FULL STACK JAVA DEVELOPER
(MASTERS PROGRAM)

PHASE 1 - OOPS USING JAVA DATA STRUCTURES

PROJECT WRITE-UP

TOPIC : VIRTUAL KEY REPOSITORY APPLICATION PROTOYPE

Submitted by

Ms. Suruthi Suresh

suruthi8695@gmail.com

Batch **SB014**

simplilearn

Get Certified. Get Ahead.

Phase 1 Overview

The main objectives of Phase 1 of the Simplilearn Full Stack Java Development program were

- to gain an understanding of core concepts of the Java Programming Language (abstraction, polymorphism, inheritance, and encapsulation),
- embrace the Eclipse Integrated Development Environment (IDE),
- understand the Agile software development life cycle, and
- gain familiarity with Java data structures for object-oriented applications.

Phase 1 ended with a culminating project to demonstrate application of the concepts. The purpose of this paper is to document the project in detail.

Problem Statement

Lockers Pvt. Ltd. aims to digitize their product catalog. For the first phase of the project, they wish to develop a prototype of the application. The prototype of the application will be then presented to the relevant stakeholders for the budget approval, with the goal of delivering a high-end quality product as early as possible.

Lockers Pvt. Ltd. would like a presentation on the following topics in the next 15 working days (3 weeks):

- Specification document - Product’s capabilities, appearance, and user interactions
- Number and duration of sprints required
- Setting up Git and GitHub account to store and track your enhancements of the prototype
- Java concepts being used in the project
- Data Structures where sorting and searching techniques are used
- Generic features and three operations:
    - Retrieving all the files in ascending order
    - Business-level operations:
        - Option to add a user specified file to the application
        - Option to delete a user specified file from the application
        - Option to search a user specified file from the application

- ✦ Navigation option to close the current execution context and return to the main context

- Option to close the application
-

Agile Project Management

This section will cover the project management details surrounding the software development life cycle for the virtual keys repository application prototype. Table below provides an overview of the project and high-level software project management milestones.

Project Overview	
Client	Lockers Pvt. Ltd.
Consultant	Suruthi Suresh Full Stack Java Developer
Application Name	LockedMe.com
Application Phase	Prototype
Phase 1 Project Deliverable	Console-based virtual keys repository Application intended for Budget Approval
Planning Project Management	
Project Duration	3 Weeks (15 working days)
Number of Sprints	3
Planned Sprint Duration	1 Week (5 working days)
Total Number of Product Backlog Items	8

Application User Roles

The Table below captures the roles for targeted software for the LockedMe.com virtual keys repository application prototype.

Role	**Description**	**Software Version**

General User	The General User will use the prototype application for file handling.	Initial Release
Admin	The Admin will use the protype application for maintaining users and setting user directory & file permissions.	Future Release

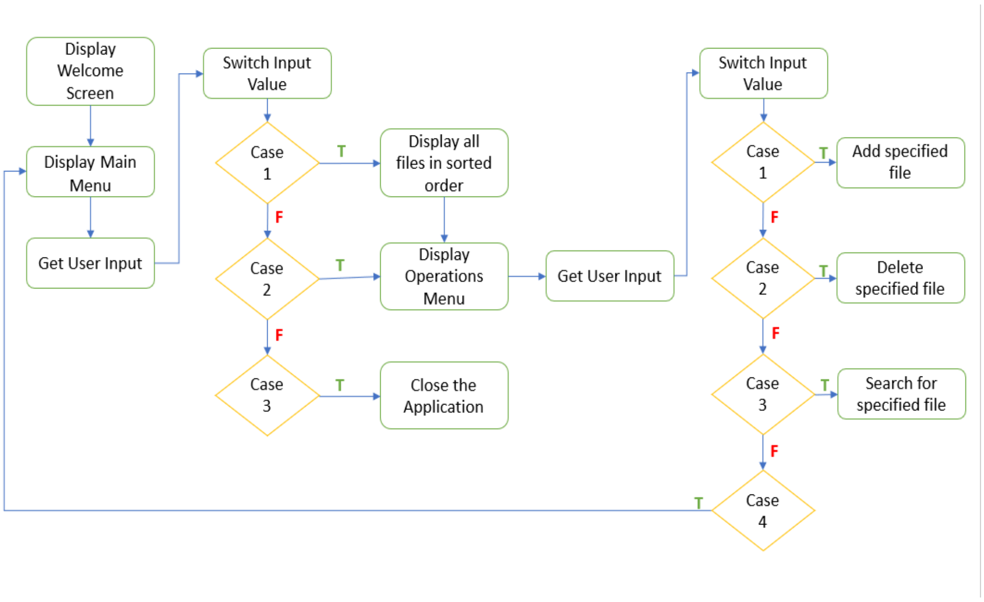
Project Planning Details

The following Table captures the project planning management overview for the virtual keys repository application prototype.

Sprint Number	Sprint Duration	Product Backlog Items
1	1 Week	Welcome Screen
		Main Menu Options List
		Business Level Menu Options List
		Business Level Option to go back to main menu
2	1 Week	File Retrieval Options & Sorting Implementation
		File Addition Implementation
3	1 Week	File Deletion Implementation
		File Search Implementation

Flow Of the Application

The following figure shows the flowchart of the Application and explains in detail the processes involved.



Moving the Application into Git Hub

An account was created in GitHub and a new repository for the filed connecting to the application were moved into it. The Git link is as follows:

https://github.com/suruthiram/Phase1Project

Java Concepts Involved

The Java code was written inside a package created called “package2”. The class name was “Lab1”.

The Java built-in packages used were as follows.

java.io.File;

java.io.IOException;

java.util.Arrays;

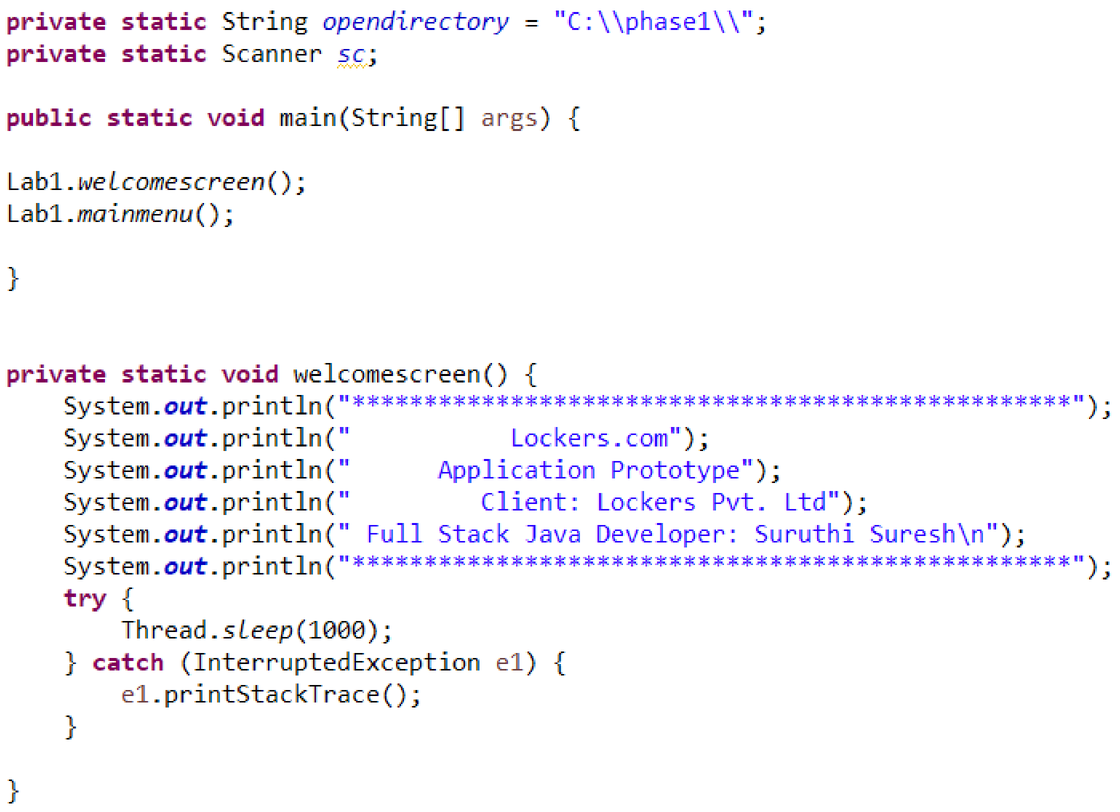
java.util.InputMismatchException;

java.util.Scanner;

The main() method was used to execute the program.

All the other methods used in the code were declared as private and static so there is no need to create an object separately to execute it.

Following are the code screenshots.

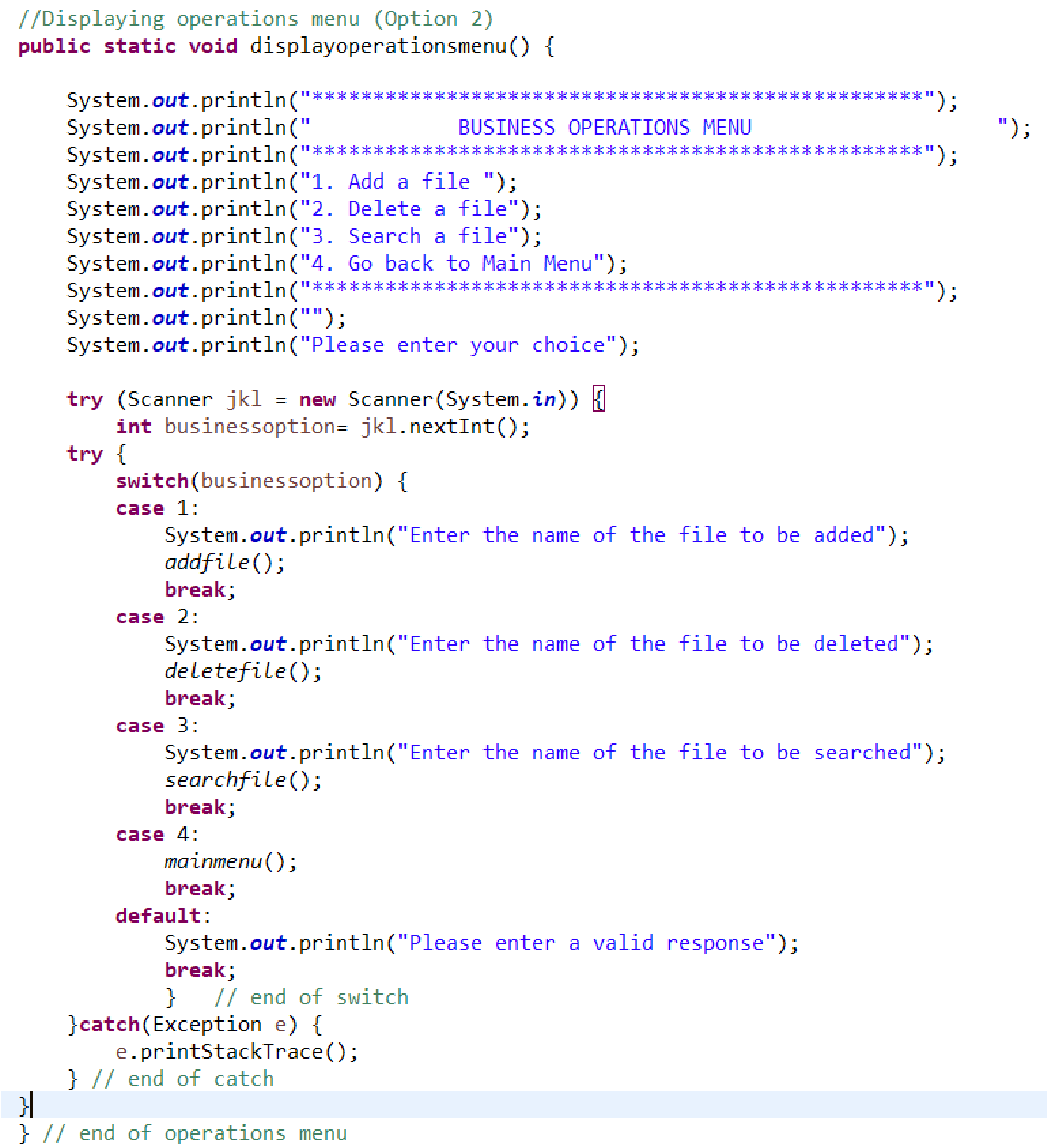


```
private static String opendirectory = "C:\\phase1\\";
private static Scanner sc;

public static void main(String[] args) {

Lab1.welcomescreen();
Lab1.mainmenu();

}


private static void welcomescreen() {
    System.out.println("**************************************************");
    System.out.println("           Lockers.com");
    System.out.println("      Application Prototype");
    System.out.println("         Client: Lockers Pvt. Ltd");
    System.out.println(" Full Stack Java Developer: Suruthi Suresh\n");
    System.out.println("**************************************************");
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e1) {
        e1.printStackTrace();
    }

}
```

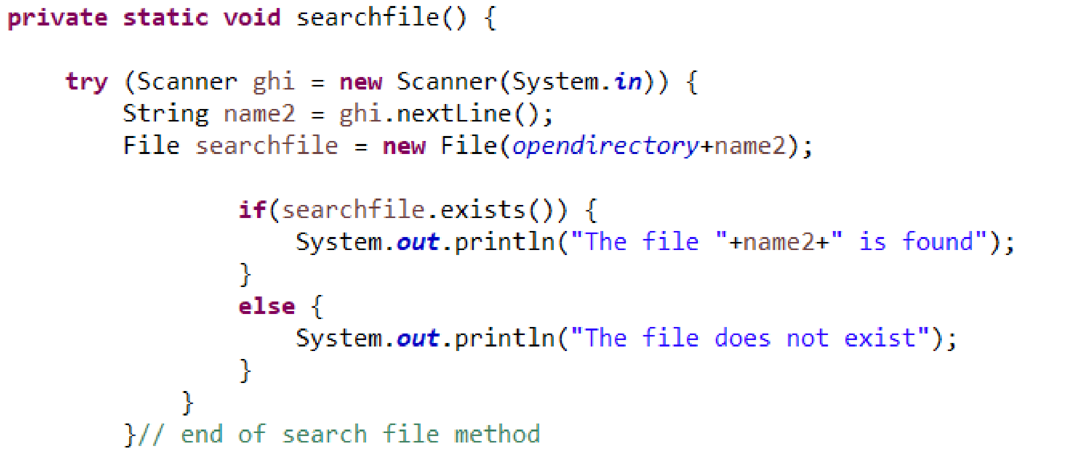
```java
private static void mainmenu() { // start of MAIN MENU method

    System.out.println("**************************************************");
    System.out.println("                MAIN MENU                     ");
    System.out.println("**************************************************");
    System.out.println("1. Display the available files in Ascending order ");
    System.out.println("2. Perform Business level operations");
    System.out.println("3. Close the application");
    System.out.println("**************************************************");
    System.out.println("");
    System.out.println("Please enter your choice");

    try (Scanner xyz = new Scanner(System.in)) {
        int mainmenuchoice = xyz.nextInt();


            try {

            switch(mainmenuchoice)
            {
            case 1:
                System.out.println("Displaying all files from "+ opendirectory);
                displayallfiles();
                break;
            case 2:
                displayoperationsmenu();
                break;
            case 3:
                System.out.println("Closing the Application");
                System.out.println("See you next time");
                break;

            }//end of switch

            }catch(Exception e) {
                System.out.println("Please enter a valid response");
                e.printStackTrace();
            }
    }
}// end of main menu method
```

```java
//Displaying files in Ascending order (Option 1)
public static void displayallfiles(){

    File directory = new File(opendirectory);
    String files[] = directory.list();

    Arrays.sort(files);
    int numberoffiles = files.length;

    for(int k=0;k<files.length;k++) {
        System.out.println(files[k]);

    }

    System.out.println("There are " +numberoffiles+ " files in the directory.");

}//end of display all files method



private static void deletefile() {

    try (Scanner def = new Scanner(System.in)) {
        String name1 = def.nextLine();
        File deletefile = new File(opendirectory+name1);

            if(deletefile.exists()) {
                deletefile.delete();
                System.out.println("The file "+name1+" has been deleted");
            }
            else {
                System.out.println("This file does not exist");
            }
         }
     } //end of delete file method
```

```
private static void addfile() {

    try (Scanner abc = new Scanner(System.in)) {
        String name = abc.nextLine();
        File addfile = new File(opendirectory+name);
        try {
            boolean isnewfilecreated = addfile.createNewFile();
            if(isnewfilecreated) {
                System.out.println("The new file is added");
            }
            else {
                System.out.println("The file already exists");
            }
        } catch (IOException e) {

            e.printStackTrace();
            System.out.println("That is an ivalid file name");
        }
    }
} //end of add file method
```



```
//Displaying operations menu (Option 2)
public static void displayoperationsmenu() {

    System.out.println("***************************************************");
    System.out.println("            BUSINESS OPERATIONS MENU                    ");
    System.out.println("***************************************************");
    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. Go back to Main Menu");
    System.out.println("***************************************************");
    System.out.println("");
    System.out.println("Please enter your choice");

    try (Scanner jkl = new Scanner(System.in)) {
        int businessoption= jkl.nextInt();
    try {
        switch(businessoption) {
        case 1:
            System.out.println("Enter the name of the file to be added");
            addfile();
            break;
        case 2:
            System.out.println("Enter the name of the file to be deleted");
            deletefile();
            break;
        case 3:
            System.out.println("Enter the name of the file to be searched");
            searchfile();
            break;
        case 4:
            mainmenu();
            break;
        default:
            System.out.println("Please enter a valid response");
            break;
            }   // end of switch
    }catch(Exception e) {
        e.printStackTrace();
    } // end of catch
}
} // end of operations menu
```



```
private static void searchfile() {

    try (Scanner ghi = new Scanner(System.in)) {
        String name2 = ghi.nextLine();
        File searchfile = new File(opendirectory+name2);

                if(searchfile.exists()) {
                    System.out.println("The file "+name2+" is found");
                }
                else {
                    System.out.println("The file does not exist");
                }
            }
        }// end of search file method
```