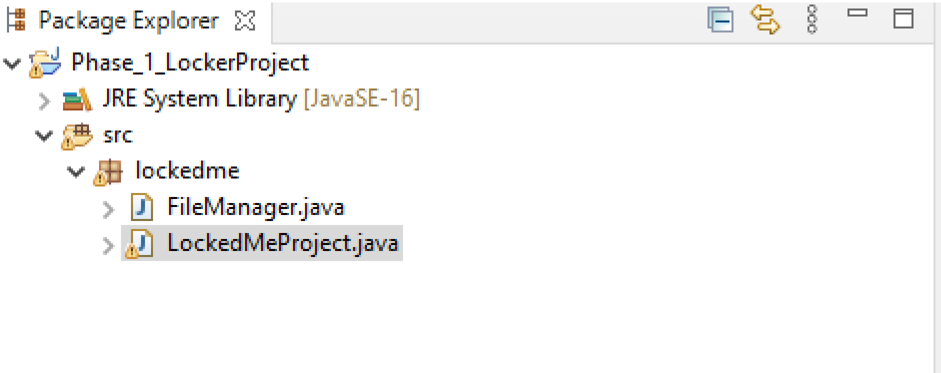
LOCKEDME.COM

SOURCE CODE

Author	Date	Version
Tushar Binnar	15-08-2021	1.0

Contents

Project Folder Structure



Package Explorer
Phase_1_LockerProject
JRE System Library [JavaSE-16]
src
lockedme
FileManager.java
LockedMeProject.java

LockedMeProject.java

```
package lockedme;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class LockedMeProject
{
        static final String folderpath="G:\\Tushar Softies\\Git_Repo\\Project\\Phase_1_Project\\Locked_Me_Files";
```

Main Method

```
        public static void main(String[] args)
        {
                //Variables
                int IsContinueMainMenu = 1, IsContinueSubMenu = 1;
                int MainMenu_ch = 0,SubMenu_ch=0;

                do // Do while to display Main Menu again & again
                {

                        MainMenu_ch = ReadUserInput("MainMenu");
                        switch(MainMenu_ch)
                        {
                                case 1 : getAllFiles();
                                                        break;
                                case 2 : IsContinueSubMenu = 1;
                                                do { // Do while to display Sub Menu again & again
                                                        SubMenu_ch = ReadUserInput("SubMenu");
                                                                switch(SubMenu_ch)
                                                                {
                                                                        case 1 : addFiles();
                                                                                        break;
                                                                        case 2 : deleteFile();
                                                                                        break;
                                                                        case 3 : searchFile();
                                                                                        break;
                                                                        case 4 : IsContinueSubMenu = -1;
                                                                                        break;
                                                                default : System.out.println("Invalid Option");

                                                                }
                                                        }while(IsContinueSubMenu > 0);
                                                        break;
                                case 3 : System.exit(0);
                                                        break;
                                default : System.out.println("Invalid Option");
                        }

                }while(IsContinueMainMenu>0);
```

```
}
```

Main Menu Display Method

```
/**
	 * Method to print display Menu
	 * @return
	 */
	public static void MainMenuDisplay()
	{
		System.out.println("****************************************************************************");
		System.out.println("\t\tLocked Me.Com");
		System.out.println("\tDeveloper :- Tushar Binnar");
		System.out.println("****************************************************************************");
		System.out.println("1. Display List Of Files");
		System.out.println("2. File Opeartions List");
		System.out.println("3. Exit");

	}
```

Sub Menu Display Method

```
/**
	 * Method Sub Menu Display
	 */
	public static void SubMenuDisplay()
	{
		System.out.println("****************************************************************************");
		System.out.println("\t\tFile Operation Menu");
		System.out.println("****************************************************************************");
		System.out.println("1. Add New File");
		System.out.println("2. Delete a File");
		System.out.println("3. Search a file");
		System.out.println("4. Return to Main Menu");
	}
```

Read User Input Method

```
/**
	 * Method to Read User Input
	 * @param MenuType
	 * @return
	 */
	public static int ReadUserInput(String MenuType)
	{
		int IsWrongChoice;
		int ch = 0;
		do //Do while loop to display Menu again if choice is not valid
		{
			try
			{
				//Scanner object creation
				Scanner sc = new Scanner(System.in);
				//Display Menu
				if(MenuType == "SubMenu")
					SubMenuDisplay();
				else
					MainMenuDisplay();
				System.out.println("Enter Your Choice:");
				ch =Integer.parseInt(sc.nextLine());
				IsWrongChoice = 1;
			}
			catch(Exception ex)
			{
				System.out.println("Invalid Choice. Please Enter choice again");
				IsWrongChoice = 0;
			}
		}while(IsWrongChoice ==0);

		return ch;
	}
```

Get All File List Method

```
/**
         * Method to get all file list
         */
        public static void getAllFiles()
        {
                int count = 1;
                //To Get List of files in FOlder
                List<String> fileNames = FileManager.getAllFileNames(folderpath);

                System.out.println("\n\t List Of Files");
                for(String f:fileNames)
                {
                        System.out.println(count+" " +f);
                        count++;
                }

        }
```

Add Files Method

```
        /**
         * Method add file in list
         */
        public static void addFiles()
        {
                //Variable Declaration
                String fileName;
                int linesCount;

                //Scanner object creation
                Scanner sc = new Scanner(System.in);

                //Array list object creation
                List<String> content = new ArrayList<String>();

                //Read File Name to be created from User
                System.out.println("Enter file name to be added:");
                fileName=sc.nextLine();

                //Read number of lines in file from user
                System.out.println("Enter the number of lines in file:");
                linesCount=Integer.parseInt(sc.nextLine());

                //Read Lines from user
                for(int i=1;i<=linesCount;i++)
                {
                        System.out.println("Enter line "+i+":");
                        content.add(sc.nextLine());
                }

                //save the content into the file
                boolean isSaved =FileManager.addFiles(folderpath, fileName, content);


                if(isSaved)
                        System.out.println("File Created & Saved successfully");
                else
                        System.out.println("Error occured while Creating/Saving file.");
        }
```

Delete Files Method

```
/**
	 * Method to delete file from list
	 */
	public static void deleteFile()
	{
		//Variable Declaration
		String fileName;

		//Scanner object creation
		Scanner sc = new Scanner(System.in);

		//Read File Name to be deleted
		System.out.println("Enter file name to be deleted:");
		fileName=sc.nextLine();

		//Delete the File from Folder
		boolean isDeleted =FileManager.deleteFile(folderpath, fileName);

		if(isDeleted)
			System.out.println("File Deleted successfully");
		else
			System.out.println("File Not Found");
	}
```

Search File Method

```
	/**
	 * Method to search file
	 */
	public static void searchFile()
	{
		//Variable Declaration
		String fileName;

		//Scanner object creation
		Scanner sc = new Scanner(System.in);

		//Read File Name to be search
		System.out.println("Enter file name to be Search:");
		fileName=sc.nextLine();

		//Search the File from Folder
		boolean isExists =FileManager.searchFile(folderpath, fileName);

		if(isExists)
			System.out.println("File Found successfully");
		else
			System.out.println("File Not Found");
	}


}// this closes Main Class
```

FileManager.java

```
package lockedme;

import java.io.File;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class FileManager
{
```

Get All Files Name Method

```
/**
     * This method will return file names list from the folder
     * @param folderpath
     * @return
     */
    public static List<String> getAllFileNames(String folderpath)
    {
        //File Object Creation
        File fl = new File(folderpath);

        //Getting all the files into FileArray
        File[] listofFiles = fl.listFiles();

        //List Declaration to store file names
        List<String> fileNames = new ArrayList<String>();

        //ForEach loop to add file names in Array List
        for(File f:listofFiles)
            fileNames.add(f.getName());

        // Sorting ArrayList in ascending Order
  // using Collection.sort() method
  Collections.sort(fileNames);

        //Return the List
        return fileNames;
    }
```

Add Files Method

```
    /**
     * This method will create file & write content in the file
     * @param folderpath
     * @param fileName
     * @param Content
     * @return
     */
    public static boolean addFiles(String folderpath,String fileName,List<String> Content)
    {
        try
        {
            //File Object Creation
            File fl = new File(folderpath, fileName);

            //File Writer object Creation
            FileWriter fw = new FileWriter(fl);

            //Write into file
            for(String c:Content)
            {
                fw.write(c+"\n");
            }

            //Close File Writer Object
            fw.close();

            return true;
        }
        catch (Exception Ex)
        {
            return false;
        }
    }
```

Delete Files Method

```
    /**
     * This method will delete the file from folder
     * @param folderpath
     * @param fileName
     * @return
     */
    public static boolean deleteFile(String folderpath,String fileName)
    {
        //File Object Creation with folder path & file name
        File fl = new File(folderpath+"\\"+fileName);

        try
        {
            if(fl.delete())
                return true;
            else
                return false;
        }
        catch(Exception Ex)
        {
            return false;
        }
    }
```

Search File Method

```
    /**
     * This Method will search specific file in folder
     * @param folderpath
     * @param fileName
     * @return
     */
    public static boolean searchFile(String folderpath,String fileName)
    {
        //File Object Creation with folder path & file name
        File fl = new File(folderpath+"\\"+fileName);

        try
        {
            if(fl.exists())

                return true;
            else
                return false;
        }
        catch(Exception Ex)
        {
            return false;
        }
    }
}// this closes file manager class
```