



FICHEROS DE TEXTO

EJERCICIO 1

Dado el fichero **números.txt** que almacena en cada línea una serie de números enteros, completar el método `ejercicio1` que calcule la su suma de los positivos mayores de 10.

EJERCICIO 2

Dado el fichero **empleados.txt** que contiene los nombres de los empleados y las ventas que ha realizado cada uno, cuyo contenido puede ser, por ejemplo:

```
//El tercer campo es la provincia donde trabajan
Juan,1000,Jaen
Antonio,3200,Huelva
Oscar,2500,Granada
Pedro,500,Málaga
```

Completar el método `ejercicio2` que devuelve un String con los empleados que hayan vendido más de 1000 euros y la media de ventas de la empresa.

EJERCICIO 3

Completar el método `ejercicio3` que partiendo de fichero anterior crea otro fichero donde los datos están copiados, pero que el separador se cambie por `=`.

VARIACION: Realizar una versión alternativa del `ejercicio3`, donde se copien los datos del fichero original en un fichero XML con el siguiente formato:

```
<empleados>
  <vendedor>
    <nombre>Juan</nombre>
    <ventas>1000,</ventas>
    <provincia>Jaén</provincia>
  </vendedor>
  ...
</empleados>
```

EJERCICIO 4

Completar el método `ejercicio4` que debe crear un fichero llamado **provincias.txt** con la suma de las ventas por las 8 provincias de Andalucía.



EJERCICIO 5

```
titulo:autor:precio:numero_ejemplares
Harry Potter y el caliz de fuego:JK Rowling:15.6:900
La historia interminable:Michael Ende:5.96:9
Codigo Da Vinci:Dan Brown:10.95:20
```

Crear una aplicación que guarda información completa de libros de una biblioteca usando HashMap y ficheros de texto. Los datos que se guardan de cada libro son título, autor, precio y número de ejemplares disponibles en **libros.txt**. Hacer los siguientes métodos:

- 1) El método **cargarLibros** de la clase **AccesoDatos** mete toda la información del fichero que se recibe como parámetro en objetos Libro en el HashMap disponible.
- 2) Completar el método **visualizarLibros** que devuelve un ArrayList de **String[]** con toda la información todos los libros para poder ser visualizados como una tabla de Swing. Para ello usar el método auxiliar disponible **crearTabla** que dada una colección de libros (Clase **Collection** por compatibilidad) devuelve el **ArrayList** necesario usando el método de la clase Libro **valoresAtributos** que devuelve un **String[]** con los datos ordenados de los atributos
- 3) Completar el método **buscarTitulo** que devuelve un **String[]** con la información del libro buscado por título. Para ello usar el método de la clase Libro **valoresAtributos**. Si no existe lanzar una excepción con un mensaje adecuado.
- 4) Completar el método **buscarLibrosAutor** que devuelve un ArrayList de **String[]** con los libros del autor que se pasa por parámetro. Usar para ello el método **crearTabla** como en el ejercicio en el método **visualizarLibros**. Si no existen libro del autor lanzar una excepción con un mensaje adecuado”.
- 5) Completar el método **añadirLibro** que recibe los datos de un libro y lo añade a la lista. Si ya existe el título en la biblioteca lanzar una excepción.
- 6) Completar el método **ordenarLibros** que recibe un valor indicando si se ordena por precio o por cantidad de ejemplares descendente o ascendente indicando “ASC” o “DESC” admitiendo también en minúsculas. El método devuelve un **ArrayList de String[]** de los datos ordenados usando el método **crearTabla**
- 7) Completar el método **modificarTitulo** que recibe un String con el título del libro a modificar y otro String que es el nuevo título para el libro. Si no existe el título en la biblioteca y/o si el nuevo título ya existe lanzar una excepción.



- 8) Completar el método **guardarLibros** recibe el nombre de un fichero y guarda todos los objetos que haya en el HashMap en un fichero con dicho nombre usando el mismo formato que para la lectura.

EJERCICIO 6

Crear una aplicación que guarda información acerca de jugadores de futbol en un HashMap y ficheros de texto. Los datos que se guardan de cada futbolista son nombre, posición (portero, defensa, medio, delantero), club actual y número de goles anotados. Existe un fichero llamado **jugadores.txt** con el siguiente formato:

```
nombre:posicion:goles:equipo
Messi:delantero:10:PSG
Sergio Ramos:defensa:15:PSG
Cristiano Ronaldo:delantero:21:Manchester United
Gerard Pique:defensa:5:FC Barcelona
Luka Modric:centrocampista:10:Real Madrid
Kilyan Mbappe:delantero:40:PSG
Robert Lewandowski:delantero:30:FC Barcelona
```

Completar los siguientes métodos:

- 1) El método **cargarJugadores** de la clase AccesoDatos mete toda la información del fichero que se recibe como parámetro en objetos Jugador en el **HashMap** disponible.
- 2) Completar el método **visualizarJugadores** que devuelve un **ArrayList** de **String[]** con toda la información todos los libros para poder ser visualizados como una tabla de Swing. Para ello usar el método auxiliar disponible **crearTabla**.
- 3) Completar el método **buscarJugador** que devuelve un **String[]** con la información del jugador buscado por nombre. Para ello usar el método de la clase Jugadores **valoresAtributos**. Si no existe lanzar una excepción con un mensaje adecuado.
- 4) Completar el método **añadirJugador** que recibe los datos de un jugador y lo añade al HashMap. Si ya existe el jugador lanzar una excepción.
- 5) Completar el método **defensasGoleadores** que devuelve un **ArrayList** de **String[]** con los jugadores que son defensas y que han marcado más de 5 goles. Si no existen jugadores que cumplan con los anteriores lanzar una excepción.
- 6) Completar el método **ordenarJugadores** que recibe un valor indicando si se ordena por goles o por nombre de jugador descendente o ascendente indicando "ASC" o "DESC" admitiendo también en minúsculas. El método devuelve un **ArrayList** de **String[]** de los datos ordenados usando el método **crearTabla**



- 7) Completar el método **modificarNombre** que recibe un String con el nombre del jugador a modificar y otro String que es el nuevo nombre para el jugador. Si no existe el jugador en lista y/o si el título ya existe lanzar una excepción y no cambiarlo.
- 8) Completar el método **guardarJugadores** recibe el nombre de un fichero y guarda todos los objetos que haya en el HashMap en un fichero con dicho nombre usando el mismo formato que para la lectura.
- 9) Completar el método **resumenEquipos** que devuelve un **String** con un resumen de los equipos y la suma de goles de todos sus jugadores.
- 10) Completar el método **backupJugadores** que recibe un nombre de un fichero y guarda los 5 máximos goleadores de la liga en un formato legible con todos los datos.
- 11) Hacer otra versión del método anterior creando un fichero con formato XML.
- 12) Hacer otra versión del método anterior creando un fichero con formato JSON.