Universidad Rey Juan Carlos

Estructuras de Datos Avanzadas
Examen extraordinario
19 Junio 2013

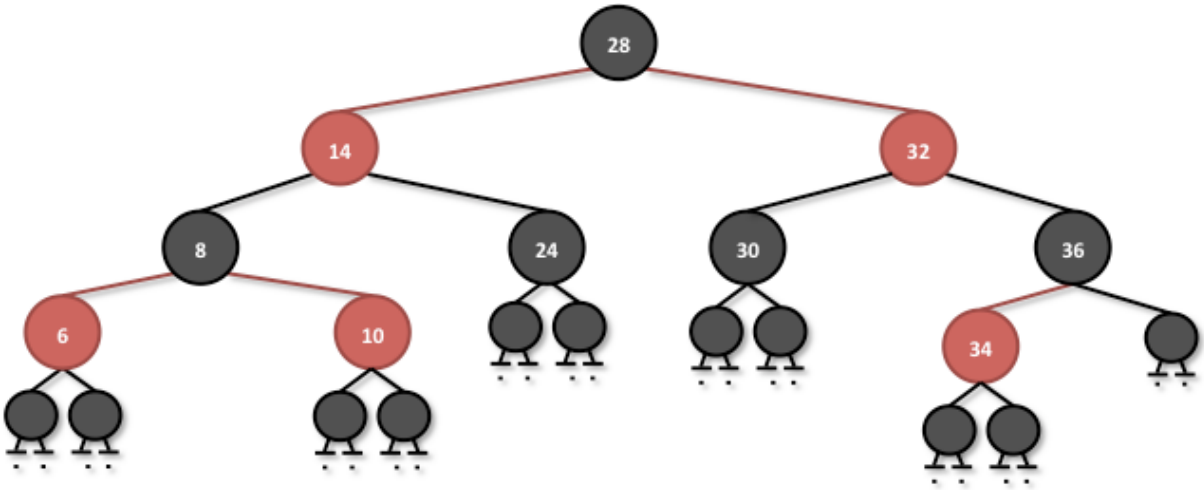
Apellidos:	***Nombre:***
Grupo:	***DNI:***
Duración: 2:30 h.	

Ejercicio 1	Ejercicio 2	Ejercicio 3	Ejercicio 4	TOTAL

Ejercicio 1 [2 puntos].

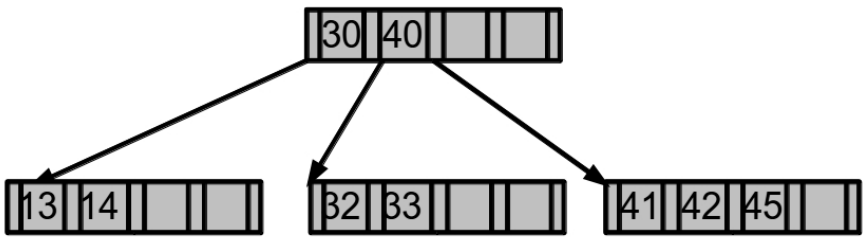
a) Dado el siguinete árbol Rojo-Negro, realizar la siguiente secuencia de borrados

6, 24, 34, 36, 30, 32



b) Dado el siguiente árbol B de orden 5, representar gráficamente dicho árbol después de las siguientes operaciones

i) Borrado de la clave 40
ii) Sobre el árbol B resultante, borrado de la clave 30
iii) Sobre el árbol B resultante, borrado de la clave 41



Ejercicio 2: [3 puntos]

Se quiere crear un diccionario electrónico que permita el acceso directo a cada letra, de tal forma que el coste de dicha operación de acceso sea O(1). Las palabras que empiecen por una determinada letra se almacenarán en una estructura de datos dinámica, ya que el número de palabras por cada letra se desconoce inicialmente. Además, se requiere que la búsqueda de una palabra que empiece por una letra determinada tenga una complejidad de O(log *n*) aunque. Se pide:

a) **[0.5 puntos]** Definir el tipo (clase) `Diccionario`, justificando la elección. También se deben definir todos los tipos de datos auxiliares que se consideren necesarios.

b) **[1 punto]** Implementar las operaciones `CrearDiccionarioVacio` e `InsertarPalabra`, indicando la complejidad de cada una de ellas. Se deberán implementar también todas las operaciones auxiliares necesarias.

c) **[1.5 puntos]** Implementar la operación Autocompletar (así como todas las operaciones auxiliares necesarias) que reciba como argumentos un diccionario y una palabra, que no tiene por qué ser exactamente la palabra almacenada, y devuelva en una estructura de datos `TResultado` (justificando su elección) todas las palabras que tengan ese mismo inicio. Por ejemplo, supóngase que se le pasa el inicio de palabra com; entonces, podría devolver como resultado:

> **com**er, **com**erciable, **com**ercial, **com**ercializable, **com**ercialización, **com**ercializar, **com**ercialmente, **com**erciante, **com**erciar, **com**ercio

Ejercicio 3 [2 puntos]

Desarrollar un programa Java que permita simular un sistema de autenticación de un usuario delante de un sistema informático para poder tener acceso a los servicios proporcionados por el mismo (por ejemplo, una cuenta bancaria por Internet). Dicha información de usuario consta de dos datos: un nombre de usuario (clave) y su contraseña (valor). Todo usuario registrado tiene una clave de acceso única. Además, la información de todos los usuarios registrados en el sistema se almacenará en una estructura de datos adecuada para este problema. El programa desarrollado deberá realizar las siguientes dos operaciones:

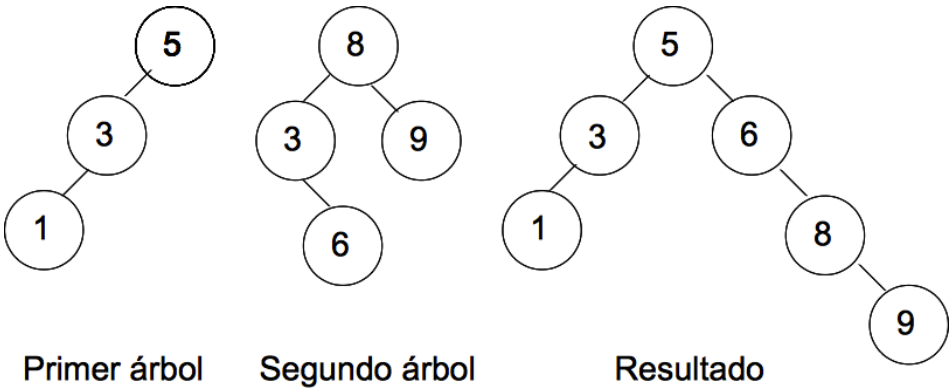
a) **[1 punto]** Impresión eficiente de toda la información contenida en la estructura de datos (suponiendo dicha información está ya previamente almacenada en ella).

b) **[1 punto]** Simulación del proceso de *login* de un usuario, quien para validarse ha de introducir correctamente su nombre de usuario y su contraseña. Si el usuario no consigue autenticarse después de tres intentos, el programa finaliza y el usuario recibe un mensaje.

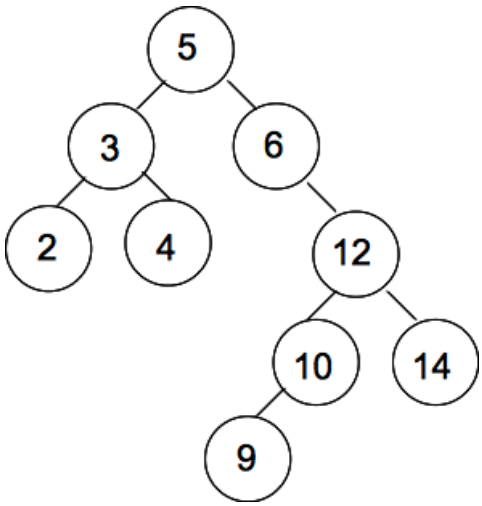
Ejercicio 4 [3 puntos]

Incluir las siguientes operaciones en el TAD Árbol Binario de Búsqueda (ABB):

a) **[1.5 puntos]** Fusionar dos ABB en uno solo tal que dicho nuevo árbol resultado contenga los elementos de ambos. Para realizar la fusión se creará un nuevo árbol que contendrá inicialmente los elementos del primer árbol, y sobre éste se insertarán los elementos nuevos del segundo árbol, recorriendo éste en inorden. La figura adjunta muestra un ejemplo de fusión de dos árboles y su resultado.



b) **[1.5 puntos]** Encontrar el elemento mediana de un ABB que contiene un número impar *n* de nodos. La mediana de un ABB es aquel nodo que ocupa la posición $\left\lceil \frac{n}{2} \right\rceil$ del árbol, si se visitan sus nodos según un recorrido en inorden. Por ejemplo, para el ABB de la figura adjunta, el nodo mediana es el que tiene valor '6'.



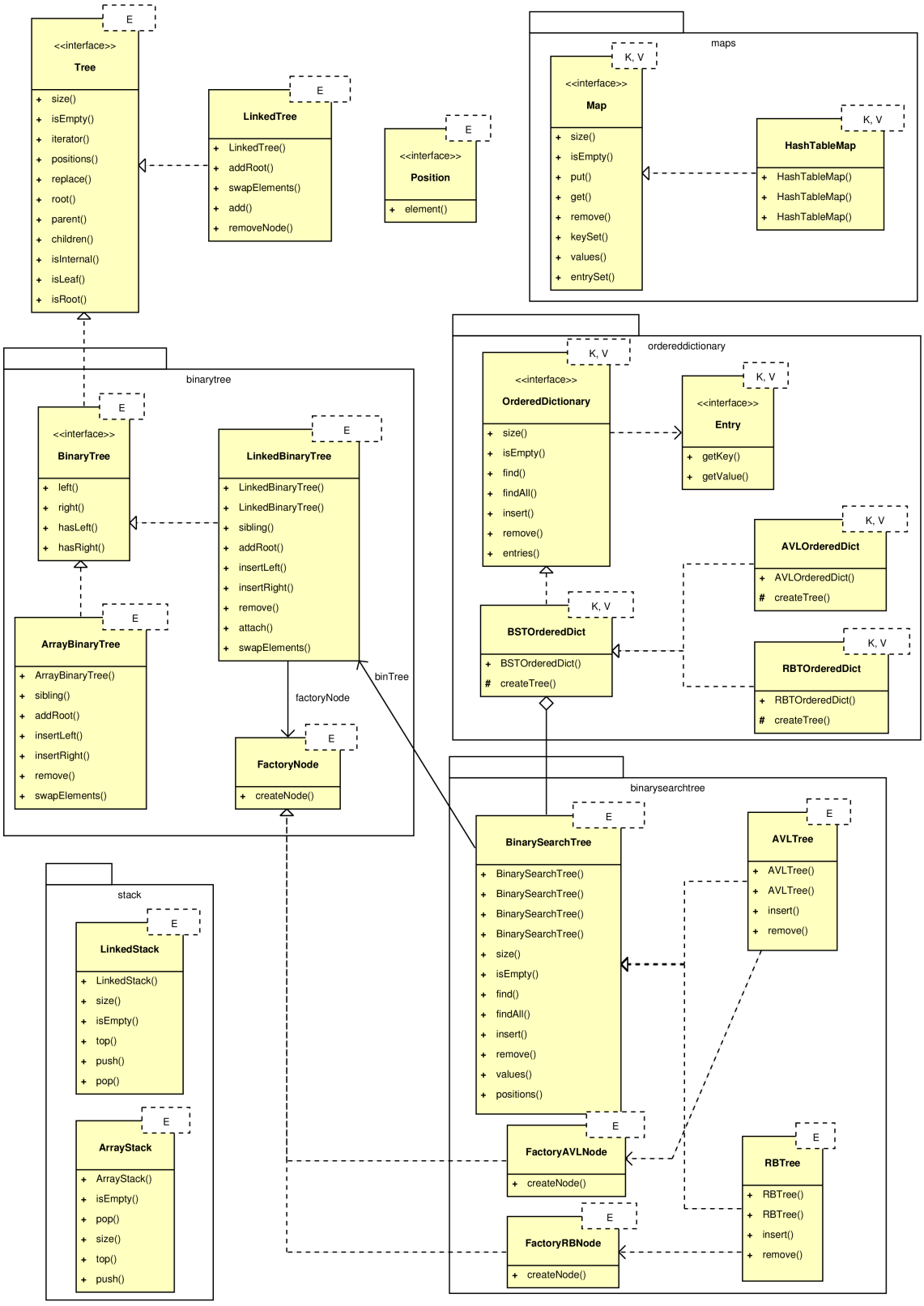


Diagrama de Clases correspondiente a las estructuras presentadas en clase

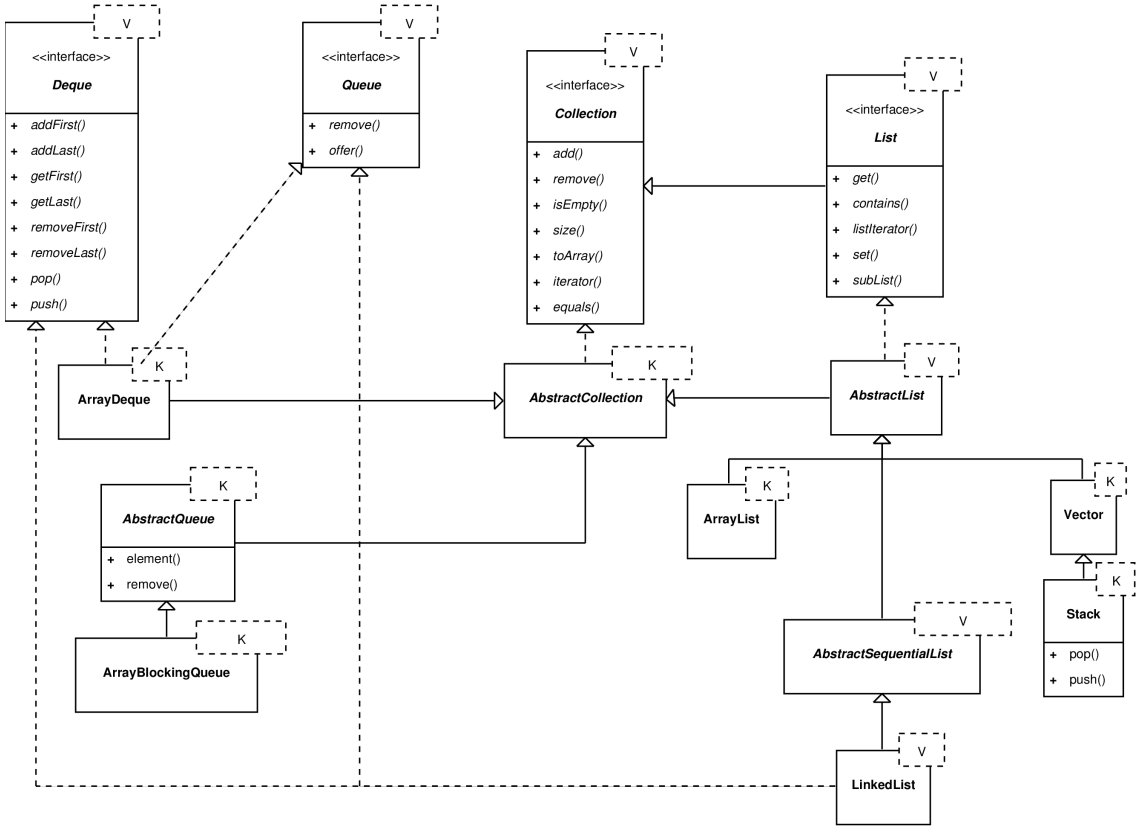


Diagrama de clases correspondiente a las estructuras de datos lineales de Java