TALLER 4 ARBOLES DE PARTICIÓN

LUIS FELIPE AYALA URQUIZA
NICOLÁS ESTEBAN BAYONA ORDOÑEZ
JUAN SEBASTIÁN HERRERA GUAITERO

ANDREA DEL PILAR RUEDA OLARTE
PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
ESTRUCTURAS DE DATOS
BOGOTA D.C
17 DE ABRIL DE 2021

Índice

TAD NodoQUAD

Conjunto mínimo de datos:

- data, dato de tipo entero, representa el dato que almacena el nodo.
- upLeft, apuntador a NodoQUAD, me representa el hijo superior izquierdo del nodo.
- upRight, apuntador a NodoQUAD, me representa el hijo superior derecho del nodo.
- downLeft, apuntador a NodoQUAD, me representa el hijo inferior izquierdo del nodo.
- downRight, apuntador a NodoQUAD, me representa el hijo inferior derecho del nodo.

Comportamiento (Operaciones):

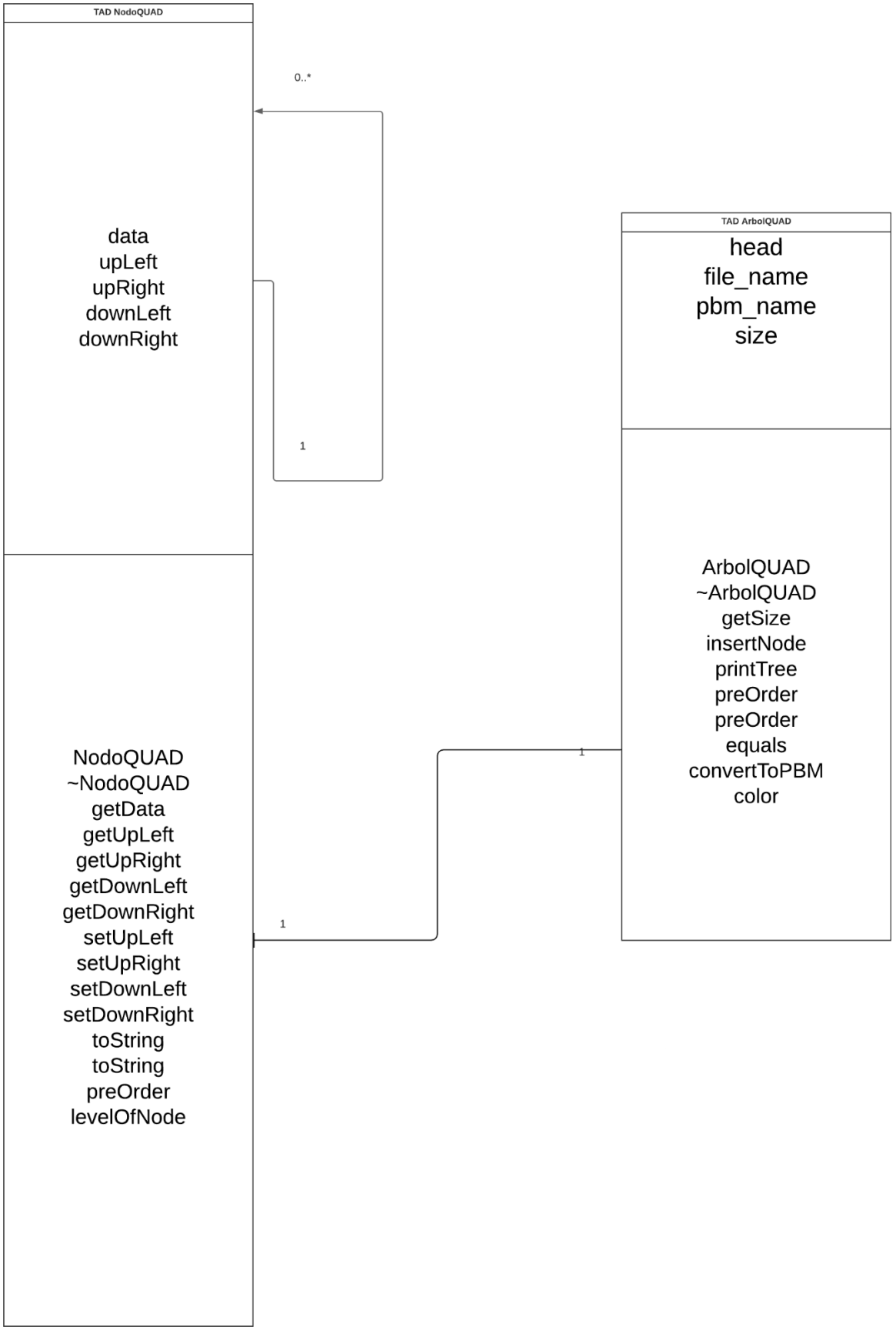
- NodoQUAD(data), crea un objeto de la clase NodoQUAD inicializando la data con el elemento pasado como parámetro.
- ~NodoQUAD(), define como se destruye el objeto de la clase NodoQUAD.
- getData(), retorna la data almacenada por el nodo
- getUpLeft (), retorna el hijo superior izquierdo del nodo.
- getUpRight (), retorna el hijo superior derecho del nodo.
- getDownLeft (), retorna el hijo superior izquierdo del nodo.
- getDownRight (), retorna el hijo superior derecho del nodo.
- setUpLeft (data), actualiza el hijo superior izquierdo del nodo con la data pasada como parámetro.
- setUpRight (data), actualiza el hijo superior derecho del nodo con la data pasada como parámetro.
- setDownLeft (data), actualiza el hijo superior izquierdo del nodo con la data pasada como parámetro.
- setDownRight (data), actualiza el hijo superior derecho del nodo con la data pasada como parámetro.
- toString(level=0, car=”Head “), imprime un nodo con sus respectivos hijos.
- toString(unique), imprime un nodo solo.
- preOrder(num), retorna una lista con el recorrido preOrden del nodo.
- levelOfNode(node, level=0), retorna el nivel de un nodo específico.

TAD ArbolQUAD
Conjunto mínimo de datos:

- head, apuntador a NodoQUAD, me representa la cabeza del árbol.
- file_name, apuntador a carácter, representa el nombre del archivo que almacena la información necesaria para llenar los datos del árbol.
- pbm_name, apuntador a carácter, representa el nombre del archivo que donde se debe guardar la información de la imagen.
- size, cadena de caracteres que representa el tamaño de la imagen.

Comportamiento (Operaciones):

- ArbolQUAD(file_name, pbm_name), crea un objeto de la clase ArbolQUAD, inicializando el campo file_name y el campo pbm_name con la data pasada como parámetro, además, este llena el árbol con la data leída desde el archivo file_name.
- ~ArbolQUAD(), define como se destruye un objeto de la clase ArbolQUAD.
- getSize(), retorna el tamaño de la imagen que almacena el árbol.
- insertNode(data), inserta un nodo inicializado con la data pasada como parámetro al árbol.
- printTree(), retorna la impresión completa del árbol.
- preOrder(string), retorna una cadena de carácteres con el recorrido inOrden del árbol completo.
- preOrder(num), retorna una lista con el listado de nodos del recorrido inOrden del árbol completo.
- convertToPBM(), retorna una cadena de caracteres conteniendo la imagen binaría.
- equals(), retorna si la data contenida en el árbol es igual a la data del archivo file_name.
- color(), guarda en una matriz la imagen pbm generada usando el recorrido en preorden para generarla.


TAD NodoQUAD
data
upLeft
upRight
downLeft
downRight
NodoQUAD
~NodoQUAD
getData
getUpLeft
getUpRight
getDownLeft
getDownRight
setUpLeft
setUpRight
setDownLeft
setDownRight
toString
toString
preOrder
levelOfNode
0..*
1
1
1
TAD ArbolQUAD
head
file_name
pbm_name
size
ArbolQUAD
~ArbolQUAD
getSize
insertNode
printTree
preOrder
preOrder
equals
convertToPBM
color

3. Diferencia entre los archivos

La principal diferencia de los archivos PBM contra los archivos de recorrido en preorden es el tamaño, ya que el archivo de recorrido en preorden esta mucho mas comprimido que el resultado final en PBM, el archivo PBM es la representación del recorrido pre-orden en ceros y unos.

4. Descripciones de las imágenes reconstruidas a partir de los archivos de texto de prueba entregados junto con el presente enunciado

No pudimos reconstruir las imágenes en su totalidad, ya que habían sectores que nos estaban dando problemas, sin embargo, logramos persuadir las figuras de una que otra imagen,sobre todo en las imágenes de mayor tamaño solían tener figuras de personas animadas. El archivo 00 era una imagen 8x8 en la que cada región tiene un color.