Laboratorio A.E.D. Ejercicio Individual 6

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 la penalización aplicada a la puntuación obtenida sobre 10:

Hasta el Lunes 20 de noviembre, 23:59 horas	0 %
Hasta el Martes 21 de noviembre, 23:59 horas	20 %
Hasta el Miercoles 22 de noviembre, 23:59 horas	40 %
Hasta el Jueves 23 de noviembre, 23:59 horas	60 %
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

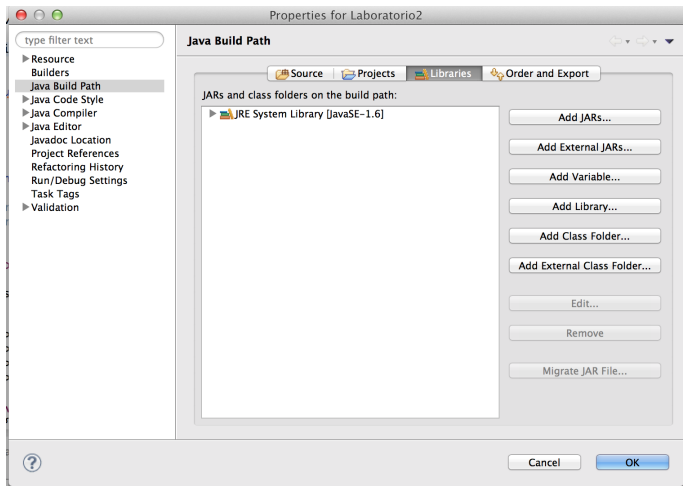
- Todos los ejercicios de laboratorio se deben entregar a través de la web `http://lml.ls.fi.upm.es/~entrega`.
- El fichero que hay que subir es `Find.java`.

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.find` en el proyecto `aed`, dentro de `src`
- Aula Virtual → AED → Laboratorios y Entregas Individuales → Individual 6 → `Individual6.zip`; descomprimidlo
- Contenido de `Individual6.zip`:
  - `aedlib.jar,FindNode.java,Find.java,TesterInd6.java`

Configuración previa al desarrollo del ejercicio.

- Importad al paquete `aed.find` los fuentes que habéis descargado
- **Añadid una nueva versión de la librería** `aedlib.jar` al proyecto `aed`. La librería esta disponible 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 `TesterInd6.java`

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/`
- También 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 botón "Apply and Close"

Tarea: buscar ficheros en una estructura de directorios representado por un árbol

- Se dispone de un árbol general `Tree<String>` que representa la estructura de directorios de un sistema operativo
- Los elementos del árbol son los nombres de los directorios y los ficheros
- Un ejemplo árbol:

```
|-- home
|   |-- aed
|       |-- files
|       |   |-- viva_las_vegas.mp3
|       |   |-- other
|       |   |   |-- TicketsMundialRusia2018.txt
|       |   |   |-- peli.mkv
|       |   |-- privado
|       |       |-- peli.mkv
```

Tarea

- Completar el método

```
public static void find(String fileName, Tree<String> directory) { ... }
```

  que imprime los "paths" (caminos) que llevan a nodos dentro el árbol `directory` con elementos iguales al parámetro `fileName`.
- El path de un nodo "n" es una `String`: la concatenación de los nombres de todos los nodos del camino que une la raíz con "n", empezando con el carácter `'/'`, y donde todos los nodos están separados por `'/'`.
- Para imprimir un camino deberías llamar al método `Print.println(String camino)` – el Tester comprueba que los caminos imprimidos son los esperados
- Notad que por defecto `Print.println(String camino)` no imprime nada a la consola. Para permitir output a la consola se puede llamar a `Print.enableOutput()` dentro vuestro método `find`.

Ejemplo

- Dado un árbol `t`:

```
|-- home
|   |-- aed
|       |-- files
|       |   |-- viva_las_vegas.mp3
|       |   |-- other
|       |   |   |-- TicketsMundialRusia2018.txt
|       |   |   |-- peli.mkv
|       |   |-- privado
|       |       |-- peli.mkv
```

- Como resultado de llamar `find` debería ser:

```
Llamada                              Imprime
=============================================================
find("examen-aed.pdf",t)

find("viva_las_vegas.mp3",t)     /home/aed/files/viva_las_vegas.mp3

find("peli.mkv",t)               /home/aed/files/other/peli.mkv
                                 /home/aed/files/privado/peli.mkv

find("privado",t)                /home/aed/files/privado
```

Reglas y Consejos

- El recorrido del árbol debería ser en orden "pre-orden".
- No es necesario distinguir entre ficheros y directorios a la hora de recorrer el árbol
- Solo se debería imprimir un camino por linea, sin lineas blancas, y sin espacio en blanco. El Tester comprueba que el formato de cada linea es correcta.
- Notad que la tarea es bastante parecido a la búsqueda de un tesoro en el laberinto; aunque no es obligatorio una solución usando recursividad puede resultar bastante elegante.
- Como pista para la solución recursiva, el método abajo podría ser un método auxiliar recursivo interesante:

```
private static void findInPos(Position<String> cursor,
                              String currentPath,
                              String fileName,
                              Tree<String> directory) { ... }
```

Notas

- El proyecto debe compilar sin errores y debe cumplirse la especificación de los métodos a completar
- Debe ejecutar `TesterInd6.java` correctamente y 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)