

# Virtual Key Repositories Project

Sachin Davis Mundassery

Source Code:

## 1. VirtualKeyApplicationMain.java

```
package virtualkey;

public class VirtualKeyApplicationMain {

    public static void main(String[] args) {

        WelcomeScreen welcome = new WelcomeScreen();
        welcome.introWelcomeScreen();
        welcome.GetUserInput();
    }
}
```

## 2. WelcomeScreen.java

```
package virtualkey;

import java.util.InputMismatchException;
import javax.swing.JOptionPane;

public class WelcomeScreen implements Screen{
    private String intro = "Welcome to LockedMe.com!";
    private String developerName = "Developer: Sachin Davis Mundassery";

    private String[] options = new String[3];

    public WelcomeScreen() {

        options[0] = "Display Files";
        options[1] = "Show File Options Menu";
        options[2] = "Quit";

    }

    public void introWelcomeScreen() {
        JOptionPane.showMessageDialog(null, intro + "\n\n" + developerName, "Welcome",
        JOptionPane.INFORMATION_MESSAGE);
    }
}
```

```

public void GetUserInput() {
    int selectedOption = 0;

    // iterate until the user quits.
    while ((selectedOption = this.getOption()) != 2) {
        this.NavigateOption(selectedOption);
    }
}

@Override
public void NavigateOption(int option) {
    switch(option) {

        case 0: // Show Files in Directory
            this.ShowFiles();
            break;

        case 1: // Show File Options menu

            ScreenService.setCurrentScreen(ScreenService.FileOptionsScreen);
            ScreenService.getCurrentScreen().GetUserInput();
            break;

        default:
            JOptionPane.showMessageDialog(null, "Invalid Option", "",
JOptionPane.INFORMATION_MESSAGE);
            System.out.println("Invalid Option");
            break;
    }
}

// to display all the files in the directory
public void ShowFiles() {
    DirectoryService.PrintFiles();
}

private int getOption() {
    int userChoiceFromMainOptions = 0;

    // exception to handle invalid input from the user
    try {
        userChoiceFromMainOptions = JOptionPane.showOptionDialog(null, "What would you like
to do?", "LockedMe", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE,null, options,
options[0]);
    }
}

```

```

        catch (InputMismatchException ex) {

        }
        return userChoiceFromMainOptions;

    }
}

```

### 3. Screen.java

```

package virtualkey;

public interface Screen {
    public void NavigateOption(int option);
    public void GetUserInput();
}

```

### 4. ScreenService.java

```

package virtualkey;

public class ScreenService {

    public static WelcomeScreen WelcomeScreen = new WelcomeScreen();
    public static FileOptionsScreen FileOptionsScreen = new FileOptionsScreen();

    public static Screen CurrentScreen = WelcomeScreen;

    public static Screen getCurrentScreen() {
        return CurrentScreen;
    }

    public static void setCurrentScreen(Screen currentScreen) {
        CurrentScreen = currentScreen;
    }
}

```

### 5. FileOptionScreen.java

```

package virtualkey;

import java.io.File;

```

```

import java.io.IOException;
import java.nio.file.FileSystems;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.InputMismatchException;
import java.util.Scanner;

import javax.swing.JOptionPane;

public class FileOptionsScreen implements Screen {

    private Directory dir = new Directory();

    private ArrayList<String> options = new ArrayList<>();

    public FileOptionsScreen() {

        options.add("1. Add a File");
        options.add("2. Delete A File");
        options.add("3. Search A File");
        options.add("4. Return to Menu");

    }

    public void GetUserInput() {
        int selectedOption;
        while ((selectedOption = this.getOption()) != 3) {
            this.NavigateOption(selectedOption);
        }
    }

    @Override
    public void NavigateOption(int option) {

        switch(option) {

            case 0: // Add File
                this.AddFile();
                break;
            case 1: // Delete File
                this.DeleteFile();
                break;
            case 2: // Search File
                this.SearchFile();

```

```

        break;
    default:
        System.out.println("Invalid Option");
        break;
    }
}

public void AddFile() {

    String fileName = this.getInputString();

    try {
        //          Path path = FileSystems.getDefault().getPath(Directory.fileLocation +
        fileName).toAbsolutePath(); // to get the path of the adding file
        File file = new File(dir.getName() + fileName);

        if (file.createNewFile()) {

            JOptionPane.showMessageDialog(null,"You have added a file
            named: " + file.getName() , "Add File", JOptionPane.INFORMATION_MESSAGE);
            dir.GetFiles().add(file);

        } else {
            JOptionPane.showMessageDialog(null, file.getName() + " already
            exists" , "Add File", JOptionPane.INFORMATION_MESSAGE);
        }
    } catch (IOException e){
        System.out.println(e);
    }
}

public void DeleteFile() {

    System.out.println("Please Enter the Filename:");

    String fileName = this.getInputString();

```

```

        Path path = FileSystems.getDefault().getPath(Directory.fileLocation +
fileName).toAbsolutePath();
        File file = pathToFile();
        if (file.delete()) {
//            System.out.println("Deleted File: " + file.getName());
            JOptionPane.showMessageDialog(null,"You have deleted the file : " +
fileName , "Delete File", JOptionPane.INFORMATION_MESSAGE);

            dir.GetFiles().remove(file);
        } else {
            JOptionPane.showMessageDialog(null,"Failed to delete the file : " +
fileName + " - NOT FOUND" , "Delete File", JOptionPane.INFORMATION_MESSAGE);

        }
    }

    public void SearchFile() {

        Boolean found = false;

//        System.out.println("Please Enter the Filename:");

        String fileName = this.getInputString();

//        System.out.println("You are searching for a file named: " + fileName);

        ArrayList<File> files = dir.GetFiles();

        for(int i = 0; i < files.size(); i++) {
            if(files.get(i).getName().equals(fileName)) {
                JOptionPane.showMessageDialog(null,"You have searched for
a file named: " + fileName + " and it is FOUND" , "Search File",
JOptionPane.INFORMATION_MESSAGE);
//                System.out.println("Found");
                found = true;
            }
        }
        if (found == false) {
            JOptionPane.showMessageDialog(null,"You have searched for a file named: " +
fileName + " and it is NOT FOUND" , "Search File", JOptionPane.INFORMATION_MESSAGE);
//            System.out.println("Not Found");

```

```

    }
}

private String getInputString() {
    String inputString = JOptionPane.showInputDialog(null, "Enter file name ", "",
JOptionPane.INFORMATION_MESSAGE);

    return(inputString);

}

private int getOption() {
    Scanner in = new Scanner(System.in);

    int returnOption = 0;
    try {

        Object[] result = options.toArray();
        returnOption = JOptionPane.showOptionDialog(null, "What would you like to do?",
"LockedMe", JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE,null, result,
result[0]);
    }
    catch (InputMismatchException ex) {
        JOptionPane.showMessageDialog(null,"Invalid Input" , "",
JOptionPane.INFORMATION_MESSAGE);
//        System.out.println("Invalid input");
    }
    return returnOption;

}

}

```

## 6. Directory.java

```
package virtualkey;
```

```
import java.util.ArrayList;
```

```
import java.nio.file.Path;
import java.nio.file.FileSystems;
import java.io.File;
```

```

import java.util.Collections;

public class Directory {

    public static final String fileLocation =
"D:\\JavaProjects\\virtualKeyProject\\LockedMe\\files\\";

    private ArrayList<File> files = new ArrayList<File>();

    Path path = FileSystems.getDefault().getPath(fileLocation).toAbsolutePath();

    File Dfiles = path.toFile();

    public String getName() {
        return fileLocation;
    }

    public void print() {
        System.out.println("Existing Files: ");
        files.forEach(f -> System.out.println(f));
    }

    public ArrayList<File> fillFiles() {

        File[] directoryFiles = Dfiles.listFiles();

        files.clear();
        for (int i = 0; i < directoryFiles.length; i++) {
            if (directoryFiles[i].isFile()) {
                files.add(directoryFiles[i]);
            }
        }

        Collections.sort(files);

        return files;
    }

    public ArrayList<File> getFiles() {

```



```

        fillFiles();
        return files;
    }

}

```

## 7. DirectoryService.java

```

package virtualkey;

import java.io.File;
import java.util.ArrayList;

import javax.swing.JList;
import javax.swing.JOptionPane;

public class DirectoryService {

    private static Directory fileDirectory = new Directory();

    public static void PrintFiles() {

        fileDirectory.fillFiles();

        ArrayList<String> fileArray = new ArrayList<String>();
        int fileNumber = 1;
        for (File file : DirectoryService.getFileDirectory().getFiles())
        {
            fileArray.add(fileNumber + " : "+file.getName()+"\n");
            fileNumber+=1;
        }

        //      System.out.println(file.getName()+"^^^");
        Object[] result = fileArray.toArray();
        JOptionPane.showMessageDialog(null, result);
    }

    public static Directory getFileDirectory() {
        return fileDirectory;
    }

    public static void setFileDirectory(Directory fileDirectory) {
        DirectoryService.fileDirectory = fileDirectory;
    }

}

```