



Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Fall 2022, B.Sc. in CSE (DAY)

LAB REPORT NO # 07

Course Title: Object Oriented Programming (JAVA)

Course Code: CSE 202

Section: CSE 213 - DA (PC)

Lab Experiment Name(s):

Interface

- Create an interface `isEmergency` with only one method - `soundSiren` which takes no arguments and returns no value.
- Write a class `FireEmergency` that implements the `IsEmergency` interface. The `soundSiren` method should print "Siren Sounded".
- Write a class `SmokeAlarm` that does not implement any interface. The class has an empty body.
- Create an array of Object class, `myArray` in the main method.
- Construct 2 `SmokeAlarm` object and add it to the array `myArray` in the main method.
- Construct 2 `FireEmergency` object and add it to the array `myArray` in the main method.
- In the main method, write a for loop, to print which array elements are instances of classes that implement the `IsEmergency` interface and if so, call the `soundSiren` method.

Student Details

Name	ID
Md. Shahidul Islam Prodhan	213902017

Lab Date: 05 December, 2022

Submission Date: 12 December, 2022

Course Teacher's Name: Dr. Muhammad Aminur Rahaman, Associate Professor

[For Teacher's use only: Don't write anything inside this box]

Lab Report Status

Marks:	Signature:
Comments:	Date:

1. TITLE OF THE LAB EXPERIMENT

Java Interface

2. OBJECTIVES

1. Using a class that implements several interfaces, we can comprehend multiple inheritance.
2. Through the use of an abstract type, an interface in java describes how a class should behave.
3. Specifying methods that a class or classes must implement.
4. Then to improve the readability and comprehension of the code.

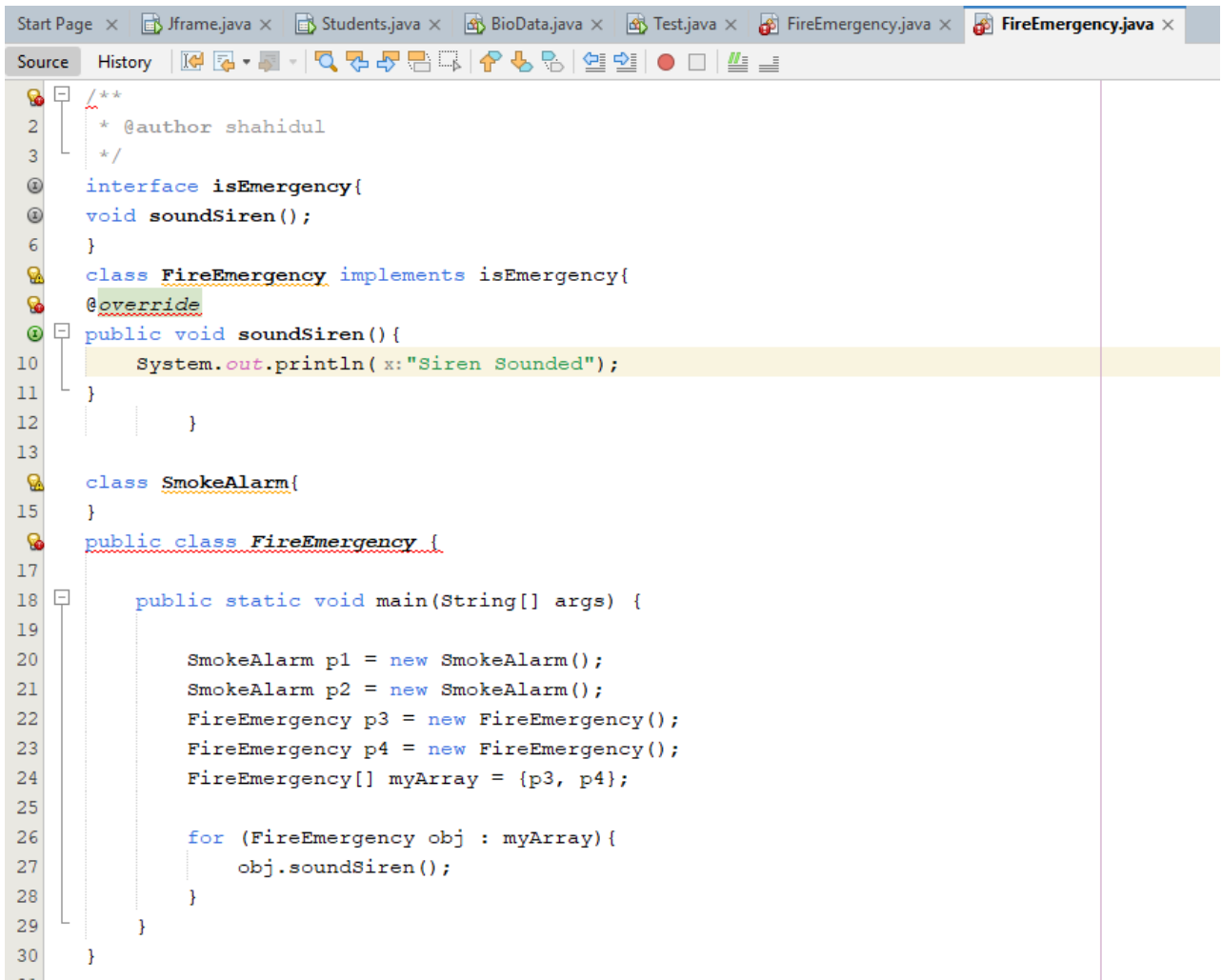
3. ALGORITHM:

Steps	Procedures / Works
Step 1	start
Step 2	Create an interface IsEmergency with an abstract technique soundSiren.
Step 3	Create a class FireEmergency that implements IsEmergency .
Step 4	Override the soundSiren method inside the FireEmergency elegance.
Step 5	Create every other magnificence SmokeAlarm with an empty frame.
Step 6	Create a primary class and put into effect the primary technique.
Step 7	Inside the principal technique create an array of item magnificence and create 2 objects of SmokeAlarm and 2 objects of FireEmergency .
Step 8	Write a for loop, to print which array elements are instances of training that implement the IsEmergency interface and if so, name the soundSiren method.
Step 9	End

4. IMPLEMENTATION & TEST RESULT

Example of Implementing Interface

CODE

A screenshot of the NetBeans IDE interface. The top toolbar shows various icons for file operations, editing, and running. The 'Source' tab is active, displaying a Java file named 'FireEmergency.java'. The code defines an interface 'isEmergency' with a method 'soundSiren()'. Two classes, 'FireEmergency' and 'SmokeAlarm', implement this interface. The 'FireEmergency' class has a 'soundSiren()' method that prints 'Siren Sounded'. The 'SmokeAlarm' class is also defined. A 'main' method is present, which creates instances of 'SmokeAlarm' and 'FireEmergency', and calls 'soundSiren()' on the 'FireEmergency' instances. The line 'System.out.println(x:"Siren Sounded");' is highlighted in yellow. The code is as follows:

```
1  /**
2   * @author shahidul
3   */
4  interface isEmergency{
5      void soundSiren();
6  }
7
8  class FireEmergency implements isEmergency{
9      @Override
10     public void soundSiren(){
11         System.out.println( x:"Siren Sounded");
12     }
13 }
14
15 class SmokeAlarm{
16 }
17
18 public class FireEmergency {
19
20     public static void main(String[] args) {
21
22         SmokeAlarm p1 = new SmokeAlarm();
23         SmokeAlarm p2 = new SmokeAlarm();
24         FireEmergency p3 = new FireEmergency();
25         FireEmergency p4 = new FireEmergency();
26         FireEmergency[] myArray = {p3, p4};
27
28         for (FireEmergency obj : myArray){
29             obj.soundSiren();
30         }
31     }
32 }
```

OUTPUT

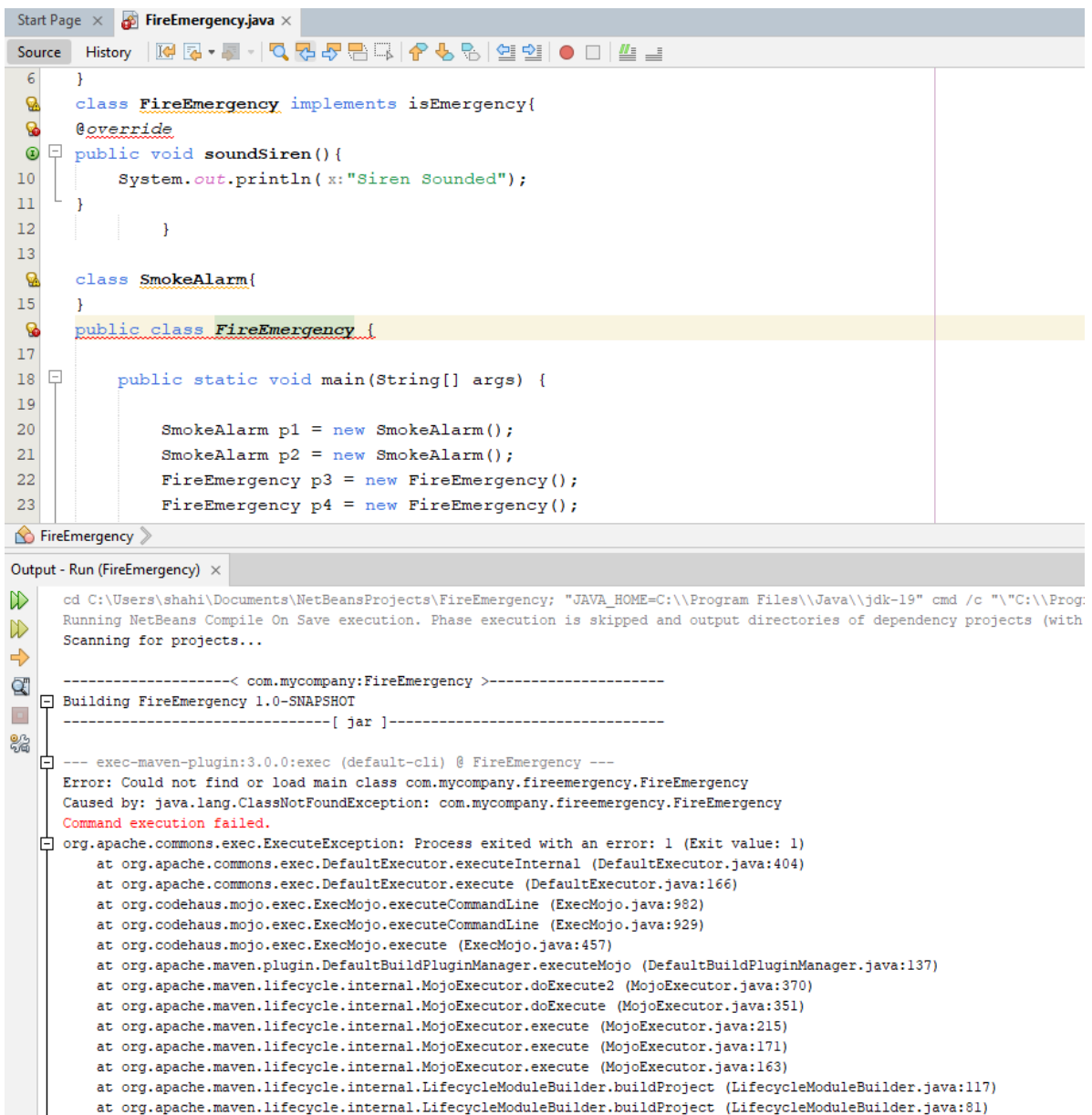
I could not successfully show expected output results in my Netbeans IDE.

4. IMPLEMENTATION & TEST RESULT

Example of Implementing Interface

OUTPUT

I could not successfully show expected output results in my Netbeans IDE.
There was error in my program.



The screenshot displays the NetBeans IDE interface. The top pane shows the source code of `FireEmergency.java`. The code defines an interface `isEmergency` with a method `soundSiren()`. The `FireEmergency` class implements this interface. The `SmokeAlarm` class is also defined. The `main` method creates instances of `SmokeAlarm` and `FireEmergency`.

```
6    }
7
8    class FireEmergency implements isEmergency{
9        @Override
10       public void soundSiren(){
11           System.out.println( x:"Siren Sounded");
12       }
13
14       class SmokeAlarm{
15       }
16
17       public class FireEmergency {
18
19           public static void main(String[] args) {
20
21               SmokeAlarm p1 = new SmokeAlarm();
22               SmokeAlarm p2 = new SmokeAlarm();
23               FireEmergency p3 = new FireEmergency();
24               FireEmergency p4 = new FireEmergency();
25           }
26       }
27   }
```

The bottom pane shows the output of the program. It indicates that the program failed to execute due to a `ClassNotFoundException`. The error message is:

```
cd C:\Users\shahi\Documents\NetBeansProjects\FireEmergency; "JAVA_HOME=C:\Program Files\Java\jdk-19" cmd /c "%C:\Prog:
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with
Scanning for projects...

-----< com.mycompany:FireEmergency >-----
Building FireEmergency 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ FireEmergency ---
Error: Could not find or load main class com.mycompany.fireemergency.FireEmergency
Caused by: java.lang.ClassNotFoundException: com.mycompany.fireemergency.FireEmergency
Command execution failed.
org.apache.commons.exec.ExecuteException: Process exited with an error: 1 (Exit value: 1)
at org.apache.commons.exec.DefaultExecutor.executeInternal (DefaultExecutor.java:404)
at org.apache.commons.exec.DefaultExecutor.execute (DefaultExecutor.java:166)
at org.codehaus.mojo.exec.ExecMojo.executeCommandLine (ExecMojo.java:982)
at org.codehaus.mojo.exec.ExecMojo.executeCommandLine (ExecMojo.java:929)
at org.codehaus.mojo.exec.ExecMojo.execute (ExecMojo.java:457)
at org.apache.maven.plugin.DefaultBuildPluginManager.executeMojo (DefaultBuildPluginManager.java:137)
at org.apache.maven.lifecycle.internal.MojoExecutor.doExecute2 (MojoExecutor.java:370)
at org.apache.maven.lifecycle.internal.MojoExecutor.doExecute (MojoExecutor.java:351)
at org.apache.maven.lifecycle.internal.MojoExecutor.execute (MojoExecutor.java:215)
at org.apache.maven.lifecycle.internal.MojoExecutor.execute (MojoExecutor.java:171)
at org.apache.maven.lifecycle.internal.MojoExecutor.execute (MojoExecutor.java:163)
at org.apache.maven.lifecycle.internal.LifecycleModuleBuilder.buildProject (LifecycleModuleBuilder.java:117)
at org.apache.maven.lifecycle.internal.LifecycleModuleBuilder.buildProject (LifecycleModuleBuilder.java:81)
```

5. ANALYSIS AND DISCUSSION

- 1) Using a class that implements various interfaces, the challenge illustrates multiple inheritance.
- 2) We learnt, how to construct an item and add it to the array while doing interface code.