

Geeks México

BLOG DE PROGRAMACIÓN EN ESPAÑOL SOBRE JAVA, FRAMEWORKS, BASES DE DATOS, CÓMPUTO EN LA NUBE, ETC. EN ESPAÑOL Y EN INGLÉS.

TUTORIALES EN ESPAÑOL

TUTORIALS IN ENGLISH

ABOUT

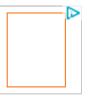
CONTACT

Anuncios

Completo Cubierta de la Pantalla de Vidrio Templado Para Xiaomi Redmi Nota 4 Pro Prot... AliExpress

€1,52

vidrio templado en de crista...



Manejo de archivos en Java

■ HACE 2 SEMANAS

₱ DEJA UN COMENTARIO

A continuación se presenta una guía rápida sobre como manejar archivos en Java de forma simple y con algunos ejemplos prácticos. Las clases necesarias para manejar archivos en Java se encuentran en el paquete java.io.

Creación de un nuevo archivo

Para crear un archivo nuevo en Java utilizaremos la clase File, veamos los puntos importantes a recordar y el ejemplo de código:

- El hecho de crear un objeto de esta clase no creará el archivo en el disco
- Se puede utilizar la clase file para crear un archivo o un directorio
- El método exists() devuelve un valor boolean que es verdadero si el archivo ya existe en el equipo y falso si no.
- El método createNewFile() creará el archivo en el disco duro.
- El método createNewFile() puede provocar una IOException así que debe ser manejada

```
File f = new File("file.txt");

if (!f.exists()) {
    f.createNewFile();
}
```

Listar archivos en un directorio

Veamos los puntos importantes a recordar y el ejemplo de código:

- El constructor de la clase File recibe como parámetro un directorio, en este caso el directorio que utilizaremos es el raíz.
- El método listFiles() devuelve un arreglo de objetos de la clase File cada uno representa un archivo contenido en el directorio.

```
File f = new File("/");
File[] listFiles = f.listFiles();
for (File file : listFiles) {
    System.out.println(file);
}
```

Principales métodos

Los métodos principales de la clase File son los siguientes:

- createNewFile(): Crea un nuevo archivo en disco
- delete(): Borra un archivo
- exists(): Devuelve verdadero o falso dependiendo si el archivo existe o no
- isDirectory(): Devuelve verdadero o falso dependiendo si es un directorio o no
- isFile(): Devuelve verdadero o falso dependiendo si es un archivo o no
- list(): Lista todos los archivos contenidos en el directorio
- mkdir(): Crear un nuevo directorio
- renameTo(): Re nombra un archivo

Ejemplo práctico

Para ejemplificar como funcionan los archivos programaremos el comando tree, el cuál lista todos los archivos y directorios del sistema operativo.

```
import java.io.File;
import java.io.IOException;

/**
@author raidentrance
```

```
6
7
8
     public class TreeCommand {
         public static void printFile(File f, 5
9
              System.out.println(spaces.concat()
10
11
              if (f.isDirectory() & amp; & amp; f.
                  for (File file : f.listFiles()
12
13
                      if (file.isDirectory()) {
14
                           printFile(file, spaces
15
16
              }
17
18
19
         }
20
21
         public static void main(String[] args)
22
              printFile(new File("/Users/raident
23
         }
     }
24
```

Si deseas ver una explicación detallada del ejemplo que se presentó te recomendamos leer el post Aprende a utilizar la recursividad programando el comando tree (https://geeks-mexico.com/2017/10/08/aprende-a-utilizar-la-recursividad-programando-el-comando-tree/).

Autor: Alejandro Agapito Bautista

Twitter: @raidentrance

Contacto:raidentrance@gmail.com

Anuncios

