**Source Code**

Company Lockers Pvt. Ltd Demo program.

Programmer: **Luis A. Silva Soto**

Three classes where defined for the program.

1) CompanyLockerMain Class

**package** phase1;

**public** **class** CompanyLockerMain {

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

// Good Practice for the main class to be simple so I only call a Menu Object

Menu obj = **new** Menu();

obj.menu();

}

}

2) Menu Class:

**package** phase1;

**import** java.io.IOException;

**import** java.util.Scanner;

**import** java.util.SortedSet;

**public** **class** Menu {

**public** **void** menu ()

{

**int** c=0;

**int** d=0;

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

SortedSet<String> temp1;

**boolean** check = **true**,check1 = **true**;

// Do While has been used so that the menu only exits when the option 3 is inputted by the user.

**do** {

System.***out***.println("Welcome to Company Lockers LTD");

System.***out***.println("Developer : Luis Silva");

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

System.***out***.println("1. Retrieve Files");

System.***out***.println("2. Business Level Operations");

System.***out***.println("3. Exit");

// Inside the do While loop I define an object of type File Operations.

FileOperations obj2 = **new** FileOperations ();

// To Avoid the program to Exit with an exception if the user inputs a character instead

// of an integer, I include the exception handling within a do While.

**do** {

**try** {

c = choice.nextInt();

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

// Exception handling for incorrect options - the system is expecting an integer but this code handles if the user inputs a character, negative integer or <3

**if** (c<0 || c>3 ) {

System.***out***.println("Incorrect choice ! Please input 1-3 \n");

}

check = **false**;

}

**catch** (Exception e) {

System.***out***.println("Incorrect Option! Enter a number! \n");

choice.nextLine();

}

} **while** ( check);

**if** (c==1) {

temp1 = obj2.retrieveFiles();

**if** (temp1.isEmpty()) {

System.***out***.println("Directory is empty! \n");

}**else** {

System.***out***.println("List of Files in Alphabetical Order:");

**for** (String value :temp1) {

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

}

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

}

}

**else** **if** (c==2) {

// I've included a sub menu for the operations with a file.

**do** {

System.***out***.println("Business Level Operations Menu");

System.***out***.println("1. Add a File");

System.***out***.println("2. Delete a File");

System.***out***.println("3. Search a File");

System.***out***.println("4. Return to Main Menu");

**do** {

// Same logic of Exception handling as in the Main menu

**try** {

d = choice.nextInt();

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

**if** (d<0 || d>4 ) {

System.***out***.println("Incorrect choice ! Please input 1-4 \n");

}

check1 = **false**;

}

**catch** (Exception e) {

System.***out***.println("Incorrect Option! Enter a number! \n");

choice.nextLine();

}

} **while** ( check1);

**if** (d == 1) {

//Exception handling for adding a file.

**try** {

obj2.addFile();

} **catch** (IOException e) {

e.printStackTrace();

System.***out***.println("There was an issue adding the file, please retry");

}

} **else** **if** (d ==2) {

obj2.deleteFile();

} **else** **if** (d==3){

obj2.search();

}

} **while** (d != 4);

}

} **while** (c != 3);

choice.close();

}

}

3) FileOperations Class

**package** phase1;

**import** java.io.File;

**import** java.util.\*;

**import** java.io.IOException;

**public** **class** FileOperations {

File path = **new** File(System.*getProperty*("user.dir"));// home directory so it can run in any environment

String set [] = path.list();

SortedSet<String> temp = **new** TreeSet<String> (String.***CASE\_INSENSITIVE\_ORDER***); // To ignore case sensitive in the sorting

Scanner choice2= **new** Scanner(System.***in***);

String d;

// Method to retrieve the files - Adding the Array with all the names of the files to a SortedSet and then returning the SortedSet

**public** SortedSet<String> retrieveFiles(){

set = path.list();

**for** (**int** i=0; i < set.length; i++) {

temp.add(set[i]);

}

**return** temp;

}

//This Method adds a file, with exception handling in case there is a issue with the method of File createNewFile

**public** **void** addFile() **throws** IOException

{

System.***out***.println("Name of the File to be created ->");

d = choice2.next();

**int** c=**this**.findFile(d);

**if** (c<1) {

File newFile = **new** File(path.getAbsolutePath()+"\\"+d);

newFile.createNewFile();

}**else** {System.***out***.println("File Already Exists! \n");}

}

//This Method Searches a specific File Name in the directory

**public** **void** search(){

System.***out***.println("Name of the File to be searched ->");

d = choice2.next();

**int** b = **this**.findFile(d);

**if** (b >= 0) {

System.***out***.println("File Found! \n");

}**else** {System.***out***.println("File Not Found! \n");}

}

**public** **void** deleteFile() {

System.***out***.println("Name of the File to be deleted ->");

d = choice2.next();

**int** a = **this**.findFile(d);

**if** (a>=0) {

File newFile1 = **new** File (path.getAbsolutePath()+"\\"+d);

newFile1.delete();

}**else** {

System.***out***.println("File does not exist! \n");

}

}

// This Function received the User inputed String - and then checks if it matches with any of the File names in

// the directory - if it does it returns the Index in the set, if it does not it returns a negative number.

**public** **int** findFile(String a) { //STRCMPI find

**int** j;

**int** c = -1;

set = path.list();

**for** (**int** i=0; i < set.length; i++) {

j = set[i].compareTo(a);

**if** (j == 0 ) {

c = i;

}

}

**return** c;

}

}