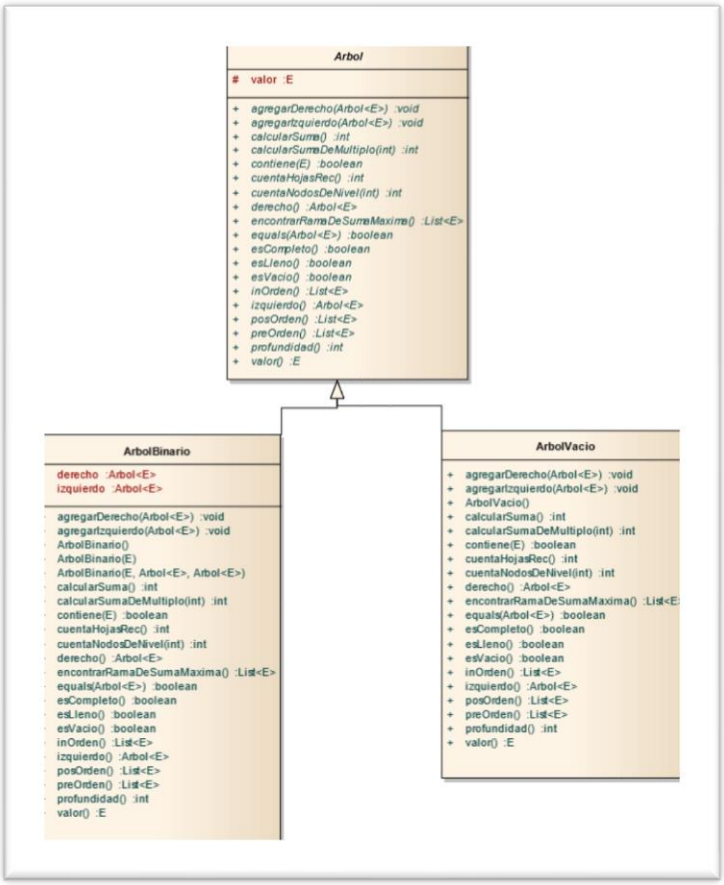
Guía de trabajos prácticos

Árboles – Árboles Binarios

Ejercicio 1:

Dado el siguiente modelo de árbol, recursivo..



Implemente en las subclases de **Arbol, ArbolVacio y ArbolBinario** los siguientes métodos.

a) `public abstract boolean contiene(E unValor)` → retorna true si un elemento existe en el árbol.
b) `public abstract boolean equals(Arbol2<E> unArbol)` :método recursivo que retorna true si un árbol binario es idéntico a un recibido como parámetro.
c) `public abstract int profundidad()`:método recursivo que cuenta las altura de un árbol binario.
d) `public int cuentaHojasRec()`:método recursivo que cuente las hojas de un árbol binario.
e) `public int cuentaNodosDeNivel(int nivel)` : método que determina el número de nodos que se encuentran en un nivel N de un árbol.

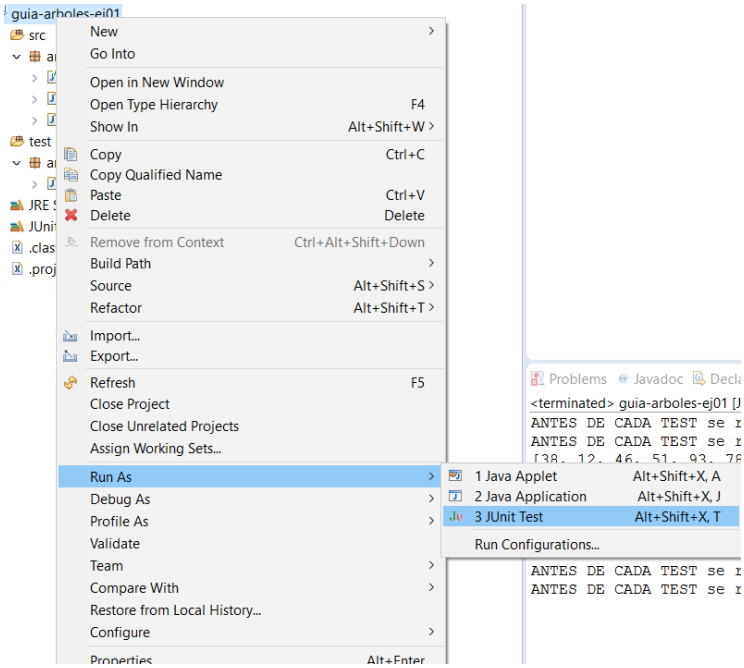
f) `public boolean esLleno()`: método que determina si el árbol binario es lleno[i]

g) `public boolean esCompleto()`: método que determina si el árbol binario es completo[ii]

h) `public int calcularSuma()`: método que retorna la suma de todos los nodos del árbol (suponiendo que todos los nodos son de tipo entero)

i) `public List<E> camino(E v1,E v2)`: retorna el camino entre v1 y v2, si existe o null si no existe.

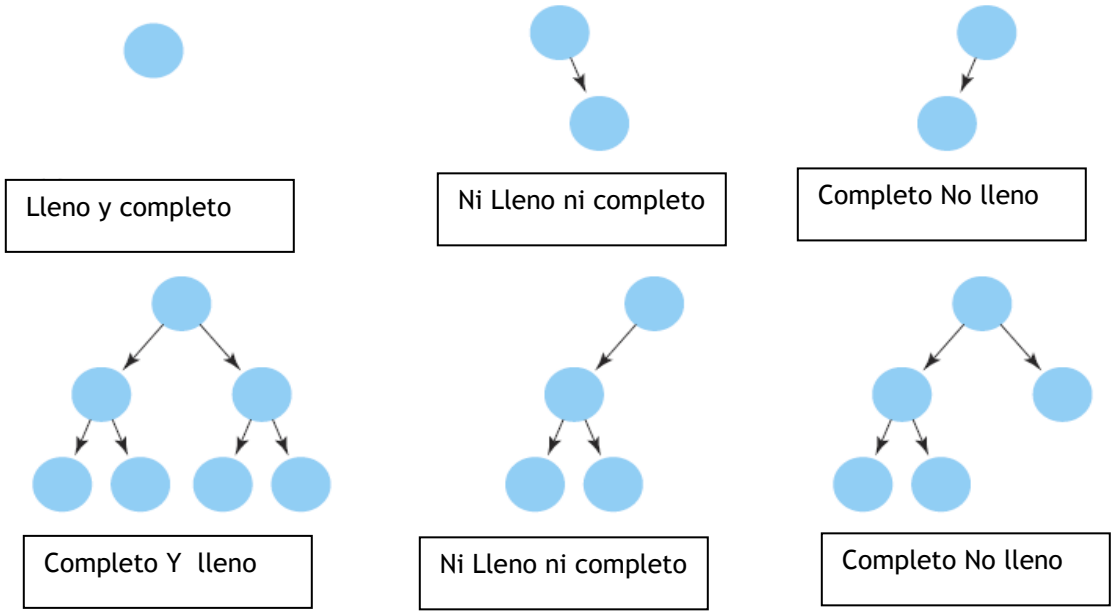
Puede implementar el código en el proyecto eclipse que se le entrega para desarrollarlo. Este proyecto tiene una clase de testing que permite probar el código desarrollado. Cuando haya resuelto todos los métodos puede ejecutar el proyecto seleccionando "Run→ Run as→ Junit Test" y le mostrará información acerca de la validez del código desarrollado



[i] Un árbol binario de nivel N es **lleno** cuando el máximo número de nodos permitidos en cada uno de los niveles..

[ii] Un árbol binario de nivel N es **completo** cuando para cada nivel desde el nivel 0 al nivel n-1, tiene un conjunto lleno de nodos (es decir tiene el máximo número de nodos permitidos para ese nivel), y en el nivel n, todos los nodos hoja ocupan las posiciones más a la izquierda del árbol.

Ejemplos:



Espejar un BTree

Ejercicio 2:

Implemente en la clase ArbolBinario y ÁrbolVacío los siguientes métodos

- ***a)*** `public boolean esSubArbol(Arbol otroArbol)`: indica si otro árbol es un subárbol del pasado por parámetro.
- ***b)*** `public Arbol espejar()`: retorna el mismo árbol binario pero intercambia los hijos a izquierda y derecha (ver ejemplo).
- ***c)*** `public List<E> recorrerPorNivel()` : muestra los nodos ordenados por nivel.
- ***d)*** `public boolean calcularSumaDeMultiplos(int n)`: método que retorna la suma de todos los nodos del árbol que son múltiplos del parámetro "n" (suponiendo que todos los nodos son de tipo entero).
- ***e)*** `public List<E> encontrarRamaDeSumaMaxima()`:muestra la rama (derecha o izquierda) cuya suma es máxima.

Agregue también 2 métodos cuya firma debe definir con el siguiente objetivo

a) un método que le permita determinar si el árbol binario es un árbol de búsqueda binario.

b) encuentre desde la raíz, la rama de mayor profundidad.

Ejercicio 3:

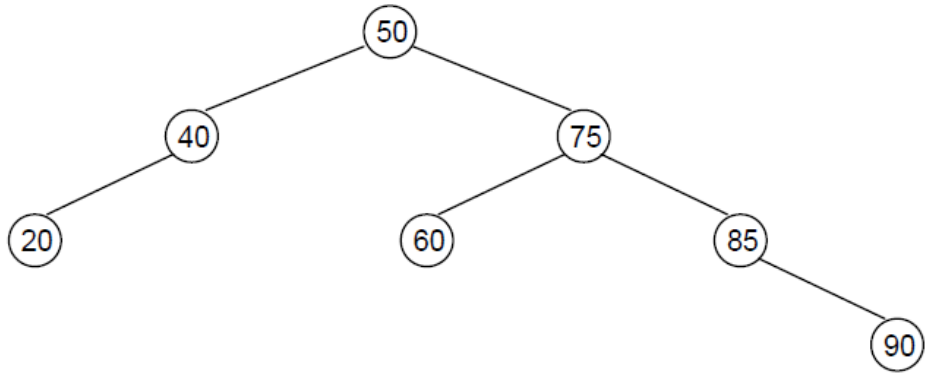
Construir un método recursivo para escribir todos los nodos de un ***árbol binario de búsqueda:***

a) Cuyo campo clave sea mayor que un valor dado.

b) Cuyo campo clave esté en comprendido dentro del rango de un valor inferior y uno superior.

Ejercicio 4:

Dado el siguiente árbol AVL, que almacena Valores Enteros.



a) Indique el factor de equilibrio de cada nodo del árbol:

Nodo	FE
50	
40	
10	
75	
60	
85	
90	

b) Dibuje el mismo árbol luego de realizar cada una de las siguientes operaciones (siempre comienze con el árbol de la figura, no con el que resulta de acumular las operaciones):

a. Insertar la clave 10.
b. Insertar la clave 95

c. Insertar la clave 80 y luego la clave 77.
d. Insertar la clave 80 y luego la clave 83
e. Insertar la clave 45
f. Insertar la clave 14 y luego borrar la clave 14.
g. Insertar la clave 30 y luego borrar la clave 30.
h. Insertar la clave 88 y luego borrar la clave 88.
i. Insertar la clave 93 y luego borrar la clave 93.

c) ¿Luego de ejecutar las operaciones f), g), h) , e i), el árbol queda de la misma manera que antes de ejecutarlas?

Ejercicio 5:

a) Dada la secuencia: 5 -10 – 15- 20 -23- 28 – 30 – 40
   i. Muestre el árbol binario de búsqueda correspondiente
   ii. Muestre el árbol binario AVL correspondiente
b) Dada la secuencia 4 19 -7 49 100 0 22 12
   i. Muestre el árbol binario de búsqueda correspondiente
   ii. Muestre el árbol AVL correspondiente.

Ejercicio 6:

Dado un ***árbol binario de búsqueda***, que contiene un número impar de claves, ninguna de las cuales está repetida, según la siguiente declaración:

a) Diseñe un método que permita retornar la clave del nodo del árbol que contiene la mediana (clave que tiene tantas claves menores que ella como claves mayores que ella). No utilice ninguna estructura de datos auxiliar.
b) Diseñar un método que liste los nodos del árbol ordenados descendentemente.
c) Diseñar un método que retorne un entero “int nivel(Comparable x)” que determina en qué nivel se encuentra un elemento.
d) Diseñar un método “boolean existe(Integer x,Integer y)” que retorna “true” si en el árbol existen dos elementos:
   a. el elemento “x” existe en el árbol
   b. el elemento “x^y” existe en el árbol.

Ejercicio 7:

Dado un árbol AVL

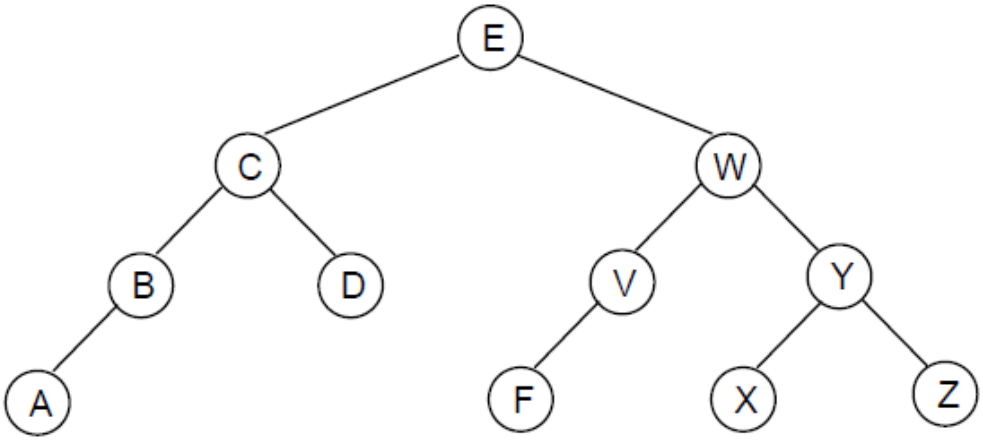
a) Dibujar la estructura del árbol que se produce luego de insertar en el orden en que aparecen los valores: 14,6,24,35,17,21,32,4,7,15,22.
b) Al árbol del punto a) eliminarle el nodo raíz. Hacerlo tantas veces como sea necesario hasta que se desequilibre un nodo y se deba aplicar rotación simple.

c) Mostrar un ejemplo donde la misma secuencia de valores ingresados, pero en distinto orden, genere dos árboles AVL distintos.

Ejercicio 8:

Escribir un programa que lea un texto de longitud indeterminada y que produzca como resultado la lista de todas las palabras diferentes contenidas en el texto ordenadas alfabéticamente, así como su frecuencia de aparición.
a) Hacer uso de la estructura de árbol binario de búsqueda para localizar cada nodo del árbol que tenga una palabra y su frecuencia.

Ejercicio 9:



a) Dibuje el mismo árbol luego de realizar cada una de las siguientes operaciones (siempre comienze con el árbol de la figura, no con el que resulta de acumular las operaciones):
    a. Borrar D.
    b. Borrar V y luego F
    c. Borrar E
    d. Borrar W
b) Dada la secuencia: 3 -7 – 12- 4 -6- 5 – 16 – 10– 15– 14
    a. Muestre el árbol binario de búsqueda correspondiente
    b. Muestre el árbol AVL correspondiente.
    c. Remueva el valor 7 y muestre ambos árboles.
c) Dada la secuencia 50, 25, 75, 10, 40, 60, 90, 35, 45, 70, 42.
    a. Muestre el árbol binario de búsqueda correspondiente
    b. Muestre el árbol AVL correspondiente.
d) Dada la secuencia 10, 75, 34, 22, 64, 53, 41, 5, 25, 74, 20, 15, 90.
    a. Muestre el árbol binario de búsqueda correspondiente
    b. Elimine el valor 25
    c. Muestre el árbol AVL correspondiente.

   d. Elimine la raíz 2 veces
e) dada la secuencia: 50 72 96 94 107 26 12 11 9 2 10 25 51 16 17 95
   a. Muestre el árbol binario de búsqueda correspondiente
   b. Muestre el árbol binario AVL
   c. Al árbol AVL borrarle el valor 51. Luego borrar el valor 94. Luego borrar el valor 26.

Ejercicio 10:

a) Dibuje un montículo luego de insertar la secuencia 15 – 18 – 3 – 7 – 41- 27 – 16 – 8 – 14 -12
b) Realice 2 operaciones eliminar
c) Inserte el valor 31 y luego 20
d) Realizar 3 operaciones eliminar

Ejercicio 11:

El recorrido de un determinado árbol binario es:

- en preorden GEAIBMCLDFKJH
- en inorden IABEGLDCFMKHJ.

iii. Dibujar el árbol binario.
iv. Dar el recorrido en postorden.
v. Diseñar método para dar el recorrido en postorden dado el recorrido en preorden e inorden