

AD – TEMA 1 – BOLETIN EJERCICIOS 1.

EJERCICIO 1

Un periódico digital organiza sus noticias mediante el uso de una clase Noticia que incluye la siguiente información:

- Fecha (entero en el siguiente formato aaaammdd, por ejemplo, 20120130 representa el 30 de enero de 2012).
- La hora (entero en el formato hhmm, por ejemplo, 1005 quiere decir 10:05).
- El texto de la noticia (String).
- Número de veces que se ha leído la noticia (entero).

Implementa la clase Noticia para poder almacenar toda esta información. Hay que tener en cuenta que las siguientes restricciones del sistema:

- Todos los atributos serán privados.
- Construir un constructor de la clase dado una fecha, hora y texto de la noticia.
- Implementar los siguientes métodos de la clase:

```
// Devolver la fecha en el formato aaaammdd  
public int getData()
```

```
// Devolver el número de lecturas  
public int getLecturas()
```

```
// Incrementar en uno el número de lecturas  
public void incLecturas()
```

```
// Comprobar si una fecha dada por d, m, a es igual a la fecha de la  
// noticia actual  
public boolean igualData(int d, int m, int a)
```

```
// Devuelve un String con la información de la noticia  
public String toString() {  
    String s = "";  
    s+=fecha%100 + "/" + (fecha/100)%100 + "/" + (fecha/10000) + " - ";  
    s+=(hora/100) + ":" + (hora%100) + "\n";  
    s+=texto + "\n";  
    s+="Leída " + lecturas + " veces\n";  
    return s;  
}
```

El periódico se compone de un conjunto de noticias (se supone que no hay más de 1000 noticias disponibles), y proporciona funcionalidades tales como obtener la primera noticia de un día determinado, mostrar las noticias más populares (las más leídas) y borrar las noticias más viejas.

Para representar este conjunto de noticias se pide escribir una clase Periódico que incluirá:

- Atributos:
 - Un atributo noticias, vector de Noticia, con capacidad para 1000 elementos.
 - Un atributo entero numNoticias, que representa el número de objetos Noticia almacenados en cada momento en el vector.

Nota: Los objetos Noticia se han de situar siempre al principio del vector y no se mantendrán ordenados por fecha. Asá, el atributo numNoticias indica también la primera posición libre del vector.

- Un constructor que inicialice los atributos, creando un vector de 1000 posiciones, sin ninguna noticia inicial.
- Implementar los siguientes métodos de la clase:

```
// Inserta una noticia
```

```
public void insertar(Noticia n)
```

Que dada una Noticia, inserta la noticia en el vector, si cabe.
En caso contrario, no hace nada.

```
// Devuelva la primera noticia
```

```
public Noticia primeraNoticia(int d, int m, int a)
```

Que devuelva la primera Noticia que se encuentre en sentido ascendente en el vector con la fecha dada por d, m y a.
Si no existe ninguna noticia de esa fecha o no existen noticias en el vector devolveremos null para indicar esta situación.

```
// Imprimir la noticias más populares
```

```
public void masPopulares()
```

Muestra por pantalla toda la información de las noticias más populares (es decir, las noticias más leídas). En el caso de que haya varias noticias con el mismo número de lecturas, se mostraran todas ellas.
En caso de que no haya noticias, se mostrara un mensaje que lo indique.

```
// Borrar noticias anteriores a una fecha  
public void borrarAnteriores(int d, int m, int a)
```

Eliminar del vector las noticias anteriores a una fecha dada por d, m y a. Después de borrar, los elementos del vector han de permanecer contiguos (es decir, sin espacios entre los elementos).

Una vez que tenemos la estructura de datos vamos a proceder al tratamiento con los ficheros. Vamos a realizar lo siguiente:

- Cuando vayamos a cerrar el programa tendremos que preguntar si queremos salvar las noticias, así como los datos de periódico. Si no queremos salvarlo cerramos el programa, pero en caso de que queramos salvarlo debemos crear dos ficheros, uno para las noticias y otro para el periódico. El fichero ha de ser de texto, y el formato de texto en el que queráis salvarlo lo debéis elegirlo vosotros, aunque se requiere que los atributos estén separados por el carácter #.
- Al abrir el programa, si existen los ficheros que hemos salvado se cargaran automáticamente en memoria, sin preguntar si queremos salvarlos o no, y a continuación se borrarán los ficheros, de forma que mantendremos toda la estructura en memoria.
- Si deseamos volver a crear los ficheros solo debemos volver a salvarlos al cerrar el programa.

EJERCICIO 2

Realiza un programa en Java que trabaje con la clase File.

```
1.- Asignar carpeta de trabajo  
2.- Listado de la carpeta de trabajo  
3.- Información detallada de un archivo  
4.- Creación de un directorio  
5.- Creación de un fichero  
6.- Eliminación de un fichero  
7.- Salir
```

Para ello realizar una función para cada una de las opciones descritas en el menú anterior.

Crearemos la clase necesaria llamada Ejercicio1.java dentro del paquete ejercicio2.

COSAS A TENER EN CUENTA:

- Crear por defecto una carpeta de trabajo inicial a / o c:\
- Listar la carpeta de trabajo indicando si son directorios o ficheros cada uno de los archivos.
- Intentar controlar el máximo número de excepciones que puedan suponer no existir un archivo del que queramos información detallada, a la hora de crearlo o eliminarlo.

EJERCICIO 3

Intenta reestructurar el ejercicio 2 mediante la programación orientada a objetos, creando una clase `GestionFicheros.java` y otra clase principal `Ejercicio2.java` donde crearemos un objeto de la anterior clase para llamar a sus métodos.

Todo las clases anteriores estarán dentro del paquete llamado `ejercicio2`.

EJERCICIO 4

Realiza un programa que pida dos números al usuario. Los guarde en un fichero binario (`ejercicio1.data`) y guarde el resultado de las operaciones de suma, resta y multiplicación.

Después de guardar la información en el fichero lo lea y lo muestre por pantalla.

EJERCICIO 5

Realiza un programa en Java que haga una copia exacta de un fichero a otro de todos sus datos. Para ello vamos a trabajar con ficheros binarios orientados a bytes.

NOTA: Los nombres del fichero a copiar será `ejercicio5.data` y su destino `ejercicio5Copia.data`.

EJERCICIO 6

Realiza un programa que lea un fichero binario y carácter a carácter de un fichero y convierta a mayúsculas en el fichero todas las letras 'a'. El programa recibirá como parámetro de entrada el fichero, y como salida generará un nuevo fichero con el resultado en texto, para poder visualizarlo, que recibirá el nombre del fichero de entrada seguido de "Copia".

EJEMPLO: De entrada recibirá el fichero `ejercicio6.dat` y el resultado será `ejercicio6Copia.txt`.

EJERCICIO 7

Realiza un programa que contenga un menú que aparezca en pantalla y muestre las siguientes opciones:

```
1.- Leer desde teclado  
2.- Leer desde fichero  
3.- Escribir en fichero  
4.- Fin
```

- Tecleando 1: Nos pedirá que le introduzcamos un texto y lo imprimiremos en pantalla indicando al final de la frase “Gracias por la frase”.
- Tecleando 2: Nos pedirá el nombre/ruta del fichero a leer e imprimiremos todo el fichero por pantalla.
- Tecleando 3: Nos pedirá el nombre/ruta del fichero a escribir y nos pedirá que introduzcamos un texto por pantalla, que luego escribiremos en el fichero. Si el fichero existe, añadiremos el texto al final del fichero.
- Tecleando 4: Finalizará el programa.

Nota: Mientras no finalicemos seguirá mostrando el menú para volver a realizar tareas.

EJERCICIO 8

Realiza un programa que pida por pantalla de un Vehículo la siguiente información:

- La marca
- El modelo del coche
- La matricula
- La potencia
- El color

Los guarde en un objeto `vehículo` y los escriba en un fichero llamado (`vehiculo.bin`). Si cerramos el programa y volvemos a abrirlo se añadirán objetos al final del fichero.

En el mismo programa incorpora una opción que permita visualizar por pantalla todos los coches del fichero:

```
Opción 1.- Insertar vehículo  
Opción 2.- Visualizar vehículos  
Opción 3.- Salir
```

EJERCICIO 9

Implementa un programa que almacena en un fichero secuencial binario registros que contengan datos informativos de los alumnos de la asignatura. Para ello, crea un método `main` donde se solicite al usuario que introduzca mediante teclado el nombre del archivo donde se almacenarán los registros.

A continuació, el usuari deberá introducir mediante teclado el dni de cada alumno, la nota que obtuvo en el primer parcial y la nota que obtuvo en el segundo parcial. Por cada alumno introducido por el usuario, el programa deberá crear y almacenar un registro que contenga el dni del alumno representado por un valor numérico entero, y las notas del primer y segundo parcial representadas por dos valores numéricos reales.

Escribir otro programa que contenga un nuevo método main donde se lea del fichero creado anteriormente todos los datos y se imprima por pantalla.

Usar las excepciones que consideréis necesarias para el buen funcionamiento de los programas, haciendo especial hincapié en aquellas relacionadas con el buen uso de los ficheros.

EJERCICIO 10

Haz una clase llamada Agenda para manejar una agenda de teléfonos. Los objetos instanciados a partir de esa clase tendrán la siguiente funcionalidad:

- Construir una agenda a partir de un número máximo de contactos posibles (constructor)
- Añadir un contacto nuevo
- Listar todos los contactos de la agenda
- Buscar un contacto por nombre
- Buscar un contacto a partir de un número de teléfono
- Borrar un teléfono dado un nombre
- Modificar un número de teléfono dado un nombre

Puedes añadir más métodos si lo consideras adecuado.

Las entradas de la agenda se modelarán mediante objetos de la clase Contacto que habrá que crear. Para cada Contacto de la agenda se almacenará la siguiente información:

- Nombre
- Dirección
- Código Postal
- Teléfono

Puedes añadir los métodos que consideres adecuados.

Para probar tus clases hay que escribir un programa TestAgenda.java donde se instancie un objeto de la clase Agenda y mediante un menú se pueda probar sus funcionalidades. En el menú se tiene que dar también la posibilidad de leer y escribir la agenda desde un fichero binario, pidiendo el nombre y de crear una nueva agenda.