

# Laboratorio A.E.D. Ejercicio Individual 6

**Guillermo Román**

guillermo.roman@upm.es

**Lars-Åke Fredlund**

lfredlund@fi.upm.es

**Manuel Carro**

mcarro@fi.upm.es

**Marina Álvarez**

marina.alvarez@upm.es

**Julio García**

juliomanuel.garcia@upm.es

**Tonghong Li**

tonghong@fi.upm.es

**Raúl Correal**

raul.correal@upm.es

# Normas.

- ▶ **¡Solo debe entregar una persona por grupo!.**
- ▶ Fechas de entrega y nota máxima alcanzable:

Hasta el Lunes 12 de diciembre, 23:59 horas	10
Hasta el Martes 13 de diciembre, 23:59 horas	8
Hasta el Miércoles 14 de diciembre, 23:59 horas	6

Después la puntuación máxima será 0
- ▶ Se comprobará plagio y se actuará sobre los detectados
- ▶ Usad las horas de tutoría para preguntar sobre programación – son oportunidades excelentes para aprender

# Entrega

- ▶ Todos los ejercicios de laboratorio se deben entregar a través de la web `http://lm1.ls.fi.upm.es/~entrega`.
- ▶ El fichero que hay que subir es `HeapOrderChecker.java`.

## Configuración previa

- ▶ Arrancad Eclipse. Debéis tener un acceso directo.
- ▶ Si trabajáis en portátil, podéis utilizar cualquier versión relativamente reciente de Eclipse. Debería valer cualquier versión a partir de la versión 3.7. Es suficiente con que instaléis la *Eclipse IDE for Java Developers*.
- ▶ Cambiad a “Java Perspective”.
- ▶ Cread un proyecto Java llamado `aed`:
  - ▶ Seleccionad separación de directorios de fuentes y binarios.
- ▶ Cread un *package* `aed.heaps` en el proyecto `aed`, dentro de `src`.
- ▶ Aula Virtual → AED → Sesiones de laboratorio → Individual 6 → Individual6.zip; descomprimido.
- ▶ Contenido de Individual6.zip:
  - ▶ `TesterInd6.java`, `HeapOrderChecker.java`

## Configuración previa al desarrollo del ejercicio.

- ▶ Importad al paquete `aed.heaps` los fuentes que habéis descargado ( `TesterInd6.java`, `HeapOrderChecker.java` )
- ▶ Añadid al proyecto `aed` la librería `net-datastructures-5.0.jar` que tenéis en Moodle
- ▶ Intentad ejecutar `TesterInd6`

# Cumple la *heap-order property*?

- ▶ Se pide implementar el método

```
static <E extends Comparable<E>> boolean satisfiesHeapOrderProperty  
                                (BinaryTree<E> tree)
```

- ▶ que recibe un árbol binario y devuelve si éste cumple la *heap-order property*
- ▶ Un árbol cumple la *heap-order property* cuando para todo nodo  $n$ , distinto de la raíz, se cumple que el valor almacenado en  $n$  es mayor o igual que el valor almacenado en el padre de  $n$
- ▶ `<E extends Comparable<E>>` indica que el tipo genérico `E` implementa el interfaz `Comparable`

# Notas

- ▶ No se debe modificar la estructura de datos recibida como parámetro
- ▶ El proyecto debe compilar sin errores y debe cumplirse la especificación de los métodos a completar
- ▶ Debe ejecutar `TesterInd6` correctamente sin mensajes de error
- ▶ Nota: una ejecución sin mensajes de error no significa que el método sea correcto (es decir, que funcione bien para cada posible entrada)
- ▶ Todos los ejercicios se comprueban manualmente antes de dar la nota final