

## Examen de Programación (Grado en Física, Grado en Matemáticas)

Junio 2011

### Primera parte (5 puntos, 50% nota del examen).

- 1) Escribe el código de una clase `PaquetePostal` que permite almacenar los datos de un paquete (destinatario y peso). La clase tendrá los campos indicados a continuación:

Atributos:

- Atributo NO estático: peso del paquete (en Kilos).
- Atributo NO estático: dirección del destinatario.
- Atributo estático: peso total de los paquetes creados.

Métodos:

- Constructor que permite inicializar los atributos no estáticos de la clase y que incrementa el peso total de los paquetes creados.
- Método estático que retorna el peso total de los paquetes creados.

Escribe además un programa sencillo (que no pida datos al usuario) que cree tres paquetes postales y muestre en la consola el peso total de los paquetes creados.

- 2) Escribir un método estático que recibe como parámetro un array de números reales en doble precisión y escribe en la consola una línea por cada número del array.

En cada línea aparece el índice del número en el array, si es positivo, negativo o cero y el valor del propio número con 3 decimales.

Por ejemplo para el array `[2.0, -34.34567, 28.0341, 0.0]` la salida deseada sería:

0	positivo	2.000
1	negativo	-34.346
2	positivo	28.034
3	cero	0.000

(La salida debe respetar la ordenación por columnas y el alineamiento).

- 3) Escribir un método con la cabecera:

```
public static int buscaMaximo(String numeros)
```

El método recibe un String formado por varios números enteros separados por comas y retorna el valor del mayor de los números.

Por ejemplo si el método es invocado con el String "20,34,8,25" debería retornar el número entero 34.

- 4) Se dispone de la clase `SensorHumedad` que permite medir el grado de humedad de una habitación. El código de dicha clase (sólo las partes relevantes para el problema) es:

```
class SensorHumedad {
    ... // atributos y métodos no relevantes para el problema

    /**
     * Retorna el grado de humedad leído por el sensor.
     * @return grado de humedad leído por el sensor
     * @throws LecturaIncorrecta cuando se ha producido un error al
     * tratar de leer.
     */
    public double leeHumedad() throws LecturaIncorrecta {
        ... // código no relevante para el problema
    }
}
```

Debido a problemas de funcionamiento del sensor en ocasiones se produce una lectura incorrecta y, para notificar esta situación, el método `leeHumedad` lanza la excepción `LecturaIncorrecta`.

Se pide escribir el código de un método de la clase `SensorHumedad` que, utilizando el patrón de excepciones recuperables, verifique el comportamiento descrito por el siguiente comentario de documentación:

```
/**
 * Invoca repetidamente el método leeHumedad hasta que se produzca
 * una lectura correcta o hasta que se alcance el número máximo de
 * intentos indicado.
 * @param numIntentos número máximo de intentos
 * @return la humedad retornada por la primera lectura correcta de
 * leeHumedad o Double.NaN en el caso de que todos los intentos
 * sean lecturas incorrectas.
 */
```