

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 ejercicios propuestos para clase, como 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.
3. Para probar el correcto funcionamiento de los métodos se puede hacer uso del fichero de 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 29 de octubre de 2024.

Tiempo estimado

3 horas

Ejercicios para resolver en clase

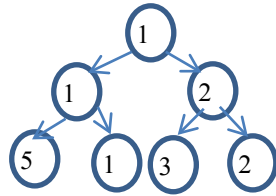
1. Escribe un método que dado un árbol binario devuelva verdadero si el árbol es **completo** y falso en otro caso. Un árbol binario es completo si todos sus nodos tienen dos descendientes, excepto las hojas.
2. Escribe un método que dados dos árboles binarios A y B, determine si son **idénticos** o no.
3. Escribe un método que **cuenta** 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.

Otros ejercicios

6. 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 esArbolSeleccion (ArbolBinario<E> arbol)
```

Ejemplo:



7. Dado un árbol binario de enteros escribe el método **existeSuma**, que devuelve cierto si existe al menos un nodo del árbol tal que la suma de todos sus nodos descendientes es igual al parámetro suma. Haciendo uso del TAD Árbol Binario disponible en el anexo, escribe dicho método:

```
public static <E> boolean existeSuma (ArbolBinario<Integer> arbol, Integer suma)
```

8. 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)
```

9. 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 esDistanciaK (ArbolBinario<E> a, E elem, int k)
```

ANEXO con información para usar los TADs de AEDII:

La librería “tads-aed-2” contiene los TAD vistos en clase de teoría. Para el TAD Árbol Binario se muestra a continuación la interface (con sus métodos), así como la clase que implementa la interface (interesa conocer los constructores):

■ 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, NullPointerException;
    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, ArbolBinario<E> hi, ArbolBinario<E> hd) {...}
    ...
}
```

Cómo usar la librería con los TADs de AEDII:

Para usar la librería en los proyectos Maven es necesario editar el fichero **pom.xml**. Para editarlo puedes:

- i) En la pestaña Projects botón derecho en el proyecto y "Open POM",
- ii) En la pestaña Projects, ir a la carpeta Project Files y ya se ve el pom.xml
- iii) En la pestaña Files, ya se ve el pom.xml

A continuación, incluir dentro de <project> el siguiente código:

```
<repositories>
  <repository>
    <id>sing-maven-releases</id>
    <name>SING Maven Releases</name>
    <url>https://maven.sing-group.org/repository/maven-releases/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
  </repository>
</repositories>
<dependencies>
  <dependency>
    <groupId>esai.aed2</groupId>
    <artifactId>tads-aed-2</artifactId>
    <version>1.0</version>
  </dependency>
</dependencies>
```

Y acordarse de importar las clases necesarias en el fichero fuente que usa los árboles:

```
import arbolBinario.ArbolBinario;
import arbolBinario.EnlazadoArbolBinario;
```