

Excepciones

Luis Fernando Llana Díaz

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid

18 de abril de 2007

Luis Fernando Llana Díaz

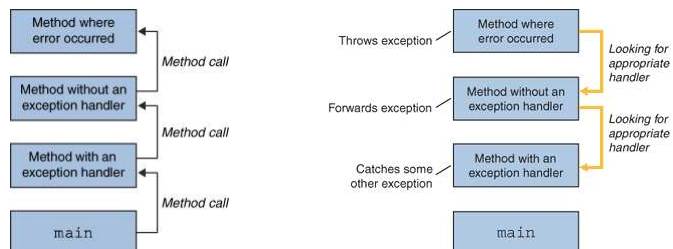
Departamento de Sistemas Informáticos y ComputaciónUniversidad Complutense de Madrid

Excepciones

¿Qué es una excepción? I

Excepción

Es un evento que ocurre durante la ejecución de un programa e interrumpe su ejecución normal



Luis Fernando Llana Díaz

Departamento de Sistemas Informáticos y ComputaciónUniversidad Complutense de Madrid

Excepciones

¿Qué es una excepción? II

Cuando se produce una excepción se puede

- Capturarla y arreglar la situación
- Transmitir el error al procedimiento llamante

Capturar/Transmitir una excepción

```
public static int leeInt() throws IOException {  
    try {  
        return Integer.parseInt(teclado.readLine());  
    }  
    catch (NumberFormatException e) {  
        System.out.println("Número incorrecto, dámelo otra vez.");  
        return leeInt();  
    }  
}
```

1
2
3
4
5
6
7
8
9

- El método `readLine` puede lanzar `java.io.IOException`
- El método `parseInt` puede lanzar `java.lang.NumberFormatException`

Instrucción `try/catch/finally`

```
try {  
    codigo  
} catch (Excepcion1 e1) {  
    codigo1  
} catch (Excepcion2 e2) {  
    codigo2  
}  
.....  
} finally {  
  
}
```

1
2
3
4
5
6
7
8
9
10
11

- Intenta ejecutar el *código*.
- Si se produce una excepción de clase *Excepcion_k* se ejecuta el código *codigo_k*
- El código en *finally* se ejecuta al final en cualquier caso.

Instrucción `try/catch/finally`

```
public void writeList() {  
    PrintWriter out = null;  
    try {  
        System.out.println("Entering try statement");  
        out = new PrintWriter(  
            new FileWriter("OutFile.txt"));  
        for (int i = 0; i < SIZE; i++)  
            out.println("Value at: " + i + " = "  
                + vector.elementAt(i));  
    } catch (ArrayIndexOutOfBoundsException e) {  
        System.err.println("Caught "  
            + "ArrayIndexOutOfBoundsException: "  
            + e.getMessage());  
    } catch (IOException e) {  
        System.err.println("Caught IOException: "  
            + e.getMessage());  
    } finally {  
        if (out != null) {  
            System.out.println("Closing PrintWriter");  
            out.close();  
        }  
        else {  
            System.out.println("PrintWriter not open");  
        }  
    }  
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

Declarar/lanzar excepciones

Requisito captura o declarar

Si una instrucción en un método puede lanzar una excepción se debe

- Capturar con instrucción `catch`
- El método debe declarar (`throws`) que la puede lanzar.

Excepciones normales deben cumplir el requisito

Errores errores externos (fallos de hardware). No necesitan cumplir el requisito.

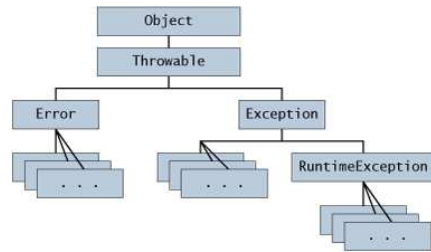
Excepciones **Runtime** errores internos que se debe a fallos, usos incorrectos de una API. No necesitan cumplir el requisito.

Lanzar excepciones

```
public void anyadirDias(int inc) {  
    int d=diasDesdeInicio+inc;  
    if (d<0) throw new FechaFueraDeRango("Día anterior al permitido");  
    diasDesdeInicio=d;  
}
```

1
2
3
4
5

Escribiendo excepciones



```
package fecha;
public class FechaFueraDeRango extends RuntimeException {
    public FechaFueraDeRango(String s) {
        super(s);
    }
}
```

1
2
3
4
5
6