

Práctica: Pruebas IDM (en parejas de alumnos)

Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación, URJC

13 de noviembre de 2019

Esta práctica se realiza en parejas. Utilizad un repositorio en GitHub para desarrollar las pruebas y el SW que pase las pruebas que se describe en las cuatro secciones siguientes.

Para cada uno de los cuatro programas tenéis que:

1. Identificar el *Software Under Test*
2. Identificar los parámetros del SUT
3. Utilizando la técnica de la modelización de las entradas, crear caracterizaciones basadas en la funcionalidad, y sus particiones en bloques.
4. Elegir valores adecuados de cada bloque que satisfagan alguno de los criterios de cobertura “razonables” vistos en clase, prestando atención a los casos frontera que pueda haber en cada caso a la hora de elegir valores concretos para los tests.
5. Escribir el código de pruebas con JUnit, **antes de escribir el código del SUT**
6. Escribir el código del SUT y mejorarlo hasta que pase todas las pruebas.
7. Utilizad los tests que habéis desarrollado para descubrir fallos en las implementaciones del SUT de otras parejas de compañeros.

1. Cálculo de años bisiestos

```
public class Bisiestos {  
  
    // Devuelve true si año es bisiesto, false en caso contrario  
    // Eleva la excepción InvalidParameter si año no es un parámetro válido  
  
    public boolean esBisiesto(int año) throws InvalidParameter {  
        //  
    }  
}
```

2. Conversión de números romanos a base diez

```
// Ver https://es.wikipedia.org/wiki/Numeraci%C3%B3n_romana  
public class RomanNumeral {  
    // Parámetro: s es un número romano  
    // Devuelve : el número s en base 10  
    // Eleva la excepción InvalidParameter si s no es un número romano  
  
    public int convierte(String s) throws InvalidParameter {
```

```

    //
}
}

```

3. Embotelladora

```

public class Embotelladora {

    // Parámetros: pequenas: número de botellas en almacén de 1L
    //              grandes : número de botellas en almacén de 5L
    //              total   : número total de litros que hay que embotellar
    // Devuelve:    número de botellas pequeñas necesarias para embotellar el total de líquido, teniendo
    //              en cuenta que hay que minimizar el número de botellas pequeñas: primero
    //              se rellenan las grandes
    // Eleva la excepción NoSolution si no es posible embotellar todo el líquido

    public int calculaBotellasPequeñas(int pequenas, int grandes, int total) throws NoSolution {
        //
    }

}

```

4. Descuento BlackFriday

```

public class DescuentoBlackFriday {

    // Parámetros: precioOriginal es el precio de un producto marcado
    //              en la etiqueta
    //              porcentajeDescuento es el descuento a aplicar expresado como un porcentaje
    // Devuelve:    el precio final teniendo en cuenta que si es black friday (29 de noviembre) se aplica
    //              un descuento de porcentajeDescuento
    // Eleva la excepción InvalidParameter si precioOriginal es negativo

    public double PrecioFinal(double precioOriginal, double porcentajeDbescuento) throws InvalidParameter {
        //
    }

}

```

Entrega de la práctica

Cada pareja tendrá que entregar un documento de texto de nombre `PRUEBA-IDM.txt` en cuya primera línea aparezca la url del proyecto en GitHub.

En este fichero hay que documentar, para cada uno de los 4 programas, el proceso de diseño de casos de pruebas (caracterizaciones, bloques, criterio de cobertura elegido, combinaciones de bloques, valores elegidos para cada test,...).

En el código de los tests es importante documentar en comentarios los nombres de las caracterizaciones de forma que se pueda saber a qué combinación de bloques de qué caracterizaciones corresponde cada caso de prueba.

Ejemplo: *“Este test implementa la combinación de los bloques c1.bx, c2.by, c3.bz de las caracterizaciones c1, c2, c3”*.

Si encontráis dificultades relacionadas con la observabilidad o la controlabilidad al realizar alguno de los tests documentadlas.

Documentad los fallos que gracias a vuestros tests habéis encontrado en el SUT desarrollado por otras parejas de compañeros, indicando la url del repo de los compañeros que identifique la versión concreta de su SUT en el que habéis localizado un fallo, y qué test de los vuestros es el que habéis usado para ello.