

Taller 02

Ejercicio 01

Completar en la clase `ArbolBinarioDeBusqueda` los siguientes métodos

Es Hoja :

- retorna verdadero si el nodo es un nodo hoja. El árbol binario vacío no es un nodo hoja. El `arbolbinarioDeBusqueda` es un nodo hoja, cuando el subárbol izquierdo y el subárbol derecho es un árbol vacío.

CantidadNodos

- retorna la cantidad de nodos que tiene un árbol, esto es, la raíz, mas la cantidad de nodos del subárbol izquierdo y la cantidad de nodos del subárbol derecho. El árbol vacío tiene 0 nodos.

Es Zurdo

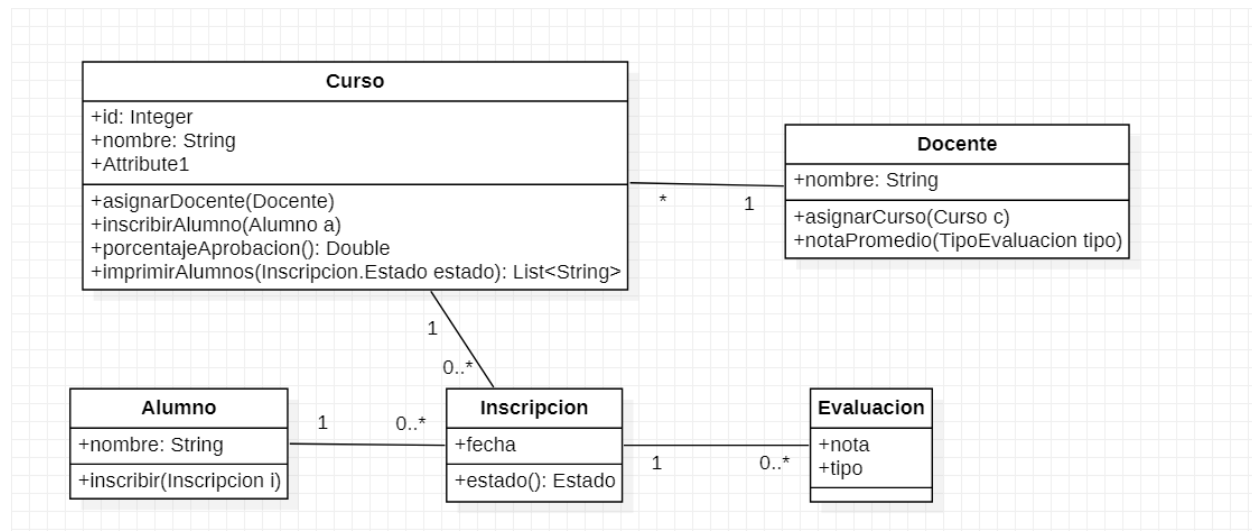
- Un árbol binario es zurdo si
 - a. Es el árbol vacío
 - b. Es una hoja
 - c. Es un árbol donde más de la mitad de sus descendientes están en el hijo izquierdo y además tanto sus hijos izquierdos y como derecho son zurdos (*es decir o son un árbol vacío, o son un nodo hoja, o cumplen el punto c).*

Desarrollar en Java métodos recursivos para los 3 métodos anteriores.

Crear en la clase `App01` las instancias necesarias para probar los 3 items anteriores lo suficiente para crear 3 árboles binarios, las 2 últimas vacías. Luego crear los árboles binarios de tal que el postorden sean [A5, A4, A3, A1, A2, A6, A7]

Ejercicio 02

El sistema de gestión de una institución educativa permite gestionar cursos donde los alumnos se inscriben.



Dentro de la clase inscripción tenemos una enum que indica el estado posible para una inscripción

```
public enum Estado { INSCRIPTO, REGULAR, PROMOCION, LIBRE }
```

Dentro de la enum TipoEvaluacion tenemos los posibles tipos de evaluaciones

```
public enum TipoEvaluacion {
    PARCIAL1, PARCIAL2, FINAL
}
```

Tareas a resolver **USANDO EL API DE STREAM**

1. Crear las excepciones `NoPuedeInscribirAlumnoException` y `NoPuedeAsignarDocenteException`
2. **Calcular el porcentaje de aprobación de un curso:** el porcentaje de aprobación de un curso es igual a la cantidad de notas de evaluaciones finales superiores al 60% sobre el total de inscripciones. Implemente este método en la **clase CURSO**
3. **Asignar un docente a un curso:** Un docente puede ser asignado a un curso, si y solo si, todos los cursos donde el docente este asignado tiene al menos un 25% del porcentaje de aprobación. En el caso de que un docente no se pueda inscribir en un curso, debe lanzar una excepción para gestionar esta situación. Estos métodos se deben implementar en las clases, **CURSO** y **DOCENTE**.
4. **Asignar un alumno a un curso:** Un alumno se puede inscribir a un curso registrando un objeto inscripción si y solo si no acumula mas de 3 inscripciones en las que el estado sea **LIBRE** o **REGULAR**. En el caso de que un alumno no se pueda inscribir en un curso, debe lanzar una excepción para gestionar esta situación. Estos métodos se deben implementar en las clases, **CURSO** y **ALUMNO**.

Crear en la clase App02 las instancias necesarias para probar los 3 items anteriores y el manejo de excepciones apropiado

Ejercicios opcionales

5. **Imprimir lista ordenada alfabéticamente de alumnos según situación:** en la clase Curso crear un método que reciba el estado de una inscripción y retorne una lista de alumnos de ese curso, que tienen ese estado ordenado alfabéticamente.
6. **Imprimir notas promedias para los cursos de un docente:** Retorna un map donde la clave es el nombre de cada curso y el valor es el promedio de todas las evaluaciones para ese curso