

## Programación III

### 13/05/2015 - Primera fecha

**IMPORTANTE:** NO implemente las estructuras de datos vistas en clase, asuma que las estructuras existen y ya tienen implementados los métodos vistos en clase; no tienen ninguna otra operación adicional.

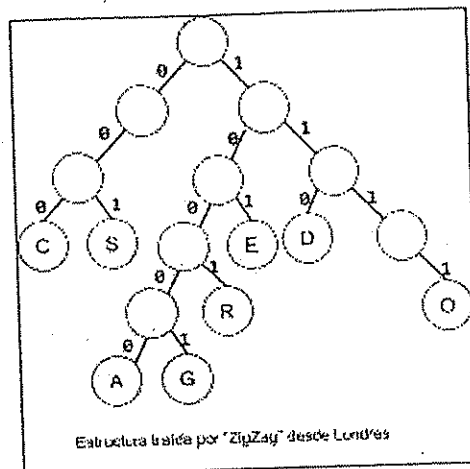
**Ejercicio 1.** El famoso espía Eddie Chapman, alias agente "ZigZag", está tratando de descifrar el mensaje que le llega desde Londres. Antes de dejar Londres, a "ZigZag" le explicaron cómo debía descifrar los mensajes que le enviaban. Los mensajes estaban formados por letras, cada letra es codificada como una secuencia de unos y ceros y de diferentes longitudes. Para poder descifrar el mensaje, a "ZigZag" le entregaron una estructura similar a la del gráfico.

En esta estructura, los 0 siempre están a la izquierda del nodo superior, y los 1 a la derecha del nodo superior. A modo de ejemplo, si "ZigZag" recibe el mensaje cifrado:

101 001 10001 1111 1001 110 1111

con la estructura traída desde Londres, el mensaje se descifra como **ESGORDO**.

Asuma que todas las secuencias cifradas que recibe "ZigZag" se corresponden con una letra, por ejemplo, "ZigZag" nunca recibe el mensaje cifrado: 00 01 porque ninguna de las secuencias 00 ni 01 se corresponden con alguna letra.



Su misión es escribir en una clase llamada **CodigoZigZag** (que NO contiene variables de instancia), un método con la siguiente firma:

en este espacio complete como parámetro la estructura que recibió "ZigZag"

```
public ListaGenerica<Character> descifrarCodigo(XXXX, ListaGenerica<String> listaDeSecuencias)
```

que sea capaz de descifrar un mensaje que ha sido cifrado como secuencias de ceros y unos. Es decir: dada la estructura traída por "ZigZag" y una lista de String (donde cada String está formado sólo por los caracteres 0 ó 1) ud. deberá devolver el mensaje descifrado, como una lista formada por letras. El programa debe funcionar para cualquier secuencia de 1 y 0, no sólo con las del ejemplo.

**Ejercicio 2.** En una ocasión el agente "ZigZag" no estaba seguro de que la estructura traída desde Londres, en realidad servía para descifrar a todas las letras. Por este motivo quiso imprimir de manera ordenada las letras contenidas en la estructura, sólo para poder verificar que no había letras faltantes. Su nueva misión es agregar en la clase **CodigoZigZag** un método con la siguiente firma:

en este espacio complete como parámetro la estructura que recibió "ZigZag"

```
public void imprimirLetrasOrdenadas(XXXX)
```

que imprima en orden alfabético las letras que están en la estructura que trajo "ZigZag". Para realizar este ejercicio **DEBERÁ USAR** alguna de las estructuras vistas en la práctica y que le ayude con el objetivo del ejercicio de imprimir en forma ordenada las letras. **NO PODRÁ** escribir ud. el método de ordenación (como una ordenación de listas o insertar ordenado en un arreglo), ni usar código de la API de Java que realice la ordenación (como `Arrays.sort(...)`).