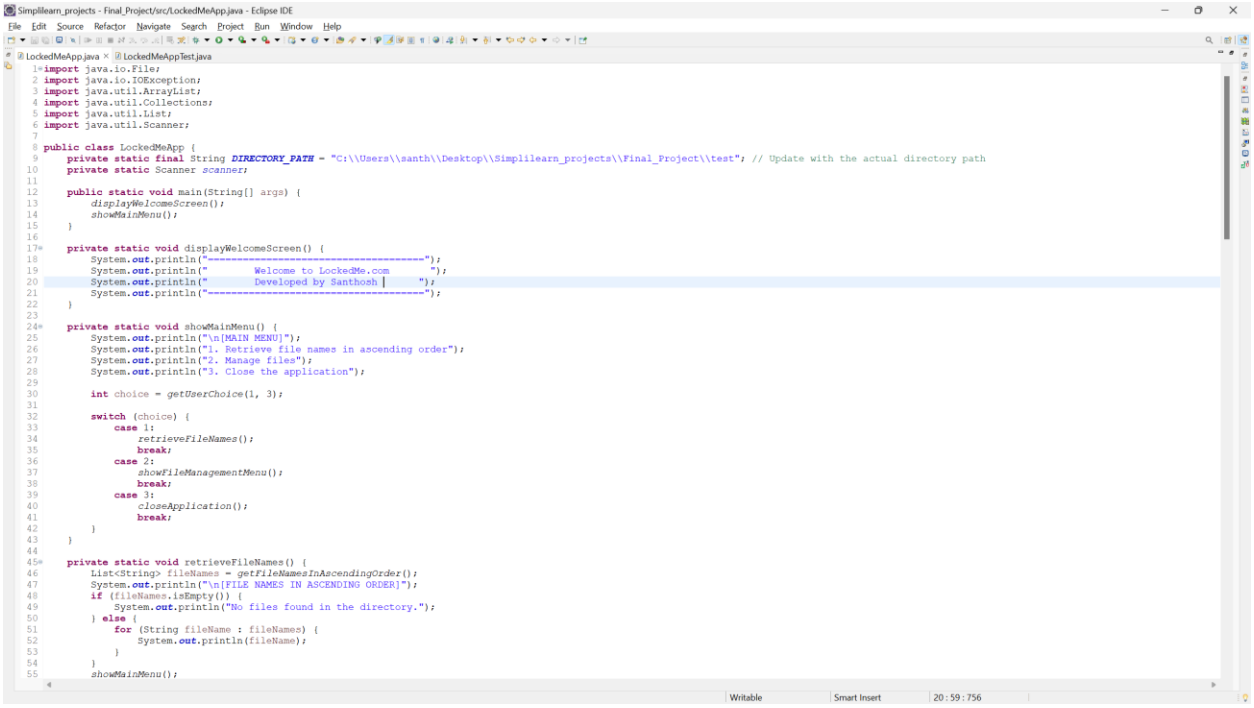
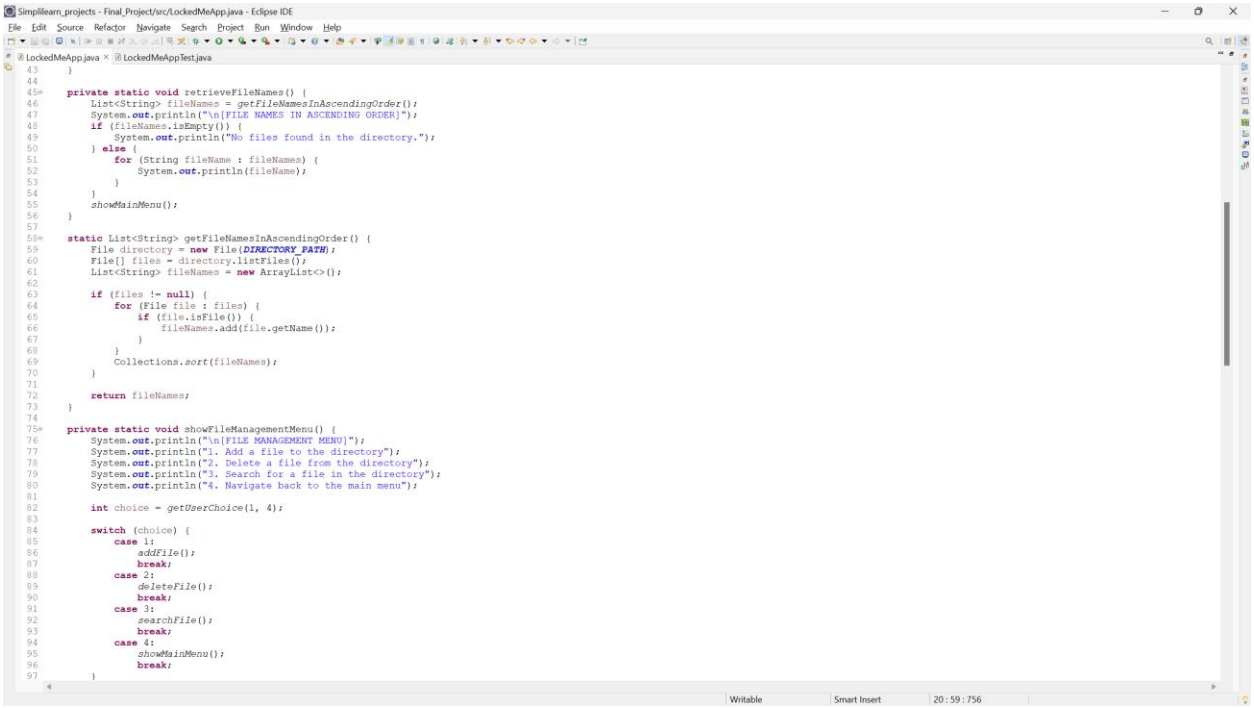
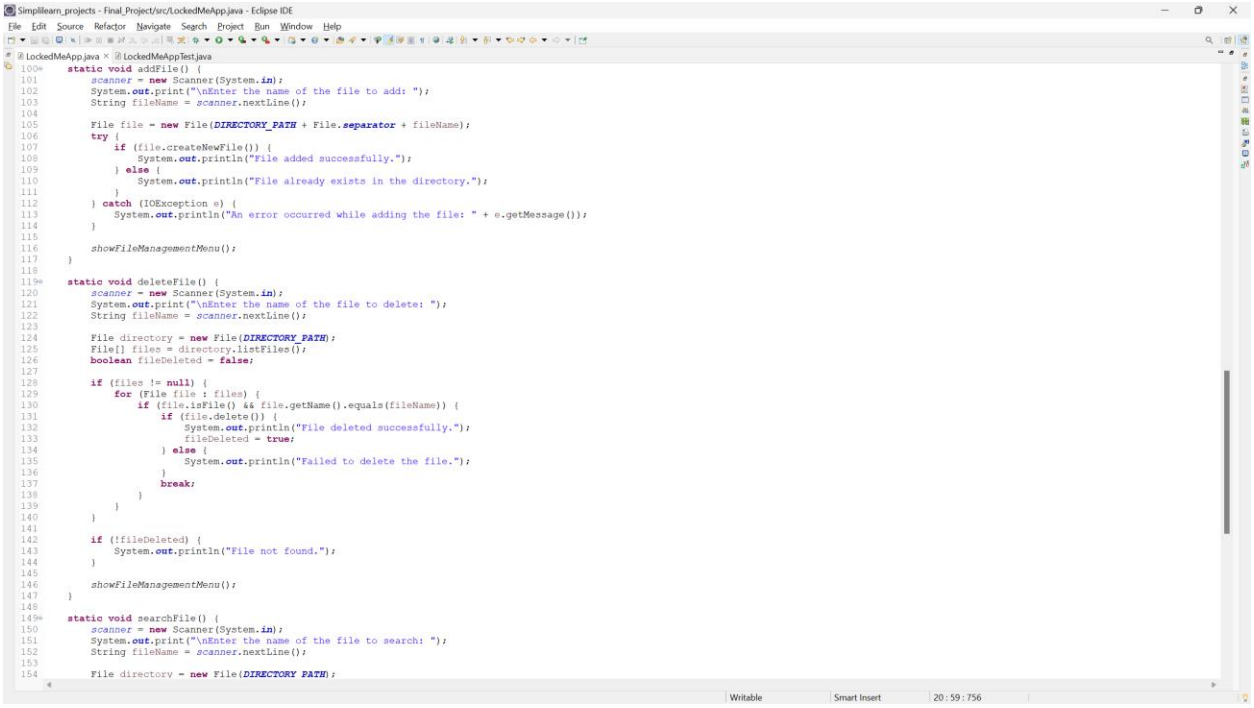
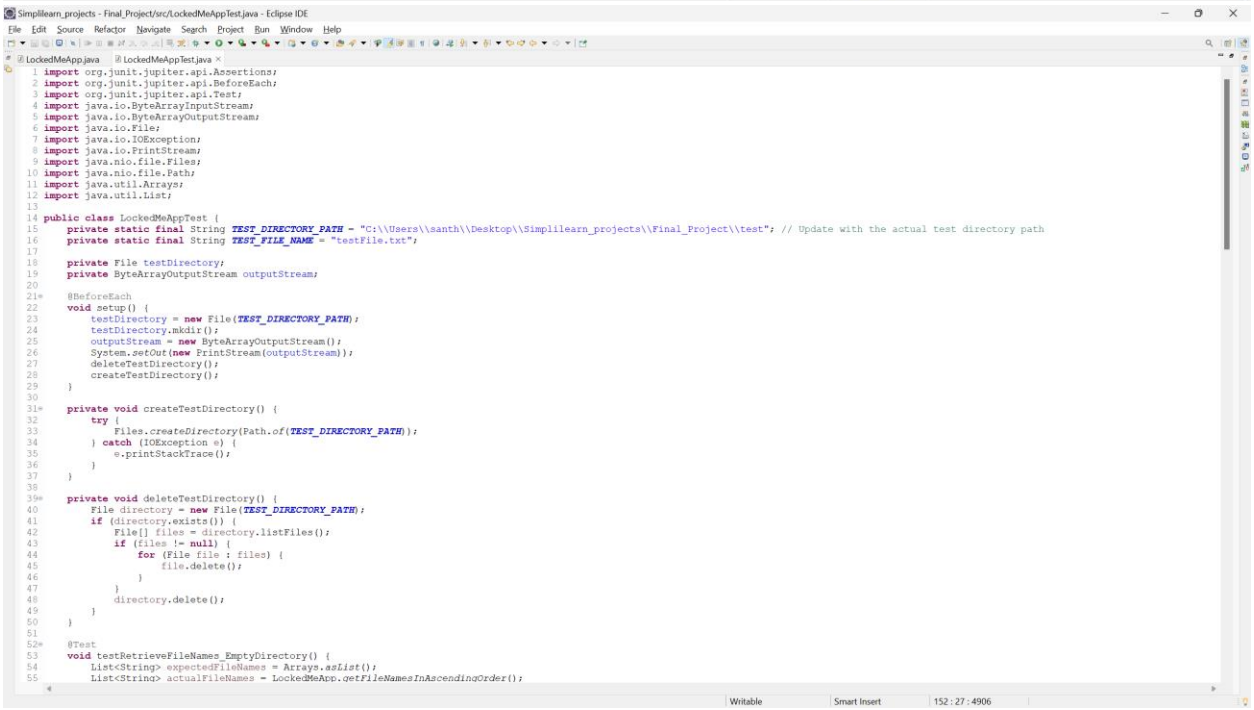
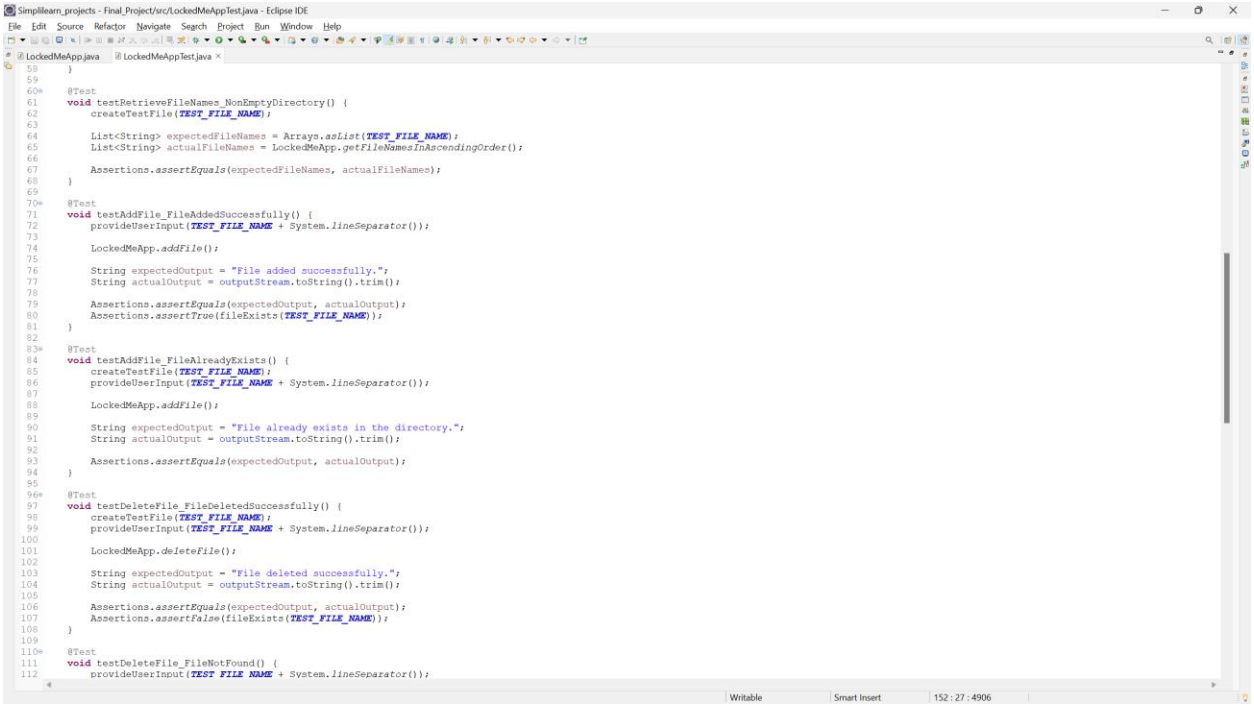

Simplilearn_projects - Final_Project/src/LockedMeApp.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
LockedMeApp.java
LockedMeAppTest.java
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Scanner;
public class LockedMeApp {
private static final String DIRECTORY_PATH = "C:\\Users\\santh\\Desktop\\Simplilearn_projects\\Final_Project\\test"; // Update with the actual directory path
private static Scanner scanner;
public static void main(String[] args) {
displayWelcomeScreen();
showMainMenu();
}
private static void displayWelcomeScreen() {
System.out.println("--");
System.out.println("   Welcome to LockedMe.com   ");
System.out.println("   Developed by Santhosh   ");
System.out.println("--");
}
private static void showMainMenu() {
System.out.println("\n[MAIN MENU]");
System.out.println("1. Retrieve file names in ascending order");
System.out.println("2. Manage files");
System.out.println("3. Close the application");
int choice = getUserChoice(1, 3);
switch (choice) {
case 1:
retrieveFileNames();
break;
case 2:
showFileManagementMenu();
break;
case 3:
closeApplication();
break;
}
}
private static void retrieveFileNames() {
List<String> fileNames = getFileNamesInAscendingOrder();
System.out.println("\n[FILE NAMES IN ASCENDING ORDER]");
if (fileNames.isEmpty()) {
System.out.println("No files found in the directory.");
} else {
for (String fileName : fileNames) {
System.out.println(fileName);
}
}
showMainMenu();
Writable
Smart Insert
20 : 59 : 756


Simplilearn_projects - Final_Project/src/LockedMeApp.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
LockedMeApp.java
LockedMeAppTest.java
}
private static void retrieveFileNames() {
List<String> fileNames = getFileNamesInAscendingOrder();
System.out.println("\n[FILE NAMES IN ASCENDING ORDER]");
if (fileNames.isEmpty()) {
System.out.println("No files found in the directory.");
} else {
for (String fileName : fileNames) {
System.out.println(fileName);
}
}
showMainMenu();
}
static List<String> getFileNamesInAscendingOrder() {
File directory = new File(DIRECTORY_PATH);
File[] files = directory.listFiles();
List<String> fileNames = new ArrayList<>();
if (files != null) {
for (File file : files) {
if (file.isFile()) {
fileNames.add(file.getName());
}
}
Collections.sort(fileNames);
}
return fileNames;
}
private static void showFileManagementMenu() {
System.out.println("\n[FILE MANAGEMENT MENU]");
System.out.println("1. Add a file to the directory");
System.out.println("2. Delete a file from the directory");
System.out.println("3. Search for a file in the directory");
System.out.println("4. Navigate back to the main menu");
int choice = getUserChoice(1, 4);
switch (choice) {
case 1:
addFile();
break;
case 2:
deleteFile();
break;
case 3:
searchFile();
break;
case 4:
showMainMenu();
break;
}
Writable
Smart Insert
20 : 59 : 756


Simplilearn_projects - Final_Project/src/LockedMeApp.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
LockedMeApp.java
LockedMeAppTest.java
static void addFile() {
scanner = new Scanner(System.in);
System.out.print("\nEnter the name of the file to add: ");
String fileName = scanner.nextLine();
File file = new File(DIRECTORY_PATH + File.separator + fileName);
try {
if (file.createNewFile()) {
System.out.println("File added successfully.");
} else {
System.out.println("File already exists in the directory.");
}
} catch (IOException e) {
System.out.println("An error occurred while adding the file: " + e.getMessage());
}
showFileManagementMenu();
}
static void deleteFile() {
scanner = new Scanner(System.in);
System.out.print("\nEnter the name of the file to delete: ");
String fileName = scanner.nextLine();
File directory = new File(DIRECTORY_PATH);
File[] files = directory.listFiles();
boolean fileDeleted = false;
if (files != null) {
for (File file : files) {
if (file.isFile() && file.getName().equals(fileName)) {
if (file.delete()) {
System.out.println("File deleted successfully.");
fileDeleted = true;
} else {
System.out.println("Failed to delete the file.");
}
break;
}
}
}
if (!fileDeleted) {
System.out.println("File not found.");
}
showFileManagementMenu();
}
static void searchFile() {
scanner = new Scanner(System.in);
System.out.print("\nEnter the name of the file to search: ");
String fileName = scanner.nextLine();
File directory = new File(DIRECTORY_PATH);
Writable
Smart Insert
20 : 59 : 756


Simplilearn_projects - Final_Project/src/LockedMeAppTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
LockedMeApp.java
LockedMeAppTest.java
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.PrintStream;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.Arrays;
import java.util.List;
public class LockedMeAppTest {
private static final String TEST_DIRECTORY_PATH = "C:\\Users\\santh\\Desktop\\Simplilearn_projects\\Final_Project\\test"; // Update with the actual test directory path
private static final String TEST_FILE_NAME = "testFile.txt";
private File testDirectory;
private ByteArrayOutputStream outputStream;
@BeforeEach
void setup() {
testDirectory = new File(TEST_DIRECTORY_PATH);
testDirectory.mkdir();
outputStream = new ByteArrayOutputStream();
System.setOut(new PrintStream(outputStream));
deleteTestDirectory();
createTestDirectory();
}
private void createTestDirectory() {
try {
Files.createDirectory(Path.of(TEST_DIRECTORY_PATH));
} catch (IOException e) {
e.printStackTrace();
}
}
private void deleteTestDirectory() {
File directory = new File(TEST_DIRECTORY_PATH);
if (directory.exists()) {
File[] files = directory.listFiles();
if (files != null) {
for (File file : files) {
file.delete();
}
}
directory.delete();
}
}
@Test
void testRetrieveFileNames_EmptyDirectory() {
List<String> expectedFileNames = Arrays.asList();
List<String> actualFileNames = LockedMeApp.getFileNamesInAscendingOrder();
Writable
Smart Insert
152 : 27 : 4906



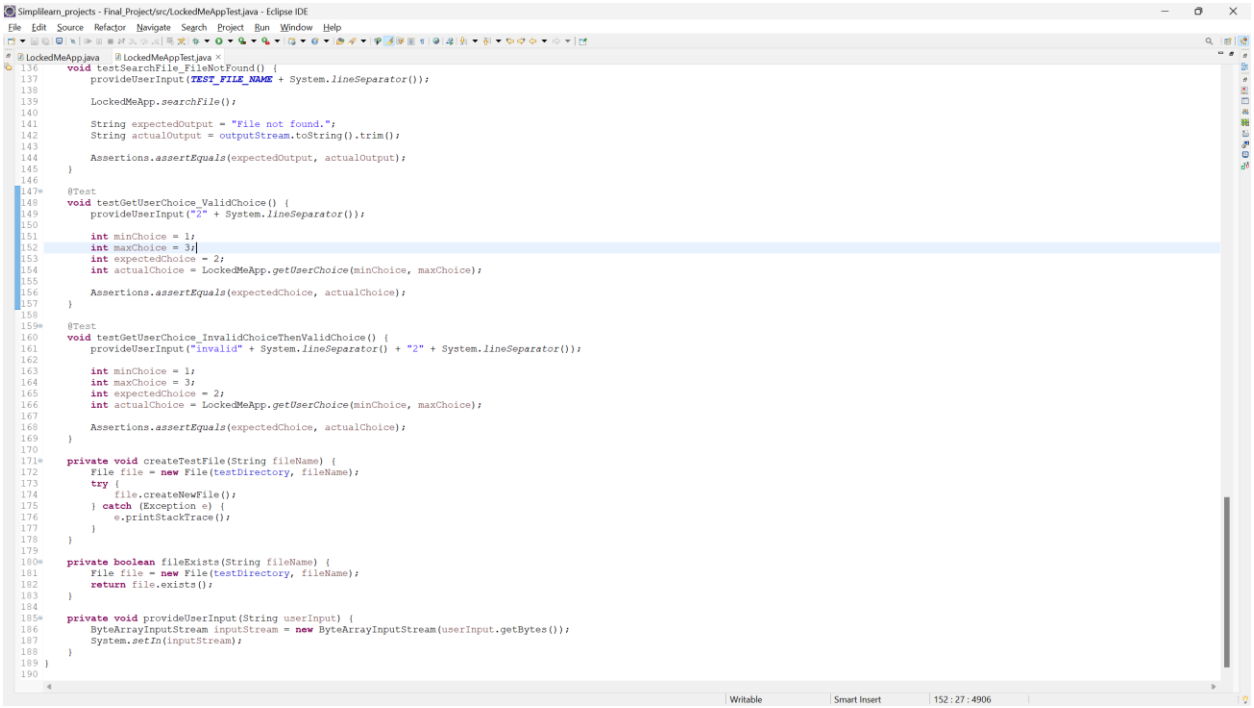
Simplilearn_projects - Final_Project/src/LockedMeAppTest.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

LockedMeApp.java LockedMeAppTest.java

```
58      }
59
60      @Test
61      void testRetrieveFileNames_NonEmptyDirectory() {
62          createTestFile(TEST_FILE_NAME);
63
64          List<String> expectedFileNames = Arrays.asList(TEST_FILE_NAME);
65          List<String> actualFileNames = LockedMeApp.getFileNamesInAscendingOrder();
66
67          Assertions.assertEquals(expectedFileNames, actualFileNames);
68      }
69
70      @Test
71      void testAddFile_FileAddedSuccessfully() {
72          provideUserInput(TEST_FILE_NAME + System.lineSeparator());
73
74          LockedMeApp.addFile();
75
76          String expectedOutput = "File added successfully.";
77          String actualOutput = outputStream.toString().trim();
78
79          Assertions.assertEquals(expectedOutput, actualOutput);
80          Assertions.assertTrue(fileExists(TEST_FILE_NAME));
81      }
82
83      @Test
84      void testAddFile_FileAlreadyExists() {
85          createTestFile(TEST_FILE_NAME);
86          provideUserInput(TEST_FILE_NAME + System.lineSeparator());
87
88          LockedMeApp.addFile();
89
90          String expectedOutput = "File already exists in the directory.";
91          String actualOutput = outputStream.toString().trim();
92
93          Assertions.assertEquals(expectedOutput, actualOutput);
94      }
95
96      @Test
97      void testDeleteFile_FileDeletedSuccessfully() {
98          createTestFile(TEST_FILE_NAME);
99          provideUserInput(TEST_FILE_NAME + System.lineSeparator());
100
101         LockedMeApp.deleteFile();
102
103         String expectedOutput = "File deleted successfully.";
104         String actualOutput = outputStream.toString().trim();
105
106         Assertions.assertEquals(expectedOutput, actualOutput);
107         Assertions.assertFalse(fileExists(TEST_FILE_NAME));
108     }
109
110     @Test
111     void testDeleteFile_FileNotFound() {
112         provideUserInput(TEST_FILE_NAME + System.lineSeparator());
```

Writable Smart Insert 152 : 27 : 4906



Simplilearn_projects - Final_Project/src/LockedMeAppTest.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

LockedMeApp.java LockedMeAppTest.java

```
136     void testSearchFile_FileNotFound() {
137         provideUserInput(TEST_FILE_NAME + System.lineSeparator());
138
139         LockedMeApp.searchFile();
140
141         String expectedOutput = "File not found.";
142         String actualOutput = outputStream.toString().trim();
143
144         Assertions.assertEquals(expectedOutput, actualOutput);
145     }
146
147     @Test
148     void testGetUserChoice_ValidChoice() {
149         provideUserInput("2" + System.lineSeparator());
150
151         int minChoice = 1;
152         int maxChoice = 3;
153         int expectedChoice = 2;
154         int actualChoice = LockedMeApp.getUserChoice(minChoice, maxChoice);
155
156         Assertions.assertEquals(expectedChoice, actualChoice);
157     }
158
159     @Test
160     void testGetUserChoice_InvalidChoiceThenValidChoice() {
161         provideUserInput("invalid" + System.lineSeparator() + "2" + System.lineSeparator());
162
163         int minChoice = 1;
164         int maxChoice = 3;
165         int expectedChoice = 2;
166         int actualChoice = LockedMeApp.getUserChoice(minChoice, maxChoice);
167
168         Assertions.assertEquals(expectedChoice, actualChoice);
169     }
170
171     private void createTestFile(String fileName) {
172         File file = new File(testDirectory, fileName);
173         try {
174             file.createNewFile();
175         } catch (Exception e) {
176             e.printStackTrace();
177         }
178     }
179
180     private boolean fileExists(String fileName) {
181         File file = new File(testDirectory, fileName);
182         return file.exists();
183     }
184
185     private void provideUserInput(String userInput) {
186         ByteArrayInputStream inputStream = new ByteArrayInputStream(userInput.getBytes());
187         System.setIn(inputStream);
188     }
189 }
190
```

Writable Smart Insert 152 : 27 : 4906