**Exercise 1: Implementing the Singleton Pattern**

**Scenario:**

You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

**SOLUTION:**

Logger.java  
  
package Singleton;

public class Logger {

private static Logger *instance*;

private Logger() {};

public static Logger getInstance() {

if(*instance*==null) {

*instance* =new Logger();

}

return *instance*;

}

public static void log(String msg) {

System.***out***.println("Message: "+msg);

}

}

Main.java  
  
package Singleton;

public class Main {

public static void main(String args[]) {

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

logger1.*log*("This is the first log message.");

logger2.*log*("This is the second log message.");

if (logger1 == logger2) {

System.***out***.println("Both logger instances are the same.");

} else {

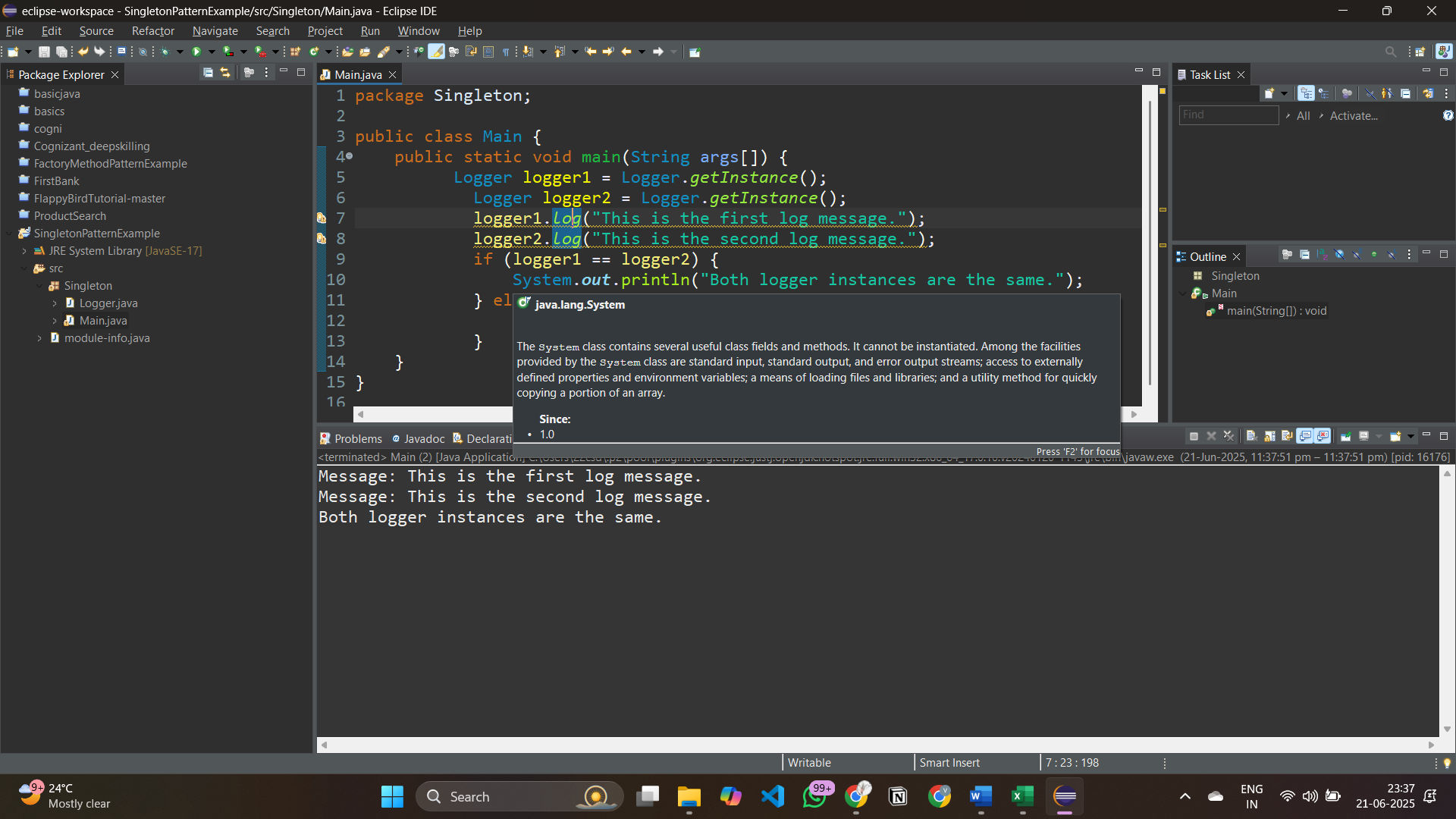
System.***out***.println("Different instances found.");

}

}

}

OUTPUT:



**Exercise 2: Implementing the Factory Method Pattern**

**Scenario:**

You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

**SOLUTION:**

Document.java

package Factory;

public interface Document {

void open();

}

wordDocx.java

package Factory;

public class wordDocx implements Document{

public void open() {

System.***out***.println("Openning a word Doc");

}

}

pdfDoc.java

package Factory;

public class pdfDoc implements Document{

public void open() {

System.***out***.println("Opens a pdf document");

}

}

excelDoc.java

package Factory;

public class excelDoc implements Document{

public void open() {

System.***out***.println("Opens a excel doc");

}

}

DocumentFactory.java

package Factory;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

wordDocxFactory.java

package Factory;

public class wordDocxFactory extends DocumentFactory{

public Document createDocument() {

return new wordDocx();

}

}

pdfDocFactory.java

package Factory;

public class pdfDocFactory extends DocumentFactory{

public Document createDocument() {

return new pdfDoc();

}

}

excelDocFactory.java

package Factory;

public class excelDocFactory extends DocumentFactory{

public Document createDocument() {

return new excelDoc();

}

}

Main.java

package Factory;

public class Main {

public static void main(String args[]) {

DocumentFactory wordFactory = new wordDocxFactory();//word doc creation

Document wordDoc = wordFactory.createDocument();

wordDoc.open();

DocumentFactory pdfFactory = new pdfDocFactory();//pdf

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

DocumentFactory excelFactory = new excelDocFactory();//excel

Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**OUTPUT:**

