

Excepciones en Java



Definición

- ▶ Una **excepción** es cualquier situación anómala que puede ocurrir durante la ejecución de un programa y que **impide que continúe el flujo normal de instrucciones**
 - Dispositivos que fallan
 - Datos que se introducen incorrectamente
 - Valores fuera de rango
 - etc.
- ▶ **Tenerlas en cuenta en el programa** lo complicaría enormemente. **No tenerlas en cuenta** haría que el programa abortase



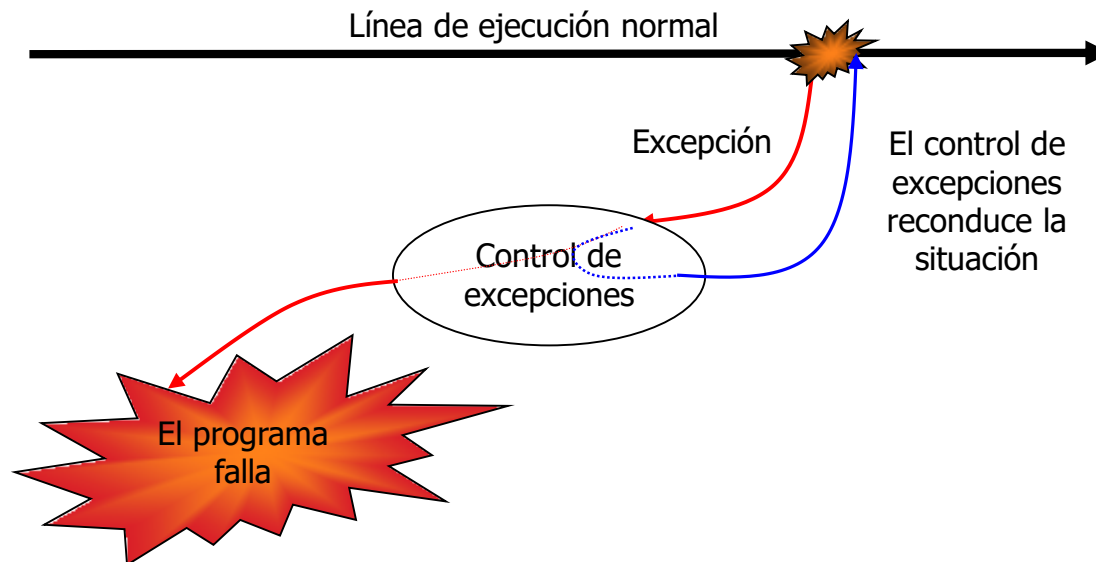
Representación de excepciones

- ▶ Muchos lenguajes disponen de mecanismos para resolver estas situaciones de forma controlada
- ▶ Las excepciones son elementos de información de los programas identificadas por un nombre y que indican dónde y por qué un proceso se interrumpe
- ▶ Las excepciones en Java están **representadas por objetos** que contienen información sobre ellas
 - El tipo de objeto determina el tipo de excepción



Ciclo de vida de las excepciones

- ▶ Fases del ciclo de existencia de una excepción
 - **Lanzamiento:** se abandona la ejecución en marcha
 - **Captura:** la ejecución continúa en un gestor de excepciones
 - **Control:** se reconduce la situación o termina el programa



Control de excepciones en Java

- ▶ El **control de excepciones** se basa en separar:
 - La parte del programa susceptible de producir alguna excepción en bloques try
 - Los gestores de excepciones que se encargarán de tratar las excepciones se definen con la instrucción catch
 - Opcionalmente, se puede tener un bloque finally que se sitúa después de los **catch** y se ejecuta siempre después del bloque **try**, haya ocurrido, o no, una excepción



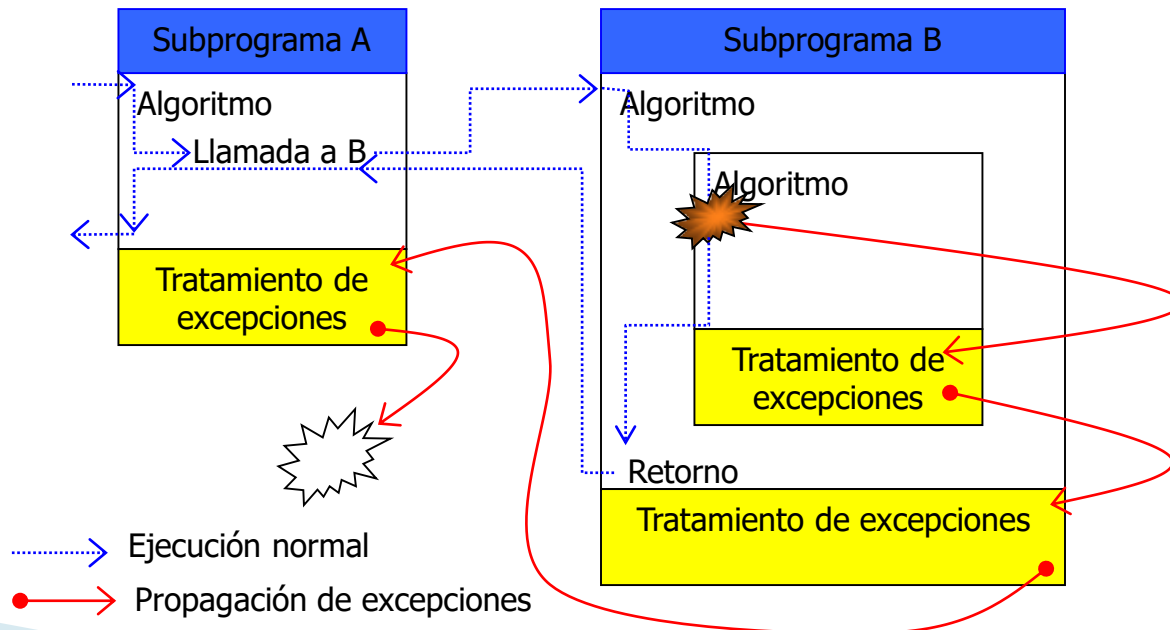
Formato de las excepciones

```
try {  
    //Código que puede lanzar excepciones  
}  
catch(TipoExcepción1 e) {  
    //Gestor de la excepción de tipo TipoExcepción1  
}  
catch(TipoExcepción2 e) {  
    //Gestor de la excepción de tipo TipoExcepción2  
}  
//Puede haber tantos catch como sean necesarios  
//...  
finally {  
    //Después de los catch  
    //Opcionalmente se puede tener este bloque  
}
```

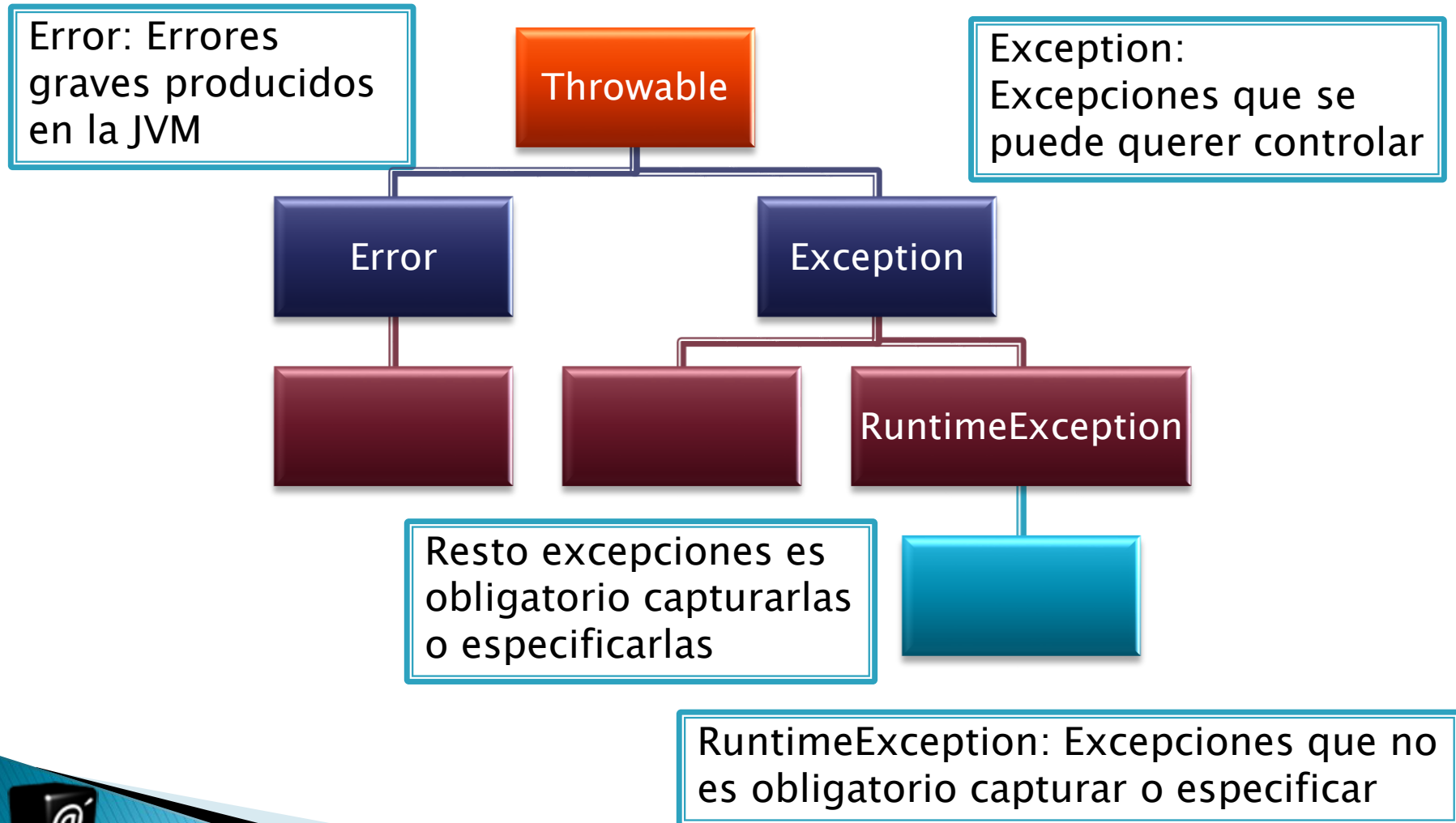


Propagación de las excepciones

- ▶ Las excepciones se pueden producir directamente o indirectamente en métodos llamados durante la ejecución de ese código
- ▶ El control de excepciones se realiza desde los bloques más internos a los más externos



Jerarquía de clases de excepciones



Controlar o especificar excepciones

- ▶ Algunas excepciones es obligatorio especificarlas o capturarlas
- ▶ Para especificar las excepciones se escribe, después del cierra paréntesis “)” de los parámetros y antes del abre llaves “{”, **throws** y las clases de excepciones que pueden salir de él
- ▶ Es obligatorio especificar qué excepciones pueden salir de un método, ya sea porque se lanza con **throw** o porque no captura la excepción que puede lanzar otro método llamado desde éste
- ▶ Las excepciones que **no es obligatorio** capturar o especificar son las derivadas de **RuntimeException**



Lanzamiento de excepciones

- ▶ Las excepciones predefinidas en el lenguaje se lanzan de forma automática
- ▶ Las excepciones definidas por el programador se lanzan mediante la instrucción **throw** objeto
 - Se debe escribir código para detectar cuándo se produce la situación anormal y efectuar su lanzamiento
 - Puede llevar asociada el nombre de la excepción y un texto como mensaje para una mejor identificación del problema ocurrido
 - Sin argumentos va dentro de un manejador y relanza la excepción recogida

```
throw new Exception("Error");
```

