**PRIMER PARCIAL**

**INF310 SX**‒ **Estructuras de Datos II. Gestión 1-2021.**

**Subgrupo: A-L**

Árbol Binario

1. En la class Arbol (binario, desordenado pero sin duplicados), escriba el procedimiento

public void changeLeaf (int h1, int h2)

el cual intercambia las hojas, cuyos Datas son h1 y h2. Si h1 o h2 no son hojas o alguno de ellos no existe, este procedimiento no hace nada.

**Por ejemplo**: (En el gráfico, no se dibujan los punteros null)

**Árbol A**

|  |  |
| --- | --- |
| **A.**changeLeaf(**150, 35);** | El 150 no existe: El árbol queda igual. |
| **A.**changeLeaf(**80, 60);** | El 80 y el 60 existen, pero el 60 no es una hoja: El árbol queda igual. |
| **A.**changeLeaf(**80, 35);**  El 80 y el 35 son hojas, por tanto, se intercambian. | **Árbol A** |
| **A.**changeLeaf(2**0, 5);**  Las hojas 20 y 5 se intercambian. | **Árbol A** |

**Listas**

2. **EL JUEGO DE LA ESCALERA SOLITARIA.** *Por simplicidad, asumimos que los naipes se enumeran de 1 al 9 y no tienen palos.*

Es un juego de naipes para UN solo jugador, el cual todo el tiempo tiene 3 naipes. Inicialmente, el Jugador tiene 3 naipes con números diferentes, luego saca del mazo una carta y bota otra (método add). Luego de hacer esto, verifica si tiene una escalera (3 números consecutivos): si es así el juego finaliza, caso contrario el jugador sigue sacando naipes del mazo.

**public Juego(){**

//**Constructor**. El Jugador tiene 3 naipes diferentes (cargar al azar tres números entre 1 y 9)

}

**public boolean add(int nuevoNaipe, int naipeBotar){**

//Asuma que nuevoNaipe y naipeBotar son números entre 1 y 9 y que nuevoNaipe≠naipeBotar.

//Si naipeBotar no existe, esta función no hace nada y return false.

//Si naipeBotar existe, **se reemplaza** el naipeBotar por el nuevoNaipe. Luego de hacer esto, si existe una escalera,

//return true; caso contrario, return false;

}

Represente e implemente este Juego, usando **Single’s – List’s**.

**Por ejemplo (en el main):**

P = new Juego(); //Se crea el Juego con 3 naipes, supongamos P = [5, 1, 2]

boolean b = P.add(9, **3**); //El naipe a botar **3**, NO está en P; por tanto, P queda igual y b=false (porque add, return false)

b = P.add(9, **1**); //Se sustituye el 9 por el 1. P = [5, **9**, 2]. La función devuelve false, porque no hay una escalera (b=false).

b = P.add(5, **2**); //Se sustituye el 5 por el 2. P = [5, 9, **5**]. La función devuelve false, porque no hay una escalera (b=false).

b = P.add(7, **5**); //Se sustituye el 7 por el primer 5. P = [**7**, 9, 5]. La función devuelve false, porque no hay una escalera (b=false).

b = P.add(8, **5**); //Se sustituye el 8 por el 5. P = [7, 9, **8**]. La función devuelve TRUE, porque ya hay una escalera (7, 8, 9).