

# Fundamentos de Java

## Manejo de Excepciones

# Excepciones

Una excepción es un evento que ocurre durante la ejecución de un programa que rompe el flujo normal de ejecución.

Cuando se habla de excepciones nos referimos a evento excepcional.

Muchas cosas pueden generar excepciones: Errores de hardware (falla de disco), de programa (acceso fuera de rango en arreglo), apertura de archivo inexistente, etc.

# Manejo de excepciones en Java

Cuando se produce una excepción dentro de un método de Java, éste crea un objeto que contiene información sobre la excepción y lo pasa al código llamador.

La rutina receptora de la excepción es responsable de reaccionar a tal evento inesperado.

Cuando creamos un objeto para la excepción y lo pasamos al código llamador decimos que lanzamos una excepción (Throw an exception)

Si el método llamador no tiene un manejador de la excepción se busca hacia atrás en la pila de llamados anidados hasta encontrarlo.

Decimos que el manejador atrapa la excepción (catch the exception)

# Ventajas de Excepciones

Separación entre el "código normal" y el código de manejo de errores.

Propagación de errores hacia "arriba" en el stack de llamadas (y detención automática del programa si una situación de error no es manejada).

Facilidades para la agrupación de tipos de errores.

Facilidades para entregar información del error producido, sin que se produzca interferencia con el retorno normal.

# Captura de Excepciones

```
try { //sentencias}
catch (e-type1 e )
{ // sentencias}
catch (e-type2 e )
{ // sentencias} ...
finally { //sentencias}
```

# Ejemplo captura de excepciones

```
public static void doio (InputStream in, OutputStream out) {  
    int c;  
    try {  
        while (( c=in.read()) >=0 ) {  
            c= Character.toLowerCase( (char) c);  
            out.write( c );  
        }  
    } catch ( IOException e ) {  
        System.err.println("doio: I/O Problem");  
        System.exit(1);  
    }  
}
```

# Propagando Excepciones

```
public static void doio2 (InputStream in, OutputStream out) throws Throwable {  
    int c;  
    while (( c=in.read()) >=0 ) {  
        c= Character.toLowerCase( (char) c);  
        out.write( c );  
    }  
}
```

# Tipos de excepciones

Aquellas generadas por el lenguaje Java. Éstas se generan cuando hay errores de ejecución, como al tratar de acceder a métodos de una referencia no asignada a un objeto, división por cero, etc.

Aquellas no generadas por el lenguaje, sino incluidas por el programador



# Excepción lanzada por el usuario

```
if (b !=0 )  
    result = a/b;  
else {  
    throw new ArithmeticException("b no es valido, causa resultado infinito");  
}
```