**ENUM:**

MainMenu:

package com.constants;  
  
public enum MainMenu {  
  
 *LIST*("1", "Display all files in directory"),  
 *SUBMENU*("2", "List the options"),  
 *EXIT*("3", "Exit");  
  
 String menuId;  
 String menuMessage;  
  
 MainMenu(String menuId, String menuMessage) {  
 this.menuId = menuId;  
 this.menuMessage = menuMessage;  
 }  
  
 public static void printMainMenu() {  
 for (MainMenu mainMenu : MainMenu.*values*()) {  
 System.*out*.println(mainMenu.menuId + " : " + mainMenu.menuMessage);  
 }  
 }  
  
 public boolean isExit(){  
 return menuId.equals(*EXIT*.menuId);  
 }  
  
 public static MainMenu from(String choice){  
 for(MainMenu mainMenu : MainMenu.*values*()){  
 if(mainMenu.menuId.equals(choice)){  
 return mainMenu;  
 }  
 }  
 System.*out*.println("Invalid Choice! Please, rerun the application! GOODBYE !");  
 return *EXIT*;  
 }  
}

SubMenu:

package com.constants;  
  
public enum SubMenu {  
 *ADD*("A", "Create New File"),  
 *DELETE*("B", "Delete File"),  
 *SEARCH*("C", "Search File"),  
 *RETURN\_TO\_MENU*("D", "Return to Main Menu");  
  
 String menuId;  
 String menuMessage;  
  
 SubMenu(String menuId, String menuMessage) {  
 this.menuId = menuId;  
 this.menuMessage = menuMessage;  
 }  
  
 public static void printSubMenu() {  
 for (SubMenu subMenu : SubMenu.*values*()) {  
 System.*out*.println("\t" + subMenu.menuId + " : " + subMenu.menuMessage);  
 }  
 }  
  
 public static SubMenu from(String choice){  
 for(SubMenu subMenu : SubMenu.*values*()){  
 if(subMenu.menuId.equals(choice)){  
 return subMenu;  
 }  
 }  
 System.*out*.println("Invalid Choice! You will redirected to main menu!");  
 return *RETURN\_TO\_MENU*;  
 }  
}

DOMAIN

Developer

package com.domain;  
  
public class Developer {  
 private String name;  
 private String surname;  
 private String email;  
  
 public Developer(String name, String surname, String email) {  
 this.name = name;  
 this.surname = surname;  
 this.email = email;  
 }  
  
 @Override  
 public String toString() {  
 return "Name: " + this.name + "\nSurname: " + this.surname + "\nemail: " + this.email;  
 }  
}

**OPERATIONS**

DeveloperOperations

package com.operations;  
  
import com.domain.Developer;  
  
public class DeveloperOperations {  
  
 public static void printDeveloperDetails() {  
 Developer developer = new Developer("Ozge", "Akkol", "email@domain.com");  
 System.*out*.println("-------------Developer Information-------------");  
 System.*out*.println(developer.toString());  
 System.*out*.println("-----------------------------------------------");  
 }  
}

MenuOperations

package com.operations;  
  
import com.constants.MainMenu;  
import com.constants.SubMenu;  
  
import static com.operations.DeveloperOperations.*printDeveloperDetails*;  
  
public class MenuOperations {  
  
 public static void printApplicationInfo() {  
 *printWelcomeMessage*();  
 *printDeveloperDetails*();  
 }  
  
 private static void printWelcomeMessage() {  
 System.*out*.println("\nWelcome to File Directory Project!\n");  
 }  
  
 public static void printMainMenu() {  
  
 System.*out*.println("Make a choice: ");  
 *printLineSeparator*();  
 MainMenu.*printMainMenu*();  
  
 }  
  
 public static void printSubMenu() {  
 *printLineSeparator*();  
 SubMenu.*printSubMenu*();  
 *printLineSeparator*();  
  
 }  
  
 public static void printLineSeparator() {  
 System.*out*.println("-----------------------------------------------");  
 }  
}

FileOperations

package com.operations;  
  
import java.io.File;  
import java.io.IOException;  
import java.util.Arrays;  
  
public class FileOperations {  
  
 public static void listAllFilesInDirectory(String pathName) {  
 File directory = new File(pathName);  
 File[] files = directory.listFiles();  
 if (files.length == 0) {  
 System.*out*.println("Empty directory!");  
 } else {  
 for (File file : files) {  
 System.*out*.println(file.getName());  
 }  
 }  
 }  
  
 public static void createFileInDirectory(String pathName, String fileNameToBeCreated) {  
 File newFile = new File(pathName, fileNameToBeCreated);  
 if (newFile.exists()) {  
 System.*out*.println("File is already exists!");  
 } else {  
 try {  
 newFile.createNewFile();  
 System.*out*.println("File created!");  
 } catch (IOException e) {  
 System.*out*.println("Exception occurred while creating file with name %s" + fileNameToBeCreated);  
 }  
 }  
 }  
  
 public static void deleteFileFromDirectory(String pathName, String fileName) {  
 if (*isFileAlreadyExist*(pathName, fileName)) {  
 File fileToBeDeleted = new File(pathName, fileName);  
 fileToBeDeleted.delete();  
 System.*out*.println("File deleted successfully!");  
 } else {  
 System.*out*.println("File not found!");  
 }  
 }  
  
 public static void searchFileInDirectory(String pathName, String fileName) {  
 if (!fileName.isEmpty()) {  
 if (*isFileAlreadyExist*(pathName, fileName)) {  
 System.*out*.println("File found in the directory! ");  
 } else {  
 System.*out*.println("There is no file named " + fileName);  
 }  
 } else {  
 System.*out*.println("Invalid file name! Please enter correct file name for search!");  
 }  
 }  
  
 private static boolean isFileAlreadyExist(String pathName, String fileName) {  
 File directory = new File(pathName);  
 File[] files = directory.listFiles();  
 return Arrays.*stream*(files).anyMatch(file -> file.getName().equals(fileName));  
 }  
}

ApplicationClass

package com;  
  
import com.constants.MainMenu;  
import com.constants.SubMenu;  
  
import java.util.Locale;  
import java.util.Scanner;  
  
import static com.operations.FileOperations.\*;  
import static com.operations.MenuOperations.\*;  
  
public class FileOperationsApp {  
  
 private static String *PATH* = "Phase1/fileDirectory";  
  
 public static void main(String[] args) {  
 *printApplicationInfo*();  
  
 *printMainMenu*();  
  
 Scanner sc = new Scanner(System.*in*);  
 String choice = sc.next();  
 MainMenu mainMenuChoice = MainMenu.*from*(choice);  
  
 while (!mainMenuChoice.isExit()) {  
 switch (mainMenuChoice) {  
 case *LIST*: {  
 *listAllFilesInDirectory*(*PATH*);  
 *printMainMenu*();  
 mainMenuChoice = MainMenu.*from*(sc.next());  
 break;  
 }  
 case *SUBMENU*: {  
 *printSubMenu*();  
 String subChoice = sc.next();  
 SubMenu subMenuChoice = SubMenu.*from*(subChoice.toUpperCase(Locale.*ROOT*));  
  
 switch (subMenuChoice) {  
 case *ADD*: {  
 System.*out*.println("Enter new file name:");  
 String fileNameToBeCreated = sc.next();  
 if (fileNameToBeCreated.isEmpty()) {  
 System.*out*.println("Please enter correct file name: ");  
 fileNameToBeCreated = sc.next();  
 }  
  
 *createFileInDirectory*(*PATH*, fileNameToBeCreated);  
 break;  
 }  
 case *DELETE*: {  
 System.*out*.println("Enter the name of the file to be deleted:");  
 String fileName = sc.next();  
 *deleteFileFromDirectory*(*PATH*, fileName);  
 break;  
 }  
 case *SEARCH*: {  
 System.*out*.println("Enter the file name to be searched: ");  
 String fileName = sc.next();  
 *searchFileInDirectory*(*PATH*, fileName);  
 break;  
 }  
 case *RETURN\_TO\_MENU*: {  
 *printMainMenu*();  
 mainMenuChoice = MainMenu.*from*(sc.next());  
 break;  
 }  
 }  
 break;  
 }  
 default:  
 System.*out*.println("Please enter correct option!");  
 mainMenuChoice = MainMenu.*from*(sc.next());  
 break;  
 }  
 }  
  
 }  
  
}