

Laboratorio A.E.D. Ejercicio Individual 2

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

Sergio Paraiso

sergio.paraiso@upm.es

Normas.

- ▶ Fechas de entrega y nota máxima alcanzable:

Hasta el Lunes 16 de septiembre, 23:59 horas	10
Hasta el Martes 17 de septiembre, 23:59 horas	8
Hasta el Miércoles 18 de septiembre, 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

`http://costa.ls.fi.upm.es/entrega`

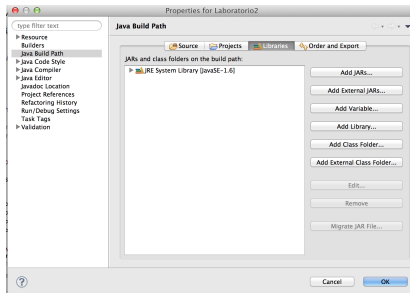
- ▶ El fichero que hay que subir es `Utils.java`.

Configuración previa

- ▶ Arrancad Eclipse
- ▶ 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.indexedlist` en el proyecto aed, dentro de `src`
- ▶ Aula Virtual → AED → Laboratorios y Ejercicios Individuales → Individual 2 → Individual2.zip; descomprimidlo
- ▶ Contenido de Individual2.zip:
 - ▶ `Utils.java`, `TesterInd2.java`

Configuración previa al desarrollo del ejercicio.

- ▶ Importad al paquete `aed.indexedlist` los fuentes que habéis descargado (`Utils.java`, `TesterInd2.java`)
- ▶ Añadid al proyecto `aed` la librería `aedlib.jar` que tenéis en Moodle (en Laboratorios y Entregas Individuales). Para ello:
- ▶ Project → Properties → Java Build Path. Se abrirá una ventana como esta:



- ▶ Usad la opción “Add External JARs...”.
- ▶ Intentad ejecutar `TesterInd2`

Documentación de la librería aedlib.jar

- ▶ La documentación de la API de la librería aedlib.jar esta disponible en
`http://costa.ls.fi.upm.es/entrega/aed/docs/aedlib/es/upm/aedlib/indexedlist/IndexedList.html`.
- ▶ Tambien se puede añadir la documentación de la librería a Eclipse (*no es obligatorio*): en el “Package Explorer”:
“Referenced Libraries” → aedlib.jar y elige la opción
“Properties”. Se abre una ventana donde se puede elegir
“Javadoc Location” y ahí se pone como “javadoc location
path:”
`http://costa.ls.fi.upm.es/entrega/aed/docs/aedlib/`
y presionar el boton “Apply and Close”

Tarea: Borrar elementos repetidos en una lista indexada

- ▶ Se pide implementar el método

```
static <E> IndexedList<E> deleteRepeated(IndexedList<E> l)
```

dentro la clase Utils.

- ▶ Recibe una lista indexada, y devuelve una **nueva** lista donde los elementos repetidos dentro de l han sido borrados.
- ▶ El orden de elementos en l debe preservarse en el resultado.
- ▶ No se debe modificar la lista de entrada l.
- ▶ Las listas indexadas son las de la asignatura Programación II. Están disponibles en la librería aedlib.jar.
- ▶ La documentación de su API está en <http://costa.ls.fi.upm.es/entrega/aed/docs/aedlib/es/upm/aedlib/indexedlist/IndexedList.html>.
- ▶ Para crear una lista indexada se puede usar

```
IndexedList<E> list = new ArrayIndexedList<E>();
```

Ejemplos

Ejemplos:

<code>deleteRepeated([])</code>	-->	<code>[]</code>
<code>deleteRepeated([1])</code>	-->	<code>[1]</code>
<code>deleteRepeated([1,4,3])</code>	-->	<code>[1,4,3]</code>
<code>deleteRepeated([1,4,3,4,4,3])</code>	-->	<code>[1,4,3]</code>
<code>deleteRepeated([1,4,4,3,4,2,3])</code>	-->	<code>[1,4,3,2]</code>

Notas importantes

- ▶ El valor de `l` no será `null` y no contendrá elementos `null`
- ▶ 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 `TesterInd2` 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 cualquier entrada posible).
- ▶ Todos los ejercicios se comprueban manualmente antes de dar la nota final.