Práctica 2A: Análisis teórico de los TADs AB y ABB

Contenido

[1) TAD AB 1](#_Toc483495516)

[a) Corrección TAD AB 1](#_Toc483495517)

[b) Operaciones básicas con TAD AB 1](#_Toc483495518)

[2) TAD ABB 1](#_Toc483495519)

[a) Corrección TAD ABB 1](#_Toc483495520)

[b) Operaciones básicas TAD ABB 1](#_Toc483495521)

# TAD AB

## Corrección TAD AB

Corregimos el TAD modificando y añadiendo los siguientes metodos:

* Constructores EnlazadoArbolBinario.- Modificamos el constructor que recibe como parametros el elemento, hijo izquierdo e hijo derecho para que nos permita crear árboles con el elemento y sus dos hijos asi como tambien, con el elemento y un solo hijo sea izquierdo o derecho.
* Ademas añadimos otro constructor que nos deje crear un árbol solo con elemento y sin hijos.
* Métodos InOrder y PostOrder.- Agregamos estos métodos por que solo tenia el metodo PreOrder para listar los nodos.
* Método anchura: Modificamos el método para utilizar la cola integrada de java

(java.util.Queue).

## Operaciones básicas con TAD AB

* Devolver nodo almacenado en la raíz.
* Comprobar si el árbol está vacio.
* Devolver árbol izquierdo.
* Devolver árbol derecho.
* Comprobar si un elemento está en el árbol.
* Asignar objeto a la raíz del árbol.
* Asignar un árbol como subarbol izquierdo del arbol raiz.
* Asignar un árbol como subarbol derecho del arbol raiz.
* Vaciar el árbol.

# TAD ABB

## Corrección TAD ABB

## Operaciones básicas TAD ABB

* Devolver nodo almacenado en la raíz.
* Comprobar si el árbol está vacio.
* Devolver árbol izquierdo.
* Devolver árbol derecho.
* Insertar un objeto al árbol.
* Eliminar un objeto del árbol.
* Buscar objeto en el árbol.
* Comprobar si un elemento está en el árbol.
* Asignar objeto a la raíz del árbol.
* Asignar un árbol como subarbol izquierdo del arbol raiz.
* Asignar un árbol como subarbol derecho del arbol raiz.
* Vaciar el árbol.