

# Curso de Programación en Java



Juan Francisco Maldonado León  
**Arquitecto de Software**



# Uso de **Exceptions**



Juan Francisco Maldonado León  
**Arquitecto de Software**



# Introducción a Java

Manejo de Excepciones

## Exceptions

Manejo de Errores

En Java las situaciones que pueden provocar un fallo en el programa se denominan excepciones.

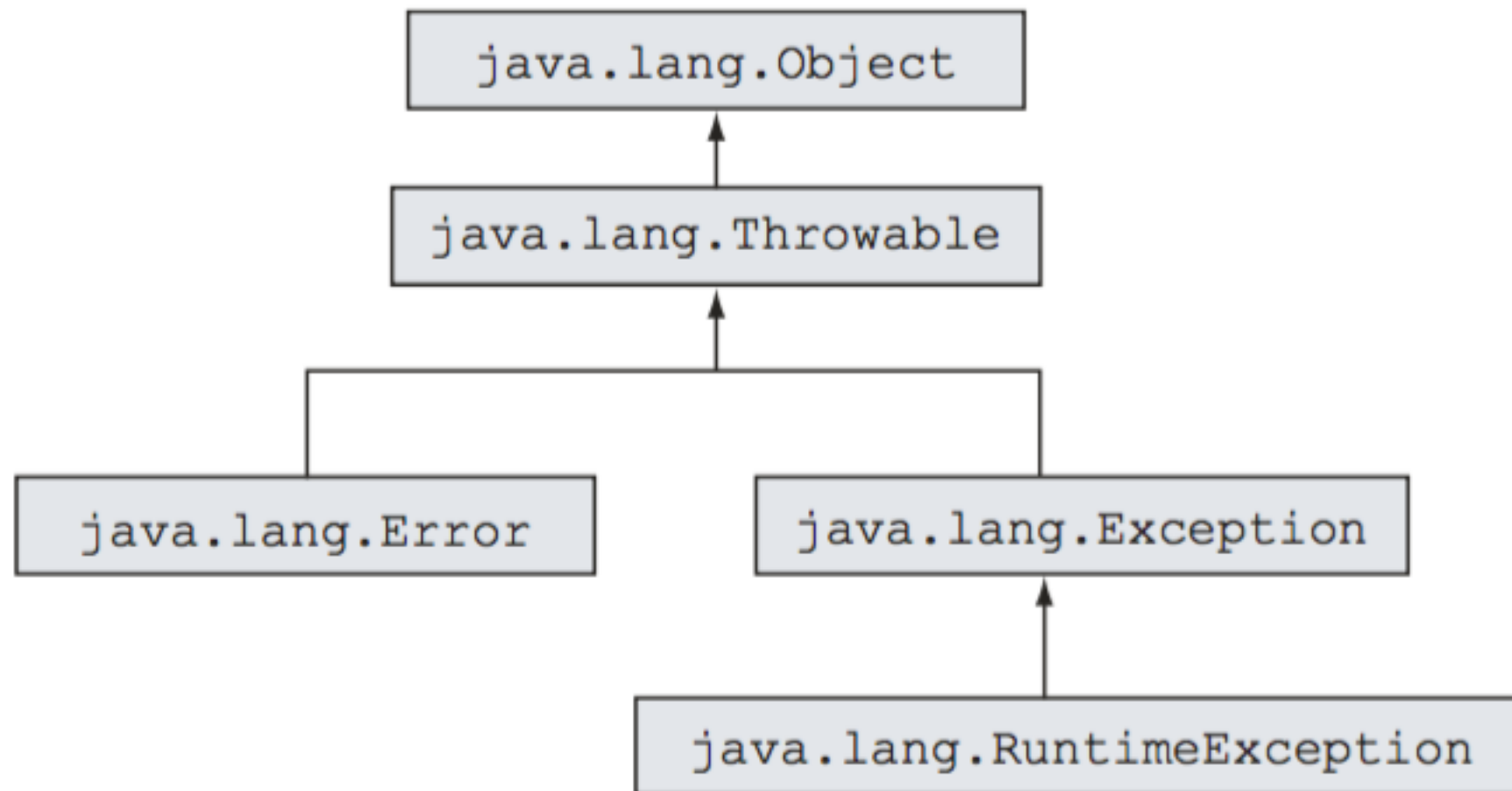


# Introducción a Java

Manejo de Excepciones

## Exceptions

Manejo de Errores

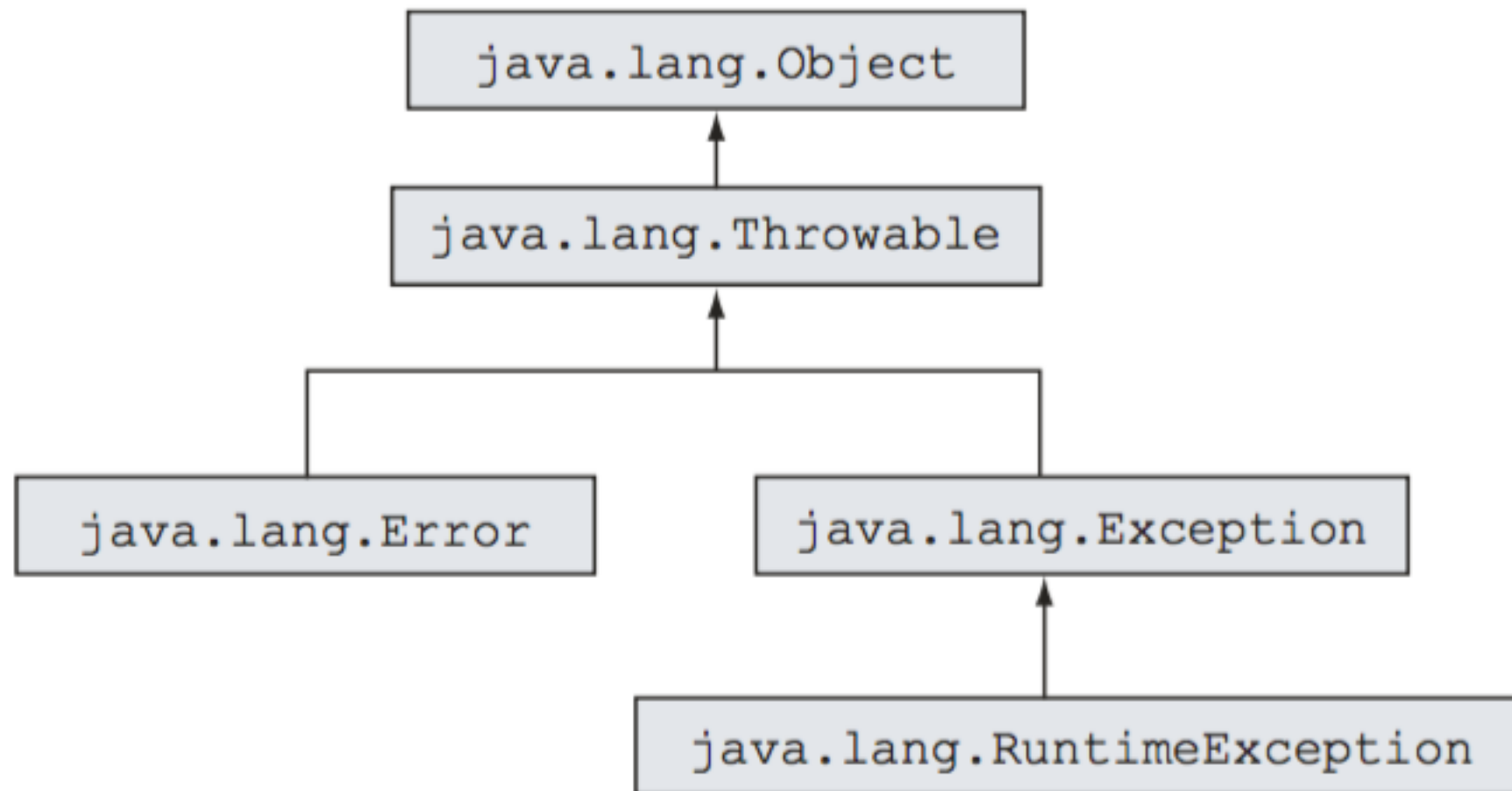


# Introducción a Java

Manejo de Excepciones

## Exceptions

Manejo de Errores

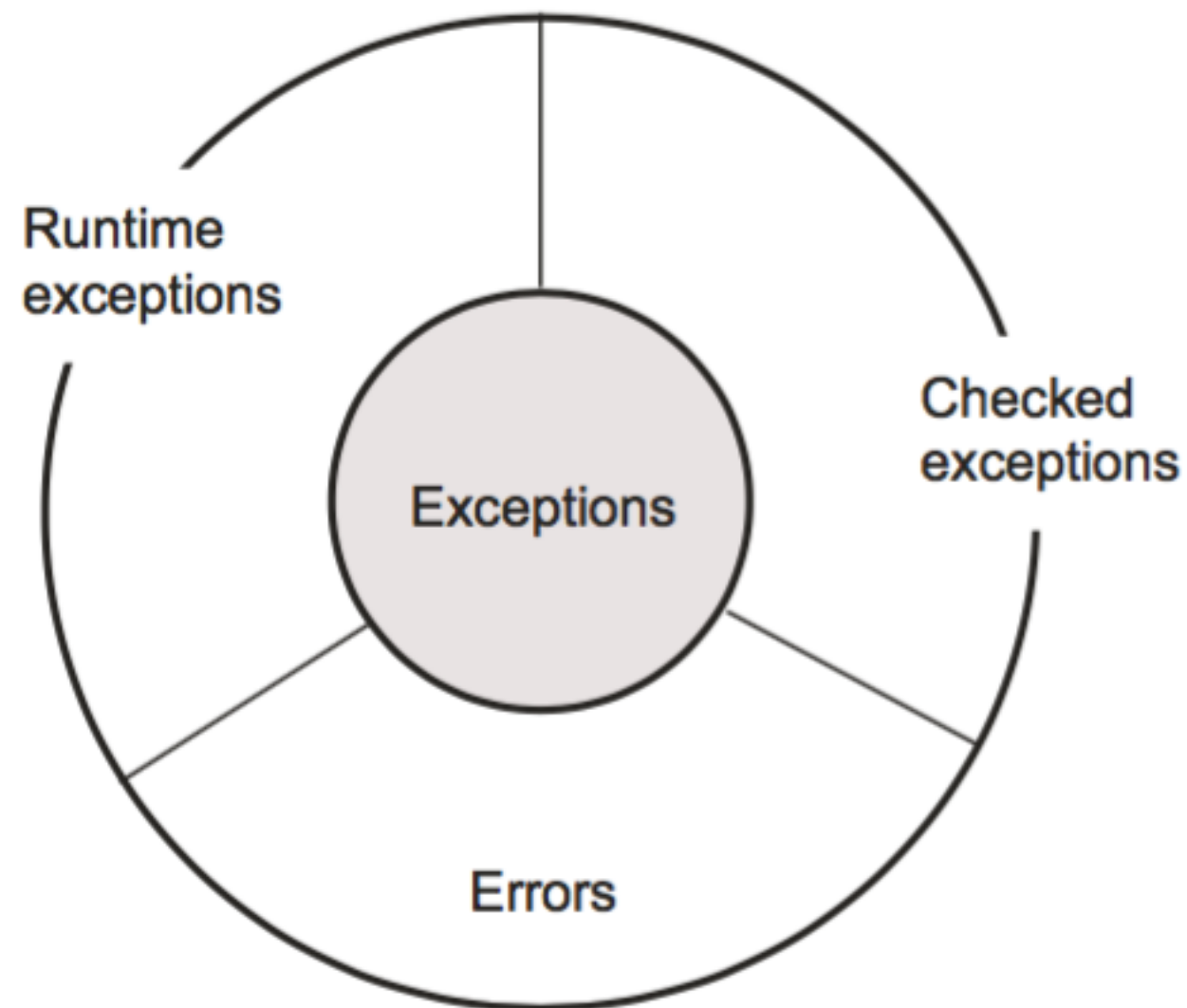


# Introducción a Java

Manejo de Excepciones

## Categorías de Excepciones

Manejo de Errores



# Introducción a Java

Manejo de Excepciones

## **Excepciones Chequeadas**

Categoría de excepciones

Son todas aquellas que heredan directamente de la clase  
Exception.

Necesitan ser capturadas o lanzarlas, caso contrario no se  
podrá compilar el código.



# Introducción a Java

Manejo de Excepciones

## Bloque Try-Catch

Capturar Excepciones

```
public void test()
{
    try {
        Thread.sleep(1000); //Lanza una excepcion
    } catch (InterruptedException e) {
        // Bloque de codigo para
        // capturar la excepcion lanzada
    }
}
```



# Introducción a Java

Manejo de Excepciones

## Lanzar una excepción

Capturar Excepciones

```
public void test() throws InterruptedException  
{  
    Thread.sleep(1000);  
}
```

# Introducción a Java

Manejo de Excepciones

## Lanzar una excepción

Capturar Excepciones

Cuando una excepción es lanzada, el proximo método que llame al método que lanzo la excepción deberá capturar la excepción o lanzarla nuevamente

# Introducción a Java

Manejo de Excepciones

## Lanzar una excepción

Capturar Excepciones

```
public void metodoA() throws InterruptedException {  
    Thread.sleep(1000);  
}  
  
public void metodoB() throws InterruptedException {  
    metodoA();  
}
```

# Introducción a Java

Manejo de Excepciones

## Lanzar una excepción

Capturar Excepciones

```
public void metodoA() throws InterruptedException {  
    Thread.sleep(1000);  
}
```

```
public void metodoB() {  
    try {  
        metodoA();  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
}
```

# Introducción a Java

Manejo de Excepciones

## Excepciones no Chequeadas

Categorías de Excepciones

Son todas aquellas que heredan de RuntimeException.  
Las excepciones no chequeadas no nos obligan a capturar o lanzar la excepción que pueda provocar un método.

# Introducción a Java

Manejo de Excepciones

## Errors

Categorías de Excepciones

Todas aquellas excepciones que heredan de la clase Error.

Un Error es una excepción grave la cual es lanzada por la JVM como resultado de un error.

# Introducción a Java

Manejo de Excepciones

## Errors

Categorías de Excepciones

### **NoClassDefFoundError**

Se produce cuando la jvm no encuentra un archivo .class en tiempo de ejecución.

### **StackOverflowError**

Se produce cuando la lista de ejecuciones de un programa, necesita mas memoria de lo que tiene asignado.  
Generalmente son producido por bucles infinitos



# Introducción a Java

Manejo de Excepciones

## Errors

Categorías de Excepciones

```
public static void metodoA() {  
    metodoA();  
}  
  
public static void main(String[] args) {  
    metodoA();  
}
```

Exception in thread "main" java.lang.StackOverflowError  
at cl.java.curso.threads.saludo.Personal.metodoA([Personal.java:21](#))

# Introducción a Java

## Manejo de Excepciones

### Crear Excepciones Propias

#### Excepciones

```
public class CuentaBloqueadaException extends RuntimeException {  
  
    private static final long serialVersionUID = 4944876324427607785L;  
  
    /**  
     * @param message  
     */  
    public CuentaBloqueadaException(String message) {  
        super(message);  
    }  
}
```

# Introducción a Java

## Manejo de Excepciones

### Crear Excepciones Propias

#### Excepciones

```
public class CuentaBloqueadaException extends Exception {  
  
    private static final long serialVersionUID = 4944876324427607785L;  
  
    /**  
     * @param message  
     */  
    public CuentaBloqueadaException(String message) {  
        super(message);  
    }  
}
```