Actividad 2. Uso del TAD Árbol Binario

Objetivo

Saber usar el TAD árbol binario para resolver problemas de programación.

Procedimiento

1. Ver el video o leer la presentación sobre árboles binarios que están disponibles en Moodle, Tema 1/ Sección 1.2 TAD Árbol Binario/ Recursos didácticos.
2. Resolver la relación de cinco ejercicios propuestos para autoevaluación con el objetivo de conocer el nivel de comprensión del tema. Para resolver los ejercicios es necesario conocer la interfaz del TAD árbol binario, disponible en el Anexo. Para su implementación se hará uso de la librería *aed2.jar* (disponible en Moodle).
3. Para probar el correcto funcionamiento de los métodos se puede hacer uso del test disponible en Moodle, Tema 1/ Sección 1.2 TAD Árbol Binario/ Actividades Grupo Grande.

Evaluación

Estos contenidos serán evaluados mediante una prueba individual el 21 de octubre de 2022

Tiempo estimado

3 horas

Ejercicios para resolver en clase

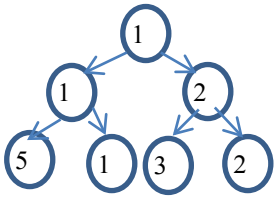
1. Escribe un método que dado un árbol binario devuelva verdadero si el árbol es **completo** y falso en otro caso. Un árbol binario es completo si todos sus nodos tienen dos descendientes, excepto las hojas.
2. Escribe un método que dados dos árboles binarios A y B, determine si son **idénticos** o no.
3. Escribe un método que **cuente** el número de nodos que están en el **nivel** *n* de un árbol binario.
4. Escribe un método que dado un árbol binario A, cree un árbol binario B igual que A pero **sin las hojas**.
5. Escribe un método que calcule la **altura** de un árbol binario.
6. Escribe un método que dados los recorridos en preoden e inorden de un árbol binario, reconstruya el árbol. Suponemos que los recorridos son String y que no hay caracteres repetidos.

Otros ejercicios

7. Un **árbol de selección** es un árbol binario donde cada nodo representa al menor de sus dos hijos, excepto las hojas. Construir un método que, dado un árbol binario, indique si es o no un árbol de selección.

public static <E extends Comparable<E>> boolean arbolSeleccion (ArbolBinario<E> arbol)

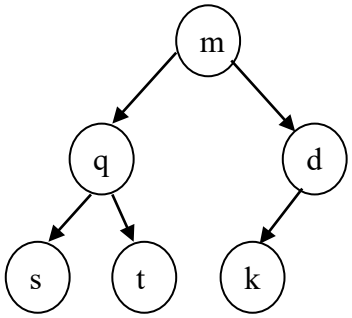
Ejemplo:



8. Escribe un método booleano que dados un árbol binario y un camino expresado en forma de String determine si existe dicho **camino** en el árbol, teniendo en cuenta que el camino debe comenzar necesariamente en la raíz.

   public static <E> boolean esCamino(ArbolBinario<E> arbol, String camino)

   Por ejemplo, para el árbol que sigue existen los caminos m-q-t y m-d, pero no existen los caminos r-q-t ni d-k.



9. Escribe un método que dado un árbol binario y un nivel n del árbol, realice una **copia** del árbol hasta dicho nivel.

   public static <E> ArbolBinario<E> copiaHastaNivel(ArbolBinario<E> a, int nivel)

10. Un elemento de un árbol se dirá de nivel k si aparece en el árbol a distancia k de la raíz. Escribe un método que determine si un elemento está a **distancia k**.

    public static <E> boolean nivelK (ArbolBinario<E> a, E elem, int k)

Anexo:

- **TAD Árbol Binario:**

```
public interface ArbolBinario<E>{
   public boolean esVacio();
   public E raiz() throws ArbolVacioExcepcion;
   public ArbolBinario<E> hijoIzq()throws ArbolVacioExcepcion;
   public ArbolBinario<E> hijoDer()throws ArbolVacioExcepcion;
   public boolean esta(E elemento);
   public void setRaiz(E elemRaiz) throws ArbolVacioExcepcion;
   public void setHijoIzq(ArbolBinario<E> hi) throws ArbolVacioExcepcion, NullPointerException;
   public void setHijoDer(ArbolBinario<E> hd) throws ArbolVacioExcepcion, NullPointerException;
   public void suprimir();
}

public class EnlazadoArbolBinario<E> implements ArbolBinario<E>{
    public EnlazadoArbolBinario(){…}
    public EnlazadoArbolBinario(E elemento, EnlazadoArbolBinario<E> hi, EnlazadoArbolBinario<E> hd) {…}
    …
}
```