Student: Abilio Junior

SimpliLearn Project LockedMe.com – Virtual Key Repositories

Contents

[Sprint Planning 1](#_Toc125970814)

[Java concepts used in the project 2](#_Toc125970815)

[Flow Diagram 2](#_Toc125970816)

[GIT Repository 2](#_Toc125970817)

[Application Development Process: 3](#_Toc125970818)

[Class: Menu 4](#_Toc125970819)

[Method 1: getInput() 4](#_Toc125970820)

[Method 2: info() 5](#_Toc125970821)

[Method 3, 4: displayMainMenu(), mainMenuSwitch(int input) 5](#_Toc125970822)

[Method 5, 6: operationsMenu(), operationsSwitch(int input) 6](#_Toc125970823)

[Class: FileOperations 6](#_Toc125970824)

[Method 1: listFiles() 7](#_Toc125970825)

[Method 2: addFile() 7](#_Toc125970826)

[Method 3: removeFile() 8](#_Toc125970827)

[Method 4: searchFile() 9](#_Toc125970828)

[Class: LockedMeMain 9](#_Toc125970829)

[Conclusion 10](#_Toc125970830)

# Sprint Planning

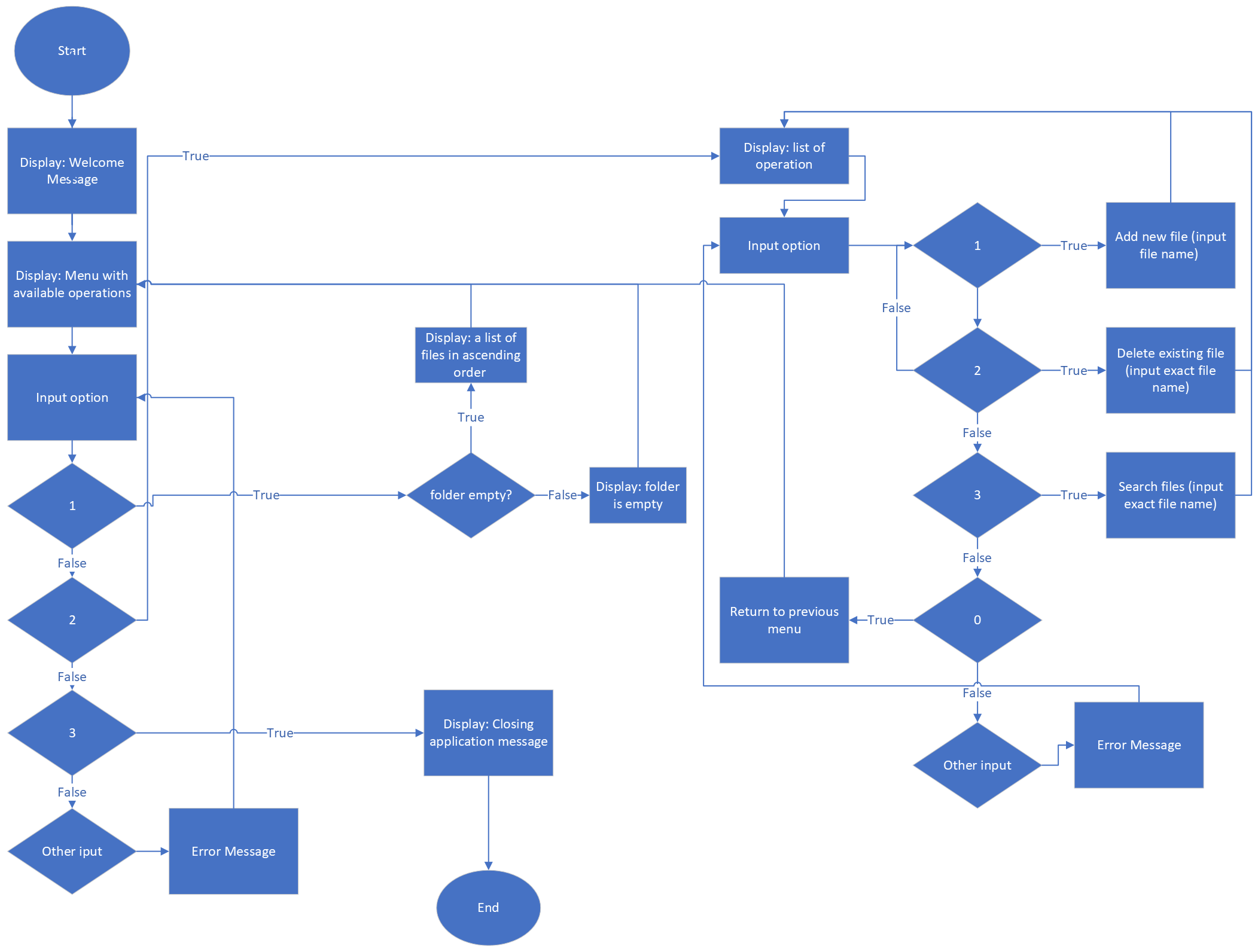
The project was divided in two sprints.

1. Understand the project design the flow of the application according to the requirements
2. Initialize Git and Connect GitHub, Develop the application and document the application.

## Java concepts used in the project

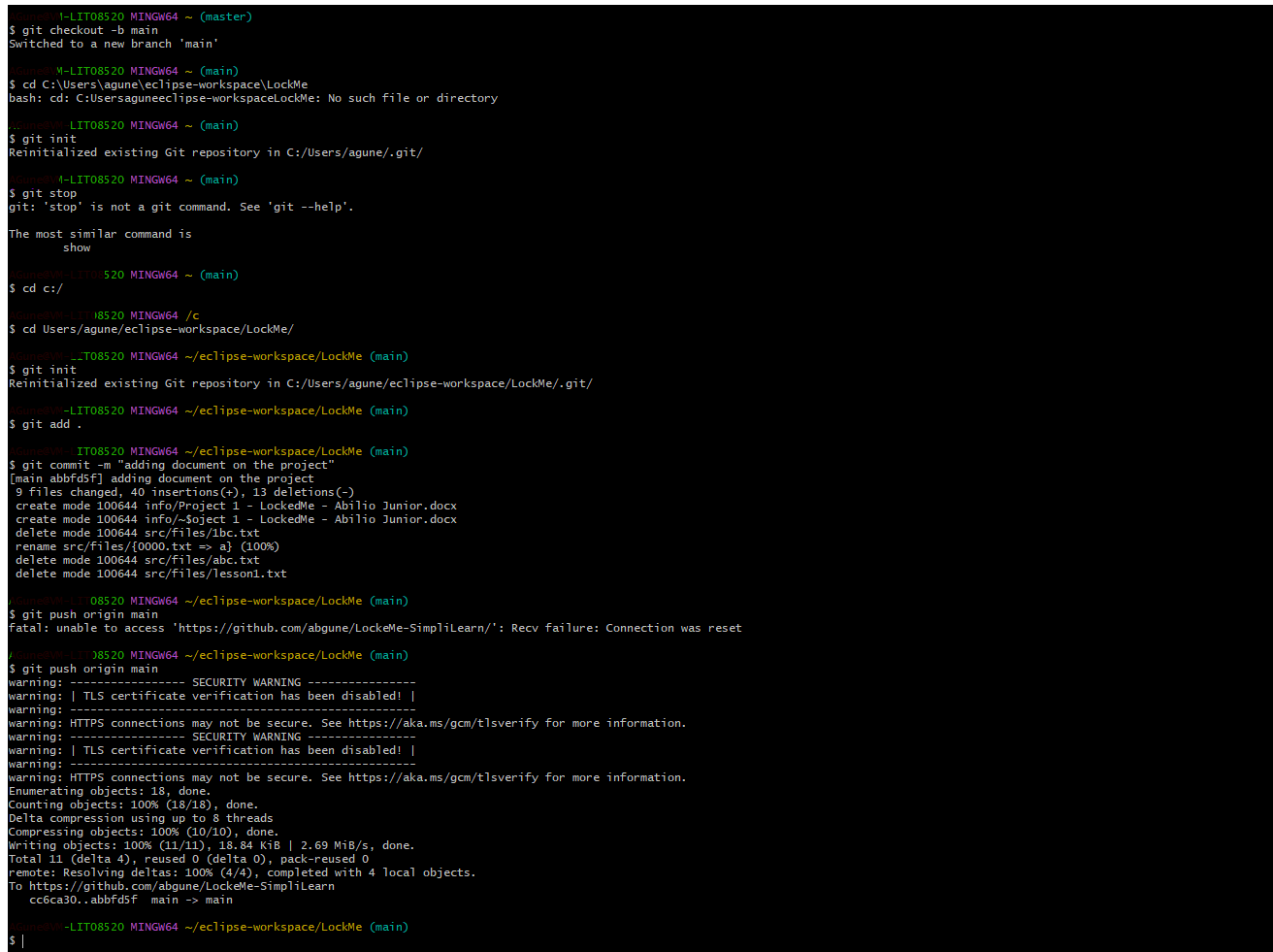
* Collections Framework (List)
* Exception Handling
* File Handling

# Flow Diagram



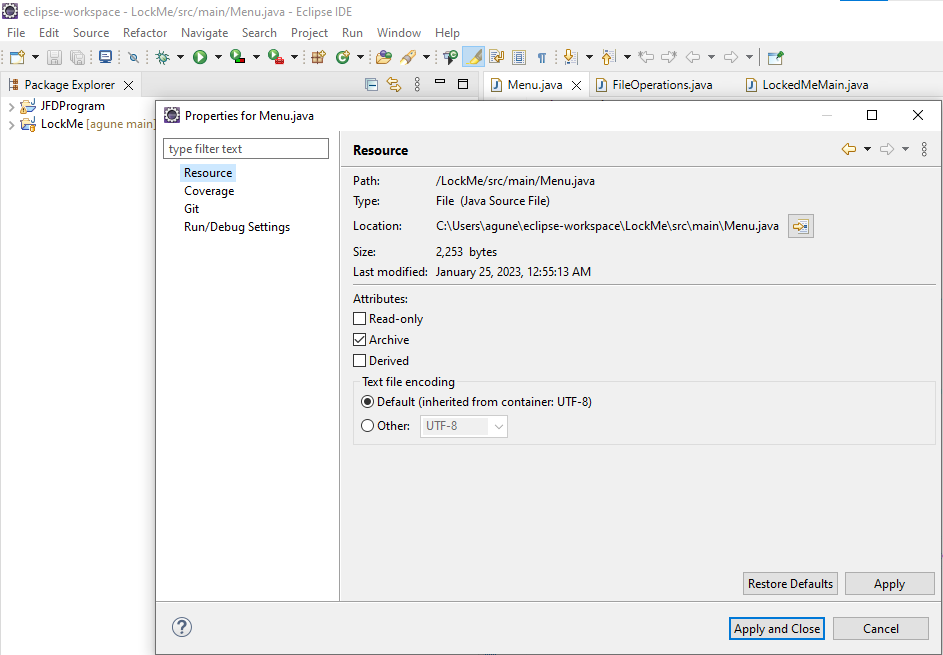
# GIT Repository

GIT - <https://github.com/abgune/LockeMe-SimpliLearn>



# Application Development Process:

Application was developed using eclipse and JDK version 11

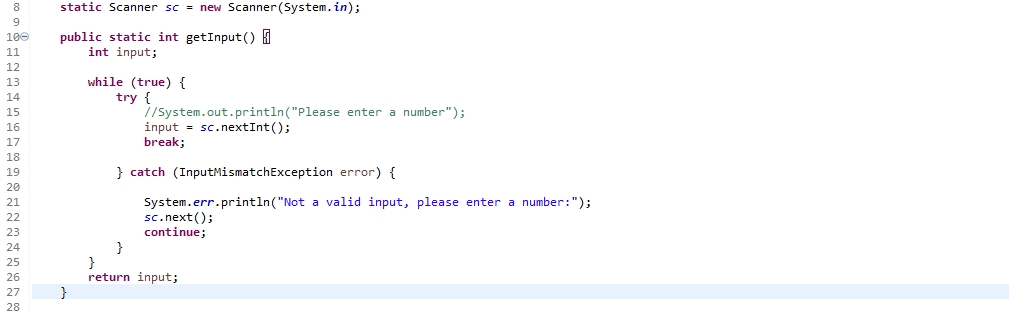


**The project was implemented in 3 classes:**

# Class: Menu

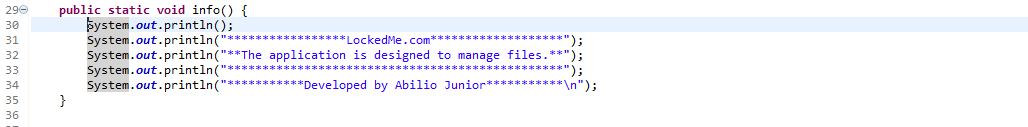
## Method 1: getInput()

Responsible to receive a correct integer input to be used on the menus switch.

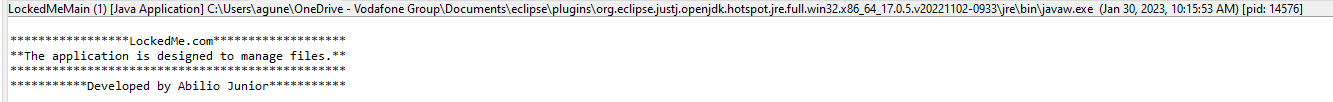


## Method 2: info()

Displays the information of the application and the developer name

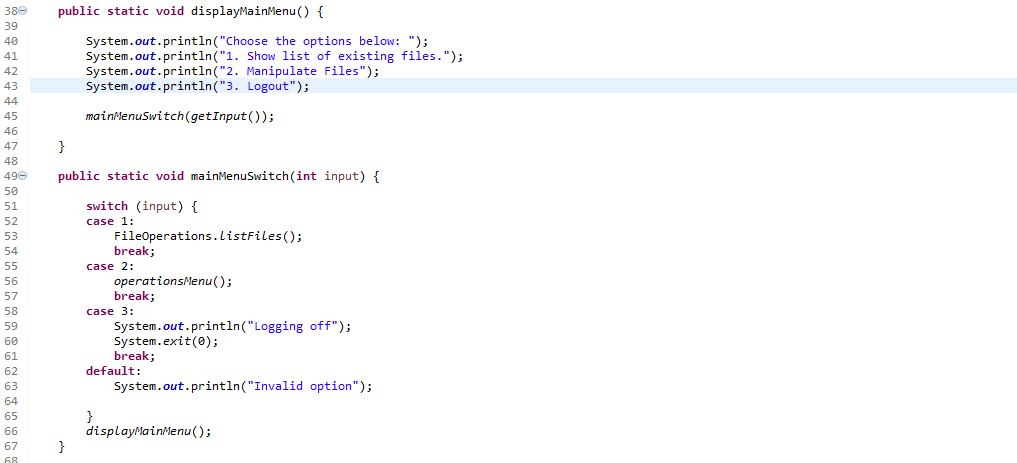


Output:



## Method 3, 4: displayMainMenu(), mainMenuSwitch(int input)

Displays the information on the initial menu, and the method 3 is the switch which helps navigate on the menu options

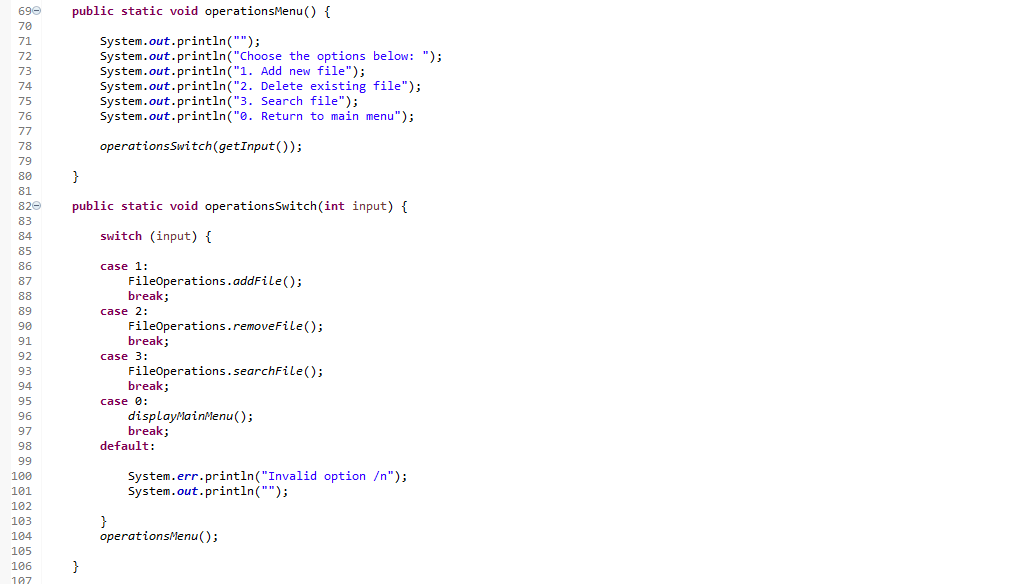


Output:



## Method 5, 6: operationsMenu(), operationsSwitch(int input)

Displays the second menu for operations with the fille, and the method 6 is the switch which helps navigate on the menu options



Output:

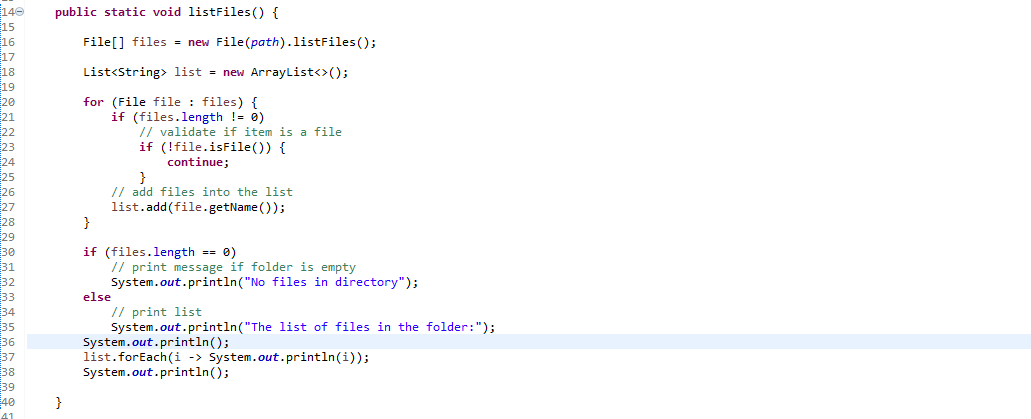


# Class: FileOperations

Defined a static path that the files will be searched, created and deleted from.

**static** String *path* = "src/files/";

## Method 1: listFiles()



Output:

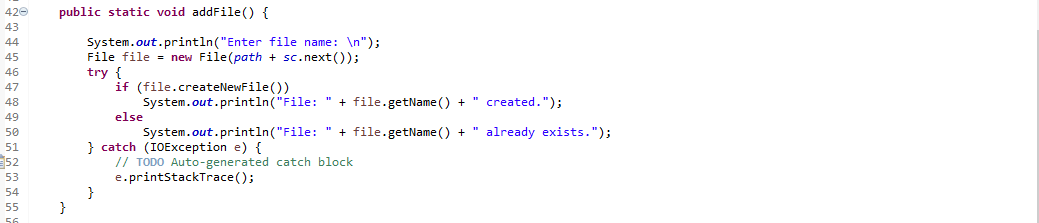
When not empty:



When empty:



## Method 2: addFile()

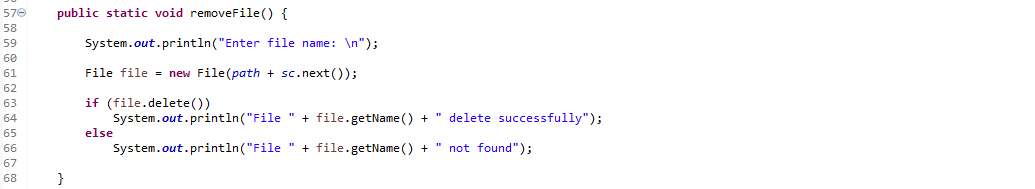




When filename already exists on the folder:

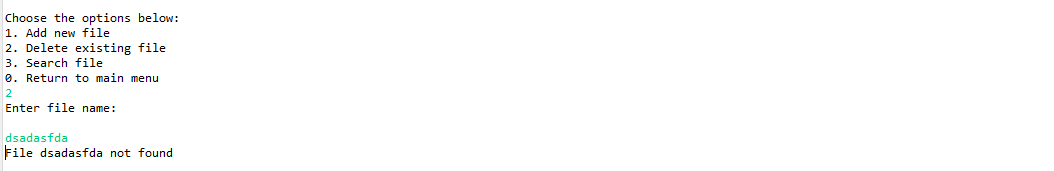


## Method 3: removeFile()



Output:

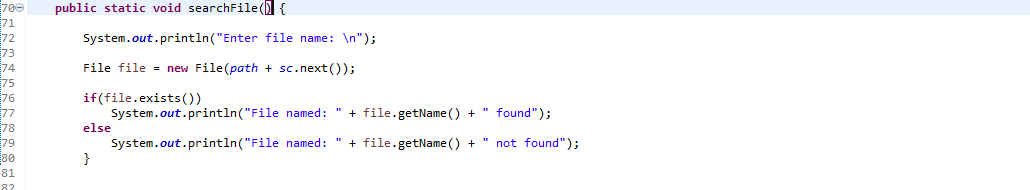
When name not matched



When name is matched



## Method 4: searchFile()



When file name is not matched



When file name is matched



# Class: LockedMeMain

Main class to execute the aplication



# Conclusion

The application is designed to allow customer to add, search and list all files in a predefined folder to not confuse him with how you name the path.

Files are listed in ascending order when displayed.

It can add any type of file without restrictions, it doesn’t allow to create an existing file as it might delete existing file content.

Doesn’t crash if the user inputs incorrect charters while navigating in the menu and it only closes the application if the user select the option to do it.

Enhancements:

Consider adding am option to the customer to input the folder in which he would like to do the operations of listing files, add, remove, or search.

Create user management class to give users access to which type of permissions they have on the folder or files.

Ability to list all files that the name match or are contained in the input of the user input.