FILE HANDLING APP

PROJECT: FilesApp

DEVELOPED BY: Lokesh

GITHUB REPOSITORY LINK: https://github.com/lokesh05l/phase1_project

PREREQUISITES:

- Installation of Java (Java 8 is preferable)
- IDE for running Java (preferably Eclipse, Spring Tool Suite 4)
- Git for running a Maven Project

DESCRIPTION:

FilesApp is an application for file handling where we can do the following operations in the specified directory:

- Add (create) a new file
- Delete a file
- Search for files
- Display the contents of the directory

SPRINTS PLANNED:

Sprint 1: Methods to add and search a file in the specified directory.
Sprint 2: Methods to delete a specified file in the directory and display the contents of the directory
Sprint 3: Additional methods to ensure the app works without any errors.

ALGORITHM:

Step 1: Check whether the specified directory is valid. If invalid exit, else proceed further.

Step 2: a) Display Files in the Directory

```
if (directory is not empty)
     display the contents of the folder
else
     throw exception with appropriate message
     (“directory is empty”)
```

b) Search Files in the Directory

```
if (filename is valid)
```

```
        if (file exists)
             print "file exists"
        else
             throw exception with appropriate message
             ("file not found")
   else
        throw  exception  with  appropriate  message
        ("invalid filename")
```

c) Add a file in the Directory

```
   if (filename is valid)
        if (file does not exist)
             create new file
             print "file create successfully"
        else
             throw exception with appropriate message
             ("file already exists")
   else
        throw  exception  with  appropriate  message
   ("invalid filename")
```

d) Delete a file in the Directory

```
   if (directory not empty)
        if (filename is valid)
             if (file exists)
                  delete file
                  print "file deleted successfully"
        else
             throw exception with appropriate message
             ("invalid filename")
   else
        throw  exception  with  appropriate  message
        ("directory is empty")
```

e) Exit from the app

Print a thank you message upon exit.

EXECUTION STEPS:

After downloading the .zip from the GitHub repository, follow the following steps for execution:

Step 1: Extract the folder to specific location.

Step 2: Open your IDE, then click on File -> Open Projects from File System -> Import Source (in case of Eclipse or STS4).

- Choose the location of your folder.
- Click Finish

Step 3: Navigate to the package that contains the main method.
filesapp -> src/main/java -> com.app.main-> FilesMain.java
Other packages contain the classes that required for running the main method.

Step 4: Click on Run button on the toolbar or press Ctrl+F11
Now you can execute the program by providing valid inputs.

Step 5: To run as Maven Project

- Check filesapp -> JRE System Library is using jdk. If it is with jdk, continue the next step or else follow the following steps:
  - If not, right click on JRE System Library, choose Build Path -> Configure Build Path -> Libraries -> Add Library -> JRE System Library -> Alternate JRE -> Select jdk from drop down -> Installed JREs.
  - Select jdk from the list and click on Apply and Close. If jdk is not available, click on Add -> Standard VM. Choose the location of jdk and click Finish. Now select jdk and click on Apply and Close.
  - Select Workspace default JRE and click Finish.
  - Select JRE System Library with jdk and click on Apply and Close.
- Right click on project and choose Run as -> Maven Build. Enter the Goals and click on Run.
- If the build is successful, a .jar file will be generated which will be available under target (filesapp -> target).
- Right click on the .jar file. Select Properties and click on Show in System Explorer icon (Icon will be displayed at the right end of Location).
- You can see the .jar file there. Right click anywhere in that pane and select Git Bash Here.
- Now type the following command

  ```
  java -cp <jar file with extension> <main class name including the package>
  ```

  In my case, it will be,

  ```
  java -cp filesapp-0.0.1-SNAPSHOT.jar com.app.main.FilesMain
  ```

- Now you can run the application via Console.

GITHUB REPOSITORY LINK: https://github.com/lokesh05l/phase1_project