# Examen 2ª evaluación Programación - Dual

Vamos a implementar el **patrón Composite**, que:

*"Permite a un grupo de objetos ser tratado como un objeto simple".*

*"Básicamente, implica implementar tu propia clase colección, la cual posee los mismos métodos que sus miembros".*

* "El patrón Composite se ha diseñado para implementar el tratamiento uniforme de objetos en una jerarquía, la cual contiene otros objetos sin importar si a su vez contienen otro objetos (como parte de la jerarquía) o no.
* Estos objetos reciben el nombre de **composite**.
* Los objetos composite tienen la misma clase base tanto para los objetos individuales (**no-composite**) como para aquellos que son secciones de objetos (composite).
* Ambos objetos, composite y no-composite normalmente **tienen el mismo conjunto de métodos**. Los objetos composite además tienen métodos adicionales para soportar la **adición** y **sustracción** de elementos hijo, así como la **iteración** sobre ellos."

Los elementos con los que trabajaremos serán la liga, conferencias, divisiones y equipos de la NBA:<https://es.wikipedia.org/wiki/National_Basketball_Association>

* Una liga NBA tiene dos conferencias: Este y Oeste (objetos compuestos).
* Cada conferencia se divide en divisiones (objetos compuestos). Hay 3 divisiones en la conferencia Este y otras 3 en la Oeste.
* En cada división hay 5 equipos (objetos simples).

Utiliza el capítulo 12 pag. 475 del libro "Beginning Java Programming" si quieres leer algo sobre el patrón Composite.

## **Proyecto**

1. Crea un proyecto con tu nombre y apellidos.
2. Descarga la captura de imagen de la salida de la aplicación. **Utilízala como guía en el proceso TDD**.
3. Completa las clases que necesites **implementando los casos test que se proponen en la función principal**. (captura de imagen).
   1. Si utilizas JUnit tu nota se verá recompensada.
   2. Si no funcionan las cosas de este modo, hazlo a la manera tradicional, situando los casos test en la función main().
4. Descarga la imagen que contiene el **diagrama de clases** que representa el diseño del programa. A partir de este diagrama debes generar el código.

### 

### **Cómo entregar el código**

1. Accede al workspace de Eclipse y busca la carpeta que tiene igual nombre que tu proyecto.
2. Comprime esa carpeta.
3. Envíame el archivo por correo electrónico.

## **Clase ITEM**

### **Atributos**

* nombre: nombre del elemento.
* compuesto: indica si el elemento es compuesto o no.
* hijos: contiene la colección de objetos hijo (compuestos y simples).

### **Métodos**

* gettersy setters.
* getHijos(): devuelve la colección de hijos
* anhadir(item) : añade elementos simples y compuestos a la colección de hijos de un elemento compuesto.
* quitar(nombre): elimina elementos simples y compuestos a la colección de hijos de un elemento compuesto, buscándolos por el **nombre** ("Boston" o "Atlantic", por ejemplo).
* quitar(item): elimina elementos simples y compuestos a la colección de hijos de un elemento compuesto, buscándolos por referencia al **objeto**.
* iterable(): recorre la colección de elementos hijo de un elemento compuesto, mostrando en consola el nombre del elemento y el de sus hijos.
* composite(nombres): construye la colección de elementos hijo de un elemento compuesto, a partir de un array de strings con los nombres de los elementos hijo.
* composite(items) : construye la colección de elementos hijo de un elemento compuesto, a partir de un array de objetos de elementos hijo.