

Auto-Documentación en Java: Javadoc

- La herramienta Javadoc genera de manera automática documentación a partir de los programas fuente Java. Para ello, estos programas deben tener comentarios y etiquetas entendidas por Javadoc `/** */`

Etiqueta	Quién lo utiliza	Objetivo
@author name	Clases e interfaces	Autor del código. Se pone una etiqueta para cada autor.
@deprecated	Clases, métodos	Método anticuado. Mejor no utilizarlo.
@exception name description	Métodos	Excepciones que el método puede elevar. Se pone una etiqueta para cada excepción posible.
@param name description	Métodos	Para describir los parámetros, su utilización y su tipo. Se pone una etiqueta para cada parámetro.
@return description	Métodos	Para describir los valores devueltos por cada método y su tipo.
@since	Clases, métodos	Desde qué versión está. Ej: desde .JDK 1.1
@ see ClassName	Clases, interfaces, métodos y atributos.	Pondrá la dirección para conectarse con esta clase en la documentación
@see ClassName#NombreMétodo	Clases, interfaces, métodos y atributos.	Pondrá la dirección para conectarse con este método en la documentación.
@version text	Clases, interfaces.	Información acerca de la versión.

```

import java.io.*;
import java.net.*;
import java.util.*;
/**
 * Los objetos de esta clase leen de una entrada y escriben en
 * una salida. Cuando se crea cada objeto, la entrada se asocia a la
 * entrada estándar y la salida a la salida estándar. Se
 * puede conectar la entrada a ficheros o URLs y la
 * salida a ficheros.
 * @author Prof. ISO
 */
public class LectorEscritorTexto {

    InputStreamReader input;
    OutputStreamWriter output;
    Vector vector;

    public final static int MAXIMO = 10000;

/**
 * Constructor de la clase. Asocia la entrada a la
 * entrada estándar y la salida a la salida estándar
 */
    public LectorEscritorTexto() {
        input = new InputStreamReader(System.in);
        output = new OutputStreamWriter(System.out);
        vector = new Vector();
    }
/**
 * Método que conecta la entrada a una URL
 * @param url String que contiene la URL
 */
    public void addLectorURL(String url)
        throws java.net.MalformedURLException,
            java.io.IOException{
        URL laUrl = new URL(url);
        InputStream ent = laUrl.openStream();
        input = new InputStreamReader(ent);
}

```

```

    * Método que conecta la entrada a un fichero
    * @param fich String que contiene el nombre del fichero
    */
    public void addLectorFichero(String fich) throws java.io.FileNotFoundException {
        {
            input = new FileReader(fich);
        }
    }
/**
 * Método que conecta la salida a un fichero
 * @param fich String que contiene el nombre del fichero
 */
    public void addEscritorFichero(String fich) throws java.io.IOException {
        output = new FileWriter(fich); }

/**
 * Método que lee de la entrada definida en ese momento y almacena
 * el contenido dentro del propio objeto.
 */
    public void leer() throws java.io.IOException {
        BufferedReader bufInput = new BufferedReader(input);
        String linea;
        vector.removeAllElements();
        while ((linea=bufInput.readLine())!=null)
            vector.addElement(linea);
        bufInput.close(); }

/**
 * Método que escribe en la salida definida en ese momento el
 * contenido que previamente se ha leído al ejecutar el método leer()
 */
    public void escribir() throws java.io.IOException {
        PrintWriter printOutput = new PrintWriter(output);
        for (int i=0;i<vector.size();i++)
            printOutput.println((String)vector.elementAt(i));
        printOutput.close();
    }
    public static void main(String argv[])
        { .... }
}

```

Class LectorEscritorTexto - Netscape

Inicio | Edición | Ver | Ir | Comunicador | Ayuda

Anterior | Siguiente | Recarga | Volver a cargar esta página desde el servidor | Seguridad | Parar

Marcadores | Dirección: file:///D:/alfredo/ADSI/jdeveloper/fuentes/ejISO/ejISO/LectorEscritorTexto.html

Internet | Novedades | Interesante | Miembros | Conexiones | Compras

class Tree Deprecated Index Help

BY CLASS NEXT CLASS

PRIMARY INDEX FIELD CONSTR METHOD

PRIMARY INDEX FIELD CONSTR METHOD

DETAIL FIELD CONSTR METHOD

ISO

Class LectorEscritorTexto

java.lang.Object

---ejISO.LectorEscritorTexto

public class **LectorEscritorTexto**
 extends java.lang.Object

Los objetos de esta clase leen de una entrada y escriben en una salida. Cuando se crea cada objeto, la entrada se asocia a la entrada estándar y la salida a la salida estándar. Se puede conectar la entrada a ficheros o URLs y la salida a ficheros.

Autor:
 Prof. ISO

Field Summary

private int MAXIMO

Constructor Summary

LectorEscritorTexto ()
 Constructor de la clase.

Method Summary

void addEscritorFichero(java.lang.String fich)
 Método que conecta la salida a un fichero

void addLectorFichero(java.lang.String fich)
 Método que conecta la entrada a un fichero

void addLectorURL(java.lang.String url)
 Método que conecta la entrada a una URL



Warning: Cannot access property of undefined variable: java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

MAXIMO

public static final int MAXIMO

Constructor Detail

LectorEscritorTexto

public LectorEscritorTexto()

Constructor de la clase. Asocia la entrada a la entrada estándar y la salida a la salida estándar

Method Detail

addLectorURL

public void addLectorURL (java.lang.String url)
throws java.net.MalformedURLException,
java.io.IOException

Método que conecta la entrada a una URL

Parameters:

url - String que contiene la URL

addLectorFichero

public void addLectorFichero (java.lang.String fich)
throws java.io.FileNotFoundException

Método que conecta la entrada a un fichero

Parameters:

fich - String que contiene el nombre del fichero