PRÁCTICA 8. Aplicaciones.Entrada/Salida de datos (java.io.*)

Archivos binarios y de texto. Serialización.

ENTRADA/SALIDA DE DATOS EN JAVA.

Los programas necesitan comunicarse con su entorno, tanto para recoger datos e información que deben procesar, como para devolver los resultados obtenidos. La manera de representar estas entradas y salidas en Java es a base de streams (flujos de datos). Un stream es una conexión entre el programa y la fuente o destino de los datos. La información se traslada en serie (un carácter a continuación de otro) a través de esta conexión. Esto da lugar a una forma general de representar muchos tipos de comunicaciones.

Por ejemplo, cuando se quiere imprimir algo en pantalla, se hace a través de un stream que conecta el monitor al programa. Se da a ese stream la orden de escribir algo y éste lo traslada a la pantalla. Este concepto es suficientemente general para representar la lectura/escritura de archivos, la comunicación a través de Internet o la lectura de la información de un sensor a través del puerto en serie.

Nota:

En esta práctica la resolución de los distintos ejercicios se basa esencialmente en el conocimiento y reutilización de las distintas clases del paquete java.io

Consulte y estudie mediante la documentación de la API las siguientes clases del paquete java.io

- **File**
- **FileInputStream, DataInputStream**
- **FileOutputStream, DataOutputStream**
- **FileReader, FileWriter, PrintWriter**
- **ObjectInputStream, ObjectOutputStream**

Ejercicios a realizar

1) Escriba un programa, UtilidadDirectorios, que genere un lista(do) con todos los directorios y archivo descendientes a partir de un directorio del disco. También debe comprimir a partir de un directorio todos los archivos que dependen de él, generando un nuevo archivo .zip. (ver los tests). Consulte y ayúdese en http://javaalmanac.com los ejemplos para el paquete java.io

   Nota: Incluir en el paquete: **com.mp.practica8.ejercicio1**

   Nota: Los tests se situaran en: **com.mp.practica8.ejercicio1.test**

2) A partir del archivo (**notas.txt**), con varias líneas de datos, que puede modificar con un editor. En cada línea, los datos se estructuran de la siguiente forma:

   Dni**;**NombreApellidos;NotaTeoria;NotaPracticas.

   Escriba un programa que lea el archivo procese los datos y genere un nuevo archivo de texto (**notasfinales.txt**), con la siguiente estructura:

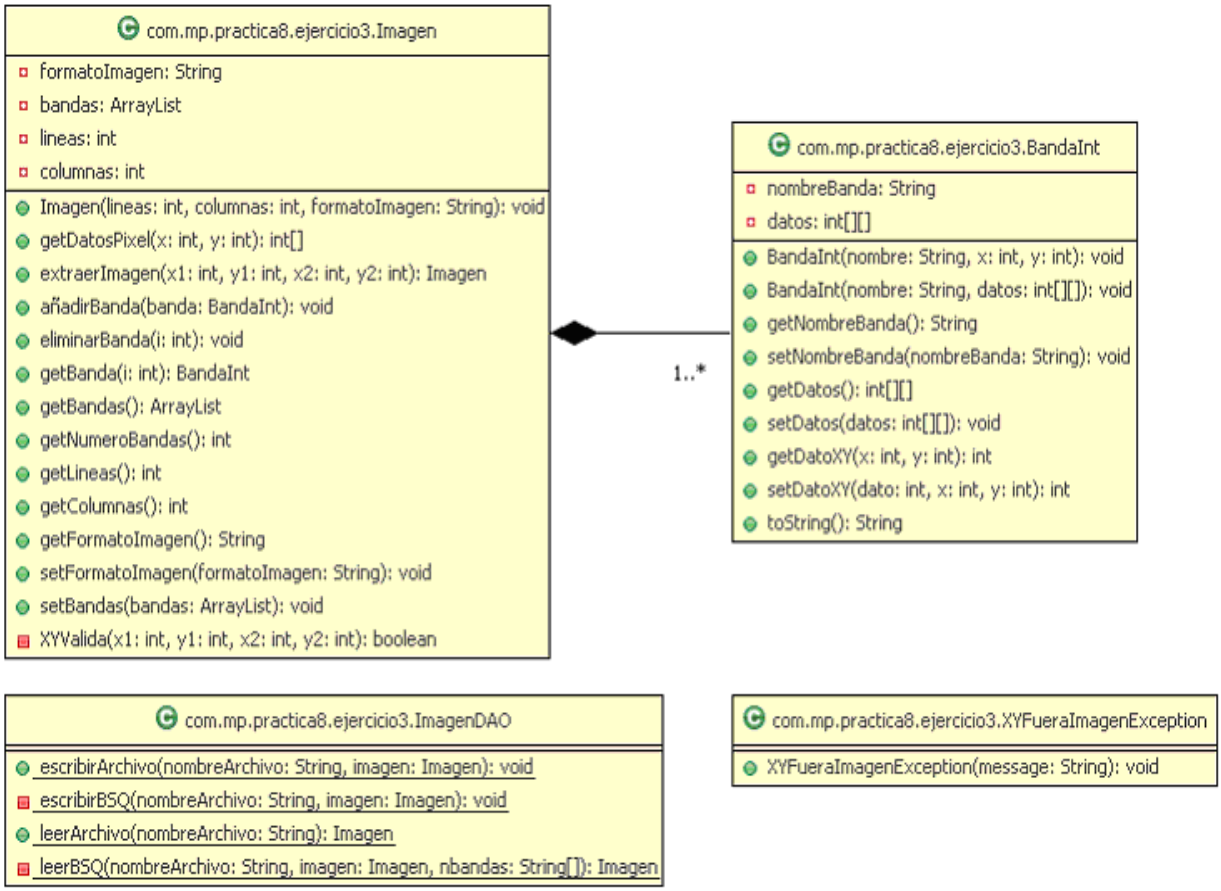
```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
Dni       Nombre                      Teoría   Prácticas       Final     Nota Acta
--------------------------------------------------------------------------------
54886796  Juan Lopez Garcia             2,80        5,60        3,78      Suspenso
74658923  Luis Cabrera Gomez            5,40        6,80        5,89      Aprobado
56231589  Jose Ruiz Ruiz                8,20        9,00        8,48 Sobresaliente

Suspensos:               1
Aprobados:               1
Notables:                0
Sobresalientes:          1

Número Alumnos:          3
Nota Media:           6,05
```

   Para calcular NotaFinal (numérico), se debe leer un archivo auxiliar coeficientes.txt, donde en la primera línea esta el coeficiente para teoría (por ejemplo 0,65) y en la segunda el coeficiente para prácticas (por ejemplo 0,35). En las siguientes líneas encontraremos los valores para discriminar las notas, por ejemplo en la línea 3, tendremos un 5, que se procesará de forma que la nota final entre 0 hasta 5 (no incluido) generará la NotaActa (texto) de Suspenso. La siguiente línea, con un 7, generará la Nota de Acta de Aprobado. Y así las demás para Notable y Sobresaliente.

   Nota: Incluir en el paquete: **com.mp.practica8.ejercicio2**

   Nota: Los tests y los archivos de datos se situaran en: **com.mp.practica8.ejercicio2.test**

3) Escriba un programa que debe procesar un fichero binario que almacena datos de tipo int, podría representa una imagen con datos físicos de una área de un mapa. El archivo esta organizado por distintas bandas, todas del mismo tamaño y tipo. Todas las bandas están caracterizadas por número de líneas y píxeles por línea (filas, columnas). Esta información junto con el nombre del archivo esta almacenada en un archivo de texto cabecera. (Vea los tests).



Nota: Incluir en el paquete: **com.mp.practica8.ejercicio3**

Nota: Los tests se situaran en: **com.mp.practica8.ejercicio3.test**

4) Introduzca una nueva funcionalidad al ejercicio sexto ejercicio de la séptima práctica (residencia). Ahora se pide que además de la funcionalidad original, el programa permita la persistencia de los datos mediante archivos. Diseñe una nueva clase ResidenciaDAO que permita las operaciones de guardar y recuperar los datos relevantes (objetos) de la aplicación en archivos. Utilice la interfaz Serializable y las clases ObjectInputStream y ObjectOutputStream.

Nota: Incluir en el paquete: **com.mp.practica8.ejercicio4**

Nota: Los tests se situaran en: **com.mp.practica8.ejercicio4.test**

5) En este último ejercicio deberá rellenar una encuesta sobre las prácticas. Como resultado de la ejecución se genera un archivo de texto con los datos.

   Debe rellenar cada alumno de la pareja una encuesta independiente. EncuestaTestAlumnoA para el primer alumno de la pareja y EncuestaTestAlumnoB para el segundo.

   Lea con atención como rellenar por código cada tipo de pregunta.

   Para la aceptación de las prácticas es necesario que cada alumno rellene esta encuesta que se asocia al proceso de presentación de las prácticas.