

# Resumen Detallado: Excepciones en Java

## Introducción a la Gestión de Errores en Java

- ¿Qué es una Excepción? Una excepción es una condición anormal que ocurre durante la ejecución de un programa. Java utiliza la clase `Exception` para gestionar errores.
- Beneficios de las Excepciones:
  - Encapsulan errores en clases.
  - Separan el flujo normal del programa del tratamiento de errores.
  - Las excepciones pueden ser manejadas o relanzadas para que otro método en la pila de llamadas las gestione.

## Jerarquía de Clases de Excepciones

- Clase `Throwable`: Es la superclase de todas las clases que pueden ser lanzadas como excepción.
- Tipos de Clases `Throwable`:
  - `java.lang.Error`: Representa errores de compilación y del sistema.
  - `java.lang.Exception`: Representa excepciones generadas por la aplicación.
  - `java.lang.RuntimeException`: Un tipo especial de `Exception` cuya gestión no es obligatoria.

## Excepciones Verificadas (Checked) vs. No Verificadas (Unchecked)

- Excepciones Verificadas (Checked):
  - Su tratamiento es obligatorio y el compilador lo verifica.
  - Son subclases de `java.lang.Exception` que no son subclases de `java.lang.RuntimeException`.
- Excepciones No Verificadas (Unchecked):
  - Su tratamiento no es obligatorio y el compilador no lo verifica.
  - Son subclases de `java.lang.Error` o `java.lang.RuntimeException`.

## Manejo de Excepciones

- Bloques `try & catch`: Se utilizan para manejar excepciones.
  - `try`: Contiene el código que puede lanzar una excepción.
  - `catch`: Contiene el código para tratar una excepción específica.

- `finally`: Contiene el código que se ejecuta siempre, independientemente de si se produce o no una excepción.
- Anidamiento: Los bloques `try` & `catch` pueden estar anidados.
- Variables Locales: Las variables definidas dentro de los bloques `try`, `catch` o `finally` no son accesibles desde fuera de esos bloques.

## Lanzamiento de Excepciones

- `throws`: Se utiliza en la declaración de un método para indicar que puede lanzar una excepción.
- `throw`: Se utiliza para crear y lanzar una nueva excepción.
- Relanzamiento: Si una excepción se relanza consecutivamente hasta la JVM, el programa se detendrá mostrando la excepción.
- Note: `java.lang.Exception` tiene varios constructores, pero el más usado es el que recibe un `String` con el motivo de la excepción. Este `String` será accesible mediante el método `getMessage()`.

## Métodos Útiles para Excepciones

- `printStackTrace()`: Imprime la pila de llamadas (stack) en la salida estándar, incluyendo los números de línea y ficheros donde se produjo la excepción.
- `getMessage()`: Devuelve una cadena con la descripción de la excepción.
- `toString()`: Devuelve una representación en cadena de la excepción.

## Ejemplos de Excepciones y Errores Comunes

- No Verificadas (Unchecked):
  - `java.lang.NoClassDefFoundError`: No se encuentra la clase al intentar instanciarla.
  - `java.lang.IllegalArgumentException`: Se llama a un método con un parámetro erróneo.
  - `java.lang.IndexOutOfBoundsException`: Se intenta acceder a una posición inexistente en una colección.
  - `java.lang.NoSuchFieldError`: Se intenta acceder a un atributo inexistente.
  - `java.lang.NoSuchMethodError`: Se intenta acceder a un método inexistente.
  - `java.lang.NumberFormatException`: Se intenta convertir una cadena no numérica a un número.
  - `java.lang.OutOfMemoryError`: La JVM no puede instanciar un objeto por falta de memoria.

- `java.lang.ClassCastException`: Se realiza un casting incorrecto de un objeto.
- `java.lang.ArithmeticException`: Ocurre una operación matemática anómala (ej., división por cero).
- `java.lang.NullPointerException`: Se accede a un objeto con valor `null`.
- Verificadas (Checked):
  - `java.lang.InterruptedException`: Un thread en estado de espera es interrumpido.
  - `java.io.IOException`: Ocurre un problema de I/O genérico.
  - `java.io.EOFException`: Se detecta el final de un stream inesperadamente.
  - `java.io.FileNotFoundException`: Se intenta acceder a un fichero en modo lectura y no se encuentra.