**PRIMER PARCIAL**

**INF310 SX**‒ **Estructuras de Datos II. Gestión 1-2019.**

**Subgrupo: G-R**

Árbol Binario

1. En un árbol binario B (desordenado pero sin duplicados), llamamos “cuasi-hoja” a un Nodo del árbol que tiene un solo hijo y ése único hijo es una hoja. En la class Árbol, escriba el procedimiento

**public void delCuasiHoja(int x)**

el cual elimine del árbol el cuasi-hoja x (si x no existe o x no es un cuasi-hoja, este procedimiento no hace nada).

Tome en cuenta que cada vez que se elimina un cuasi-hoja, su único hijo toma su lugar.

**Por ejemplo**: (En el gráfico, se encuentran sombreados los cuasi-hojas y no se grafican los punteros null)

**Árbol A**

|  |  |
| --- | --- |
| **A.delCuasiHoja(100);** | El 100 no existe: “No pasa nada” (el árbol queda igual) |
| **A.delCuasiHoja(90);**  El 90 es un cuasi hoja, por tanto él será eliminado, pero el 20 ocupará su lugar. |  |
| **A.delCuasiHoja(50);** | El 50 existe, pero **no** es un cuasi-hoja: “No pasa nada” (el árbol queda igual) |

**Listas**

2. **El Juego de los Tarros**.

Se tiene un conjunto de Tarros, numerados de 0 a N-1 (N es una variable static), los cuales inicialmente están vacíos.

El juego consiste en lanzar fichas, numeradas del 1 al 9, las cuales caerán en algunos de los Tarros (esto se simula con el método **lanzar**).

* No se permite que dos monedas de la misma numeración estén en el mismo Tarro (i.e. Un tarro no admite “repetidos”).
* Aquel Tarro cuya suma de sus monedas sea 45 será el ganador.

Usando Singles-Lists, represente e implemente este Juego, en la siguiente class:

public class Juego {

private static final int N= ;

public Juego(){ //Constructor. Todos los tarros están vacíos.

}

public void lanzar(int i, int moneda){

//Al Tarro i, se le inserta la moneda. Si la moneda ya existe en el Tarro i, “no pasa nada”.

//(Por comodidad, asuma que moneda es un número entre 1 y 9.

}

public boolean gano(int i){

//Devuelve true si y solo si, la suma de las monedas del Tarro i es 45.

}

}

**Por ejemplo**: Asumiendo que N=3 (Entonces los tarros tienen los índices de 0 a 2)

**Juego J = new Juego();** //El Juego **J** tiene sus tres tarros vacíos

**J.lanzar(0, 9);** //La moneda 9 se inserta en el tarro **0**. Tarro0 = {**9**}, Tarro1=(vacío), Tarro2=(vacío)

**J.lanzar(1, 9);** //La moneda 9 se inserta en el tarro **1**. Tarro0 = {9}, Tarro1={**9**}, Tarro2=(vacío)

**J.lanzar(0, 6);** //La moneda 6 se inserta en el tarro **0**. Tarro0 = {9, **6**}, Tarro1={9}, Tarro2=(vacío)

**J.lanzar(2, 6);**

**J.lanzar(2, 3);**

**J.lanzar(2, 9);** //Las monedas 6, 3, 9 se insertan al tarro **2**. Tarro0 = {9, 6}, Tarro1={9}, Tarro2={**6**, **3**, **9**}

**J.lanzar(2, 3);** //La moneda **3**, ya **existe** en el tarro **2**: “No pasa nada”.

**J.gano(2)=false;** //Es false, porque la suma de las monedas del tarro **2** = 6 + 3 + 9 = 18 (debe ser 45).