Asignatura: Programación y Estructuras de Datos
Grado en Ingeniería Informática de Servicios y Aplicaciones / Doble Grado INFOMAT
E.I. Informática (Campus de Segovia) – Universidad de Valladolid
Profesores: * Pilar Grande González (teoría y responsable de la asignatura)
* Francisco Hernando Gallego (Laboratorio)

Práctica 3: TAD ABB (Árbol Binario de Búsqueda)

Objetivos:

- Diseñar, implementar y usar un TAD.

ENUNCIADO

Se desea diseñar el TAD ABB siguiendo las distintas etapas de la "técnica de abstracción de datos" que hemos estudiado. Se pide:

1.- Realizar la **especificación lógica del TAD ABB** teniendo en cuenta las operaciones básicas que hemos considerado.

2.- **Implementar el TAD ABB en Java**. Trabajaremos con el TAD implementado en memoria dinámica de la forma estudiada en clase *(clase NodoABB y clase ABB).*

a) Diseñar la **Interfaz del TAD ABB** *(Interfaz Modelo_ABB<E>, paquete "modelos")*

b) Implementar en Java el TAD ABB en memoria dinámica *(Clase NodoABB y clase ABB, paquete "jerarquicos").* La clase ABB debe implementar la Interfaz anterior.

3.- Repita el proceso anterior para diseñar el TAD ABBEnteros que modele un ABB en el que se almacenan claves enteras.

a) Diseñar la **Interfaz del TAD ABBEnteros** *(Interfaz Modelo_ABBEnteros<Integer>, paquete "modelos").* Puede incluir como funciones básicas los métodos que ejecutarán las opciones 5, 6 y 7 del menú principal, además de todas las funciones básicas que todo ABB debe proporcionar.

b) Implementar en Java el TAD ABBEnteros en memoria dinámica *(Clase NodoABB y clase ABBEnteros, paquete "jerarquicos").* La clase ABBEnteros debe implementar la Interfaz anterior.

Nota: Considere la posibilidad de aplicar herencia entre interfaces y herencia entre clases y justifique en la memoria la decisión que ha tomado al respecto al realizar el diseño de la interfaz Modelo_ABBEnteros y la clase ABBEnteros.

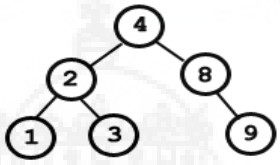
4.- **Uso de los TADs diseñados**: Realizar un programa que presente, de forma repetitiva, el siguiente menú de opciones:

MENÚ PRINCIPAL

1.- Crear ABB de enteros positivos (Equilibrado)
2.- Listado de claves en orden ascendente
3.- Listado de claves en orden descendente
4.- Listado de claves por niveles (converso)
5.- Transformar árbol ABB de enteros
6.- Sumar claves menores que un número dado
7.- Calcular el antecesor de un nodo
8.- Mostrar hojas (posiciones impares)
0.- Salir

A continuación se detalla la tarea que se debe llevar a cabo al ejecutar cada una de las opciones del menú principal de la aplicación:

Opción 1.- ***"Crear ABB de enteros positivos (Equilibrado)"***: Al ejecutar esta opción se crea un ABB de claves enteras positivas introducidas por el usuario por teclado. Se supone un máximo de 50 claves. El ABB debe quedar equilibrado *(Consultad el ejercicio 9 estudiado en clase).*



Opción 2.- ***"Listado de claves en orden ascendente"***: Realiza un recorrido del ABB anterior mostrando en pantalla las claves en orden ascendente.

Salida en pantalla: 1 2 3 4 8 9

Opción 3.- ***"Listado de claves en orden descendente"***: Realiza un recorrido del ABB anterior mostrando en pantalla las claves en orden descendente.

Salida en pantalla: 9 8 4 3 2 1

Opción 4.- ***"Listado de claves por niveles (converso)"***: Realiza un recorrido del ABB anterior mostrando en pantalla las claves por niveles, y dentro de cada nivel, de derecha a izquierda. Puede utilizar la estructura de datos auxiliar que considere necesaria.

Salida en pantalla: 4 8 2 9 3 1

Opción 5.- ***"Transformar árbol ABB de enteros"***: Transforma el ABB anterior considerando todas sus claves negativas. El árbol resultante de este proceso deberá seguir siendo ABB. Tenga especial cuidado en mantener la condición de orden que todo ABB debe cumplir. Puede comprobar que el proceso de transformación ha sido correcto ejecutando los listados de claves de las opciones 2, 3 y 4 del menú principal. **==>** ***Método de la clase ABBEnteros***

Listado de claves ascendente => Salida en pantalla: -9 -8 -4 -3 -2 -1
Listado de claves descendente => Salida en pantalla: -1 -2 -3 -4 -8 -9
Listado de claves por niveles (converso) => Salida en pantalla: -4 -2 -8 -1 -3 -9

Opción 6.- ***"Sumar claves menores que un número dado"***: Al ejecutar esta opción se solicita al usuario un número entero y se recorre el ABB sumando todas las claves que sean menores que dicho número. **==>** ***Método de la clase ABBEnteros***

Ej: Considerando el ABB de la imagen *(opción 1)*, la salida en pantalla será la siguiente:

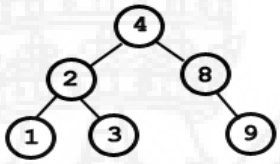
Introduzca una clave: 7
La suma de elementos menores que 7 es 10

Opción 7.- ***"Calcular el antecesor de un nodo":*** Al ejecutar esta opción, se solicita al usuario una clave entera y se busca dicha clave en el árbol. Si existe, se muestra en pantalla la clave almacenada en su nodo antecesor. Si no existe, se muestra en pantalla el mensaje "Esta clave no existe en el árbol". ==> ***Método de la clase ABB***

Ej: Considerando el ABB de la imagen *(opción 1)*, la salida en pantalla será la siguiente:

Introduzca una clave: 8
El antecesor del nodo 8 es el 4

Opción 8.- ***"Mostrar hojas (posiciones impares)":*** Al ejecutar esta opción, se muestran en pantalla las claves que se encuentran en las hojas del árbol que se encuentran en posiciones impares (contadas de izquierda a derecha).



Hoja 1 (1) Hoja 2 (3) Hoja 3 (9)

La salida en pantalla será:

Hoja 1 – Clave 1
Hoja 3 – Clave 9

Opción 0.- "*Salir*": Finaliza la ejecución del programa mostrando un mensaje de despedida. *"Gracias por utilizar nuestros TAD ABB y ABBEnteros".*

Observaciones:

- Las opciones 2, 3 , 4 , 5 , 6, 7 y 8 del "Menú Principal" sólo podrán ser ejecutadas si con anterioridad se ha ejecutado la opción 1 *("Crear ABB de enteros (Equilibrado)").*

- Puede diseñar las excepciones que considere oportunas.

NORMAS DE REALIZACIÓN Y ENTREGA DE LA PRÁCTICA:

La memoria y el código fuente implementado deberán ser entregados a través del campus virtual de acuerdo a las instrucciones que allí se detallan.

DOCUMENTACIÓN A ENTREGAR

Será preciso realizar una **MEMORIA** que contenga la siguiente información:

*** Portada que incluya:** título de la práctica, nombre de los alumnos, nº de grupo de prácticas, turno de laboratorio y fecha de entrega

*** Especificación lógica** de los TADs ABB y ABBEnteros diseñados (similar a la que realizamos en el tema 2).

*** Diseño de los métodos recursivos** *(Análisis del problema /Identificación de caso/s base / Identificación de caso/s recursivo/s).*
* Descripción detallada del funcionamiento de cada método **recursivo** (Breve comentario)**.**
*** Listado del código fuente** de la aplicación desarrollada. El código fuente deberá estar correctamente comentado *(Cabeceras descriptivas en cada método, etc)*
*** Pruebas de ejecución** de la aplicación (capturas de pantallas de ejecución de cada una de las opciones del menú principal).

FECHAS A CONSIDERAR:

- Publicación de la práctica:Martes, 2 de Mayo de 2023
- Entrega de la práctica: Lunes, 29 de Mayo de 2023, 22 h. (en el campus virtual)

Importante:

- La memoria de la práctica y el código fuente de la misma deberán ser entregados a través del campus virtual de acuerdo a las instrucciones que allí se detallan.
- No se admitirán entregas por otra vía distinta *(correo electrónico, drive,.. etc.).*
- No se admitirá entregas fuera del plazo establecido.

AVISO
*** Cualquier intento de plagio de la práctica supondrá la obtención de una nota numérica igual a 0.0 en la misma**
*** Además, TODOS los alumnos implicados deberán realizar de forma INDIVIDUAL una nueva práctica de complejidad muy superior a la actual para poder optar a superar la asignatura en la convocatoria extraordinaria.**
*** La nota máxima que podrá obtener en esta nueva práctica será 6, siendo necesario obtener al menos un 5 para aprobar esta práctica.**
*** Se recuerda que para superar la asignatura es preciso aprobar ambas partes (Teoría y Práctica) por separado.**