Programación orientada a objetos. Ejercicios con archivos

Ejercicio 2.- Escribe un programa que permita mantener la información del fichero de clientes de una empresa. Deber incluir, al menos, altas, bajas, modificaciones, consultas y búsqueda de un elemento. Las consultas y búsquedas deben poder hacerse por apellidos y por CIF.

Se partiría de un fichero (CLIENTES.TXT) **SECUENCIAL** con la información siguiente:

- nombre: cadena. Máximo 50 caracteres
- apellidos: cadena. Máximo 50 caracteres.
- CIF: cadena. Máximo 10 caracteres
- categoría: entero.
- dirección: cadena. Máximo 50 caracteres

Los datos se encuentran ordenados ascendentemente por CIF. Cada uno de los campos del registro se encuentra separado por comas(,).

→ Puede ser importante para este ejercicio que investiguéis el método **Split** del paquete Java.lang.String

Los procesos principales serán:

- **Gestión de Clientes** que tendrá los siguientes procedimientos:
    - altaCliente: procedimiento que registra un cliente en el final de un fichero de texto. Interfaz “void altaCliente(File fichero,Cliente cliente)”
    - consultarElementoPorApellidos: procedimiento que muestra el registro al que le corresponda los apellidos pasados por parámetro. Interfaz:
      “void consultarElementoPorApellidos(File fichero, String apellidos)”
    - buscarElementoPorApellidos: funcion que devuelve el registro al que le corresponda los apellidos pasado por parámetro. Interfaz:
      “String buscarElementoPorApellidos (File fichero, String apellidos)”
    - consultarElementoPorCif: procedimiento que muestra el registro al que le corresponda el cif pasado por parámetro. Interfaz: “void consultarElementoPorCif(File fichero, String cif)”
    - buscarElementoPorCif: funcion que devuelve el registro al que le corresponda el cif pasado por parámetro. Interfaz:
      “String buscarElementoPorCif (File fichero, String apellidos)”
    - insertarModificacionCliente: procedimiento que inserta en el fichero de modificaciones un cliente para ser modificado en un futuro sobre le original de clientes. No se valida que el cliente exista en el fichero maestro. Interfaz:
      “void insertarModifacionCliente(File fichero, Cliente cliente)”

- realizarModificaciones: procedimiento que recoge en el fichero clientes todas las modificaciones registradas en el fichero de modificaciones. Interfaz:
  "void realizarModificaciones(File clientes, File modificaciones)"

- insertarBajaCliente: procedimiento que inserta en el fichero de bajas un cliente para ser dado de baja en un futuro sobre el fichero original de clientes. No se valida que el cliente exista en el fichero maestro. Interfaz:
  "void insertarBajaCliente(File fichero, Cliente cli)"

- realizarBajas: procedimiento que recoge en el fichero clientes todas las altas registradas en el fichero de bajas: Interfaz: "void realizarBajas(File clientes, File bajas)"

- realizarAltas: procedimiento que recoge en el fichero clientes todas las altas registradas en el fichero de altas: Interfaz: "void realizarAltas(File clientes, File altas)"

- transformarAObjeto: función que transforma un registro cliente en forma de cadena a un objeto cliente. Interfaz "Cliente transformarAObjeto(String registro)"

- ordenarFichero: método que ordena un fichero ascendentemente según el compareTo implementado para la clase Cliente

- altaClienteOrdenado: procedimiento que registra un cliente en el lugar que le corresponde en un fichero de texto segun cif. Interfaz:
  "void altaClienteOrdenado(File fichero, Cliente cliente)"

- modificarElemento: procedimiento que modifica un elemento del fichero directamente con ayuda de un fichero auxiliar. Interfaz:
  "void modificarElemento(File fichero, String registroViejo, Cliente registroNuevo)"

- bajaElemento: procedimiento que borra del fichero un elemento directamente con ayuda de un fichero auxiliar. Interfaz: "void bajaElemento(File fichero, String registro)"