

**Lars-Åke Fredlund**

lfredlund@fi.upm.es

**Tonghong Li**

tonghong@fi.upm.es

**Pablo Noguiera**

pablo@babel.ls.fi.upm.es

**Manuel Carro Liñares**

mcarro@fi.upm.es

**Germán Puebla Sánchez**

german@fi.upm.es

Viernes 11:00-13:00

# Normas.

- ▶ **Si ya tenéis los laboratorios aprobados no hace falta volverlos hacer**

Si repetis los laboratorios perdeis las notas ya aprobadas

- ▶ **¡Solo una entrega por grupo!**
- ▶ Se puede entregar los resultados de este laboratorio sin penalización hasta el **Lunes 1 de Octubre, a las 16 horas**
- ▶ “Los ejercicios aceptados con posterioridad tendrán una reducción en su nota del 20 % por cada 24 horas posteriores a la fecha y hora límite. Llegado al 100 % de penalización se puede seguir entregando el ejercicio pero la nota máxima del mismo será 0.”
- ▶ No copiéis – comprobamos si hay plagio
- ▶ Usad las horas de tutoría para preguntar sobre programación – son oportunidades excelentes para aprender

## Configuración previa al desarrollo del ejercicio.

- ▶ Arrancad Eclipse. Debéis tener un acceso directo. Si trabajáis en portátil, se recomienda la versión 3.7 (Indigo) o 4.2 (Juno).
- ▶ Cambiad a “Java Perspective”.
- ▶ Debéis tener instalado Java 6.
- ▶ Crear un proyecto Java llamado aed:
  - ▶ Seleccionar separación de directorios de fuentes y binarios.
- ▶ Cread un *package* integerSet en el proyecto aed, dentro de src.
- ▶ Aula Virtual → AED → Sesiones de laboratorio → Laboratorio 1 → codigo\_lab1.zip (formato zip).
- ▶ Importad al paquete integerSet las fuentes que habéis descargado
- ▶ Ejecutad Tester. Veréis que lanza el mensaje de error:  
3 should be a member of {0,1,2,3,4}.

# Sistema de Entrega

- ▶ Todas 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 `IntegerSet.java`
- ▶ El sistema de entrega comprueba que el código entregado pasa los tests antes de aceptar la entrega
- ▶ **Probad antes a subir una implementación aunque sea incorrecta para verificar que tenéis acceso.**
- ▶ **Si no lo habéis hecho, tenéis que dar de alta vuestro grupo en**  
<http://lml.ls.fi.upm.es/~entrega/cgi-bin/aed/>

## Tareas para hoy

- ▶ Se pide modificar los métodos `member` y `insert` de la clase `IntegerSet`. Está permitido añadir nuevos métodos o variables.
- ▶ El método `member` debería devolver `true` si el entero `element` se encuentra en el array `setArray`, y `false` si no.
- ▶ El método `insert` debería devolver `true` si fue posible insertar el entero `element` dentro el array `setArray`, y `false` si el array ya contenía `element`, o si no hay un espacio libre para insertar el entero. El valor `null` indica un espacio libre en el array.
- ▶ Se pide que la implementación del método `member` sea **recursiva**. Es decir no está permitido usar bucles `for`, `while` o `do`. Hace falta hacer una llamada recursiva a algún método (por ejemplo un método nuevo).
- ▶ Se pide que la implementación del método `insert` **no sea recursiva**, es decir es *obligatorio* usar un bucle `for`, `while` o `do`. La implementación no debería usar “iteradores”.

# Tareas para hoy

- ▶ El proyecto debe compilar sin errores y debe cumplirse la especificación de los métodos a completar.
- ▶ Debe ejecutar `Tester` correctamente.
- ▶ Aunque `Tester` no comprueba el uso o no uso de recursion, el sistema de entrega en el web intenta advertir si se detecta una implementación que viola los requisitos sobre el uso de recursion:
  - ▶ `""WARNING: cannot detect recursion in the implementation of member"`
  - ▶ `""WARNING: detecting recursion in the implementation of insert through the method ..."`
- ▶ El package `integerSet` esta documentado mediante `"Javadoc"`. La pagina de web:  
<http://babel.ls.fi.upm.es/~fred/courses/aed/integerSet/>  
contiene la documentación generada.