

```
//cod 1
try{
    //cod 2

}catch (ArithmeticException e){
    //cod 3
}
finally{
    //cod 4
}

// cod 5
```

- Sin excepciones: se ejecuta 1,2,4,5
- Excepción aritmética: 1,2,3,4,5
- Otra excepción: 1, 2, 4

Creación de una excepción personalizada

Objetos excepción

- Crear una clase que tenga como clase base a Exception (directa o indirectamente).
- Implementar los constructores (al menos uno sin argumentos y otro que reciba un string).
- Añadir los métodos que se estimen necesarios.

```
class MiExcepcion extends Exception{  
    public MiExcepcion() {super();}  
    public MiExcepcion(String s) {super(s);}  
}
```

Ejemplo de excepciones

```
class ClaseC{
    public void metodoC() throws MiExcepcion{
        System.out.println("C-> lanza excepcion");
        throw new MiExcepcion("Casi que no");
        //System.out.println("Aqui no se llega");
    } }
class ClaseB{
    public void metodoB() throws MiExcepcion{
        ClaseC c=new ClaseC();
        try{
            c.metodoC();
        } catch (MiExcepcion e){
            System.out.println("B-> captura excepcion");
            System.out.println("B-> propaga excepcion");
            throw e;
        } } }
class ClaseA{
    public void metodoA() throws MiExcepcion{
        ClaseB b=new ClaseB();
        b.metodoB();
        System.out.println("A-> propaga excepcion"); //no se
ejecuta
    } }
public class PruebaExcepcion{
    public static void main(String [] args){
        ClaseA a=new ClaseA();
        try{
            a.metodoA();
        } catch (MiExcepcion e){
            System.out.println("main-> captura excepcion");
            System.out.println(e.getMessage());
        } } }
```

UNIVERSIDAD
INTERNACIONAL
DE LA RIOJA

unir

www.unir.net