**// concepts of file handling, error handling, sorting and searching is used to complete this project//**

**package** fileHandling;

**import** java.io.IOException;

**import** java.nio.charset.StandardCharsets;

**import** java.nio.file.DirectoryNotEmptyException;

**import** java.nio.file.Files;

**import** java.nio.file.Path;

**import** java.nio.file.Paths;

**import** java.nio.file.StandardOpenOption;

**import** java.io.File;

**import** java.io.FileFilter;

**import** java.util.Arrays;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Scanner;

**public** **class** virtualKey {

**public** **static** **void** main(String [] args) {

*optionsSelection*();

}

**public** **static** **void** retrieveFile()

{

Scanner scanner = **new** Scanner( System.***in*** );

System.***out***.println("Enter the file path: ");

String dirPath = scanner.nextLine(); // Takes the directory path as the user input

File folder = **new** File(dirPath);

**if**(folder.isDirectory())

{

File[] fileList = folder.listFiles();

Arrays.*sort*(fileList);

System.***out***.println("\nTotal number of items present in the directory: " + fileList.length );

FileFilter fileFilter = **new** FileFilter()

{

@Override

**public** **boolean** accept(File file) {

**return** !file.isDirectory();

}

};

fileList = folder.listFiles(fileFilter);

// Sort files by name

Arrays.*sort*(fileList, **new** Comparator()

{

@Override

**public** **int** compare(Object f1, Object f2) {

**return** ((File) f1).getName().compareTo(((File) f2).getName());

}

});

//Prints the files in file name ascending order

System.***out***.println("FILE NAMES IN ASCENDING ORDER");

**for**(File file:fileList)

{

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

}

}

}

**public** **static** **void** createFileUsingNIO() **throws** IOException

{

Scanner scanner = **new** Scanner( System.***in*** );

System.***out***.println("Enter the file path: ");

String dirPath = scanner.nextLine();

System.***out***.println("Enter the file name ");

String strName=scanner.nextLine();

File folder = **new** File(dirPath+ "" + strName + ".txt");

Path path=Paths.*get*(dirPath + "" + strName + ".txt");

System.***out***.println("Enter content of file ");

String content = scanner.nextLine();

**byte** array[]=content.getBytes();

Files.*write*(path, array, StandardOpenOption.***CREATE***,StandardOpenOption.***APPEND***);

System.***out***.println("Data Written Successfully");

Path path1=Paths.*get*(dirPath + "" + strName + ".txt");

}

**public** **static** **void** readDataUsingNIO() **throws** IOException

{

Scanner scanner = **new** Scanner( System.***in*** );

System.***out***.println("Enter path of the file to be read ");

String dirPath = scanner.nextLine();

List<String> list=Collections.*emptyList*();

Path path= Paths.*get*(dirPath);

list= Files.*readAllLines*(path,StandardCharsets.***UTF\_8***);

Iterator<String> it= list.iterator();

**while**(it.hasNext()) {

System.***out***.println(it.next());

}

}

**private** **static** **void** closeApp() {

System.***out***.println("Closing your application... \nThank you!");

}

**public** **static** **void** DeleteFile() {

Scanner scanner = **new** Scanner( System.***in*** );

System.***out***.println("Enter the file path: ");

String dirPath = scanner.nextLine();

**try** {

Path path= Paths.*get*(dirPath);

**if**(Files.*deleteIfExists*(path))

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

**else**

System.***out***.println("File not Deleted");

} **catch** (DirectoryNotEmptyException e) {

System.***out***.println("Directory is not empty");

}

**catch** (IOException e) {

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

}

}

**private** **static** **void** optionsSelection() {

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

String[] arr = {"1. Retrieve current file name in ascending order",

"2. Business level operation menu",

"3. Exit from the application",

};

**int**[] arr1 = {1,2,3};

**int** slen = arr1.length;

**for**(**int** i=0; i<slen;i++){

System.***out***.println(arr[i]);

}

System.***out***.println("\nEnter your choice:\t");

Scanner sc = **new** Scanner(System.***in***);

**int** options = sc.nextInt();

**for**(**int** j=1;j<=slen;j++){

**if**(options==j){

**switch** (options){

**case** 1:

*retrieveFile*();

*optionsSelection*();

**break**;

**case** 2:

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

String[] arr2 = {"1. ADD A FILE AND ITS CONTENT TO DIRECTORY",

"2. DELETE A FILE FROM DIRECTORY",

"3. SEARCH A FILE AND SHOW ITS CONTENTS",

"4. EXIT FROM BLO MENU",

};

**for**(**int** i=0; i<=3;i++){

System.***out***.println(arr2[i]);

// display the all the Strings mentioned in the String array

}

System.***out***.println("\nEnter your choice:\t");

**int** option2 = sc.nextInt();

**for**(**int** k=1;k<=4;k++){

**if**(option2==k){

**switch** (option2){

**case** 1:

**try** {

*createFileUsingNIO*();

} **catch** (IOException e) {

e.printStackTrace();

}

*optionsSelection*();

**break**;

**case** 2:

*DeleteFile*();

*optionsSelection*();

**break**;

**case** 3:

**try** {

*readDataUsingNIO*();

} **catch** (IOException e) {

System.***out***.println("File not available");

}

*optionsSelection*();

**break**;

**case** 4:

System.***out***.println("You have exited from BLO MENU...");

*optionsSelection*();

**break**;

**default**:System.***out***.println("You have made an invalid choice!");

**break**;

}

}

}

**break**;

**case** 3:

*closeApp*();

**break**;

**default**:

System.***out***.println("You have made an invalid choice!");

**break**;

}

}

}

}

}