

Ejercicio Agencia Inmobiliaria

NORMAS:

Queda prohibido modificar los ficheros csv y el main del programa principal

Si se tiene que implementar algún método necesario para la correcta ejecución del Código, se declarara siempre como privado, por no estar en la descripción del ejercicio.

No se admiten System.out o System.in fuera de la clase Main. Los mensajes o avisos se realizan mediante "paso de mensajes" utilizando a técnica que se prefiera

Introducción

Implementamos una aplicación para la gestión de los inmuebles que tiene en cesión una inmobiliaria.

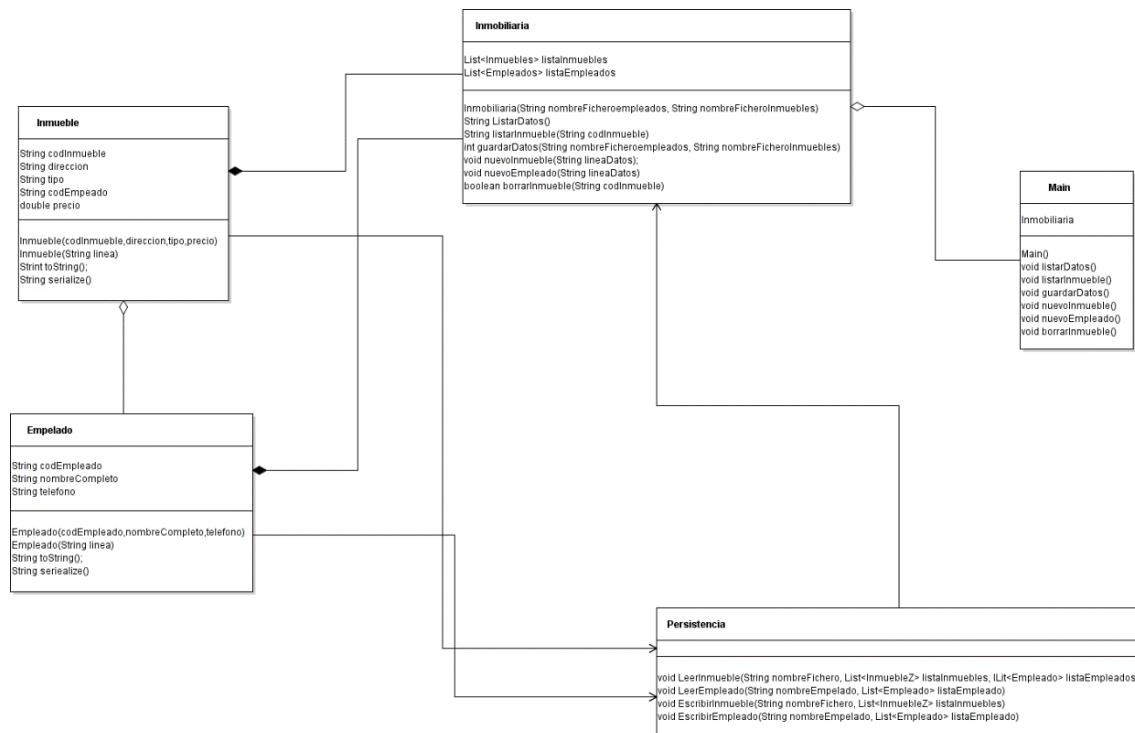
Por un lado la empresa almacena la información de sus empleados

- String codEmpleado
- String nombreCompleto
- String teléfono

Cada inmueble tiene asociado un empleado que es el responsable de su gestión. La información que almacenamos de cada inmueble es:

- String codInmueble
- String direccion
- String tipo
- **String codEmpleado** : Es una clase ajena de Empleado
- double precio

La gestión de la información se realiza en combinación con excel y la nueva aplicación que estamos creando



Gestión de empleados

En este módulo gestionamos toda la información relativa a los empleados que se encuentra almacenada en empleados.csv

Empleados
Empleado(codEmpleado,nombreCompleto,telefono)
Constructor de la clase. Inicializa los atributos de la clase
Empleado(String línea)
Constructor de la clase, que recibe la información en un string con formato csv codEmpleado;nombreEmpleado;teléfono
String toString();
método sobrecargado que retorna la información del empleado
String serialize()
Método que retorna un string con la información del empleado en formato csv codEmpleado;nombreEmpleado;teléfono

Gestión de Inmueble

En este módulo gestionamos toda la información relativa a los empleados que se encuentra almacenada en inmuebles.csv

Inmueble
Empleado(codInmueble,direccion,tipo,codEmpleado,precio)
Constructor de la clase. Inicializa los atributos de la clase
Empleado(String línea)
Constructor de la clase, que recibe la información en un string con formato csv codInmueble,dirección,tipo,codEmpleado,precio)
String toString();
método sobrecargado que retorna la información del empleado
String serialize()
Método que retorna un string con la información del empleado en formato csv codInmueble,dirección,tipo,codEmpleado,precio)

Gestor Inmobiliaria

Este módulo se encarga de la gestión global de los datos, tanto de los inmuebles como de los empleados

Inmobiliaria
Inmobiliaria(String nombreFicheroempleados, String nombreFicheroInmuebles)
Constructor de la clase que recibe los nombre de los dos ficheros que contienen la información de empleados e inmuebles. La secuencia del constructor: <ul style="list-style-type: none"> • inicializar contenedores • Cargar Empleados mediante la llamada al gestor de persistencia pasándole el nombre del archivo empleados.csv y la lista donde se guardarán los empleados • Cargar Inmueble mediante la llamada al gestor de persistencia pasándole el nombre del archivo inmuebles.csv y la lista donde se guardarán los datos
Importante: Cada inmueble tiene asociado un empleado, es necesario por parte del gestor de persistencia

verificar que el cod empelado asociado al inmueble existe en la lista
Si se detecta un cod empelado que no existe es un ERROR GRAVE
String ListarDatos()
Retorna una cadena con el listado de todos los datos; es decir inmuebles y empleados
String listarInmueble(String codInmueble)
Retornar una cadena con el listado de los datos asociados a un inmueble. En este listado se deberá retornar también la información asociada al empleado que se encarga del inmueble
int guardarDatos(String nombreFicheroempleados, String nombreFicheroInmuebles)
Realizar las llamadas al gestor de persistencia para almacenar los datos de cada lista en su fichero correspondiente pasado como parámetro.
Retorna el número total de registros guardados en fichero
void nuevoInmueble(String lineaDatos);
Se recibe desde el menú una cadena con los datos asociados a un nuevo inmueble. Este inmueble debe ser incluido siempre que se verifique el no existe un código igual asociado a otro inmueble y además el empelado existe dentro de la base de datos
void nuevoEmpleado(String lineaDatos)
Recibe una cadena de caracteres con los datos asociados a un nuevo empleado, verificar que no existe otro empleado con el mismo código
Booelan borrarInmueble(String codInmueble)
Eliminar los datos almacenados en la base de datos correspondiente a un inmueble cuyo código se recibe como parámetro.
Retorna true si se borra con éxito el empleado.

Gestor de persistencia:

Modulo encargado del procesamiento de los ficheros.

Persistencia Interfaz

void LeerInmueble(String nombreFichero, List<Inmueble> listaInmuebles, List<Empleado> listaEmpleados)

Realiza una lectura secuencial por el fichero que recibe como parámetro donde se almacenan los datos del inmueble. Retorna en la lista que se le pasa como parámetro cada uno de los inmuebles

Importante:

Se deberá tener en cuenta que para insertar un nuevo inmueble es necesario que el empleado exista previamente como dato en la lista empleado

Importante:

Si una línea es incorrecta se deberá mostrar el mensaje, pero NUNCA romper la lectura secuencial del fichero, es decir, la línea errónea se ignora y se continua con la carga de datos

`void LeerEmpleado(String nombreEmpleado, List<Empleado> listaEmpleado)`

Realiza la lectura secuencial del fichero empleados, almacenando los datos en la lista que se le pasa como parámetro

`void EscribirInmueble(String nombreFichero, List<Inmueble> listaInmuebles)`

Guarda los datos almacenados en la lista inmueble en el fichero que se le pasa como parámetro

`void EscribirEmpleado(String nombreEmpleado, List<Empleado> listaEmpleado)`

Guarda los datos almacenados en la lista empleados en el fichero que se le pasa como parámetro