SIMPLILEARN PROJECT: CODE

Alexandra Arseni, Vodafone

alexandra.arseni@vodafone.com

Contents

Personal Information

Name: Alexandra Arseni

Email: alexandra.arseni@vodafone.com

Github URL: alexarseni/PhaseEndProject (github.com)

About this document

This document contains the code of the application. No documentation or execution screenshots are presented in this document. For the screenshot document please find the "Alexandra-Arseni-Screenshots" document. For the documentation please find the "Alexandra-Arseni-Documentation" document.

Note: the code has been tested in the provided lab's environment. The code has not been tested in a windows environment.

The app's architecture

The code is organized in four different packages:

- virtualkey.main, which contains the main method of the application
- virtualkey.util, which contains the mainMenu and subMenu user interfaces
- virtualkey.service, which contains all the business methods of the application

The code

VirtualKeyApp

```
package virtualkey.main;
import virtualkey.util.*;

public class VirtualKeyApp {

    public static void welcome() {
        System.out.println("Welcome to my application");
        System.out.println("Application name: VirtualKeyApp");
        System.out.println("Developer: Arseni Alexandra");
        System.out.println("Email: alexandra.arseni@vodafone.com");
        System.out.println("Github URL:
https://github.com/alexarseni/PhaseEndProject");
        System.out.println("----------------------------------------
----------------\n");
    }

    public static void main(String[] args) {

        welcome(); //prints the Welcome Screen
        MainMenu.mainMenu();     //calls the main menu method
        System.out.println("Bye Bye");

    }

}
```

MainMenu

```
package virtualkey.util;
import java.util.InputMismatchException;
import java.util.Scanner;
import virtualkey.service.VirtualKeyService;

public class MainMenu {

    public static void mainMenu() {

        Scanner sc = new Scanner(System.in); //initialize the scanner
object to take input from keyboard
        boolean exit = false; //this variable will be used to exit
the do..while loop later on
        VirtualKeyService vks = new VirtualKeyService(); //create a
VirtualKeyService object to gain access to the business methods.

        //this method is optional. it creates two test files for
easier testing.
        vks.createSomeFiles();

        //prompt the user with his choices.
        System.out.println("Here are your choices:");
        System.out.println("1. Display all file names 2. Access
Business Methods 3. Exit the application\n");

        int input; //this variable will store the user's input

        do {
            System.out.println("Please enter your choice by writing a
number in the range 1-3");
            try {
            input = sc.nextInt(); //take value from keyboard
                switch(input) {
                    case 1:
                    String sortedFiles[] = vks.getFiles();
                    //we need to to iterate through the array of
strings to get each file name, only if the array is not empty.
                    if(sortedFiles.length!=0) {
                        for(String s:sortedFiles) {
                            System.out.println(s);
                        }
                        System.out.println();
                    }
                    else {
                        System.out.println("There are no files to
display\n");
                    }
                    break;

                    case 2:
                    //we call the submenu method and pass the
VirtualKeyService object as an argument
                    SubMenu.subMenu(vks);
                    break;

                    case 3: System.out.println("We will exit the
application");
```

```
                    exit = true; //change the variable's value to
true, to exit the while loop.
                    break;
                    default:
                    System.out.println("The number you typed is out
of range");
                }
            }
            catch(InputMismatchException e) {
                System.out.println("What you typed was not a number.
Try again\n");
            }
            sc.nextLine(); //read and discard anything else the user
may have inputed.
        }while(exit==false);
        sc.close(); //close the scanner object
    }
}
```

SubMenu

```
package virtualkey.util;
import java.util.InputMismatchException;
import java.util.Scanner;
import virtualkey.service.VirtualKeyService;

public class SubMenu {

    public static void subMenu(VirtualKeyService vks) {
        Scanner sc = new Scanner(System.in); //initialize the scanner
object to take input from keyboard
        boolean exit = false; //this variable will be used to exit
the while loop later on

        System.out.println("Here are the sub-menu choices:");
        System.out.println("1. Add a file 2. Delete a file 3. Search
for a file 4. Exit the submenu\n");

        int input; //these two variables will be used to save the
user's input.
        String filename;

        do {
            System.out.println("Please enter your choice by writing a
number in the range 1-4");
            try {
            input = sc.nextInt(); //take value from keyboard
            sc.nextLine(); //discard anything else the user may have
typed
                switch(input) {
                    case 1:
                    System.out.println("Please type the name of the
file you want to add");
                    filename = sc.next(); //take the name of the file
from user
                    System.out.println(vks.addFile(filename)); //call
the addFile business method and print the returned statement
                    System.out.println();
```

```
                    sc.nextLine(); //discard anything else the user
may have typed
                    break;
                    case 2: System.out.println("Please type the name
of the file you want to delete.");
                    filename = sc.next(); //take the name of the file
from user
                    System.out.println(vks.deleteFile(filename));
//call the deleteFile method and print the returned statement
                    System.out.println();
                    sc.nextLine(); //discard anything else the user
may have typed
                    break;
                    case 3: System.out.println("Please type the name
of the file you want to search for.");
                    filename = sc.next();//take the name of the file
from user
                    System.out.println(vks.searchFile(filename));
//call the searchFile method and print the returned statement
                    System.out.println();
                    sc.nextLine(); //discard anything else the user
may have typed
                    break;
                    case 4: System.out.println("Back to main
menu\n");
                    exit = true; //change the variable's value to
true to exit the do...while loop.
                    break;
                    default:
                    System.out.println("The number you typed is not
in the range 1-4. Try again\n");

                }
            }
            catch(InputMismatchException e) {
                System.out.println("What you typed was not a number.
Try again\n");
                sc.nextLine();
            }


        }while(exit==false);

        return;
    }

}
```

VirtualKeyService

```
package virtualkey.service;
import java.io.File;
import java.io.IOException;
import java.util.Arrays;

public class VirtualKeyService {
    File destination = new File("./VirtualKeyFiles");

    //The constructor will create the directory for our files
    public VirtualKeyService(){
```

```
        if(destination.exists() == false) {
            destination.mkdir();
        }
    }
    //We use this function to create some test files. Optional
    public void createSomeFiles() {
        File file1 = new File(destination+"/maria.txt");
        File file2 = new File(destination+"/alex.txt");

        try {
            file1.createNewFile();
            file2.createNewFile();
        }
        catch (IOException e){
            System.out.println("Files could not be created - IO
Exception");
        }
    }

    //This method will display the contents of the destination
directory
    public String[] getFiles() {

        String[] sortedFiles = destination.list(); //get the list of
file names in the destination directory

        if(sortedFiles.length!=0) { //if the returned array is not
empty, proceed to sort it.
            Arrays.sort(sortedFiles); //sort the names
        }
        return sortedFiles;
    }
    //The add method takes the user-specified name as an argument
    public String addFile(String filename){
        //create newFile object with the correct path.
        File newFile = new File(destination+"/"+filename);

        try{ //if the .createNewFile() command creates the file
successfully it returns true.
            if(newFile.createNewFile()) {
                return "File "+filename+" created successfully";
            }
            else {
                return "The file already exists";
            }
        }
        catch (IOException e){
            return "Files could not be created - IO Exception";
        }
    }
    //The searchFile takes the user-specified name as an argument.
    public String searchFile(String filename) {
        //create the correct object/path for the file
        File newFile = new File(destination+"/"+filename);
        //the .exists() method return true, if the file already
exists.
        if(newFile.exists()) {
            return "Found the file you are searching for";
        }
        else {
            return "The file you are searching for does not exist";
```

```
        }
    }
    //The deleteFile takes the user-specified name of the file as an
argument.
    public String deleteFile(String filename) {
        File newfile = new File(destination+"/"+filename);
        if (newfile.exists()) { //we first check whther the file
exists
            newfile.delete(); //if it does then we proceed to delete
it
            return "File deleted sucessfully";
        }
        else { //if it does not exists, we notify the user.
            return "The file you want to delete does not exist";
        }
    }
}
```