

Actividad 5. Uso del TAD Árbol General

Objetivo

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

Procedimiento

1. Leer y comprender los apuntes de árboles generales que están disponibles en **Tema**, sección **Documentos e Ligazóns / Teoría / ÁrbolGeneral.pdf**
2. Emplear técnicas de aprendizaje colaborativo para resolver los ejercicios que se indican en esta actividad, utilizando el lenguaje java. En concreto se utilizará la **técnica del rompecabezas**.
 - a. Se formarán grupos base de 6 personas.
 - b. Cada miembro del grupo será responsable de la resolución de un ejercicio (trabajo individual no presencial).
 - c. En clase se reunirán los expertos de cada ejercicio para puesta en común.
 - d. Después cada persona volverá a su grupo base para explicarles a cada compañero lo que aprendieron.
3. Los *otros ejercicios propuestos* se trabajarán de manera no presencial. Para probar su correcto funcionamiento se puede hacer uso del test disponible en Tema, sección Documentos e Ligazóns / Actividades / Test / **Actividad5Test.java**
4. Para resolver todos los ejercicios es necesario conocer la interfaz del TAD árbol general, disponible en el anexo.

Evaluación

El aprendizaje colaborativo se evaluará mediante una prueba grupal el 20 de octubre.

Estos contenidos serán evaluados mediante una prueba individual el 3 de noviembre.

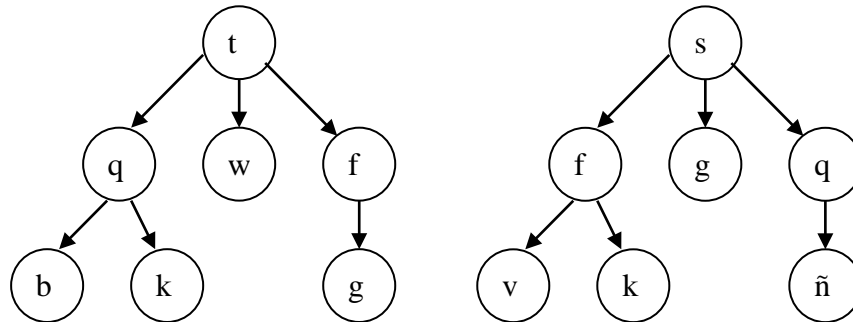
Tiempo estimado

6 horas

Ejercicios

1. Escribe un método que dada una referencia a un árbol general de enteros, calcule la **suma** de los valores almacenados en sus nodos.

2. Escribe un método booleano que dados dos árboles generales determine si tienen la **misma estructura**. Por ejemplo, los árboles generales que siguen tienen la misma estructura, aunque, como puede observarse, no coincidan los valores que se almacenan en los nodos.



3. Escribe un método que dado un árbol general, devuelva cierto si se trata de un **árbol 2-3**. Un árbol 2-3 es aquel en el que cada nodo tiene exactamente 2 o 3 hijos, excepto las hojas.
4. Un árbol de **selección** es un árbol donde cada nodo representa al menor de sus hijos, excepto las hojas. Construir un método que, dado un árbol general, indique si es o no un árbol de selección.
5. Escribe un método que determine el **nivel** de un elemento concreto en el árbol general.
6. Escribe un método que devuelve el **grado** de un árbol general. El grado de un árbol es el máximo de los grados de sus nodos.

Otros ejercicios propuestos

7. Escribe un método que devuelva la **altura** de un árbol general.
- ```
public static <E> int altura(ArbolGeneral<E> a)
```
8. Haciendo uso de una cola, escribe un método que muestre el recorrido en **anchura** de un árbol general.
- ```
public static <E> void anchura(ArbolGeneral<E> a)
```
9. Escribe un método que dado un árbol general, devuelva cierto si se trata de un **cuadArbol**. Un cuadArbol es aquel en el que cada nodo tiene exactamente 4 hijos, excepto las hojas.
- ```
public static <E> boolean cuadArbol(ArbolGeneral<E> arbol)
```
10. Escribe un método que cuente el **número de nodos pares** en un árbol de enteros que se pasa como parámetro: el siguiente método:
- ```
public static int numPares (ArbolGeneral<Integer> arbol)
```
11. Escribe un método que dado un árbol general y una lista que se pasa como parámetro, guarde en la lista las **hojas** del árbol general.
- ```
public static <E> void hojas(ArbolGeneral<E> a, List<E> l)
```

**Anexo:**

- **TAD Árbol General:**

```
public interface ArbolGeneral<E>{
 public boolean esVacio();
 public E raiz() throws ArbolVacioExcepcion;
 public ArbolGeneral<E> hijoMasIzq() throws ArbolVacioExcepcion ;
 public ArbolGeneral<E> hermanoDer() throws ArbolVacioExcepcion;
 public boolean esta(E elemento);
 public void setRaiz(E elemRaiz) throws ArbolVacioExcepcion;
 public void setHijo(ArbolGeneral<E> hijo) throws ArbolVacioExcepcion, NullPointerException;
 public void suprimir();
}
public class EnlazadoArbolGeneral<E> implements ArbolGeneral<E>{
 public EnlazadoArbolGeneral(){...}
 public EnlazadoArbolGeneral(E elemento, ArbolGeneral<E> ... hijos) {...}
 ...
}
```

**Ejemplo de uso:**

```
import arbolGeneral.*;

public class Prueba{
 public static void main (String[] args) {
 ArbolGeneral <Integer> a = new EnlazadoArbolGeneral <Integer>();
 }
}
```