

PROGRAMACIÓN ORIENTADA A OBJETOS (Laboratorio de Prácticas)

Titulaciones de Grado en Ingeniería de Computadores, Ingeniería Informática y Sistemas de Información

Sesión 6 EXCEPCIONES EN JAVA

Para familiarizarse con los distintos mecanismos para el tratamiento de las excepciones realizar los ejemplos del capítulo 5 de teoría sobre Java.

Ejemplo de uso de Excepciones:

En el siguiente ejemplo calculamos la media de unas calificaciones de examen, contemplando todos sus posibles errores en tiempo de ejecución.

```
import java.io.*;

public class Notas {

    public static void main(String[] args) {
        double media = 0, total = 0, notanum = 0;
        int contador = 0;
        String notatxt = "";
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (!notatxt.equals("Z")) {
            try {
                System.out.print("\nTeclee calificación (0-10), Z para terminar: ");
                notatxt = entrada.readLine().toUpperCase();
                notanum = Double.parseDouble(notatxt);
                if (notanum < 0) {
                    throw new NotaMal("No se admiten calificaciones menores que 0.");
                } else if (notanum > 10) {
                    throw new NotaMal("No se admiten calificaciones mayores que 10.");
                }
                total += notanum;
                contador++;
            } catch (NotaMal nm) {
                System.out.println("\n" + nm.getMessage());
            } catch (NumberFormatException nfe) {
                if (!notatxt.equals("Z")) {
                    System.out.println("\nError de formato numérico: " +
nfe.toString());
                }
            } catch (IOException ioe) {
                System.out.println("\nError de entrada/salida: " + ioe.toString());
            } catch (Exception e) {
                // Captura cualquier otra excepción
                System.out.println("Excepción: " + e.toString());
            }
        }

        if (contador != 0) {
            media = (double) total / contador;
        }
    }
}
```

```

        System.out.println("\nEl promedio del grupo es: " + media);
    } else {
        System.out.println("\nNo se introdujeron calificaciones.");
    }
}

}

}

class NotaMal extends Exception {

    public NotaMal() {
        super("Excepción definida por el usuario: NOTA INCORRECTA.");
    }

    public NotaMal(String msg) {
        super("Excepción definida por el usuario: " + msg);
    }

}

```

1. Ejercicio Media

Reescribir el ejercicio de la **“Media”** de la Sesión 2, de forma que se controle mediante excepciones si el usuario introduce una letra diferente a A, B, C, D y E o F (para finalizar), imprimiendo un mensaje de **“nota introducida no valida”**.

2. Ejercicio Calculadora

Escribir un programa llamada Calculadora que realice las funciones típicas de una calculadora. Para ello pedirá al usuario tres valores: operación (+,-,*,/), operando1, operando2 y a partir de ellos muestre el resultado de la operación. Cuando el usuario introduzca una Z como valor de la operación el programa parará. Si se introduce un carácter no permitido, se debe producir una excepción definida por el usuario imprimiendo un mensaje de error.

Se deberán intentar utilizar todos los mecanismos de tratamiento de excepciones dados en teoría.

3. Ejercicio Publicaciones

Se desea desarrollar un programa para la gestión de publicaciones en una biblioteca. Todas las publicaciones deben tener asociado un autor (String), un título (String) y un año de publicación (int). Deben considerarse tres tipos de publicaciones distintos: libros, revistas y tesis doctorales. En el caso de los libros se debe agregar el nombre de la editorial correspondiente (String). Las revistas deben añadir el número (int) y el volumen (int). Por su parte, las tesis doctorales deberán incluir la Universidad (String) y la calificación (int). Se deben codificar las clases Publicación, Libro, Revista y Tesis con los atributos mencionados, los métodos get y set correspondientes y al menos un constructor. Además, debe generarse una excepción definida por el usuario (FechaMal) cuando se intente establecer como año de publicación un año posterior al existente en el sistema o un año negativo. Se debe tener en cuenta la generación de esta excepción tanto en el método set del atributo correspondiente como en el constructor.

4. Ejercicio Aparcamiento (ejercicio obligatorio para II y SI, voluntario para IC)

Reescribir el código del ejercicio “Aparcamiento” de la sesión 4 de forma que los métodos `introducirVehiculo` y `sacarVehiculo` puedan producir excepciones en las siguientes circunstancias:

- No se puede introducir un vehículo si ya está dentro o el aparcamiento está lleno.
- No se puede sacar un vehículo que no está dentro del aparcamiento.

La aplicación tiene que hacer uso de la clase `AparcamientoException` para la utilización de excepciones. *Ejemplo de utilización: Si detectamos que queremos sacar un vehículo del aparcamiento y éste no está, ejecutaremos una sentencia como la siguiente:*

`throw new AparcamientoException(AparcamientoException.VEHICULO_FUERA);`

```
public class AparcamientoException extends Exception {  
    public static String VEHICULO_DENTRO = "El vehículo ya se encuentra en el  
aparcamiento.";   
    public static String VEHICULO_FUERA = "El vehículo no se encuentra en el  
aparcamiento.";   
    public static String APARCAMIENTO_LLENO = "El aparcamiento está lleno.";   
  
    public AparcamientoException() {  
        super("Se ha producido una excepción en la aplicación.");  
    }  
  
    public AparcamientoException(String txt) {  
        super(txt);  
    }  
}
```