


Ejercicios de manejo de excepciones

1. Dado el siguiente código:

```
public class Ejercicio01 {
    public static void main(String[] args) {
        Bomba b=new Bomba();
        try {
            System.out.println("Inicio bloque");
            
            System.out.println("Fin bloque");
        }
        catch (Exception e) {
            System.out.println(e);
            System.out.println(e.getMessage());
        }
        System.out.println("Fin programa");
    }
}

class Bomba {
    public void a() throws A {
        throw (new A());
    }
    public void b() throws B {
        throw (new B());
    }
    public void c1() throws C {
        throw (new C());
    }
    public void c2() throws C {
        int n=10;
        if (n<10) {throw (new C());}
    }
}

class A extends Exception {
    public String toString() {
        return "aS";}
    public String getMessage() {
```

```

        return "aM";}
    }
    class B extends Exception {
        public String getMessage() {
            return "bM";}
    }
    class C extends Exception {
        public String toString() {
            return "cS";}
    }
}

```

Anticipar qué saldrá por la pantalla, si en la zona marcada como código se incluye cada una de las siguientes sentencias:

1. b.a();
2. b.b();
3. b.c1();
4. b.c2();

Comprobar (compilando y ejecutando el código) que la previsión que se hizo fue correcta y razonar el resultado.

2. Escribir el código de las clases E1 y E2 para conseguir que se imprima por pantalla "Hola amigos", sin modificar nada del código escrito

```

public class Ejercicio02 {
    public static void main(String[] args) {
        Bomba b=new Bomba();
        try {
            b.f();
            b.g();
            System.out.println("Hola");
        }
        catch (E1 e) {
            System.out.println(e.getMessage());
        }
        catch (E2 e) {
            System.out.println(e);
        }
    }
}

```

```

class Bomba {
    public void f() throws E1 {
        int x=30;
        if (x>30) {
            System.out.println("Hola amigos");
        }
    }
}

```

```

        throw (new E1());
    }
}
public void g() throws E2 {
    throw (new E2());
}
}

```

3. Hacer una clase Ejercicio03 con un método estático **examinar(x:int)** que lance una excepción personalizada de tipo **NumeroException**.

examinar(x:int) lanzará una excepción con el mensaje (asociado a getMessage())) CERO si x valiera cero, PAR si x es par, e IMPAR si x es impar. Desde el método main de Ejercicio03, probar el método "examinar", leyendo un número por teclado, y examinando dicho número.

El programa mostrará un mensaje del estilo "**El número es [PAR | IMPAR | CERO]**", pero sin necesidad de haber escrito ningún "if" en Ejercicio03

4. Hacer una versión del programa anterior en el que se hubiera mostrado por pantalla "**El número es [nulo | divisible por dos | NO divisible por dos]**"

NOTA: No se puede modificar el código de **examinar(...)** ni el de la clase **NumeroException**

5. Hacer una versión del programa anterior, pero lanzando 3 tipos de excepciones distintas: **ParException**, **ImparException** y **CeroException**