Laboratorio A.E.D. Ejercicio Individual 5

Guillermo Román
guillermo.roman@upm.es
Lars-Åke Fredlund
lfredlund@fi.upm.es
Manuel Carro Liñares
mcarro@fi.upm.es
Marina Álvarez
marina.alvarez@upm.es
Tonghong Li
tonghong@fi.upm.es
Raúl Correal
raul.correal@upm.es

Normas.

- **¡Solo debe entregar una persona por grupo!**.
- Fechas de entrega y nota máxima alcanzable:

Hasta el martes 29 de noviembre, 23:59 horas	10
Hasta el miércoles 30 de noviembre, 23:59 horas	8
Hasta el jueves 1 de diciembre, 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 `DiskUsage.java`.

Configuración previa

- Arrancad Eclipse. Debéis tener un acceso directo.
- 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.diskusage` en el proyecto `aed`, dentro de `src`.
- Aula Virtual $\rightarrow$ AED $\rightarrow$ Sesiones de laboratorio $\rightarrow$ Individual 5 $\rightarrow$ `Individual5.zip`; descomprimidlo.
- Contenido de Individual5.zip:
  - `TesterInd5.java`, `DiskUsage.java`, `Ejemplo.java`, `FileNode.java`, `Printer.java`, `AEDTree.java`

Configuración previa al desarrollo del ejercicio.

- Importad al paquete `aed.diskusage` los fuentes que habéis descargado (`TesterInd5.java`, `DiskUsage.java`, `Ejemplo.java`, `FileNode.java`, `Printer.java`, `AEDTree.java`)
- Añadid al proyecto `aed` la librería `net-datastructures-5.0.jar` que tenéis en Moodle
- Intentad ejecutar `TesterInd5`

Imprimir tamaño ficheros y directorios

- Se dispone de un árbol general Tree<FileNode> que representa la estructura de directorios de un sistema operativo
- Cada uno de los elementos del árbol es de tipo FileNode
  - Cada objeto de tipo FileNode tiene el nombre del directorio o fichero y el tamaño de dicho nodo

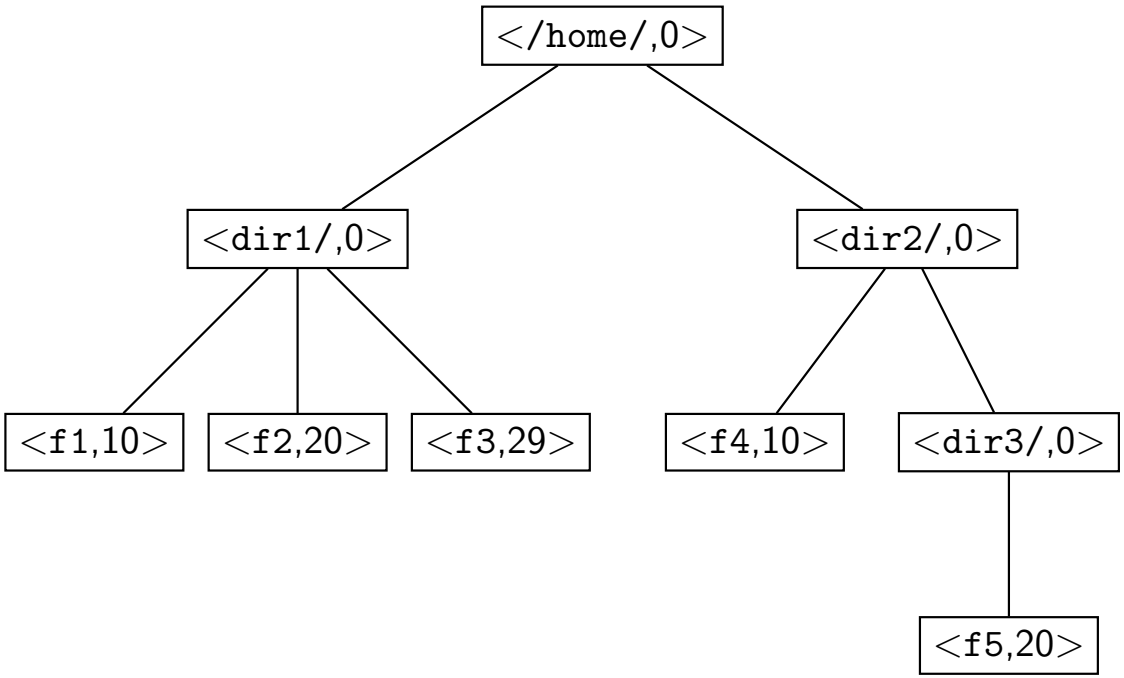
```
public class FileNode {
  private String name;
  private Integer size;

  public FileNode(String name, Integer size) {...}

  public String getName() { return name; }

  public Integer getSize() { return size; }
  ...
}
```

Imprimir tamaño ficheros y directorios

- Debéis implementar el método:
  `void printDiskUsage (Tree<FileNode> tree)`
- Dicho método imprime la estructura de directorios completa (directorios y ficheros)
- Lo hará en profundidad de izquierda a derecha
- Por cada fichero, imprimirá el tamaño que ocupa cada fichero y su path completo
- En el caso de los directorios, imprimirá el tamaño de todos los ficheros que contiene y el path completo del directorio
  - El path de un nodo "n" es la concatenación de los nombres de todos los nodos del camino que une la raiz con "n"
- Si un directorio contiene otros subdirectorios, el tamaño del directorio incluirá el tamaño de todos los ficheros contenidos en los subdirectorios
- Si un directorio no contiene ficheros también se debe imprimir

Ejemplo

Árbol de entrada:



Resultado:

```
10 /home/dir1/f1
20 /home/dir1/f2
29 /home/dir1/f3
59 /home/dir1/
10 /home/dir2/f4
20 /home/dir2/dir3/f5
20 /home/dir2/dir3/
30 /home/dir2/
89 /home/
```

Notas

- Usad la función `Printer.print` para imprimir
  - Podéis usar `System.out.print` para depurar
  - El Tester comprueba lo que imprimís usando `Printer.print`
- Cuidado con los espacios en blanco. Sólo hay un espacio en blanco entre el tamaño y el nombre del fichero/directorio
- El tester comprueba el orden de impresión de los ficheros/directorios
- NO debéis incluir las '/' para generar el path completo, ya están incluidas en los nodos del árbol que son directorios
- No es necesario distinguir entre ficheros y directorios a la hora de recorrer el árbol

Notas

- 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 `TesterInd5` 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