Agustín Nicolas Fernández 2.C

TP 3 Documento.

El programa realiza la carga de 2 tipos de dispositivos (Celulares y notebooks) en una fábrica, también puede quitar los dispositivos y ver su stock. Además, tiene 2 funcionabilidades de guardado de archivos en formato .txt y .XML.

Clase 15 - Excepciones:

Las Excepciones las use en la clase "Excepciones" y cree 2 tipos, una para archivos (ArchivosException) y otra para Dispositivos (DispositivoRepetidoException).

**ArchivosException**: Es Lanzada en caso de que haya algún problema en la lectura/escritura de algún archivo sea el formato que sea.

**DispositivoRepetidoException**: Esta excepción es lanzada en la sobrecarga de operadores "+" cuando trata de agregar un dispositivo a la lista. Si el dispositivo esta repetido se lanza la excepción.

Clase 16 – Test Unitario:

Realice 2 test unitarios en el proyecto TestUnitarios.

**TestProductoRepetidoException**(): En este test verifico que se lance la excepción "**DispositivoRepetidoException**" al agregar un dispositivo repetido.

**TestVerificarListaNula**(): En este test verifico que la lista de dispositivos de la fábrica sea diferente a nula.

Clase 17 – Tipos Genéricos:

Se utiliza en la interfaz **IArchivo** para permitir serializar objetos de cualquier tipo, y así crear métodos genéricos de lectura/escritura.

Clase 18 – Interfaces:

Las interfaces las use para crear 2 métodos de lectura/escritura de archivos, implementada por las clases **XML** y **ArchivoTxt**.

Clase 19 – Archivos y Serialización:

En el proyecto Archivos, están las 2 clases encargadas de leer/escribir archivos de texto y en XML.

En la clase fabrica, cree un método estático "Guardar" que recibe una fábrica por parámetro y devuelve true si se pudo guardar el archivo en XML.

En cambio, para crear un archivo del tipo texto tengo que instanciar la clase porque no le genere otro método a la fábrica para los archivos de texto.

Clase 20/21 - Bases de datos:

En la clase DispositivoDAO se utiliza todo el código relación a bases de datos para hacer las consultas e ir actualizando la lista de dispositivos.

En la propiedad get de la lista de dispositivos, se usa DispositivoDAO.LeerTodo() que llena la lista con la base de datos.

Al agregar un nuevo dispositivo a la lista también se agrega a la base de datos.

Clase 23 - Hilos.

En la biblioteca de clases "Test" se crean 2 hilos para simular la carga de 3 celulares y 3 notebooks simultáneamente.

Clase 24 - Delegados y Eventos:

En la clase fabrica cree un delegado que retorna void y no recibe ningun parametro, que esta enlazado con el evento "miEventoHarcodeo", lo que se me ocurrio hacer es un evento que harcodee productos a la lista.

Clase 25 – Métodos De Extensión.

En la biblioteca de clases "Entidades" se creó una clase llamada ListExtension, lo que hice fue cambiarle el namespace al de " System.Collections.Generic" para agregar el siguiente método de extension.

public static string StockTotal(this List<Dispositivos> listaDispositivos)

Lo que hace es obtener el stock total de la lista de dispositivos y lo devuelve en forma de un string usando stringbuilder.