

Guillermo Viguera

guillermo.viguera@imdea.org

Julio García

juliomanuel.garcia@upm.es

Lars-Åke Fredlund

lfredlund@fi.upm.es

Manuel Carro Liñares

mcarro@fi.upm.es

Marina Álvarez

marina.alvarez@upm.es

Tonghong Li

tonghong@fi.upm.es

Normas.

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

Hasta el lunes 30 de noviembre, 15:00 horas	10
Hasta el martes 1 de diciembre, 15:00 horas	8
Hasta el miércoles 2 de diciembre, 15:00 horas	6
Hasta el jueves 3 de diciembre, 15:00 horas	4

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.

Sistema de Entrega

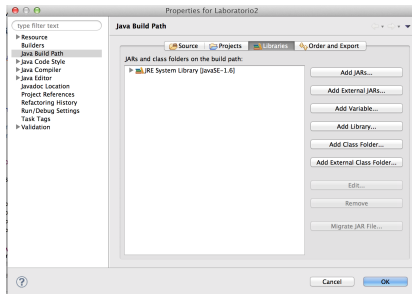
- ▶ Todos los ejercicios de laboratorio se deben entregar a través de la web <http://lm1.ls.fi.upm.es/~entrega>.
- ▶ Hoy, el fichero que hay que subir es `MergeLists.java`.

Configuración previa al desarrollo del ejercicio.

- ▶ Arrancad Eclipse. Debéis tener un acceso directo.
- ▶ Si trabajáis en portátil, podéis utilizar cualquier versión relativamente reciente de Eclipse. Debería valer cualquier versión entre la versión 3.7 (Indigo) o 4.3 (Kepler). 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* mergeLists en el proyecto aed, dentro de src.
- ▶ Aula Virtual → AED → Sesiones de laboratorio → Laboratorio 8 → Laboratorio8.zip; descomprimidlo.
- ▶ Contenido de Laboratorio7.zip:
 - ▶ Tester.java y MergeLists.java
 - ▶ positionList.jar

Configuración previa al desarrollo del ejercicio.

- ▶ Importad al paquete `mergeLists` las fuentes que habéis descargado (`Tester.java` y `MergeLists.java`).
- ▶ Añadid al proyecto `aed` la librería `positionList.jar` que habéis descargado. Para ello:
- ▶ Project → Properties. Se abrirá una ventana como esta:



- ▶ Java Build Path → Libraries → Add external JARs → Seleccionad el fichero `positionList.jar` que os habéis descargado.

Tarea para hoy

- ▶ Implementar el método

```
PositionList<Integer>  
    mergeLists(PositionList<PositionList<Integer>>  
               listOfLists)
```

dentro el fichero MergeLists.java.

- ▶ El método recibe una lista `listOfLists` cuyos elementos son listas de enteros **ordenados**, con elementos **duplicados**.
- ▶ El método devuelve una lista nueva, también **ordenada**, que contiene los mismos elementos como las sublistas de `listOfLists`, pero **sin elementos duplicados**.
- ▶ **Importante:** el método no debe cambiar el argumento `listOfLists`.
- ▶ Se puede asumir que `listOfLists` no es `null`, que sus sublistas no son `null`, y los elementos de las sublistas no son `null` tampoco.

Ejemplo

```
mergeLists([])          ---> []  
mergeLists([], [])     ---> []  
mergeLists([[1,2],[1]]) ---> [1,2]  
mergeLists([[3],[1,1,1],[],[2]]) ---> [1,2,3]
```

Implementación

- ▶ Es **obligatorio** usar recursión para implementar el método `mergeLists`.
- ▶ Está **prohibido** usar iteradores, bucles `while`, bucles `do...while`, bucles `for`, “for each”, etc.
- ▶ Probablemente será necesario implementar métodos auxiliares. Como siempre, está permitido.

Notas

- ▶ El proyecto debe compilar sin errores y debe cumplirse la especificación de los métodos a completar.
- ▶ Debe ejecutar `Tester` 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.