

Laboratorio A.E.D. Ejercicio Individual 1

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

Juan José Moreno

juanjose.moreno@upm.es

Normas

- Fechas de entrega y nota máxima alcanzable:

Hasta el Martes 21 de septiembre, 12:00 horas	10
Hasta el Miércoles 22 de septiembre, 12:00 horas	8
Hasta el Jueves 23 de septiembre, 12:00 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

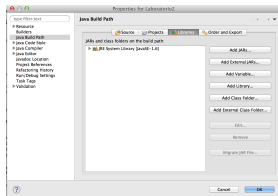
- Todos los ejercicios de laboratorio se deben entregar a través de
`http://vps142.cesvima.upm.es`
- El fichero que hay que subir es `Utils.java`.

Configuración previa

- Arrancad Eclipse
- Si trabajáis en portátil, podéis utilizar cualquier versión reciente de Eclipse. Es suficiente con que instaléis la *Eclipse IDE for Java Developers*.
- Cambiad a “Java Perspective”.
- Debéis tener instalado al menos Java JDK 8.
- Cread un proyecto Java llamado aed:
 - ▶ Seleccionad separación de directorios de fuentes y binarios.
 - ▶ **No debéis elegir la opción de crear el fichero** module-info.java
- Cread un *package* aed.loops en el proyecto aed, dentro de src
- Aula Virtual → AED → Laboratorios y Entregas Individuales → Individual 1 → Individual1.zip; descomprimidlo
- Contenido de Individual1.zip:
 - ▶ Utils.java, TesterInd1.java

Configuración previa

- Importad al paquete `aed.loops` los fuentes que habéis descargado (`Utils.java`, `TesterInd1.java`)
- Añadid al proyecto `aed` la librería `aedlib.jar` que tenéis en Moodle (en Laboratorios y Entregas Individuales).

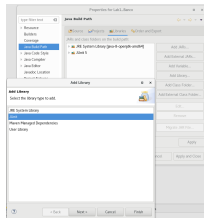


Para ello:

- Project → Properties → Java Build Path. Se abrirá una ventana como la de la izquierda
- Usad la opción “Add External JARs...”.
- Si vuestra instalacion distingue `ModulePath` y `ClassPath`, instalad en `ClassPath`

Configuración previa

- Añadid al proyecto aed la librería JUnit 5



- Project → Properties → Java Build Path. Se abrirá una ventana como la de la izquierda;
- Usad la opción “Add Library...” → Seleccionad “JUnit” → Seleccionad “JUnit 5”
- Si vuestra instalacion distingue ModulePath y ClassPath, instalad en ClassPath
- En la clase TesterInd1 tenéis las pruebas, para ejecutarlas, abrid el fichero TesterInd1, pulsando el botón derecho sobre el editor, seleccionar “Run as...” → “JUnit Test”
- NOTA: Si al ejecutar, no aparece la vista “JUnit”, podéis incluirla en “Window” → “Show View” → “Java” → “JUnit”

Documentación de la librería aedlib.jar

- La documentación de la API de aedlib.jar está disponible en <http://costa.ls.fi.upm.es/entrega/aed/docs/aedlib/>
- También 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: Calcular el numero máximo de ocurrencias **consecutivas** de un elemento en un array

- Se pide implementar el método

```
static int maxNumRepeated(Integer[] array, Integer elem)
```

dentro la clase Utils que recibe un *array* de enteros *array* y un *Integer elem*, y devuelve el numero máximo de ocurrencias **consecutivas** de *elem* en *array*.

- Ejemplos:

```
maxNumRepeated([],4)           --> 0    // array vacio
maxNumRepeated([1],4)          --> 0    // 4 no ocurre en [1]
maxNumRepeated([1,4,3],4)      --> 1
maxNumRepeated([1,4,3,4,4,3],4) --> 2    // debido a 4,4 en el array
maxNumRepeated([1,4,4,4,3],4)  --> 3    // debido a 4,4,4 en el array
maxNumRepeated([1,4,3,4,3,4],4) --> 1    // 4 solo aparece aislado
```


Notas importantes

- El valor de array 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 `TesterInd1` correctamente sin mensajes de error
- **Nota:** un test sin mensajes de error **no** significa que el método sea correcto (es decir, que funcione bien para cualquier posible entrada).
- Todos los ejercicios se comprueban manualmente antes de dar la nota final.