package Question3;

import java.io.\*;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.nio.file.attribute.BasicFileAttributes;

import java.text.SimpleDateFormat;

import java.util.Date;

public class FileManagement {

public static void main(String[] args) {

System.out.println("CreateDirectory\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

String directoryPath = "\\Users\\pooja\\Downloads\\new\_directory2";

// Create a Path object representing the directory

Path directory = Paths.get(directoryPath);

try {

// Create the directory

Files.createDirectory(directory);

System.out.println("Directory created successfully.");

} catch (IOException e) {

System.out.println("Failed to create directory: " + e.getMessage());

}

System.out.println("CreateTextFile\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//Create a new text file and write content to it.

String fileName = "new\_file.txt";

String content = "This is some text content to write to the file.";

try {

// Create a FileWriter object with the specified file name

FileWriter writer = new FileWriter(fileName);

// Write the content to the file

writer.write(content);

// Close the writer to release resources

writer.close();

System.out.println("Content written to the file successfully.");

} catch (IOException e) {

System.out.println("An error occurred: " + e.getMessage());

}

System.out.println("ReadTheContent\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//Read the content from an existing text file.

try {

// Create a FileReader object with the specified file name

FileReader fileReader = new FileReader(fileName);

// Create a BufferedReader object wrapping around the FileReader

BufferedReader reader = new BufferedReader(fileReader);

// Read each line from the file and print it

String line;

System.out.println("Content of the file:");

while ((line = reader.readLine()) != null) {

System.out.println(line);

}

// Close the reader to release resources

reader.close();

} catch (IOException e) {

System.out.println("An error occurred: " + e.getMessage());

}

System.out.println("AppendContent\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//Append new content to an existing text file.

String content1 = " \nNew Content.";

try {

// Create a FileWriter object with the specified file name and append mode

FileWriter fileWriter = new FileWriter(fileName, true);

// Write the content to the file

fileWriter.write(content1);

// Close the writer to release resources

fileWriter.close();

System.out.println("Content appended to the file successfully.");

} catch (IOException e) {

System.out.println("An error occurred: " + e.getMessage());

}

System.out.println("CopyTheContent\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

// try {

// // Create a FileReader object with the specified file name

// FileReader fileReader = new FileReader(fileName);

//

// // Create a BufferedReader object wrapping around the FileReader

// BufferedReader reader = new BufferedReader(fileReader);

//

// // Read each line from the file and print it

// String line;

// System.out.println("Content of the file:");

// while ((line = reader.readLine()) != null) {

// System.out.println(line);

// }

// // Close the reader to release resources

// reader.close();

// } catch (IOException e) {

// System.out.println("An error occurred: " + e.getMessage());

// }

//Copy the content from one text file to another.

String sourceFileName = "new\_file.txt";

String destinationFileName = "destination\_file.txt";

try {

// Create a FileReader object for the source file

FileReader fileReader = new FileReader(sourceFileName);

BufferedReader reader = new BufferedReader(fileReader);

// Create a FileWriter object for the destination file

FileWriter fileWriter = new FileWriter(destinationFileName);

BufferedWriter writer = new BufferedWriter(fileWriter);

// Read each line from the source file and write it to the destination file

String line;

while ((line = reader.readLine()) != null) {

writer.write(line);

writer.newLine(); // Add newline character after each line

}

// Close the readers and writers to release resources

reader.close();

writer.close();

System.out.println("Content copied from '" + sourceFileName + "' to '" + destinationFileName + "' successfully.");

} catch (IOException e) {

System.out.println("An error occurred: " + e.getMessage());

}

System.out.println("DeleteTextFile\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

// try {

// // Create a FileReader object with the specified file name

// FileReader fileReader = new FileReader(destinationFileName);

//

// // Create a BufferedReader object wrapping around the FileReader

// BufferedReader reader = new BufferedReader(fileReader);

//

// // Read each line from the file and print it

// String line;

// System.out.println("Content of the file:");

// while ((line = reader.readLine()) != null) {

// System.out.println(line);

// }

// // Close the reader to release resources

// reader.close();

// } catch (IOException e) {

// System.out.println("An error occurred: " + e.getMessage());

// }

//Delete a text file.

String fileName2 = "\\Users\\pooja\\Downloads\\file\_to\_delete.txt";

// Create a File object representing the file to delete

File fileToDelete = new File(fileName2);

// Check if the file exists

if (fileToDelete.exists()) {

// Attempt to delete the file

boolean deleted = fileToDelete.delete();

// Check if the file was successfully deleted

if (deleted) {

System.out.println("File '" + fileName2 + "' deleted successfully.");

} else {

System.out.println("Failed to delete file '" + fileName2 + "'.");

}

} else {

System.out.println("File '" + fileName2 + "' does not exist.");

}

System.out.println("ListFileDirectories\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//List all files and directories in a given directory.

// Specify the directory path

String directoryPath1 = "\\Users\\pooja\\Downloads\\Java Assignments";

// Create a File object representing the directory

File directory1 = new File(directoryPath1);

// Check if the specified path exists and is a directory

if (directory1.exists() && directory1.isDirectory()) {

// Get a list of all files and directories in the directory

File[] files = directory1.listFiles();

// Print the names of all files and directories

System.out.println("Files and directories in '" + directoryPath1 + "':");

for (File file : files) {

System.out.println(file.getName());

}

} else {

System.out.println("The specified directory '" + directoryPath1 + "' does not exist or is not a directory.");

}

System.out.println("SearchFile\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//Search for a specific file in a directory and its subdirectories.

String fileNameToSearch = "destination\_file.txt";

// Create a Path object representing the directory

Path directory2 = Paths.get(directoryPath1);

try {

// Search for the file recursively in the directory and its subdirectories

Files.walk(directory2)

.filter(Files::isRegularFile)

.filter(path -> path.getFileName().toString().equals(fileNameToSearch))

.forEach(path -> System.out.println("Found file: " + path));

} catch (IOException e) {

System.out.println("An error occurred: " + e.getMessage());

}

System.out.println("RenameFile\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*8");

//Rename a file.

String oldFilePath = "\\Users\\pooja\\Downloads\\Java Assignments\\destination\_file.txt";

String newFilePath = "\\Users\\pooja\\Downloads\\Java Assignments\\destination\_file\_renamed.txt";

// Create Path objects for old and new file paths

Path oldPath = Paths.get(oldFilePath);

Path newPath = Paths.get(newFilePath);

try {

// Rename the file

Files.move(oldPath, newPath);

System.out.println("File renamed successfully.");

} catch (IOException e) {

System.out.println("Failed to rename file: " + e.getMessage());

}

System.out.println("GetInfoAboutFile\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//Get information about a file (e.g., file size, last modified time).

String filePath = "\\Users\\pooja\\Downloads\\Java Assignments\\destination\_file\_renamed.txt";

// Create a Path object representing the file

Path path = Paths.get(filePath);

try {

// Get file attributes

BasicFileAttributes attributes = Files.readAttributes(path, BasicFileAttributes.class);

// Print file size

long fileSize = attributes.size();

System.out.println("File size: " + fileSize + " bytes");

// Print last modified time

long lastModifiedTimeInMillis = attributes.lastModifiedTime().toMillis();

Date lastModifiedDate = new Date(lastModifiedTimeInMillis);

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

System.out.println("Last modified time: " + dateFormat.format(lastModifiedDate));

} catch (IOException e) {

System.out.println("An error occurred: " + e.getMessage());

}

}

}

Output :

