package com.company;

import java.io.File;

import java.io.FileWriter;

import java.util.\*

public class LockMe

{

static File folder = new File("D:\\project1");

static Scanner scanner = new Scanner(System.in).useLocale(Locale.US);

public static void main(String[] args) {

//Boolean to identify if the loop continues or not

boolean bExit = false;

// Option Selected

Integer iResponse = 0;

// Exits when user selects option 5

while (!bExit) {

// The menu is painted on screen

menu();

try {

// User input is read

iResponse = Integer.parseInt(scanner.nextLine());

PrintMessage(" > Selected: " + iResponse);

// Choose the option selected by user

switch (iResponse) {

case 1: {

//All files are painted on screen

Random random = new Random();

int rn = random.nextInt(6) + 1;

switch (rn) {

case 1:

// Get all files using a for loop

getAllFiles\_forLoop();

break;

case 2:

// Get all files using a while loop

getAllFiles\_while();

break;

case 3:

// Get all files using a for each loop

getAllFiles\_forEachLoop();

break;

case 4:

// Get all files using an iterator of a

linkedlist

getAllFiles\_Iterator();

break;

case 5:

// Get all files using an iterator of a

linkedlist

getAllFiles\_LambdaExpression();

break;

case 6:

// Get all files using the Enumeration

Interface

getAllFiles\_EnumerationInterface();

break;

}

break;

}

case 2: {

//User defines name and contents of a new file

createFile(scanner);

break;

}

case 3: {

//User chooses a file to delete

deleteFile(scanner);

break;

}

case 4: {

//Determine is a file exists

searchFiles(scanner);

break;

}

case 5: {

//Exit the program

bExit = true;

break;

}

default: {

//On error the program ends

bExit = true;

}

}

}catch(NumberFormatException e){

PrintMessage("Please print only numbers");

}

}

}

/\*\*

\* Get All files using a for loop

\*/

public static void getAllFiles\_forLoop() {

//System.out.println("---> getAllFiles\_forLoop");

//Name of files are stored in an array

File[] listOfFiles = folder.listFiles();

//We use an Arraylist to contain the list of files

List<File> alListOfFiles = new ArrayList<File>();

Collections.addAll(alListOfFiles, listOfFiles);

//Read the Arraylist using a for loop with the name of the files1

try {

for (int i = 0; i < alListOfFiles.size(); i++) {

System.out.println(alListOfFiles.get(i));

}

//On error an exception is raised

} catch (Exception e) {

PrintMessage("Error: file not found");

}

}

/\*\*

\* Get all files using a while loop and a linkedlist

\*/

public static void getAllFiles\_while() {

//System.out.println("---> getAllFiles\_while");

File[] listOfFiles = folder.listFiles();

LinkedList<File> llListOfFiles = new LinkedList<File>();

//Get the list into the linkedlist

Collections.addAll(llListOfFiles, listOfFiles);

int counter = 0;

//Traverse the linkedlist

while (llListOfFiles.size() > counter) {

    System.out.println(llListOfFiles.get(counter));

counter++;

}

}

/\*\*

\* Get all the files using a for each loop

\*/

public static void getAllFiles\_forEachLoop() {

//System.out.println("---> getAllFiles\_forEachLoop");

File[] listOfFiles = folder.listFiles();

//Use a simplified loop

for (File myFile : listOfFiles) {

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

}

}

/\*\*

\* Get all the files using a lambda expression

\*/

public static void getAllFiles\_LambdaExpression() {

//System.out.println("---> getAllFiles\_LambdaExpression");

File[] listOfFiles = folder.listFiles();

List<File> alListOfFiles = new ArrayList<>();

Collections.addAll(alListOfFiles, listOfFiles);

//The lambda expression get the file and prints it

alListOfFiles.forEach((file) -> {

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

});

}

public static void getAllFiles\_EnumerationInterface() {

//System.out.println("---> getAllFiles\_EnumerationInterface");

File[] listOfFiles = folder.listFiles();

List<File> alListOfFiles = new ArrayList<>();

Collections.addAll(alListOfFiles, listOfFiles);

//Use the interface to iterate through the list elements

Enumeration<File> e = Collections.enumeration(alListOfFiles);

while (e.hasMoreElements()) {

System.out.println(e.nextElement().getName());

}

}

public static void getAllFiles\_Iterator() {

    File[] listOfFiles = folder.listFiles();

//We use an Arraylist to contain the list of files

List<File> alListOfFiles = new ArrayList<File>();

Collections.addAll(alListOfFiles, listOfFiles);

LinkedList<File> llListOfFiles = new LinkedList<>(alListOfFiles);try {

Iterator<File> alListOfFilesIterator = llListOfFiles.iterator();

while (alListOfFilesIterator.hasNext()) {

System.out.println(alListOfFilesIterator.next());

}

//On error an exception is raised

} catch (Exception e) {

PrintMessage("Error: file not found");

}

}

public static void deleteFile(Scanner scannerdelete) {

try {

//Read the name of the file to delete

System.out.println("Write the name of the file you want to

delete:");

File fileToDelete = new File(folder + "//" +

scannerdelete.nextLine());

//On success

if (fileToDelete.delete()) {

PrintMessage("File deleted successfully.");

} else {

//On error

PrintMessage("There was an error deleting the file");

}

//On error an exception is raised

} catch (Exception e) {

PrintMessage("There was an error deleting the file");

}

}

/\*\*

\* Search for a specific file

\*/

public static void searchFiles(Scanner scannerSearch) {

try {

//Name of the file to find

System.out.println("Write the name of the file you want to find:");

File fileTofind = new File(folder + "//" +

scannerSearch.nextLine());

// If the file existes

if (fileTofind.exists()) {

PrintMessage("File exists");

} else {

//If not a message is sent

PrintMessage("File does not exist");

}

//On error an exception is raised

} catch (Exception e) {

PrintMessage("There was an error searching the file.");

}

}

/\*\*

\* User creates a new file

\*

\* @param scannerCreate

\*/

public static void createFile(Scanner scannerCreate) {

//Writer object to use

FileWriter writer = null;

try {

//Read the name of the file to create

System.out.println("What is the name of your new file?");

File newFile = new File(folder + "//" + scannerCreate.nextLine());

writer = new FileWriter(newFile);

//Read the contents of the file to create

System.out.println("And the contents of your file are...");

writer.write(scannerCreate.nextLine());

writer.close();

PrintMessage("File created successfully");

//On error an exception is raised

} catch (Exception ex) {

ex.printStackTrace();

}

}

/\*\*

\* The menu is painted on screen

\*/

public static void menu() {

System.out.println("\n");

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

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome to LockMe.com

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Designed by Karishma Pawar\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" 1) Display all the files");

System.out.println(" 2) Add a new file ");

System.out.println(" 3) Delete a file ");

System.out.println(" 4) Search a file ");

System.out.println(" 5) Exit ");

System.out.println("");

System.out.println(" > Select an option...");

}

static void PrintMessage(String message){

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

System.out.println(" " + message);

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

}

}