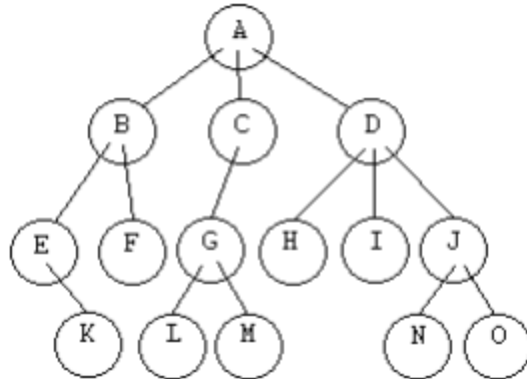


Arboles Binarios:

Un árbol binario, requiere de una estructura de NODO que permita almacenar el dato correspondiente, además de una referencia al hijo izquierdo y una más al hijo derecho. El árbol será una liga al nodo raíz, a partir del cual, se podrá acceder al resto de la estructura.



Recorridos:

Pre-orden:

En este tipo de recorrido, el valor del nodo se procesa antes de recorrer las ramas:

```
void PreOrden(NodoDeArbol a) {  
    if (a == NULL) return;  
    Procesar(dato); //método que lee el dato  
    RecorrerArbol(a.izq);  
    RecorrerArbol(a.centro);  
    RecorrerArbol(a.der);  
}
```

Si seguimos el árbol del ejemplo en pre-orden, y el proceso de los datos es sencillamente mostrarlos por pantalla, obtendremos algo así: A B E K F C G L M D H I J N O

In-orden:

En este tipo de recorrido, el valor del nodo se procesa después de recorrer la primera rama y antes de recorrer la última. Esto tiene más sentido en el caso de árboles

binarios, y también cuando existen ORDEN-1 datos, en cuyo caso procesaremos cada dato entre el recorrido de cada dos ramas (este es el caso de los árboles-b):

```
void InOrden(NodoDeArbol a) {  
    if (a == NULL) return;  
    RecorrerArbol(a.izq);  
    Procesar(dato); //método que lee el dato  
    RecorrerArbol(a.centro);  
    RecorrerArbol(a.der);  
}
```

Si seguimos el árbol del ejemplo en in-orden, y el proceso de los datos es sencillamente mostrarlos por pantalla, obtendremos algo así: K E B F A L G M C H D I N J O

Post-orden:

En este tipo de recorrido, el valor del nodo se procesa después de recorrer todas las ramas:

```
void PostOrden(NodoDeArbol a) {  
    if (a == NULL) return;  
    RecorrerArbol(a.izq);  
    RecorrerArbol(a.centro);  
    RecorrerArbol(a.der);  
    Procesar(dato); //método que lee el dato  
}
```

Si seguimos el árbol del ejemplo en post-orden, y el proceso de los datos es sencillamente mostrarlos por pantalla, obtendremos algo así: K E F B L M G C H I N O J D A

Fuente:

http://www.utim.edu.mx/~svalero/docs/ED_Java.pdf?fbclid=IwAR13yzgRJHsx6aN7IH7jfgymCpio05jG71K2YqGrGON83ssbLNu4Sl5zxxk

Referencias del autor:

- [1] Nell Dale, Object-oriented Data Structures using Java, Jones and Bartlett Publishers, 2002.
- [2] Robert Lafore, Data Structures and Algorithms in Java, Sams, 1998.