



UNIVERSIDADE DA CORUÑA

PROGRAMACIÓN ORIENTADA A OBJETOS. CURSO 2008-2009

Boletín de Ejercicios (Instrucciones)

Para desarrollar correctamente los ejercicios y que su contenido sea tenido en cuenta en la evaluación de la asignatura es **obligatorio** seguir las siguientes instrucciones:

- Sobre los boletines de ejercicios
 - La **fecha límite de entrega** de los boletines se especifica en cada boletín, se harán entregas escaladas intentando fomentar el trabajo continuo.
 - Los boletines **se entregarán en los directorios destinados a tal efecto por el CECAFI** (en nuestro caso es \PRACTICAS\OPT\POO en los servidores xurxo y limia). El directorio P1 corresponde con el primer boletín, el P2 con el segundo y el P3 con el tercero. Para dudas sobre el procedimiento contactar directamente con el CECAFI. **NOTA IMPORTANTE: Esto puede cambiar una vez esté en marcha la nueva plataforma de la Facultade Virtual.**
 - Deberá entregarse **un proyecto NetBeans completo (no sólo los fuentes) por cada boletín y SIN COMPRIMIR.**
 - Aunque los ejercicios se ordenan, más o menos, según la materia vista en los distintos temas de teoría, **no es necesario seguir ningún orden predeterminado** al resolverlos.
- Estructura de los ejercicios
 - Los ejercicios son **individuales** y deberán de ser desarrollados mediante la herramienta **NetBeans** disponible en los ordenadores de prácticas.
 - Se deberá crear un único proyecto por boletín cuyo nombre será el login del alumno más un guión y el nombre del boletín, por ejemplo: “infabc00-b1”
 - Se creará un paquete por cada ejercicio del boletín con los nombres: ejercicio1, ejercicio2, etc. Sólo crear paquetes para los ejercicios que hayáis resuelto, **no crear paquetes vacíos si dentro no vais a incluir ningún ejercicio.**
- Comprobación de la ejecución correcta de los ejercicios
 - Cada ejercicio deberá llevar asociada una clase de prueba que permita al profesor comprobar que su funcionamiento es correcto. Es importante que esta prueba sea **NO INTERACTIVA**, es decir, **no debe preguntar nada al usuario.**
 - Para llevar a cabo esa prueba utilizaremos el **framework JUnit** que permite realizar tests de unidad en Java (en las transparencias de la asignatura tenéis un fichero que explica en detalle el funcionamiento de JUnit en NetBeans)
 - La clase de prueba, además de las instrucciones típicas de JUnit para comprobar el funcionamiento de la clase tiene que imprimir unos mensajes en la consola que

permitan comprobar visualmente el comportamiento de los tests. Además debe mostrar al principio del test un mensaje que indique qué ejercicio se está probando.

- A continuación se muestra el código de un test que comprueba si un año es bisiesto o no. Como veis se incluye una cabecera y diversas pruebas JUnit (assertTrue y assertFalse) acompañadas de su correspondiente mensaje en consola. Además el juego de prueba no tiene por qué ser exhaustivo pero tampoco tiene que limitarse a unos pocos casos. Tiene que intentar ser comprobar las distintas situaciones que pueden darse (por ejemplo, en el caso de los años bisiestos, si el año es divisible por 4, si el año es divisible por 100 o si el año es divisible por 400).

```
package ejercicio2;

import org.junit.*;
import static org.junit.Assert.*;

public class YearUtilitiesTest
{
    @Test
    public void testIsLeap()
    {
        System.out.println("*****");
        System.out.println("***** Boletín 1 - Ejercicio 2 *****");
        System.out.println("*****");
        System.out.println();
        System.out.println("El año 4 es bisiesto");
        assertTrue(YearUtilities.isLeap(4));
        System.out.println("El año 100 NO es bisiesto");
        assertFalse(YearUtilities.isLeap(100));
        System.out.println("El año 400 es bisiesto");
        assertTrue(YearUtilities.isLeap(400));
        System.out.println("El año 2007 NO es bisiesto");
        assertFalse(YearUtilities.isLeap(2007));
        System.out.println("El año 2008 es bisiesto");
        assertTrue(YearUtilities.isLeap(2008));
    }
}
```

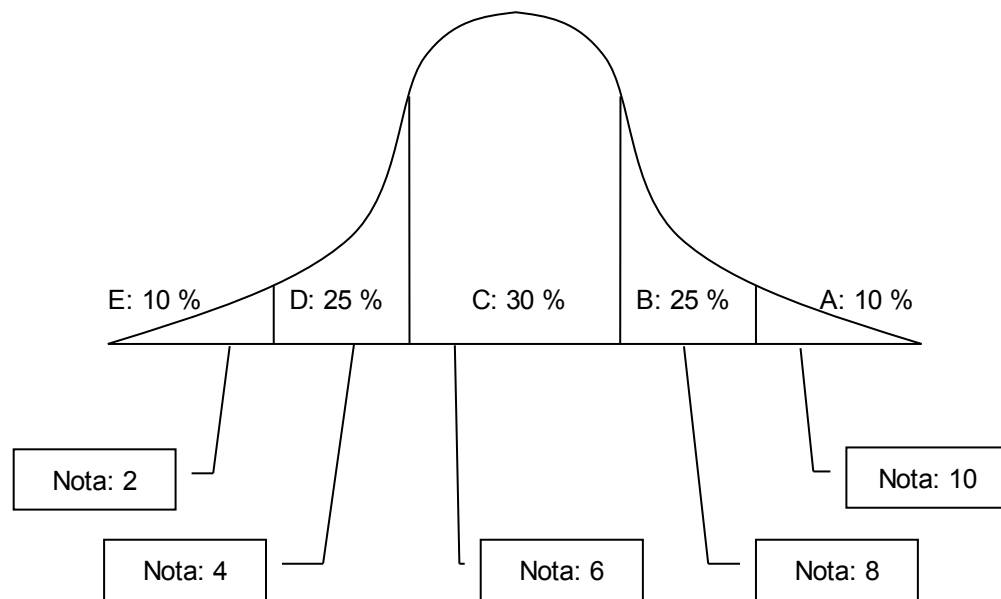
- **IMPORTANTE:** La inexistencia de la clase de test implicará que el ejercicio no se compute, aunque esté realizado correctamente.

- Aspectos a tener en cuenta

- No existe problema con que los nombres de las clases se repitan en distintos boletines porque al estar en distintos paquetes no surgirán conflictos. Si algún ejercicio reutiliza clases de otros ejercicios lo correcto es copiar esas clases en el nuevo ejercicio.
- Cada ejercicio indica el tema del que trata y, a veces, aporta datos que os serán de utilidad para su resolución. Podéis encontrar más información en el API de Java (descargándolo o accediendo a <http://java.sun.com/javase/6/docs/api/>).
- Para hacer más portable el proyecto entre distintas plataformas asegurados de que la codificación de los caracteres es UTF-8. Se puede cambiar en las propiedades del proyecto (pulsar con el botón derecho del ratón y escoger la opción en el menú contextual) en el apartado de Sources → Encoding. También podéis tratar de evitar el uso de caracteres especiales en el código y los comentarios.
- En la **Facultad Virtual** existe un **ejemplo de proyecto con varios ejercicios resueltos** que os puede servir de plantilla para crear vuestro proyecto.

- Evaluación de los ejercicios:

- La realización de los ejercicios contará **hasta tres puntos en la nota final** de la asignatura en la convocatoria ordinaria de Febrero.
- Los ejercicios son **voluntarios** pero de no realizarlos el alumno sabrá que está perdiendo puntos en la calificación de la asignatura.
- Hacer todos los ejercicios **de forma correcta** implica obtener la máxima nota.
- Para los que no hagan todos los ejercicios la asignación de notas se hará siguiendo los **criterios estadísticos que rigen los criterios ECTS**. Es decir, se ordenarán los alumnos según el número de ejercicios que hayan realizado y se les asignará la nota según el siguiente gráfico:



- Como **EJERCICIO OPCIONAL** se os propone hacer un informe de la **cobertura de vuestros tests** con la herramienta Cobertura. Si los tests alcanzan un nivel de cobertura cercano al 80% (se considera más fiable la cobertura de ramas condicionales) se os **gratificarán con DOS EJERCICIOS EXTRA en la suma de vuestros ejercicios**. Es condición necesaria que exista un objetivo (target) en el proyecto NetBeans denominado **“cobertura-coverage-report”** que genere el informe de cobertura y que funcione correctamente (acordaros de incluir la librería de cobertura en vuestro proyecto).
- Los ejercicios son **individuales** por lo tanto no se admite desarrollar ejercicios en grupos y luego realizar cambios “cosméticos” para diferenciarlos.
- **Todo ejercicio en el que haya sospechas fundadas de copias entre alumnos implicará LA ANULACIÓN DE TODO EL BOLETÍN (tanto para el alumno que lo haya desarrollado originalmente como para el que lo haya copiado).**
- Además ten en cuenta que si dejas circular un ejercicio que hayas realizado habrá más copias del mismo y por lo tanto tu nota “estadística” no será tan elevada. Deja el trabajo en grupo para la práctica final.
- Todos los ejercicios tienen el mismo valor (a no ser que se indique de otra forma en el propio ejercicio). Para evitar que sólo se hagan ejercicios de los boletines que resulten más sencillos se **penalizará el no realizar al menos 3 ejercicios de cada boletín** (restando 5 al número de ejercicios desarrollados).