**SUPERSET ID: 6412083**

**Design Patterns and Principles**

**Exercise 1: Implementing the Singleton Pattern**

**CODE:**

package singletonpatternexample;

public class Logger {

private static Logger *instance*;

private Logger() {

System.*out*.println("Logger initialized.");

}

public static Logger getInstance() {

if (*instance* == null) {

*instance* = new Logger();

}

return *instance*;

}

public void log(String message) {

System.*out*.println("[LOG] " + message);

}

}

package singletonpatternexample;

public class LoggerTest {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

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

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

if (logger1 == logger2) {

System.*out*.println("Both logger instances are the same. Singleton is working!");

System.*out*.println("Logger 1 hash: " + logger1.hashCode());

System.*out*.println("Logger 2 hash: " + logger2.hashCode());

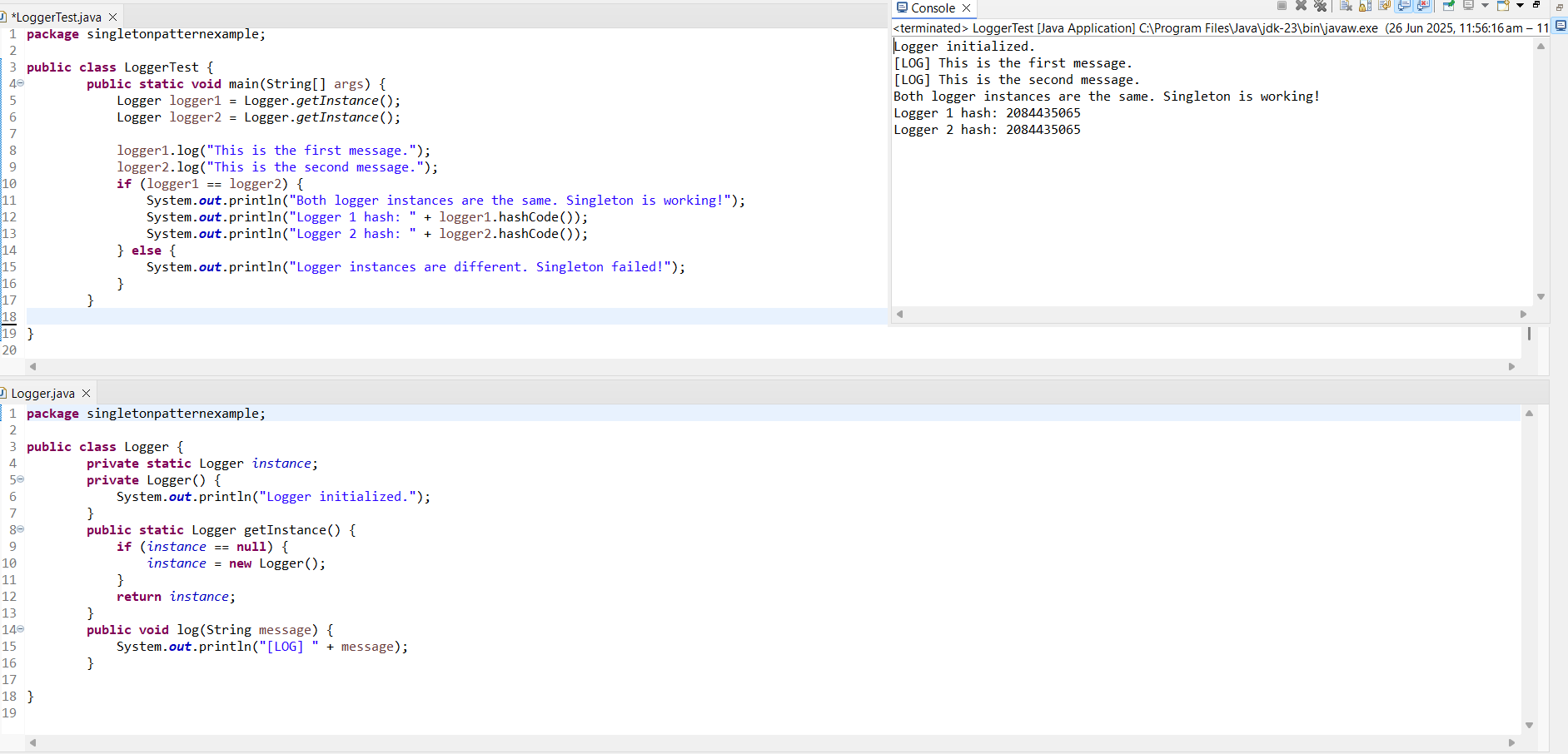
} else {

System.*out*.println("Logger instances are different. Singleton failed!");

}

}

}

****

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

package factorymethod;

public interface Documents {

void open();

}

package factorymethod;

public abstract class DocumentFactory {

public abstract Documents createDocument();

}

package factorymethod;

public class ExcelDocument implements Documents {

public void open() {

System.*out*.println("Opening Excel document...");

}

}

package factorymethod;

public class ExcelDocumentFactory extends DocumentFactory {

public Documents createDocument() {

System.*out*.println("Creating Excel Document...");

return new ExcelDocument();

}

}

package factorymethod;

public class PdfDocument implements Documents {

public void open() {

System.*out*.println("Opening PDF document...");

}

}

package factorymethod;

public class PdfDocumentFactory extends DocumentFactory {

public Documents createDocument() {

System.*out*.println("Creating PDF Document...");

return new PdfDocument();

}

}

package factorymethod;

public class WordDocument implements Documents {

public void open() {

System.*out*.println("Opening WORD document...");

}

}

package factorymethod;

public class WordDocumentFactory extends DocumentFactory {

public Documents createDocument() {

System.*out*.println("Creating Word Document...");

return new WordDocument();

}

}

package factorymethod;

public class Main {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

Documents wordDoc = wordFactory.createDocument();

wordDoc.open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Documents pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Documents excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

