```
package locker_project;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.InputMismatchException;
import java.util.List;
import java.util.NoSuchElementException;
import java.util.Scanner;

public class Locker {
	static Scanner sc = new Scanner(System.in);
	static boolean fileOcc = false;
	static String dirInputMsg = "Please Enter the User Directory Name: ";
	static String fileInputMsg = "Please Enter the File Name: ";
	static String retryMenuOptionMsg = "Please try the menu option again...";
	static String rtyValIpMsg = "Please try again with valid input as Y or N";

	public static void main(String[] args) {
		String selectedOption;
		String rootPath;

	System.out.println("**********************************************************
********");
		System.out.println("******************      Locker Project
**********************");
```

```
        System.out.println("************************************************************
********");
                System.out.println();
                System.out.println("Welcome To The Locker Project...");
                try {
                        while(true) {
                                System.out.println();
                                System.out.println("Please enter root path... (e.g.
C:\\Users\\Desktop)");
                                rootPath = sc.nextLine();
                                File file = new File(rootPath);
                                if(file.exists()) {
                                        break;
                                }
                                System.out.println("This is not a valid path. Please try again with
valid path...");
                        }

                        while(true) {
                                Locker.showMenu();
                                selectedOption = sc.nextLine();
                                Locker.inputValidation(selectedOption);

                                if(selectedOption.equalsIgnoreCase("1")) {
                                        Locker.getAllFiles(rootPath);
                                }else if(selectedOption.equalsIgnoreCase("2")){
                                        Locker.directoryHandling(rootPath);
                                }else if(selectedOption.equalsIgnoreCase("3")){
                                        Locker.fileDeleteHandling(rootPath);
                                }else if(selectedOption.equalsIgnoreCase("4")){
                                        Locker.fileSearching(rootPath);
                                }else if(selectedOption.equalsIgnoreCase("5")){
                                        System.out.println();
```

```
                                        System.out.println("Thanks For Visiting Us...");
                                        System.out.println("Have a Good Day... :)");
                                        break;
                                }
                        }
                }catch(FileNotFoundException e) {
                        System.out.println("Resource not found...");
                }catch(InputMismatchException e) {
                        System.out.println("Please provide correct/valid inputs");
                }catch (IOException e) {
                        System.out.println("IOException occured. Please contact administrator...");
                }catch (NoSuchElementException e) {
                        System.out.println("NoSuchElementException occured. Please provide valid
values...");
                }catch(Exception e){
                        System.out.println("Some error occured. Please contact administrator");
                }finally {
                        sc.close();
                }

        }

        /*
         * This Method will display the Menu to User
         */
        public static void showMenu() {
                System.out.println();
                System.out.println("----------Locker Menu----------");
                System.out.println("1. Get All Files");
                System.out.println("2. Add New File");
                System.out.println("3. Delete Existing File");
                System.out.println("4. Search Existing File");
                System.out.println("5. Close Application");
```

```
		System.out.println("Select Any One of the Above Option...");
	}

	/*
	 * This method will do the Validation for Menu Option selected by the user.
	 */
	public static void inputValidation(String selectedOption) {
		if(!selectedOption.equalsIgnoreCase("1") && !selectedOption.equalsIgnoreCase("2") && !selectedOption.equalsIgnoreCase("3") && !selectedOption.equalsIgnoreCase("4") && !selectedOption.equalsIgnoreCase("5")) {
			System.out.println("You have selected " + selectedOption + ". Please select valid option between 1 to 5...");
		}
	}

	/*
	 * This method will fetch all the files present in the root path.
	 * It will be displayed in the form of Array.
	 */
	public static void getAllFiles(String rootPath) throws FileNotFoundException {
		// Creating List and File Objects
		List<String> fileNames = new ArrayList<String>();
		File file = new File(rootPath);

		// Fetching All the files Available in Locker Files Folder
		File[] fileArray = file.listFiles();

		// Getting File Names for all the files fetched
		for(File f: fileArray) {
			if(f.isDirectory()) {
				String path = rootPath + "\\" + f.getName();
				File userFile = new File(path);
				File [] userFileArr = userFile.listFiles();
```

```
                        for(File fi: userFileArr) {
                                fileNames.add(fi.getName());
                        }

                }else {
                        fileNames.add(f.getName());
                }
        }

        // Sorting list of File Names in ascending order
        Collections.sort(fileNames);
        System.out.println();
        System.out.println("File present in Path: " + rootPath);
        System.out.println("Files -> " + fileNames);
        System.out.println();
    }

    /*
     * This method will check if directory or some file with same name as directory name given by user exists or not.
     * For Directory name provided by user --->>>
     * 1. If directory exists - This method will return a character with value as 'Y'.
     * 2. If directory doesn't exists - This method will return a character with value as 'N'.
     * 3. If some file exists with same name as directory name provided by user ->
     *          This method will return character with value as 'F'.
     */
    public static char dirExistsCheck(String rootPath, String dirName) {
        // Variable initialization.
        char isDirExistsFlag;
        String dirPath = rootPath + "\\" + dirName;

        // Checking if directory exists or not.
        File fileObj = new File(dirPath);
```

```
		if(fileObj.exists()) {
			if(fileObj.isDirectory()) {
				isDirExistsFlag = 'Y';
			}else {
				isDirExistsFlag = 'F';
			}
		}else {
			isDirExistsFlag = 'N';
		}
		return isDirExistsFlag;
	}


	/*
	 * This method will check if file or directory with same name as file name given by user exists or not.
	 * For File name provided by user --->>>
	 * 1. If file exists - This method will return a character as 'Y'.
	 * 2. If file doesn't exists - This method will return a character value as 'N'.
	 * 3. If some directory exists with same name as file name provided by user -> This method will return character as 'F'.
	 */
	public static char fileExistsCheck(String rootPath, String dirName, String fileName) {
		// Variable initialization.
		char isFileExistsFlag;
		String filePath = rootPath + "\\" + dirName + "\\" + fileName;


		// Checking if file exists or not.
		File fileObj = new File(filePath);
		if(fileObj.exists()) {
			if(fileObj.isFile()) {
				isFileExistsFlag = 'Y';
			}else {
				isFileExistsFlag = 'D';
```

```
                }
        }else {
                isFileExistsFlag = 'N';
        }
        return isFileExistsFlag;
}

/*
 * This method will create a directory with the directory name provided by user.
 * This method will return true if directory is created successfully or else it will return false.
 */
public static boolean createDirectory(String rootPath, String dirName) {
        // Creating Directory.
        String dirPath = rootPath + "\\" + dirName;
        File fileObj = new File(dirPath);
        return fileObj.mkdir();
}

/*
 * This method will create a file with the file name provided by user.
 * This method will return true if file is created successfully or else it will return false.
 */
public static boolean createFile(String rootPath, String dirName, String fileName) throws
IOException {
        // Creating file.
        String filePath = rootPath = rootPath + "\\" + dirName + "\\" + fileName;
        File fileObj = new File(filePath);
        boolean fileCreationFlag = fileObj.createNewFile();
        if(fileCreationFlag) {
                Locker.writeToFile(filePath);
        }
        return fileCreationFlag;
}
```

```
	/*
	 * This method will validate if directory exists or not using method - dirExistsCheck(String rootPath, String dirName)
	 * 1. If directory exist - it will call method fileHandling(String rootPath, String dirName)
	 * 2. If directory does not exist - As per user input it will create directory using method createDirectory(String rootPath, String dirName)
	 *                                                                  and then it will call method fileHandling(String rootPath, String dirName) or else traverse to Main Menu
	 * 3. If already some file exists with same name given by user for directory - It will ask user to try again with different directory name
	 */
	public static void directoryHandling(String rootPath) throws IOException{
		// Variable initialization.
		char isDirExistsFlag;
		boolean dirCreateFlag;
		boolean iterationFlag = true;
		String dirCreateInput = "";

		// Taking directory name as input
		System.out.println();
		System.out.println(Locker.dirInputMsg);
		String dirName = sc.nextLine();
		String dirPath = rootPath + "\\"+ dirName;

		// Validating if directory already exists or not and creating directory or file accordingly
		isDirExistsFlag = Locker.dirExistsCheck(rootPath, dirName);

		if(isDirExistsFlag == 'Y') {
			Locker.fileHandling(rootPath, dirName);
		}else if(isDirExistsFlag == 'F') {
			System.out.println("File already exists with same name in this path, So unable to create new directory with same name....");
```

```
                System.out.println("Please try again with some another user directory
name....");
                Locker.directoryHandling(rootPath);
            }else {
                while(iterationFlag) {
                    System.out.println();
                    System.out.println("Do you want to create this user directory first
??? (Y/N)");
                    dirCreateInput = sc.nextLine();
                    System.out.println();
                    System.out.println();
                    if(dirCreateInput.equalsIgnoreCase("Y") ||
dirCreateInput.equalsIgnoreCase("N")) {
                        if(dirCreateInput.equalsIgnoreCase("Y")) {
                            dirCreateFlag =
Locker.createDirectory(rootPath, dirName);
                            if(dirCreateFlag) {
                                System.out.println("Directory " +
dirName + " created successfully...");
                                System.out.println("You can check
your directory in Path : " + dirPath);
                                Locker.fileHandling(rootPath,
dirName);
                            }else {
                                System.out.println("Unable to
create a directory in this path. Please contact Administrator.");
                            }
                        }else {
                            System.out.println(Locker.retryMenuOptionMsg);
                        }
                        iterationFlag = false;
                    }else {
                        System.out.println(Locker.rtyValIpMsg);
                    }
                }
```

```
		}
	}

	/*
	 * This method will validate if file exists or not using method - fileExistsCheck(String rootPath, String dirName, String fileName).
	 * 1. If file exist - based on user input it will overwrite already existing file.
	 * 2. If file does not exist - it will create file using method createFile(String rootPath, String dirName, String fileName).
	 * 3. If already some directory exists with same file name given by user - It will ask user to try again with different file name.
	 */
	public static void fileHandling(String rootPath, String dirName) throws IOException {
		// Variable Initialization
		char isFileExistsFlag;
		boolean fileCreateFlag;
		boolean iterationFlag = true;
		String overwriteInput = "";

		// Taking file name as input
		System.out.println();
		System.out.println(Locker.fileInputMsg);
		if(Locker.fileOcc) {
			Locker.fileOcc = true;
			sc.nextLine();
		}
		String fileName = sc.nextLine();
		String filePath = rootPath + "\\" + dirName + "\\" + fileName;
		System.out.println();

		// Validating if directory already exists or not and creating directory or file accordingly
		isFileExistsFlag= Locker.fileExistsCheck(rootPath, dirName, fileName);
		System.out.println();
```

```
		if(isFileExistsFlag == 'Y') {
			System.out.println("File already exist with this name...");
			while(iterationFlag) {
				System.out.println("Do you want to overwrite existing file with new content??? (Y/N)");
				overwriteInput = sc.nextLine();
				if(overwriteInput.equalsIgnoreCase("Y") || overwriteInput.equalsIgnoreCase("N")) {
					if(overwriteInput.equalsIgnoreCase("Y")) {
						Locker.overwriteFile(filePath);
					}else {
						System.out.println(Locker.retryMenuOptionMsg);
					}
					iterationFlag = false;
				}else {
					System.out.println(Locker.rtyValIpMsg);
				}
				System.out.println();
			}
		}else if(isFileExistsFlag == 'D') {
			System.out.println("Directory already exists with same name, So will not be able to create new file with same name....");
			System.out.println("Please try again with some another file name....");
			Locker.fileHandling(rootPath, dirName);
		}else {
			fileCreateFlag = Locker.createFile(rootPath, dirName, fileName);
			if(fileCreateFlag) {
				System.out.println();
				System.out.println("File Created Successfully!!!");
				System.out.println("You can check the path : " + filePath);
			}else {
				System.out.println("Unable to create a file. Please contact administrator.");
			}
```

```
            }
}

/*
 * This method will overwrite the existing file
 */
public static void overwriteFile(String filePath) throws IOException {
            File fileObj = new File(filePath);

   // if file doesn't exists, then create it
   if (!fileObj.exists()) {
            fileObj.createNewFile();
   }
   Locker.writeToFile(filePath);
   System.out.println("File Overwritten Successfully!!!");
   System.out.println("You can check file in Path: " + filePath);
}

/*
 * This method will write content to the file
 */
public static void writeToFile(String filePath) throws IOException {
            // Variable Initialization
   int lines = 0;

            File fileObj = new File(filePath);

   FileWriter fileWriter = new FileWriter(fileObj.getAbsoluteFile());
   BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);

   // Taking input content from user and writing to file
   System.out.println("How many lines you want to enter into the file?");
```

```
            lines = sc.nextInt();
            sc.nextLine();
            for(int i = 0; i < lines; i++) {
                        String content = "";
                                System.out.println();
                    System.out.println("Please enter content for line number "+ (i+1));
                    content = sc.nextLine();
                    bufferedWriter.write(content + "\n");
            }
            bufferedWriter.close();
        }

        /*
         * This method will delete a file with the file name provided by user.
         * This method will return true if file is deleted successfully or else it will return false.
         */
        public static boolean deleteFile(String rootPath, String dirName, String fileName) throws
IOException {
                    // Variable initialization.
                    boolean fileDeleteFlag;
                    String filePath = rootPath + "\\" + dirName + "\\" + fileName;

                    // Creating file.
                    File fileObj = new File(filePath);
                    fileDeleteFlag = fileObj.delete();
                    return fileDeleteFlag;
        }

        /*
         * This method will validate if directory exists or not using method - dirExistsCheck(String
rootPath, String dirName).
         * 1. If Directory exist - it will check if file exists and accordingly it will delete that file
         * 2. If directory file does not exist - it will display message this file does not exist.
```

```
	 */
	public static void fileDeleteHandling(String rootPath) throws IOException {
		// Variable Initialization
		char isDirExistsFlag,isFileExistsFlag;
		boolean dirIterationFlag = true;
		boolean fileIterationFlag = true;
		boolean fileDeleteResult;
		String dirName, fileName;
		String dirDeleteInput = "";
		String fileDeleteInput = "";

		// Taking directory name as input
		System.out.println();
		System.out.println(Locker.dirInputMsg);
		dirName = sc.nextLine();
		System.out.println();
		// Validating if directory already exists or not and creating directory or file
accordingly
		isDirExistsFlag= Locker.dirExistsCheck(rootPath, dirName);

		if(isDirExistsFlag == 'Y') {
			// Taking file name as input
			System.out.println(Locker.fileInputMsg);
			fileName = sc.nextLine();
			System.out.println();
			// Performing file exists validation
			isFileExistsFlag= Locker.fileExistsCheck(rootPath, dirName, fileName);

			if(isFileExistsFlag == 'Y') {
				fileDeleteResult = Locker.deleteFile(rootPath, dirName, fileName);
				if(fileDeleteResult) {
					System.out.println("File Deleted Successfully!!!");
				}else {
```

```
                                        System.out.println("Unable to delete the file. Please contact
administrator.");
                                }


                        }else {
                                while(fileIterationFlag) {
                                        System.out.println("Do you want to try again to delete some
other file??? (Y/N)");
                                        fileDeleteInput = sc.nextLine();
                                        if(fileDeleteInput.equalsIgnoreCase("Y") ||
fileDeleteInput.equalsIgnoreCase("N")) {
                                                if(fileDeleteInput.equalsIgnoreCase("Y")) {
                                                        Locker.fileDeleteHandling(rootPath);
                                                }else {

        System.out.println(Locker.retryMenuOptionMsg);
                                                }
                                                fileIterationFlag = false;
                                        }else {
                                                System.out.println(Locker.rtyValIpMsg);
                                        }
                                        System.out.println();
                                }
                        }
                }else {
                        while(dirIterationFlag) {
                                System.out.println("Do you want to try again with different
directory name ??? (Y/N)");
                                dirDeleteInput = sc.nextLine();
                                if(dirDeleteInput.equalsIgnoreCase("Y") ||
dirDeleteInput.equalsIgnoreCase("N")) {
                                        if(dirDeleteInput.equalsIgnoreCase("Y")) {
                                                Locker.fileDeleteHandling(rootPath);
                                        }else {
                                                System.out.println(Locker.retryMenuOptionMsg);
```

```
                                        }
                                        dirIterationFlag = false;
                                }else {
                                        System.out.println(Locker.rtyValIpMsg);
                                }
                        }
                        System.out.println();
                }
        }

        /*
         * This method will validate if file exists or not using method - dirExistsCheck(String rootPath, String dirName)
         * 1. If Directory exist - it will check it file exists then it will delete that file
         * 2. If file does not exist - it will display message this file does not exist.
         */
        public static void fileSearching(String rootPath) throws IOException {
                // Variable Initialization
                char isDirExistsFlag;
                char isFileExistsFlag;
                boolean dirSearchFlag = true;
                boolean fileSearchFlag = true;
                String dirName;
                String fileName;
                String filePath;
                String dirSearchInput = "";
                String fileSearchInput = "";

                // Taking directory name as input
                System.out.println();
                System.out.println(Locker.dirInputMsg);
                dirName = sc.nextLine();
```

```
		// Validating if directory already exists or not and creating directory or file accordingly
		isDirExistsFlag= Locker.dirExistsCheck(rootPath, dirName);

		if(isDirExistsFlag == 'Y') {
			// Taking file name as input
			System.out.println();
			System.out.println(Locker.fileInputMsg);
			fileName = sc.nextLine();
			filePath = rootPath + "\\" + dirName + "\\" + fileName;
			// Performing file exists validation
			isFileExistsFlag= Locker.fileExistsCheck(rootPath, dirName, fileName);

			if(isFileExistsFlag == 'Y') {
				System.out.println();
				System.out.println("File found...!!!");
				Locker.getFile(filePath,fileName);
			}else {
				while(fileSearchFlag) {
					System.out.println("File not found in path : " + filePath);
					System.out.println("Do you want to try again to search for some other file??? (Y/N)");
					fileSearchInput = sc.nextLine();
					System.out.println();
					if(fileSearchInput.equalsIgnoreCase("Y") || fileSearchInput.equalsIgnoreCase("N")) {
						if(fileSearchInput.equalsIgnoreCase("Y")) {
							Locker.fileSearching(rootPath);
						}else {

	System.out.println(Locker.retryMenuOptionMsg);
						}
						fileSearchFlag = false;
					}else {
```

```
                                        System.out.println(Locker.rtyValIpMsg);
                                }
                        }
                }
        }else {
                while(dirSearchFlag) {
                        System.out.println("Do you want to try again with different
directory name ??? (Y/N)");
                        dirSearchInput = sc.nextLine();
                        System.out.println();
                        if(dirSearchInput.equalsIgnoreCase("Y") ||
dirSearchInput.equalsIgnoreCase("N")) {
                                if(dirSearchInput.equalsIgnoreCase("Y")) {
                                        Locker.fileDeleteHandling(rootPath);
                                }else {
                                        System.out.println(Locker.retryMenuOptionMsg);
                                }
                                dirSearchFlag = false;
                        }else {
                                System.out.println(Locker.rtyValIpMsg);
                        }
                }
        }
}

/*
 * This method will display the content of file mentioned at the input path
 */
public static void getFile(String filePath, String fileName) throws IOException {
        // Variable Initialization
        String string = "";
        String inputValue = "";
        boolean inputFlag = true;
```

```
		// Creating File, FileReader, BufferReader object to read the file
		File fileObj = new File(filePath);
		FileReader fileReader = new FileReader(fileObj);
		BufferedReader bufferReader = new BufferedReader(fileReader);

		// Validating user input to display content of file and displaying content in the
console
		while(inputFlag) {
			System.out.println("Do you want to read the content of the file??? (Y/N)");
			inputValue = sc.nextLine();
			System.out.println();
			if(inputValue.equalsIgnoreCase("Y") || inputValue.equalsIgnoreCase("N")) {
				if(inputValue.equalsIgnoreCase("Y")) {

	System.out.println("**********************************************************
***");
					System.out.println("Displaying content of file : " + fileName);

	System.out.println("**********************************************************
***");
					while((string = bufferReader.readLine()) != null) {
						System.out.println(string);
					}

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

				}else {
					System.out.println(Locker.retryMenuOptionMsg);
				}
				inputFlag = false;
			}else {
				System.out.println(Locker.rtyValIpMsg);
			}
```

```
                }
                bufferReader.close();
        }

}
```