

Laboratorio A.E.D. Viernes 13:00 - 15:00 y 15:00 - 17:00

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

Pablo Nogueira

pnogueira@fi.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 19 de octubre 13:00 (15:00) horas	10
Hasta el martes 20 de octubre, 13:00 (15:00) horas	8
Hasta el miércoles 21 de octubre, 13:00 (15:00) horas	6
Hasta el jueves 22 de octubre, 13:00 (15:00) horas	4

Después la máxima puntuación será 0.
- ▶ Explicamos la solución tras jueves 22 de octubre.
- ▶ 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

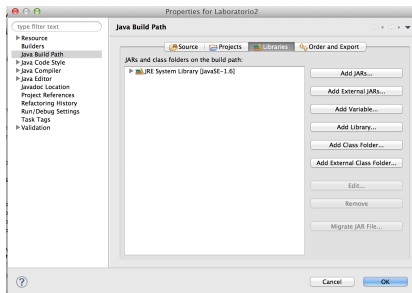
- ▶ Todos los ejercicios de laboratorio se deben entregar a través de la web <http://lml.ls.fi.upm.es/~entrega>.
- ▶ Hoy, el fichero que hay que subir es `RemoveLists.java`.

Configuración previa al desarrollo del ejercicio.

- ▶ Arrancad Eclipse. Debéis tener un acceso directo.
- ▶ Si trabajáis en portátil, podeis utilizar cualquier version relativamente reciente de Eclipse. Debería valer cualquier versión entre la versión 3.7 (Indigo) o 4.3 (Kepler). Es suficiente con que instaleis 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* RemoveLists en el proyecto aed, dentro de src.
- ▶ Aula Virtual → AED → Sesiones de laboratorio → Laboratorio 4 → Laboratorio4.zip; descomprimidlo.
- ▶ Contenido de Laboratorio4.zip
 - ▶ Tester.java y RemoveLists.java
 - ▶ positionList.jar

Configuración previa al desarrollo del ejercicio.

- ▶ Importad al paquete RemoveLists las fuentes que habéis descargado (Tester.java y RemoveLists.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
- ▶ Ejecutad Tester. Veréis que imprime un mensaje de error.

Tarea para hoy

- ▶ Realizar una implementación del método:

```
void removeAllSubList(PositionList<E> list,  
                      PositionList<E> subList)
```

dentro la clase `RemoveLists<E>`.

- ▶ Recibe una lista (`list`) y una sublista (`subList`) a buscar dentro de `list`.
- ▶ Debe eliminar de `list` todas las ocurrencias de `subList`, empezando en el principio de la lista y moviéndose hacia el final.
El método devuelve el resultado en el parámetro de entrada `list`. **No debe** cambiar el parámetro `subList`.
- ▶ Se asume que `list` y `subList` no contienen elementos `null`.

Tarea para hoy (2)

- ▶ Resultado del método para distintos valores de entrada:

```
removeAllSubList(null, [3,7]) ~> null
```

```
removeAllSubList([3,4,5,3], null) ~> [3,4,5,3]
```

```
removeAllSubList([3,2,3,4,5,3], [3]) ~> [2,4,5]
```

```
removeAllSubList([2,3,4,5,3,4], [3,4]) ~> [2,5]
```

```
removeAllSubList([3,2,3,4,7], [3,7]) ~> [3,2,3,4,7]
```

```
removeAllSubList([3,4,5,3], []) ~> [3,4,5,3]
```

```
removeAllSubList([], [3,7]) ~> []
```

```
removeAllSubList([1,0,1,0,1], [1,0,1]) ~> [0,1]
```

```
removeAllSubList([0,1,0,1,2,2], [0,1,2]) ~> [0,1,2]
```

Tareas para hoy (3)

- ▶ 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.