(Optional) Coding Challenge: File and Directory Management in Java

Contacts applicaion

Estimated time needed: 20 minutes

This optional self-guided lab provides exercises that align with skills you learned in this module. Follow the instructions to write code using your knowledge to create the described apps. After you complete these lab exercises, you can validate that your code is correct with the solution provided.

Prerequisites (optional)

You must have completed all the labs in Modules 1,2,3 and 4.

Clone the challenges repo

1. Run the following command to clone the repository with the challenges and solution.

   ```
   git clone https://github.com/ibm-developer-skills-network/flgsb-oop-additional-practice.git
   ```

2. Change to `OOPs/Module 4` directory.

   ```
   cd flgsb-oop-additional-practice/OOPs/Module\ 4
   ```

3. View the directories by running the `ls` command. Each directory has one challenge in it.

   ```
   ls
   ```

Read a text file

In this challenge, you'll create a Java program that reads contact information from a text file and displays the data in a formatted way. This exercise will reinforce your understanding of file I/O operations in Java, including:

- Opening and reading text files
- Using Java I/O packages effectively
- Handling exceptions that occur during file operations
- Processing and displaying file contents

You'll build a simple contact book application that reads contact information from a file, parses the information, and presents the information in a user-friendly format.

Challenge Description

Create a Java program that:

1. Prompts the user to enter the name of a contact file
2. Reads the contact file, which contains names and phone numbers
3. Displays the contacts in a formatted way
4. Handles any exceptions that might occur during file operations

The contact file format will be simple:

- Each line represents one contact
- Names and phone numbers are separated by a colon (:)
- Example: John Doe:+1-555-123-4567

Solve the challenge

1. In the terminal, change to `Read a text file`.

   ```
   cd Read\ a\ text\ file
   ```

2. Select the Coding Challenge: File and D to open the Java coding challenge.

Open **ContactReader.java** in IDE

3. Make changes in the file based on the specifications.

4. Compile the Java file.

5. Set the CLASSPATH.

6. Run the class `ContactReader` to ensure the class is created as per instructions.

Click here for the solution

Use Directory Explorer

In this challenge, you'll create a File and Directory Management System using Java. This hands-on exercise will help you apply key concepts from Java's file handling capabilities. You'll build a console application that allows users to navigate and manipulate the file system, reinforcing concepts of object-oriented programming, recursive methods, and file I/O operations.

By completing this challenge, you'll gain practical experience with:

- Working with Java's java.io.File class
- Navigating directory structures
- Creating, renaming, and deleting files and directories
- Using recursion to process nested directories
- Building an interactive command-line interface

This challenge extends the lab content by asking you to create a more feature-rich application with a different approach to the user interface.

Challenge Description

Create a Java application called FileSystemManager that provides a file explorer with the following capabilities:
1. Display the current working directory
2. List files and directories in the current directory
3. Navigate into subdirectories (cd command)
4. Navigate up to the parent directory (cd .. command)
5. Create new files and directories
6. Delete files and directories
7. Rename files and directories
8. Search for files by name pattern
9. Show detailed file information (size, last modified date, etc.)

Instead of using numbered menu options similar to those in the lab, your application should implement a command-line interface similar to shell commands. For example, the user could type `ls` to list files, `cd documents` to change directory, or `mkdir new_folder` to create a directory.

Solve the challenge

1. In the terminal, change to `Create your own class`.

```
cd Use\ Directory\ Explorer
```

2. Select the following button to open `FileSystemManager.java`

Open **FileSystemManager.java** in IDE

3. Make changes in the file based on the specifications.

4. Compile the Java file.

5. Set the CLASSPATH.

6. Run the class `FileSystemManager` to verify that you created the class based on the instructions.

Click here for the solution

Conclusion

After completing these labs, you should be comfortably able to implement File and Directory Management using Java.

Author(s)

Lavanya