Java-Getting File Information

Project Abstract

The purpose of this project is to demonstrate how to retrieve and display various information about a file in Java. This task helps you to understand how to use the `File` class to get file details like its name, path, size, readability, writability, and directory status. The project focuses on exploring different methods provided by the `File` class to retrieve important file information.

This project focuses on:
1. Understanding how to retrieve information about a file using the `File` class in Java.
2. Using various `File` class methods to get file name, path, size, readability, writability, and directory status.
3. Handling the case when the file does not exist by checking if the file exists before retrieving information.
4. Demonstrating how to print file information to the console.

Tasks Overview

Task 1: Retrieve and Display File Information

Objective: Retrieve and display various information about a file using the `File` class in Java.

Detailed Description: In this task, you will use the `File` class to retrieve information about a file named "fileInfo.txt". The program will check if the file exists and then retrieve and print its name, path, size, readability, writability, and whether the file is a directory. If the file does not exist, the program will print a message indicating that the file does not exist.

Steps:
1. **Create a File Instance:**
   - Declare a `File` object named `file` using the constructor `new File("fileInfo.txt")`. This represents the file "fileInfo.txt" that you will retrieve information about.

2. **Check if the File Exists:**
   - Use the `exists()` method to check if the file exists. If the file does not exist, print the message "File does not exist."

3. **Retrieve and Display File Information:**
   - If the file exists, retrieve and print the following information:
     - **File Name:** Use `file.getName()` to get and print the name of the file as `"File Name: " + file.getName()`.
     - **File Path:** Use `file.getAbsolutePath()` to get and print the full path of the file as `"File Path: " + file.getAbsolutePath()`.
     - **File Size:** Use `file.length()` to get and print the size of the file in bytes as `"File Size: " + file.length() + " bytes"`.
     - **File Readability:** Use `file.canRead()` to check and print if the file is readable as `"Is Readable: " + file.canRead()`.
     - **File Writability:** Use `file.canWrite()` to check and print if the file is writable as `"Is Writable: " + file.canWrite()`.
     - **Directory Status:** Use `file.isDirectory()` to check and print if the file is a directory as `"Is Directory: " + file.isDirectory()`.

4. **Handle File Non-Existence:**
   - If the file does not exist, print the message "File does not exist." and exit the program.

Execution Steps to Follow:

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) → Terminal →New Terminal.**
3. **This editor Auto Saves the code.**
4. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
5. **To run your project use command:**

   sudo JAVA_HOME=$JAVA_HOME /usr/share/maven/bin/mvn compile exec:java -Dexec.mainClass="com.yaksha.assignment.FileInfo"

   ***If it asks for the password, provide password : pass@word1**

6. **To test your project test cases, use the command**

   sudo JAVA_HOME=$JAVA_HOME /usr/share/maven/bin/mvn test

   ***If it asks for the password, provide password : pass@word1**