

Excepciones en JAVA

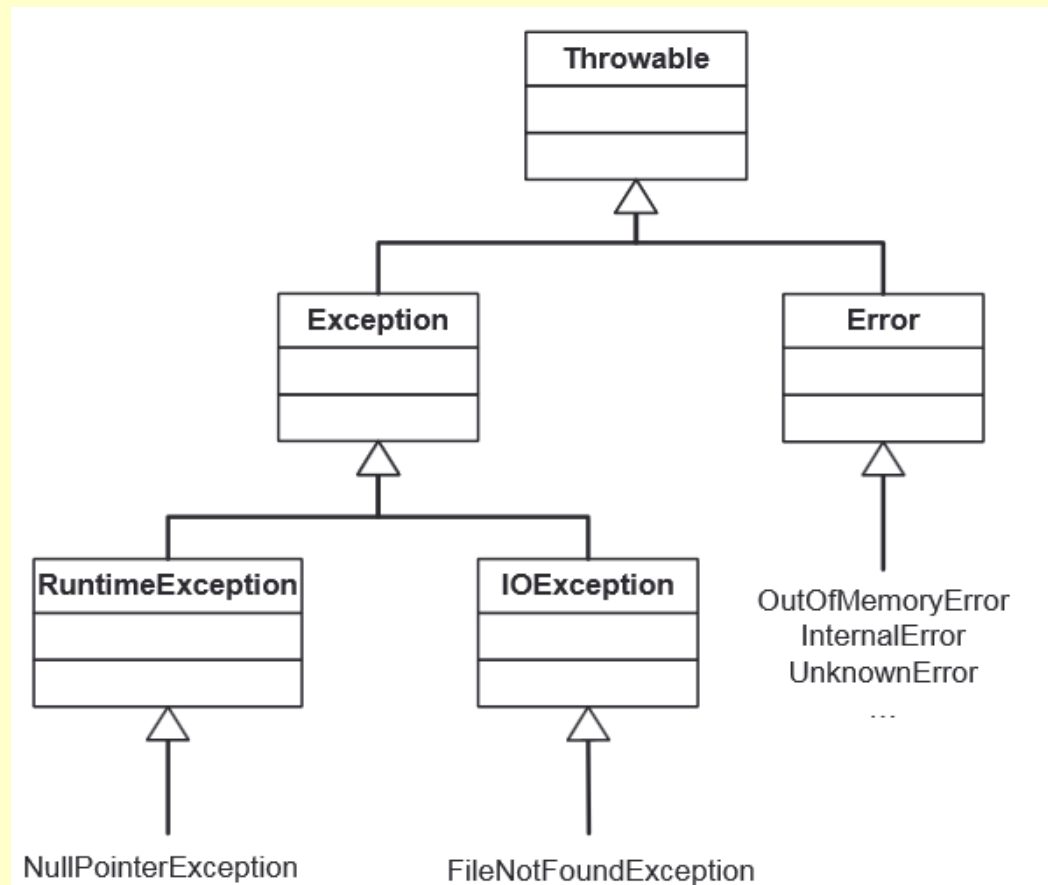


<devAcademy>

Excepción

- Una excepción, es el resultado de un comportamiento anómalo de un programa, que altera el flujo normal de ejecución
- En Java, una excepción, se traduce en una clase, que expresa el tipo concreto de problema o error que produjo su propia creación. En otras palabras, para cada tipo de error, existe un tipo de excepción

Jerarquía Excepciones



Excepción

- En realidad, una excepción, es la creación de un objeto, cuya clase hereda de Throwable
- Por ello, al crearse, tendrá acceso al método `printStackTrace()` y un campo `Message`, donde almacena el detalle del fallo.

Gestión de excepciones

- Cuando realizamos una operación susceptible de una excepción, Java nos ofrece dos posibilidades:
 - Captura
 - Propagación

Captura de excepciones

- El código susceptible de lanzar una excepción, queda envuelto en un bloque `try { }`
- Ante el posible fallo, hemos definido uno o varios bloques `catch { }` donde automáticamente irá a parar el flujo de ejecución

Captura de excepciones

```
try {  
    int direccion = System.in.read();//lanza una Excepción  
    ...  
}  
  
} catch (Exception e) { //la recojo y la proceso  
    e.printStackTrace();  
}
```

Bloque finally

- El bloque finally, es opcional y puede añadirse a la estructura de un bloque try-catch y se ejecuta haya habido un error o no en el bloque try {}
- Se describe a continuación, y normalmente, se usa para asegurarnos que hemos liberado recursos, que pueden quedar abiertos al producirse una excepción

Bloque finally

```
try {  
    ...  
}  
} catch (Exception e) { ...}  
finally  
{ //falle o no try, finally se ejecutará siempre!  
}
```

Propagación

- Es la opción que nos da Java, cuando en la cabecera del método usamos

```
public void startElement(String uri, String  
localName, String name,  
Attributes attributes) throws SAXException
```

Gestión de excepciones

- Capturar y luego propagar una excepción, no son métodos excluyentes.
- De hecho, es aconsejable, en proyectos de cierta magnitud, emplearse conjuntamente

Capturo y lanzo

```
public boolean comprobar () throws IOException
```

```
    try {  
    }  
    catch (Exception e)  
    {  
        throw e;  
    }
```

Tipos propios de exceptions

- Java nos permite definir nuestros propios tipos de excepciones.
- Para ello, definiremos nuestras propias clases, que heredan de `Exception`

Tipos propios de exceptions

```
public DivideByZeroException
    extends ArithmeticException
{
    public DivideByZeroException(String Message)
    {
        super(message);
    }
}
```

Tipos propios de exceptions

```
public double dividir(int num, int den)
    throws DivideByZeroException
{
    if (den==0)
        throw new DivideByZeroException("Error!");

    return ((double) num/ (double)den);`
}
```

Práctica

- Definir una excepción propia, llamada `InsertarPersonasException` y que se lance al sobrepasar el número de personas que incrementamos en nuestra Lista.
- Nota: debemos previamente a lanzar nuestra excepción, recoger la `ArrayIndexOutOfBoundsException`