

Ejercicios de Archivos

1. Crear una clase Cuenta con un método llamado "cuentaPalabras()" que cuente las palabras existentes en un archivo de texto pasado como parámetro.
El método recibirá como parámetro un objeto File y retornará la cantidad de palabras que componen el texto.
El proceso será ir leyendo líneas del archivo y contar cuantas palabras hay en cada una.
Recordad que esto lo hicimos con un StringTokenizer.
2. En el vivero "La Planta feliz", disponen de un archivo donde está guardada toda la información de las plantas que tienen a la venta. Este fichero se llama "Plantas.venta", y contiene objetos de la clase Planta.

Archivo "Plantas.venta"

Objeto Planta	Objeto Planta	Objeto Planta	Objeto Planta	...
------------------	------------------	------------------	------------------	-----

Ante la inminente llegada del verano, el vivero ha decidido hacer una campaña de rebajas, haciendo descuentos del 20, 30 y hasta el 40%.

Realiza un programa que se encargue hacer la modificación de cada uno de los objetos planta aplicando los descuentos en el precio de cada planta, sabiendo que los descuentos se aplican así:

- 20% para las plantas con una cantidad inferior a 10
- 30% para las plantas con una cantidad entre 10 y 50
- 40% para las plantas cuya cantidad supere los 50

La clase Planta contiene los siguientes atributos:

```
private int codigo;  
  
private String nombreVulgar;  
  
private String nombreCientifico;  
  
private double precio;  
  
private int cantidad;
```

En la carpeta encontrarás la clase Planta y un pequeño programa para generar de forma automática el archivo "Plantas.venta".

3. El ayuntamiento de Toledo contrata a varios empleados para realizar la limpieza de parques y jardines de la ciudad. El capataz de la cuadrilla de limpieza anota cada día en un archivo los trabajadores que han acudido al trabajo, a fin de realizar al final de mes un informe con las jornadas de trabajo que hay que pagarle a cada empleado.

El archivo donde el capataz realiza las anotaciones se llama "Peonadas.txt" y contiene:

```
1/5/2024:Juan,Pedro,Daniel,Carlos
2/5/2024:Pedro,Carlos,Luis,Daniel
3/5/2024:Carlos,Pedro
4/5/2024:Juan,Luis,Pedro
```

Y el archivo donde se guardan los nombres de los trabajadores se llama "Trabajadores.txt" y contiene:

```
Juan
Pedro
Daniel
Carlos
Luis
```

Cuando el mes acaba, el capataz nos entrega los dos archivos y nosotros debemos generar un archivo con la cantidad de peonadas que hay que pagar a cada trabajador.

Debemos realizar un programa que genere el archivo "TotalPeonadas.txt" que contenga la siguiente información:

```
Juan:2
Pedro:4
Daniel:2
Carlos:3
Luis:2
```

Nota: El archivo de Peonadas contiene una línea por cada día de trabajo. Cada trabajador está separado del siguiente por una coma. No existen espacios en blanco en el interior de las líneas.

4. La página de venta de coches usados “www.cochesdeocasion.com” tiene un archivo donde guarda toda la información de sus vehículos. Este archivo contiene objetos de la clase Vehiculo, que tienen los siguientes atributos:

```
public class Vehiculo{  
    private String matricula;  
    private String modelo;  
    private float precio;  
    private LocalDate entradaEnExposicion;  
  
    ...  
}
```

La política de esta página es que, si un coche lleva 5 días en exposición y no se ha vendido, debe rebajarse su precio un 15%.

Realiza un programa que actualice los precios de todos los vehículos, teniendo en cuenta su fecha de entrada en la exposición.

Nota: Suponemos que el archivo de vehículos está en perfectas condiciones, y que todos los objetos tienen toda su información correcta. No será necesario realizar ningún tratamiento de errores.

5. En una web de información sobre selecciones de Futbol Mundiales disponen de varios archivos con la información de todos los jugadores, pero necesitamos un archivo donde esté organizada de manera conjunta todos los datos de jugadores.

Disponemos de varios archivos de texto que contienen la siguiente información:

Archivo de Jugadores: Contiene el nombre del jugador y un código de demarcación y otro de la selección a la que pertenece.

```
David de Gea:D1:S1  
Antoine Griezmann:D4:S3  
Sergio Busquets:D3:S1  
Cristiano Ronaldo:D4:S2  
Lionel Messi:D4:S4
```

Archivo de Demarcaciones: Contiene una lista con las posibles demarcaciones de un jugador.

```
D1:Portero
D2:Defensa
D3:Centrocampista
D4:Delantero
```

Archivo de Selecciones: Códigos de cada una de las selecciones.

```
S1:España
S2:Portugal
S3:Francia
S4:Argentina
```

Los tres archivos contienen información sobre los jugadores, pero queremos reunirla toda en un solo archivo.

Debemos realizar un programa que mezcle los datos de los tres archivos y cree un archivo con toda la información. El archivo resultante debe contener una línea para cada jugador con su demarcación, y todos ellos divididos por selecciones.

El archivo resultante debe contener el nombre de la selección y luego todos sus jugadores, otra selección y todos sus jugadores, etc.

El archivo resultante debe tener la siguiente estructura:

Archivo de Resultado:

```
España
David de Gea:Portero
Sergio Busquets:Centrocampista

Portugal
Cristiano Ronaldo:Delantero

Francia
Antoine Griezmann:Delantero

Argentina
Lionel Messi:Delantero
```

Nota: Supondremos que los archivos tienen todos los datos correctos, no sobra ni falta ninguna línea. Cuando creamos el archivo resultado, no deben estar ordenados los futbolistas dentro de cada país. Simplemente agruparemos los jugadores de cada país sin tener en cuenta nada más.