DISEÑO Y ANÁLISIS DE ALGORITMOS

Grado en Ingeniería del Software, 2ºcurso

ESTÁ PRÁCTICA SÓLO PUEDE SER REALIZADA POR QUIENES **NO** ASISTIERON AL LABORATORIO DE LA ASIGNATURA EL 22 DE MARZO DE 2013

PRÁCTICA 3/4

Esta práctica pertenece al tema 5 de la asignatura "Diseño y Análisis de algoritmos". Tiene como objetivo probar la capacidad del alumno al emplear la técnica de "Algoritmos Voraces" a la hora de diseñar algoritmos para obtener una solución.

La fecha límite de entrega es: **5 de abril de 2013, 15:00h**.

El modo de entrega será a través de **Campus Virtual** (apartado "Trabajos"). La práctica puede realizarse en grupos de hasta tres personas (sólo uno de los miembros del grupo realizará la entrega a través del Campus Virtual).

Se desea implementar el algoritmo de Huffman para obtener una codificación eficiente de una serie de caracteres o símbolos que se pasen como parámetro al algoritmo.

La cabecera de la función que implementa el algoritmo debe ser la siguiente:

```
public static Caracter[] huffman( Caracter[] caracteres )
```

La clase "Caracter" debe tener los siguientes atributos:

- "caracter" (char): será el símbolo al cual se desea aplicar una codificación binaria.
- "porcentaje" (float): será el porcentaje que supone el carácter en el total del texto.
- "codigo" (String): será el código binario calculado por el algoritmo para este símbolo o carácter.

Inicialmente, el campo del código estará siempre inicializado con una cadena vacía (longitud cero).

La clase "Caracter" podrá tener algún atributo más (que se deja a elección del alumno) que se pueda precisar para la manipulación de árboles cuyos nodos deberán ser instancias de la clase "Caracter", si bien pueden crearse otras clases necesarias para tal fin, que deberán aportarse en la entrega de la práctica.

Se pide por tanto diseñar e implementar en lenguaje Java un algoritmo que resuelva el problema planteado aplicando la mencionada clase "Caracter".

La clase principal deberá crear un vector cuyas posiciones contendrán en total todas las letras del abecedario en mayúsculas, así como el carácter punto y el carácter coma (en un primer momento se guardarán en orden alfabético). La clase principal debe asegurarse antes de pasarle el vector al método "Huffman" de que está ordenado en función de las apariciones de cada carácter.

Tras la ejecución del algoritmo, la clase principal mostrará por pantalla (terminal) el código binario generado para cada carácter con un formato similar al siguiente:

```
A = [1101]
B = [100110]
...
```

Material de entrega:

Se entregará un archivo comprimido (ZIP) que contendrá:

- **"Informe Práctica 3 XXXYYYZZZ.xxx"**: Informe de la práctica en formato DOC, DOCX, RTF o PDF, donde "XXXYYYZZZ" representa el número de expediente* de los hasta tres alumnos del grupo de trabajo. El informe deberá tener un primer apartado para explicar cómo se ha diseñado el algoritmo aportado. El segundo apartado qué atributos se han añadido para poder manejar las instancias de la clase "Caracter" en los árboles intermedios que maneja el algoritmo.
- **"Clase_Practica3_XXXYYYZZZ.java"**: Clase principal escrita en lenguaje Java que contendrá el método "main" y el método "Huffman" mencionado en el enunciado.
- **"Caracter.java"**: Clase mencionada en el enunciado.

El nombre del archivo comprimido será "Práctica3_XXXYYYZZZ.zip", siendo "XXXYYYZZZ" el número de expediente* de cada alumno del grupo alumno.

*- cada uno rellenado con ceros hasta tres dígitos.
Para el número de expediente 45 → "045".