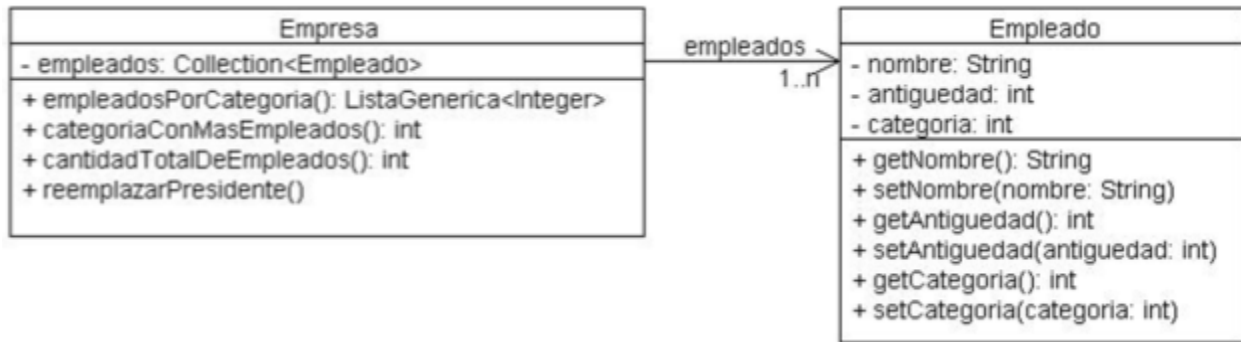


Programación III

TEMA 5: Árbol general

Práctica nº 5 - B

1. Considere la siguiente representación de la relación entre una empresa y sus empleados:



Si bien en el diagrama se muestra que la Empresa posee una colección de empleados, en realidad la relación entre empleados es de tipo jerárquica:

Categoría 1: Presidente

Categoría 2: Gerentes

Categoría 3: Sub-gerentes

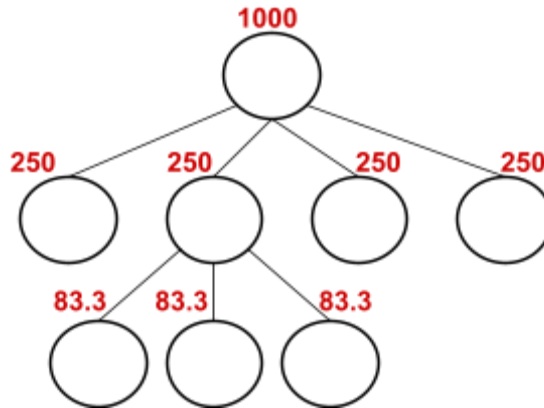
Categoría 4: Empleados que no tienen nadie a cargo

- a. Indique qué estructura de datos (en lugar de **Collection**) le permite representar esta organización jerárquica en la empresa.
- b. Realice la implementación de la empresa y sus empleados e implemente las siguientes operaciones:
- **empleadosPorCategoria(int categoria): int** Dado un número de categoría devuelve la cantidad de empleados de dicha categoría.
 - **categoriaConMasEmpleados():int** determina la categoría que cuenta con la mayor cantidad de empleados.
 - **cantidadTotalDeEmpleados():int** determina la cantidad total de empleados.
 - **reemplazarPresidente():** sea la situación en donde el presidente deja su función, reemplazarlo por la persona más antigua de sus subordinados, quién a su vez es reemplazada de la misma forma. **Pista:** Debe tomar los hijos de la raíz, y buscar el de mayor antigüedad de los hijos. Sin modificar la estructura, pase el mayor de los hijos a la raíz, y se envía el mensaje al hijo promovido. Cuando el hijo promovido no tenga hijos, se lo debe eliminar.

2. Red de Agua Potable

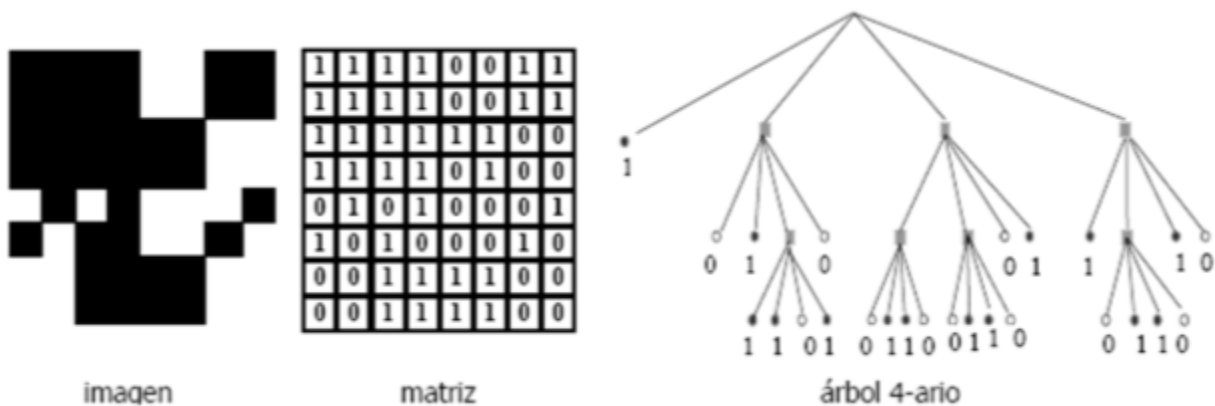
La red de agua potable comienza en un caño maestro y el mismo se va dividiendo sucesivamente hasta llegar a cada una de las casas. Por ejemplo, si por el caño

maestro ingresan 1000 litros, a medida que el caño se divide, el caudal se divide en partes iguales en cada una de las divisiones. Es decir, si el caño maestro se divide en 4 partes, cada división tiene un caudal de 250 litros. Luego, si una de esas divisiones se vuelve a dividir en 3 partes, cada una tendrá un caudal de 83,3. La situación descripta se puede modelar de la siguiente forma a través del siguiente árbol general:

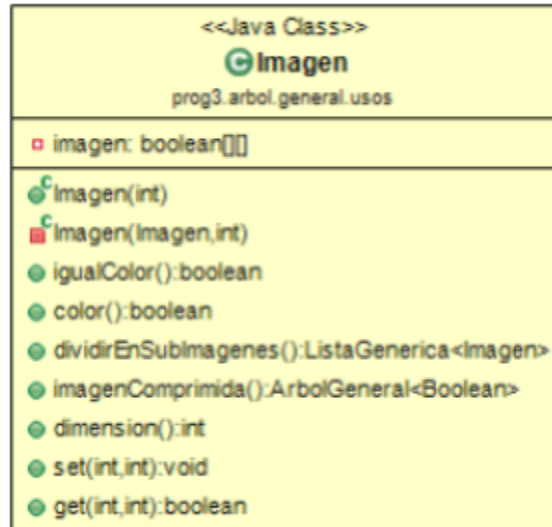


Usted debe implementar un método en la clase **RedDeAguaPotable**, que reciba una “**configuración**” con la forma de la red de agua potable y **n litros** que son los que ingresan por el caño maestro; el método calcula y devuelve cuál es el mínimo caudal que recibe una hoja.

3. **Árbol 4-ario.** Una forma de comprimir una imagen (representada como una matriz de puntos) es transformarla a un árbol 4-ario. El algoritmo es el siguiente:
 - a. Si toda la matriz tiene un mismo color, se debe definir un nodo con ese color.
 - b. En caso contrario, se divide la matriz en cuatro partes, se define un nodo con 4 hijos, y cada hijo es la conversión de cada una de las partes de la matriz.



Implemente la clase Imagen de acuerdo al siguiente diagrama:



- a. Realice un método de instancia llamado **imagenComprimida()** que devuelva la representación de la imagen en su árbol correspondiente (es decir, la matriz de booleanos se representará como árbol general).

NOTA: Teniendo en cuenta que el color blanco lo representamos con el valor falso y el negro con verdadero, implemente los métodos *igualColor()*, *color()* y *dividirEnSubimagenes()* en la clase *Imagen*.

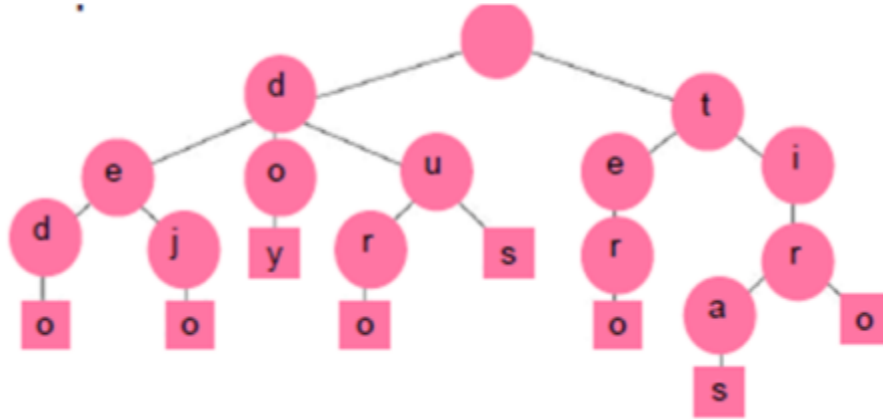
- El método *igualColor()* devuelve true si la matriz que representa la imagen posee todas sus celdas del mismo color.
- El método *color()* devuelve true si toda la matriz que representa la imagen posee el valor true (negro) y false si toda la matriz posee el valor false (blanco). Este método sólo será invocado cuando la Imagen tiene "igual color".
- El método *dividirEnSubimagenes()* devuelve una Lista con 4 Imágenes.
- Método *imagenComprimida()*: Para comprimir la imagen, debe verificar si la misma posee igual color, en ese caso debe crear y devolver un Árbol General con un único nodo. En cambio, si la imagen no posee el mismo color, debe *dividirEnSubImagenes* y enviar el mensaje *imagenComprimida()* a cada una de las nuevas Imágenes.

4. **TRIEs == re-trie-val trees.**

Definición de un TRIE

- Es una estructura de datos que permite representar conjuntos de cadenas de caracteres.
- Cada nodo de T, excepto la raíz, está etiquetado con un símbolo del alfabeto
- Los hijos de un nodo interno de T están ordenados según el ordenamiento en el alfabeto
- Cada hoja marca el final de una cadena

Una aplicación frecuente de los TRIES es el almacenamiento de diccionarios, como los que se encuentran en los teléfonos móviles.



- a. Implemente un TRIE con la siguiente operación:

public void agregarPalabra(String palabra)

Dada una palabra ingresada como parámetro, el método **agregarPalabra** deberá agregar cada uno de los caracteres que la conforman de manera de representar dicha palabra en el TRIE

- b. Agregue a la implementación del TRIE la siguiente operación:

public List<StringBuilder> palabrasQueEmpiezanCon(String prefijo)

Devuelve una lista con todas las posibles palabras que comienzan con la cadena de caracteres ingresada como parámetro