Glossary: File Management

Estimated time: 3 minutes

Welcome! This alphabetized glossary contains many terms used in this module. Understanding these terms is essential when working in the industry, participating in user groups, and participating in other certificate programs.

Term	Definition
BufferedInputStream	Class used to buffer input.
BufferedOutputStream	Class used to buffer output.
BufferedReader	Class in Java that reads text from a character-input stream, buffering characters to provide efficient reading of characters, arrays, and lines.
BufferedWriter	Class that improves the efficiency of writing data.
Byte stream	Used to read and write data as raw bytes. It handles all types of data, such as text, images, and videos, making it versatile for various tasks.
Character stream	Deals with character data or text.
Class definition	Indicates the purpose of the program.
close()	Method that closes writer, saves data, and releases resources.
create directory	Operation that allows users to create a new directory under the "Documents" folder. This functionality helps organize files by categorizing them into separate folders, making it easier to manage and access them later.
delete directory	Operation that allows users to remove a directory if it exists. However, the directory must be empty before it can be deleted.
delete()	Returns true if the deletion is successful. The developer is notified if the directory contains files or does not exist.
Directories	Folders that store files and other folders inside them. They help organize data in a clear and easy-to-use structure.
File class	Represents a file or directory path, offering methods to create and delete files or directories and functionality to check their properties.
File handling	Essential part of programming that enables reading from and writing to files on a computer.
FileInputStream	Reads bytes from a file.
FileOutputStream	Writes bytes to a file.
FileReader	Class in Java that is used to read data from files in the form of characters.
Files.createDirectories()	Method that creates the specified directory and any necessary parent directories if they don't already exist.
FileWriter	Class that handles file output, creates a file if it doesn't exist, and overwrites existing files.
Import statement	Imports the File class from the java.io package, enabling file manipulation.
InputStream	Abstract class representing an input stream of bytes. It serves as the superclass for all classes that read byte data.
Java new input or output (NIO)	Offers an improved and more efficient way to handle file and directory operations compared to the traditional Java IO.
list directory	Operation that enables users to view the contents of a specified directory. It retrieves the names of all the files and subdirectories within that directory, helping users quickly check what is stored inside and navigate their files more efficiently.
list()	Method that can retrieve an array of strings representing the names of files and directories.
Loop	Function that reads each line until there are no more lines, printing each line to the console.
main() method	Method that indicates the start of program execution.
mkdirs()	Method that creates the directory along with any missing parent directories. The program checks if the directory already exists before attempting to create it.
newline()	Method that adds a new line.
OutputStream	Abstract class representing an output stream of bytes. It serves as the superclass for all classes that write byte data.
Paths.get()	Creates a Path object representing the directory path. This Path object can then be used in various operations like reading, writing, or creating directories.
try block	Block of code that is used for exception handling, ensuring that errors during reading are caught.
write() method	Method that writes data to a file.

Skills Network