

CURSO DE PROGRAMACIÓN CON JAVA

MANEJO DE ARCHIVOS EN JAVA



Por el experto: Ing. Ubaldo Acosta



CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

Hola, te saluda Ubaldo Acosta. Bienvenida o bienvenido nuevamente. Espero que estés listo para comenzar con esta lección.

Vamos a estudiar el tema de manejo de archivos en Java.

¿Estás listo? Ok, ¡Vamos!

CREACION DE ARCHIVOS EN JAVA

Creación de archivos en Java:

```
import java.io.*;

public class ManejoArchivos {

    public static void main(String[] args){
        File archivo = new File("c:\\pruebaJava\\pruebaJava.txt");
        try {
            PrintWriter salida = new PrintWriter(new FileWriter(archivo));
            salida.close();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
        System.out.println("El archivo se ha creado correctamente");
    }
}
```

www.globalmentoring.com.mx

En esta lección vamos a estudiar el manejo de archivos en Java. Se utilizan las clases File, FileWriter y PrintWriter para poder crear un archivo. Es conveniente verificar que tengamos permisos para escribir el archivo, por ejemplo, podemos crear una carpeta y asignarle permisos de escritura, de esta manera no tendremos problemas para crear, leer o modificar el archivo creado.

La clase FileWriter se utiliza para escribir información a un archivo, y como podemos observar en el código, primeramente creamos un archivo con la clase File, sin embargo esto no crea físicamente el archivo, para ello necesitamos la clase FileWriter y finalmente la clase PrintWriter que es la que creará el archivo en el disco duro. Vemos que el archivo de prueba es creado dentro de una carpeta la cual ya debe existir para que no arroje errores de creación de archivo.

Una vez creado el archivo, podremos leer y escribir sobre él, a continuación veremos cómo realizar estos procesos.

ESCRITURA A UN ARCHIVO EN JAVA

Escritura a un archivo en Java:

```
import java.io.*;

public class ManejoArchivos {

    public static void main(String[] args){
        File archivo = new File("c:\\pruebaJava\\pruebaJava.txt");
        try {
            PrintWriter salida = new PrintWriter(new FileWriter(archivo));
            salida.close();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
        System.out.println("El archivo se ha creado correctamente");
    }
}
```

www.globalmentoring.com.mx

Vamos a estudiar ahora cómo escribir información a un archivo. La clase `FileWriter` se usa para escribir caracteres a un archivo.

La clase `PrintWriter` se usa para imprimir líneas completas, con los métodos `print()` y `println()`. La desventaja de este método es que se pierde la información que tengamos almacenada de manera previa en el archivo, por lo tanto no nos dice para agregar nueva información, sino escribir desde cero al archivo.

Más adelante veremos un ejemplo de cómo realizar este proceso.

LECTURA DE UN ARCHIVO EN JAVA

Lectura de un archivo en Java:

```
import java.io.*;

public class LecturaArchivos {

    public static void main(String args[]) {
        File archivo = new File("c:\\pruebaJava\\pruebaJava.txt");
        try {
            BufferedReader entrada = new BufferedReader(new FileReader(archivo));
            String lectura;
            lectura = entrada.readLine();
            while (lectura != null) {
                System.out.println("Lectura: " + lectura);
                lectura = entrada.readLine();
            }
            entrada.close();
        } catch (FileNotFoundException ex) {
            ex.printStackTrace();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

Veamos ahora cómo leer información de un archivo. Se utilizan las clases File, FileReader y BufferedReader para poder leer un archivo

El código es muy parecido a la lectura de información de la consola estándar.

La clase FileReader se utiliza para leer caracteres.

La clase BufferedReader se utiliza para leer líneas completas.

Así que dependiendo de lo que necesitemos leer será el tipo de clase que podemos utilizar. Más adelante veremos un ejemplo de la lectura de información de un archivo.

ANEXANDO INFORMACION A UN ARCHIVO

Anexando información a un archivo:

```
package manejoarchivos;

import java.io.*;

public class AnexarInformacionArchivo {

    public static void main(String args[]) {
        File archivo = new File("c:\\pruebaJava\\pruebaJava.txt");
        try {
            PrintWriter salida = new PrintWriter(new FileWriter(archivo, true));
            String contenido = "Anexando informacion al archivo";
            salida.println(contenido);
            salida.println();
            salida.println("Fin de anexar");
            salida.close();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
        System.out.println("Se ha anexado la informacion correctamente");
    }
}
```

www.globalmentoring.com.mx

Veamos ahora cómo anexar información a un archivo ya creado.

La diferencia con el ejercicio anterior es la llamada al constructor de la clase `FileWriter`, ahora le proporcionamos una bandera **true**, que significa que la información se va a agregar a la ya existente, y con ello no se pierde lo que ya se encuentra en el archivo, sino que anexa la información proporcionada.

Más adelante veremos un ejercicio para anexar información a un archivo.

EJERCICIOS CURSO FUNDAMENTOS DE JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio de ManejoArchivos en Java.

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

CURSO ONLINE

PROGRAMACIÓN CON JAVA

Por: Ing. Ubaldo Acosta



CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

Además agregamos nuevos cursos para que continúes con tu preparación como programador Java profesional. A continuación te presentamos nuestro listado de cursos:

- ✔ Lógica de Programación
- ✔ Fundamentos de Java
- ✔ Programación con Java
- ✔ Java con JDBC
- ✔ HTML, CSS y JavaScript
- ✔ Servlets y JSP's
- ✔ Struts Framework
- ✔ Hibernate Framework
- ✔ Spring Framework
- ✔ JavaServer Faces
- ✔ Java EE (EJB, JPA y Web Services)
- ✔ JBoss Administration
- ✔ Android con Java
- ✔ HTML5 y CSS3

Datos de Contacto:

Sitio Web: www.globalmentoring.com.mx

Email: informes@globalmentoring.com.mx

