**package** com.mainprojectphase1;

**import** java.io.File;

**import** java.io.FileOutputStream;

**import** java.util.Arrays;

**import** java.util.Comparator;

**import** java.util.Scanner;

**public** **class** Mainproject1 {

/\*ADDING OR CREATE NEW FILE METHOD STARTS\*/

**public** **static** **void** add() /\*<---- CREATE NEW FILE HERE\*/

{

**try**

{

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

{

System.***out***.println("ENTER THE FILE NAME WITH LOCATION FOR SAVE FILE (e.g --> path\\FileName.txt):-->");

String filename = sc.nextLine();

FileOutputStream fos = **new** FileOutputStream(filename, **true**);

System.***out***.println("ENTER THE FILE CONTENT :-->");

String contain = sc.nextLine();

**byte** b[] = contain.getBytes();

fos.write(b);

fos.close();

}

System.***out***.println("FILE IS SAVED IN GIVEN PATH :-->");

}

**catch** (Exception e)

{

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

e.printStackTrace();

}

}

/\*ADDING OR CREATE NEW FILE METHOD END\*/

/\*THIS METHOD FOR ASENDING ORDER\*/

**public** **static** **void** accendingorder()

{

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

{

System.***out***.println("ENTER PATH TO SHOW YOUR FILE'S IN ASENDING ORDER (e.g --> path\\FileName.txt):-->");

String filename = sc.nextLine();

File dir = **new** File(filename);

**if**(dir.isDirectory())

{

File[] files = dir.listFiles();

System.***out***.println("THIS PATH CONTAINS FOLLOWING FILES :--> ");

*extracted*(files); /\*SORT BY NAME ONLY\*/

**for**(File file:files) /\*ASENDING ORDER\*/

{

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

}

System.***out***.println("AFTER THE SORTING OF YOUR FILE'S WE GOT THIS ASENDING ORDER :-->");

**for**(File file:files)

{

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

}

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

} **else**

{

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

}

}

}

**private** **static** **void** extracted(File[] files) {

Arrays.*sort*(files, **new** Comparator<Object>()

{

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

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

}

});

}

/\*ASENDING METHOD IS END\*/

/\*DELETETION METHOD START\*/

**public** **static** **void** delete()

{

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

{

System.***out***.println("ENTER THE FILE NAME WITH LOCATION FOR DELETE THE FILE (e.g --> path\\FileName.txt):-->");

String filename = sc.nextLine();

File file= **new** File(filename);

**if** (file.delete()) {

System.***out***.println("GIVE FILE NAME IS DELETED SUCESSFULLY");

}

**else** {

System.***out***.println("FAILED TO DELETE THE FILE");

}

}

}

/\*DELETETION METHOD END\*/

/\*SEARCHING METHOD IS START\*/

**public** **static** **void** search()

{

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

{

System.***out***.println("ENTER THE FILE NAME WHICH YOU WANT TO SEARCH (e.g --> path\\FileName.txt):---> ");

String filename = sc.nextLine();

File fff = **new** File(filename);

**if**( fff.exists())

{

System.***out***.println("FILE IS AVILIABLE \n");

}

**else**

System.***out***.println("THIS FILE IS NOT HERE!!!! SORRY");

}

}

/\*SEARCHING METHOD IS END\*/

/\*GO BACK METHOD START\*/

**public** **static** **void** back()

{

String ab = "Welcome to Lockedme.com";

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

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

String DN = "Developer Name : Shivani Gokhe.\nDesignation : Java Developer.\nDate : 09/05/2022";

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

**try** (Scanner console = **new** Scanner(System.***in***))

{

**int** ch;

System.***out***.print(" \nEnter : 1 for Geting Files Name's In ASENDING ORDER. "

+ "\nEnter : 2 For BUSSINESS LEVEL OPERATION'S. "

+ "\nEnter : 3 For CLOSE the Application.\n\n\t");

ch = console.nextInt();

**switch**(ch)

{

**case** 1 :

*accendingorder*();

**break**;

**case** 2 :

**int** ch2;

System.***out***.println("FOLLOWING ARE THE BUSSINESS OPERATION'S --->");

System.***out***.print("\nEnter : a For CREATE or ADD NEW FILE.)"

+"\nEnter : b For DELETE the File."

+"\nEnter : c For SEARCH the File."

+"\nEnter : d To GO BACK.\n\n\t");

ch2 = console.next().charAt(0);

**switch**(ch2)

{

**case** 'a' :

*add*();

**break**;

**case** 'b' :

*delete*();

**break**;

**case** 'c' :

*search*();

**break**;

**case** 'd' :

System.***out***.println("THANK YOU FOR VISIT <YOU ARE GOING BACK>");

System.*exit*(ch2);

**break**;

}

**break**;

**case** 3 :

System.***out***.println("SYSTEM GONNA CLOSE\nVISIT AGAIN THANK YOU..");

System.*exit*(ch);

**default** :

System.***out***.println("SOMETHING GOES WRONG(EXCEPTION)....");

**break**;

}

}

}

/\*GO BACK METHOD END\*/

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

{

String ab = "Welcome to Lockedme.com";

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

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

String DN = "Developer Name : Shivani Gokhe.\nDesignation : Java Developer.\nDate : 05/05/2022";

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

**try** (Scanner console = **new** Scanner(System.***in***))

{

**int** ch;

System.***out***.print(" \nEnter : 1 for Geting Files Name's In ASENDING ORDER. "

+ "\nEnter : 2 For BUSSINESS LEVEL OPERATION'S. "

+ "\nEnter : 3 For CLOSE the Application.\n\n\t");

ch = console.nextInt();

**switch**(ch)

{

**case** 1 :

*accendingorder*();

**break**;

**case** 2 :

**int** ch2;

System.***out***.println("FOLLOWING ARE THE BUSSINESS OPERATION'S --->");

System.***out***.print("\nEnter : a For CREATE or ADD NEW FILE.)"

+"\nEnter : b For DELETE the File."

+"\nEnter : c For SEARCH the File."

+"\nEnter : d To GO BACK.\n\n\t");

ch2 = console.next().charAt(0);

**switch**(ch2)

{

**case** 'a' :

*add*();

**break**;

**case** 'b' :

*delete*();

**break**;

**case** 'c' :

*search*();

**break**;

**case** 'd' :

*back*();

**break**;

}

**break**;

**case** 3 :

System.***out***.println("SYSTEM GONNA CLOSE\nVISIT AGAIN THANK YOU..");

System.*exit*(ch);

**default** :

System.***out***.println("SOMETHING GOES WRONG(EXCEPTION)....");

**break**;

}

}

}

}