# Laboratorio A.E.D.

**Lars-Åke Fredlund**
lfredlund@fi.upm.es
**Tonghong Li**
tonghong@fi.upm.es
**Manuel Carro Liñares**
mcarro@fi.upm.es
**Germán Puebla Sánchez**
german@fi.upm.es
**Pablo Nogueira**
pnogueira@fi.upm.es

Viernes 11:00-13:00

# Entrega

- La fecha límite para optar a la máxima nota es
  Viernes 23 de noviembre de 2012, a las 13:00 horas
- Los ficheros que hay que subir son `CompareAlumnos.java` y `Insert.java` (son dos ficheros)
- La entrega se hace a través de la siguiente URL:
  `http://lml.ls.fi.upm.es/~entrega`
- El paquete `comparadores` esta documentado con Javadoc en
  `http://babel.ls.fi.upm.es/~fred/courses/aed/comparadores/`
- El proyecto debe compilar sin errores, cumplir la especificación y pasar el `Tester`.

# Configuración

- Arrancad Eclipse.
- Cread un paquete `comparadores` en el proyecto `aed`, dentro de `src`.
- Aula Virtual → AED → Sesiones de laboratorio → Laboratorio 6 → codigo_lab6.zip (formato zip).
- Importad al paquete `comparadores` los fuentes que habéis descargado
- Ejecutad `Tester`. Veréis que lanza una excepción:

```
Testing Comparator...

a1=Alumno {name=Jorge Valdano,matricula=01932,resultados=[7,5,10]} <
a2=Alumno {name=Maria Salvo,matricula=019959,resultados=[8,9]}
but compare(a1,a2) returns 0

Exception in thread "main" java.lang.Error
at comparadores.Tester.compareAlumnos(Tester.java:118)
at comparadores.Tester.doTest(Tester.java:58)
at comparadores.Tester.main(Tester.java:30)
```

# Tareas para hoy

- Hoy trabajaremos con comparadores, es decir clases que implementan la interfaz `Comparator<E>` del paquete `java.util`.
- Se pide completar la clase `CompareAlumnos` que implementa un comparador, y la clase `Insert` que usa un comparador para insertar un elemento en un array ordenado.
- Esta permitido añadir nuevos métodos, atributos privados, o variables locales.

# CompareAlumnos

- Se pide implementar el método `int compare(Alumno a1,Alumno a2)` de la clase `CompareAlumnos`.
- El método `int compare(Alumno a1, Alumno a2)` debe devolver un entero $< 0$ si a1 es menor que a2, 0 si a1 es igual que a2, y un entero $> 0$ si a1 es mayor que a2.
- Un objeto de la clase Alumno tiene tres métodos:
  - `String nombre()`
  - `String matricula()`
  - `int[] resultados()` – devuelve un array de enteros que corresponden a los resultados del alumno (en exámenes)
- Un alumno a1 es menor que un alumno a2 si:
  - La media de los resultados del alumno a1 es menor que la media de los resultados del alumno a2
  - o, si las medias de los resultados son iguales, si el nombre de a1 es menor que el nombre de a2
  - o, si las medias de los resultados y los nombres son iguales, si la matricula de a1 es menor que la matricula de a2

# `CompareAlumnos` (2)

- La media de un array de resultados se debe calcular usando *aritmética de enteros*.
- Ejemplo: Si un array tiene los elementos $[8, 5, 10]$ se calcula la media como $(8 + 5 + 10)/3 = 23/3 = 7$. La media del array $[7, 7]$ se calcula como $(7 + 7)/2 = 14/2 = 7$.
- Se puede asumir que un alumno tiene al menos un resultado, es decir, el método `resultado()` nunca devuelve un array con tamaño cero.
- Para comparar dos `String` (nombres o matriculas) se debe usar el método `int compareTo(String arg)` de la clase `java.lang.String`.

# insert

- Se pide implementar el método `void insert(E elem, E[] arr, Comparator<E> cmp)` de la clase `Insert`.
- `arr` es un array *ordenado*, según el comparador `cmp`, en orden *ascendente*.
  - Un ejemplo:

| 1 | 2 | 2 | 5 | null | null |
|---|---|---|---|---|---|

  - Para cada dos elementos consecutivos `arr[i]` y `arr[i+1]` que no son `null`, la llamada `cmp.compare(arr[i],arr[i+1])` devuelve un entero $\leq 0$.
  - Los elementos del array están almacenados consecutivamente y al partir de la primera posición con elemento `null`, no hay más elementos distintos de `null`.
- El método debería insertar `elem` en la posición correcta del array, para que el array sigua ordenado, desplazando los elementos una posición a la derecha del array para hacer hueco.
- En la implementación de `insert` se puede asumir que siempre hay celdas con valor `null` al final del array.

# Ejemplo `insert`

- Sea `v` un array de enteros:

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 4 | 5 | `null` | `null` |

- Después de la llamada `insert(v,3,intcmp)` – donde `intcmp` es un comparador "normal" sobre enteros – `v` tiene los elementos:

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | `null` |

  Observa que 3 ocupa el lugar de 4, y los elementos 4 y 5 han sido desplazados hacia la derecha del array.

- La siguiente llamada a `insert(v,9,intcmp)` debe dejár `v` con los siguientes elementos:

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 9 |