Laboratorio A.E.D. Laboratorio 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 25 de Septiembre, 23:59 horas	10
Hasta el Martes 26 de Septiembre, 23:59 horas	8
Hasta el Miércoles 27 de Septiembre, 23:59 horas	6
Hasta el Jueves 28 de Septiembre, 23:59 horas	4
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

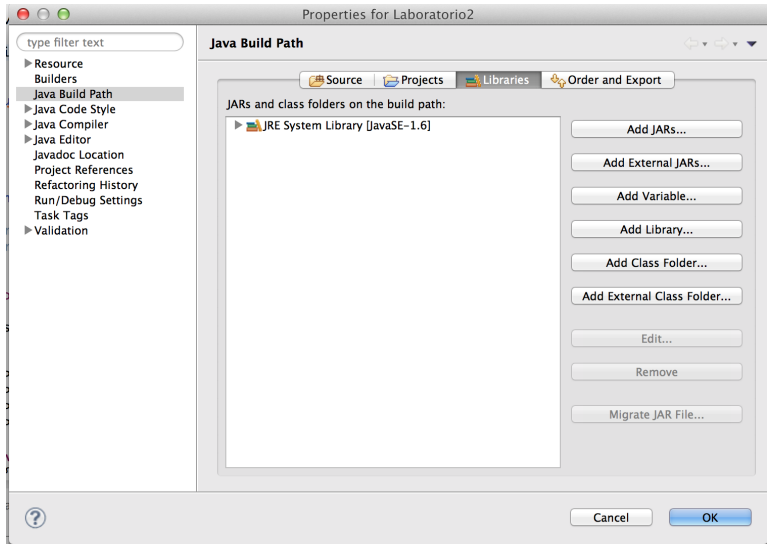
- Todos los ejercicios de laboratorio se deben entregar a través de la web `http://lml.ls.fi.upm.es/~entrega`.
- El fichero que hay que subir es `ActaNotasAED.java`.

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.actasnotas` en el proyecto `aed`, dentro de `src`
- Aula Virtual → AED → Laboratorios y Entregas Individuales → Laboratorio 1 → `Laboratorio1.zip`; descomprimidlo
- Contenido de Laboratorio1.zip:
  - `ActaNotas.java`, `Calificacion.java`, `TesterLab1.java`, `InvalidMatriculaException.java`, `CalificacionAlreadyExistsException.java`,...

Configuración previa al desarrollo del ejercicio.

- Importad al paquete `aed.actasnotas` los fuentes que habéis descargado (`ActaNotas.java`, `Calificacion.java`, `TesterLab1.java`, `InvalidMatriculaException.java`, `CalificacionAlreadyExistsException.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 `TesterLab1`

Documentación de la librería aedlib.jar

- La documentación de la API de la librería aedlib.jar esta disponible en `http://lml.ls.fi.upm.es/~entrega/aed/docs/aedlib/`.
- Tambien se puede añadir la documentación de la librería a Eclipse (*no es obligatorio*): en el "Package Explorer": "Referenced Libraries" $\rightarrow$ 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://lml.ls.fi.upm.es/~entrega/aed/docs/aedlib/`

y presionar el buton "Apply and Close"

Tarea: Implementar el interfaz `ActaNotas`

- Se predente desarrollar un sistema que permita gestionar los listados de notas de la asignatura
- Para ello se ha definido el interfaz `ActaNotas`
  - Permite dar de alta nuevas notas, actualizar, consultar y borrar notas
  - Permite obtener una lista de calificaciones que se encuentran por encima de una cierta nota
  - Permite obtener una lista de calificaciones ordenada usando diferentes criterios de ordenación. Para ello se utilizarán diferentes `Comparator<E>` (que no tendréis que implementar)

Tarea: Implementar el interfaz ActaNotas

- Se pide implementar el interfaz ActaNotas:

```
public interface ActaNotas {
  public void addCalificacion(String nombre, String matricula, int nota)
                throws CalificacionAlreadyExistsException;
  public void updateNota(String matricula, int nota)
                throws InvalidMatriculaException;
  public void deleteCalificacion(String matricula)
                throws InvalidMatriculaException;
  public int getCalificacion(String matricula)
                throws InvalidMatriculaException;

  IndexedList<Calificacion> getCalificaciones(Comparator<Calificacion> cmp);
  IndexedList<Calificacion> getAprobados(int notaMinima);
}
```

- La documentación detallada del funcionamiento de los métodos se ecuentra en el fichero ActaNotas.java, y tambien disponible en:
  http://lml.ls.fi.upm.es/~entrega/aed/docs/actasnotas/

La clase `ActaNotasAED`

- La clase `ActaNotasAED` debe implementar el interfaz `ActaNotas`

```
public class ActaNotasAED implements ActaNotas { ... }
```

- Para implementarlo podéis usar los atributos y métodos privados que consideréis necesarios
- Revisad los consejos antes de tomar decisiones

Ejemplo

```
ActaNotas acta = new ActaNotasAED ();

acta.addCalificacion ("Pepe","1",8); // [ Calificacion{"Pepe","1",8}]
acta.addCalificacion ("Luis","3",6); // [ Calificacion{"Pepe","1",8},
                                     //   Calificacion{"Luis","3",6} ]

acta.addCalificacion ("Paco","3",7); // CalificacionAlreadyExistsException

acta.getCalificacion ("1");      // returns Calificacion{"Pepe","1",8}

acta.getCalificacion ("2");      // InvalidMatriculaException

acta.updateNota("1",9);          // [Calificacion{"Pepe","1",9},
                                 //  Calificacion{"Luis","3",6}]

acta.updateNota ("2",6);         // InvalidMatriculaException

acta.addCalificacion ("Paco","5",3); // [ Calificacion{"Pepe","1",9},
                                     //   Calificacion{"Luis","3",6},
                                     //   Calificacion{"Paco","5",3} ]
```

Ejemplo (continuación

```
acta.getAprobados(5);              // returns [ Calificacion{"Pepe","1",9},
                                   //           Calificacion{"Luis","3",6} ]

acta.getAprobados (3);             // returns [ Calificacion{"Pepe","1",9},
                                   //           Calificacion{"Luis","3",6},
                                   //           Calificacion{"Paco","5",3} ]

acta.getCalificaciones(new CompNombre());
                                   // returns [ Calificacion{"Luis","3",6},
                                   //           Calificacion{"Paco","5",3},
                                   //           Calificacion{"Pepe","1",9} ]

acta.getCalificaciones(new CompNota());
                                   // returns [ Calificacion{"Paco","5",3},
                                   //           Calificacion{"Luis","3",6},
                                   //           Calificacion{"Pepe","1",9} ]

acta.deleteCalificacion("1");      // [ Calificacion{"Luis","3",6},
                                   //   Calificacion{"Paco","5",3} ]

acta.deleteCalificacion("1");      // InvalidMatriculaException
```

Consejos:

- Utilizar un atributo de tipo `IndexedList<Calificacion>` para almacenar las calificaciones
- Mantener la estructura *ordenada* usando la matrícula (recordad que `String` implementa el interfaz `Comparable`)
- Para los métodos `updateNota`, `deleteCalificacion`, `getCalificacion`:
  - Usar un método privado que devuelva la posición que ocupa la `matricula`, o -1 si no lo encuentra, con cabecera:

    ```
    private int buscarMatricula (String matricula)
    ```

  - Si la estructura está ordenada por matrícula, podeís hacer una búsqueda binaria en el método `buscarMatricula` (consiguiendo complejidad $O(log(n))$ en lugar de $O(n)$)

Seguir estos consejos os permitirá conseguir una mejor nota!

Consejos:

- Independientemente del orden utilizado para almacenar la estructura, el método

```
IndexedList<Calificacion> getCalificaciones(Comparator<Calificacion> cmp);
```

  debe devolver una **nueva lista** con todas las calificaciones
- Los elementos de la lista resultado deben estar en el orden que establece `cmp`
- No debe modificar el orden ni los elementos de la estructura interna que almacena las calificaciones
- Para crear la nueva lista podéis recorrer los elementos almacenados en el acta e ir insertándolos en el orden adecuado en la lista resultado (usando para ello `cmp`)

Seguir estos consejos os permitirá conseguir una mejor nota!

Notas

- El proyecto debe compilar sin errores y debe cumplirse la especificación de los métodos a completar
- Debe ejecutar `TesterLab1` 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