//MAIN CLASS

```
package com.Lockers;
//import java.awt.List;
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;
import java.util.regex.Pattern;
import java.util.InputMismatchException;
import java.util.Iterator;
import java.util.List;


//import javax.naming.AuthenticationException;
public class Main {
    private static int choiceBusiness = 0;
    private static int choiceMain=0;
    private static int mainMenuUpperBound=4;
    private static int businessMenuUpperBound=4;
    private static int menuType;
    private static boolean isFileFound=false;
    private static File currentDirectory = new File(".");
    private static String currentDirectoryPath = currentDirectory.getAbsoluteP
ath();
    private static boolean hasSearchedSubdir;

    public static void main (String[] args) throws Exception{
        // TODO Auto-generated method stub

        showWelcomeMessage();


        do {
            System.out.println("You are currently in the following directory")
;
            System.out.println(currentDirectory.getAbsolutePath());
            showMainMenu();
            menuType=1;
            System.out.println("\nWhats your choice?");
//          System.out.println(currentDirectoryPath);
            choiceMain=inputNumber(mainMenuUpperBound,true);

            switch (choiceMain) { //switch main
            case 1 :
                displayFiles();
                currentDirectoryPath=currentDirectory.getAbsolutePath();
                //              System.out.println("Current Directory is now s
et to "+ currentDirectory.getAbsolutePath());
                break;

            case 2:


                int sortCh=directoryChoice("sort");
                //              WorkingDirectory(sortCh);
                currentDirectory=WorkingDirectory(sortCh);
                currentDirectoryPath=currentDirectory.getAbsolutePath();

                File[] sortedDirecotry= sortFiles(currentDirectory);
                displayFiles(sortedDirecotry);
                break;

            case 3 :
                boolean breaking=false;
                boolean redirect=false;

                do{ //loop for business operation menu
                    showBusinessMenu();
                    menuType=2;
                    System.out.println("\nWhats your choice?");
                    choiceBusiness=inputNumber(businessMenuUpperBound, true);

                    switch (choiceBusiness) { //switch statement for the busin
ess operation menu
                    case 1:
                        addFile();
                        break;

                    case 2:
                        deleteFile();
                        break;
                    case 3:
                        int searchCh=directoryChoice("SEARCH");
                        currentDirectory=WorkingDirectory(searchCh);
                        System.out.println("Enter the name of the file");
                        searchFile(currentDirectory, inputFile());

                        if (!isFileFound) {
                            System.out.println("File not found");

                        }

                        else {
                            isFileFound=false;
                        }
```

```
                            break;
                        case 4:
                            System.out.println("Going back to the Main Menu");
                            breaking=true;
                            break;
                        default:
                            System.out.println("Not a valid input. Try from the op
tions above");
                        }//Close Switch for business operation menu
                        if (breaking==true) { //setting the redirect variable to t
rue so user
                            //                                          can go bac
k to the main menu.

                            redirect=true;
                            break;
                        }

                    }while(choiceBusiness !=0); //Business Menu Loop Close

                    if (redirect) {
                        continue;//for running the loop again from the start after
 redirection.
                    }

                case 4:
                    System.out.println("Thank you for using this application...\nA
borting program");
                    System.exit(0);
                default :
                    System.out.println("Not a valid choice");
                }//switch main close
            }while (choiceMain !=4); //Main Menu loop close

        }//main method close

        private static void showWelcomeMessage() {
            // TODO Auto-generated method stub
            System.out.println("\n\n");
            System.out.println("\t\t\t█░░ █▀█ █▀▀ █▄▀ █▀▀ █▀█ █▀   █▀█ █░█
▀█▀   █░░ ▀█▀ █▀▄");
            System.out.println("\t\t\t█▄▄ █▄█ █▄▄ █░█ ██▄ █▀▄ ▄█   █▀▀ ▀▄▀
░█░   █▄▄ ░█░ █▄▀");

            System.out.println("\n\n\t\t\t\t\t***Welcome to Lockers Pvt Ltd***");
            System.out.println("\t\t\t\t\t\t[By Anurag Kumar]\n\n");

        }
```

```
    static void addFile() {

        System.out.println("Enter the file name that you want to create");
        String fileName=inputFile();

        File file = new File(currentDirectoryPath+"\\"+fileName);

        boolean isFileExist;
        try {
            isFileExist=file.createNewFile();
            if (isFileExist) {
                System.out.println("File created at location "+file.getParent(
));
            }
            else {
                System.out.println("Sorry, file already exist with the same na
me.");
            }

        } catch (IOException e) {
            // TODO: handle exception

            e.printStackTrace();

        }
    }

    static void deleteFile() {

        System.out.println("Enter the file name that you want to delete");
        String fileName=inputFile();
        File file = new File(currentDirectoryPath+"\\"+fileName);

        try {
            if (file.delete()) {
                System.out.println("File "+file.getName()+ " deleted SuccessFu
lly");
            }
            else {
                System.out.println("File not present in the directory");
            }

        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
    }
    public static int directoryChoice(String word) {
        System.out.println("\nPress 1 if you want to "+word+" the files in cur
rent directory");
        System.out.println("Press 2 if you want to "+word+" files in custom di
rectory");

        return inputNumber(2);

    }

    public static void searchFile(File dir ,String fileName) {



        currentDirectoryPath=currentDirectory.getAbsolutePath();

        File[] files = dir.listFiles();
        File updatedFolder=dir;

        if (files!=null) {
            for (File file : files) {
                if (file.isFile() && file.getName().equalsIgnoreCase(fileName)
) {

                    isFileFound=true;
                    if (hasSearchedSubdir) {
                        System.out.println("\nFile found sucessfully in one of
 the subsequent direcotories"
                                + "\nThe file is located in <" + updatedFolder
.getPath());
                    }

                    else {
                        System.out.println("File found in the current director
y");
                    }
//



                }

                else {
                    hasSearchedSubdir=true;
                    updatedFolder=file;
                    searchFile(updatedFolder, fileName);
                }
            }
        }

    }


    static void displayFiles(File dir) {
        File[] filesInDirectory = dir.listFiles();
        System.out.println("\n\n*********************************************
******************************************************************************
******");
        System.out.printf("%-4s %-70s %-40s %-
50s","Sno.", "File name", "File Type", "File Size");
        System.out.println("\n\n*********************************************
******************************************************************************
******");

        double size=0;
        int count=0;
        String unit="bytes";

        String fileType="file";
        for (File f : filesInDirectory) {

            if (f.isDirectory()) { //if else block for fetching directory size
                size=getDirectorySize(f);
                fileType="Directory";
            }

            else {
                size=(double)f.length();
                String type = f.getName();
                String[] filesType=type.split("[.]");
                fileType= "."+filesType[filesType.length-1];

            }       //if else block for fetching directory size

            if (size<1024) {
                size = (int) size/1;
                unit="bytes";
            }

            else if (size>1024 && size<1024*1024) {
                unit="KB";
                size=size/1024;
                size = Math.round(size*10.0)/10.0;

            }

            else if (size>1024*1024 && size<1024*1024*1024) {
                unit="MB";
                size=size/1024/1024;
                size = Math.round(size*100.0)/100.0;
            }

            else if (size>1024*1024*1024) {
                unit="GB";
                size=size/1024/1024/1024;
                size = Math.round(size*100.0)/100.0;

            }
            System.out.printf("%-4s %-70s %-40s %-
50s\n",++count, " "+f.getName(),fileType,size+" "+unit );
        }//end of for each loop
    } //End of display files method


    static void displayFiles(File[] dir) {
        //      File[] filesInDirectory = dir.listFiles();
        System.out.println("\n*************************************************
*******************************************************************************
****");
        System.out.printf("%-4s %-70s %-40s %-
50s","Sno.", "File name", "File Type", "File Size");
        System.out.println("\n\n***********************************************
*******************************************************************************
******");

        double size=0;
        int count=0;
        String unit="bytes";

        String fileType="file";
        for (File f : dir) {

            if (f.isDirectory()) { //if else block for fetching directory size
                size=getDirectorySize(f);
                fileType="Directory";
            }

            else {
                size=(double)f.length();
                String type = f.getName();
                String[] filesType=type.split("[.]");
                fileType= "."+filesType[filesType.length-1];
            }       //if else block for fetching directory size

            if (size<1024) {
                size = (int) size/1;
                unit="bytes";
            }

            else if (size>1024 && size<1024*1024) {
                unit="KB";
                size=size/1024;
                size = Math.round(size*10.0)/10.0;
            }

            else if (size>1024*1024 && size<1024*1024*1024) {
                unit="MB";
                size=size/1024/1024;
                size = Math.round(size*100.0)/100.0;
            }

            else if (size>1024*1024*1024) {
                unit="GB";
                size=size/1024/1024/1024;
                size = Math.round(size*100.0)/100.0;
            }
            System.out.printf("%-4s %-70s %-40s %-
50s\n",++count, " "+f.getName(),fileType,size+" "+unit );
        }//end of for each loop
    } //End of display files method

    static void displayFiles()  {

        int ch=directoryChoice("view");

        currentDirectory= WorkingDirectory(ch);
        currentDirectoryPath=currentDirectory.getAbsolutePath();

        //      System.out.println("Absolute path is " +wd.getAbsolutePath());
        File[] filesInDirectory = currentDirectory.listFiles();

        System.out.println("\nDisplaying files");

        System.out.println("\n************************************************
******************************************************************************
****");
        System.out.printf("%-4s %-70s %-40s %-
50s","Sno.", "File name", "File Type", "File Size");
        System.out.println("\n\n**********************************************
******************************************************************************
******");

        double size=0;
        int count=0;
        String unit="bytes";

        String fileType="file";
        for (File f : filesInDirectory) {

            if (f.isDirectory()) { //if else block for fetching directory size
                size=getDirectorySize(f);
                fileType="Directory";
            }

            else {
                size=(double)f.length();
                String type = f.getName();
                String[] filesType=type.split("[.]");
                fileType= "."+filesType[filesType.length-1];

            }       //if else block for fetching directory size

            if (size<1024) {
                size = (int) size/1;
                unit="bytes";
            }

            else if (size>1024 && size<1024*1024) {
                unit="KB";
                size=size/1024;
                size = Math.round(size*10.0)/10.0;
            }

            else if (size>1024*1024 && size<1024*1024*1024) {
                unit="MB";
                size=size/1024/1024;
                size = Math.round(size*100.0)/100.0;
            }

            else if (size>1024*1024*1024) {
                unit="GB";
                size=size/1024/1024/1024;
                size = Math.round(size*100.0)/100.0;
            }
            System.out.printf("%-4s %-70s %-40s %-
50s\n",++count, " "+f.getName(),fileType,size+" "+unit );
        }//end of for each loop
    } //End of display files method

    public static double getDirectorySize(File dir) { // recursive function to
 get the size of directory
```

```
        double sz = 0;
        File[] files = dir.listFiles();
        if (files != null) {
            for (File file : files) {
                if (file.isFile())
                    sz += file.length();
                else
                    sz += getDirectorySize(file);
            }
        }
        return sz;
    }

    public static String inputFile() {
        Scanner scanner = new Scanner(System.in);
        String name=null;

        try {
            name = scanner.nextLine();
        } catch (Exception e) {
            // TODO: handle exception
            name = "NoName";
        }
        return name;
    }


    //  public static File inputDir() {
    //      Scanner scanner = new Scanner(System.in);
    //
    //
    //      try {
    //          File dir = dir(scanner.nextLine());
    //      } catch (Exception e) {
    //          // TODO: handle exception
    //          name = "NoName";
    //      }
    //      return dir;
    //
    //  }



    public static int inputNumber(int upperBound) { // input number method ok!
!
        Scanner sc = new Scanner(System.in);
        int choice=0;
        while (choice<=0||choice>upperBound) {
            try {
```

```java
                System.out.println(" ** NOTE**- Please enter a number between 
1 and "+upperBound+" or press 9 to quit");
                choice=sc.nextInt();
                if (choice==9) {
                    System.out.println("Shutting down application");
                    System.exit(0);
                }

            } catch (InputMismatchException e) {
                // TODO: handle exception
                System.out.println("Invalid choice!! Please try again with a n
umber only");
                sc.next();
                choice=0;

                if (menuType==1) {
                    showMainMenu();
                }

                else if(menuType==2){
                    showBusinessMenu();
                }

            }
        }
        return choice;
    } // input number close


    public static int inputNumber(int upperBound, boolean isMenu) { // input n
umber method ok!!
        Scanner sc = new Scanner(System.in);
        int choice=0;
        while (choice<=0||choice>upperBound) {
            try {
                System.out.println(" ** NOTE**- Please enter a number between 
1 and "+upperBound);
                choice=sc.nextInt();
                if (choice==9) {
                    System.out.println("Shutting down application");
                    System.exit(0);
                }

            } catch (InputMismatchException e) {
                // TODO: handle exception
                System.out.println("Invalid choice!! Please try again with a n
umber only");
                sc.next();
                choice=0;
                if (menuType==1) {
                    showMainMenu();
                }

                else if(menuType==2){
                    showBusinessMenu();
                }

            }
        }
        return choice;
    } // input number close




    static void showBusinessMenu() {
        System.out.println("\n\t\t\t BUSINESS OPERATIONS MENU\n");
        System.out.println("*Please select from the following options and pres
s enter key for your choice\n");
        System.out.println("\t1. Add a file in the current directory ");
        System.out.println("\t2. Delete a file from the current directory");
        System.out.println("\t3. Search for a specific file");
        System.out.println("\t4. Return to the main menu");
    }

    static void showMainMenu() {
        System.out.println("\n\t\t\t\tMAIN MENU\n");
        System.out.println("*Please select from the following options and pres
s enter key for your choice\n");
        System.out.println("\t1. Display the files present in a directory");
        System.out.println("\t2. Sort files in a directory (Ascending)");
        System.out.println("\t3. More options");
        System.out.println("\t4. Exit the program");

    }





    @SuppressWarnings("unchecked")
    static File[] sortFiles(File dir) {
        File[] filesInDirectory = dir.listFiles();
        System.out.println("\n Files sorted successfully");

        Arrays.sort(filesInDirectory, new FileSorter());
        return filesInDirectory;
    }

    public static File WorkingDirectory(int choice) {

        Scanner scanner=new Scanner(System.in);

        File directory = null;

        if (choice==1) {
            directory = new File(currentDirectoryPath);
        }
        else if (choice==2) {

            while (currentDirectory.canRead()) {
                //              System.out.println("CD is "+currentDirectoryPa
th);
                System.out.println("Please enter an existing directory path");

                directory = new File(scanner.nextLine());

                if (!(directory.isDirectory())) {
                    System.out.println("INPUT ERROR. Try Again");
                    continue;
                }

                else if (directory.canExecute()){
                    break;
                }

            }

        }
        return directory;

    }

}//Class Main closed
```

```java
//FILE SORTER CLASS

package com.Lockers;
import java.io.File;
import java.util.Comparator;

public class FileSorter implements Comparator
{

    private final boolean isDigit(char ch)
    {
        return ch >= 48 && ch <= 57;
    }


    private final String getChunk(String s, int slength, int marker)
    {
        StringBuilder chunk = new StringBuilder();
        char c = s.charAt(marker);
        chunk.append(c);
        marker++;
        if (isDigit(c))
        {
            while (marker < slength)
            {
                c = s.charAt(marker);
                if (!isDigit(c))
                    break;
                chunk.append(c);
                marker++;
            }
        } else
        {
            while (marker < slength)
            {
                c = s.charAt(marker);
                if (isDigit(c))
                    break;
                chunk.append(c);
                marker++;
            }
        }
        return chunk.toString();
    }

    @Override
    public int compare(Object o1, Object o2)
    {
        if (!(o1 instanceof File) || !(o2 instanceof File))
```

```
            {
                return 0;
            }
            File f1 = (File)o1;
            File f2 = (File)o2;
            String s1 = f1.getName();
            String s2 = f2.getName();

            int thisMarker = 0;
            int thatMarker = 0;
            int s1Length = s1.length();
            int s2Length = s2.length();

            while (thisMarker < s1Length && thatMarker < s2Length)
            {
                String thisChunk = getChunk(s1, s1Length, thisMarker);
                thisMarker += thisChunk.length();

                String thatChunk = getChunk(s2, s2Length, thatMarker);
                thatMarker += thatChunk.length();

                // If both chunks contain numeric characters, sort them numericall
        y

                int result = 0;
                if (isDigit(thisChunk.charAt(0)) && isDigit(thatChunk.charAt(0)))
                {
                    // Simple chunk comparison by length.
                    int thisChunkLength = thisChunk.length();
                    result = thisChunkLength - thatChunk.length();
                    // If equal, the first different number counts
                    if (result == 0)
                    {
                        for (int i = 0; i < thisChunkLength; i++)
                        {
                            result = thisChunk.charAt(i) - thatChunk.charAt(i);
                            if (result != 0)
                            {
                                return result;
                            }
                        }
                    }
                } else
                {
                    result = thisChunk.compareTo(thatChunk);
                }

                if (result != 0)
                    return result;
            }

            return s1Length - s2Length;
        }
    }
```