

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

Normas.

- Fechas de entrega y nota máxima alcanzable:

Hasta el Lunes 18 de septiembre, 23:59 horas	10
Hasta el Martes 19 de septiembre, 23:59 horas	8
Hasta el Miércoles 20 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

- Todas los ejercicios de laboratorio se deben entregar a través de la web <http://lml.ls.fi.upm.es/~entrega>.
- El fichero que hay que subir es `Utils.java`.

Configuración previa

- Arrancad Eclipse
- 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.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 al desarrollo del ejercicio.

- Importad al paquete `aed.loops` los fuentes que habéis descargado (`Utils.java`, `TesterInd1.java`)
- Intentad ejecutar `TesterInd1`

Tarea: Calcular el numero máximo de ocurrencias **consecutivas** de un elemento en un array

- Se pide implementar el método

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

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

- Ejemplos:

```
maxNumRepeated([],4)           --> 0    // lista vacia
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 nunca es repetido
```

- 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: 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