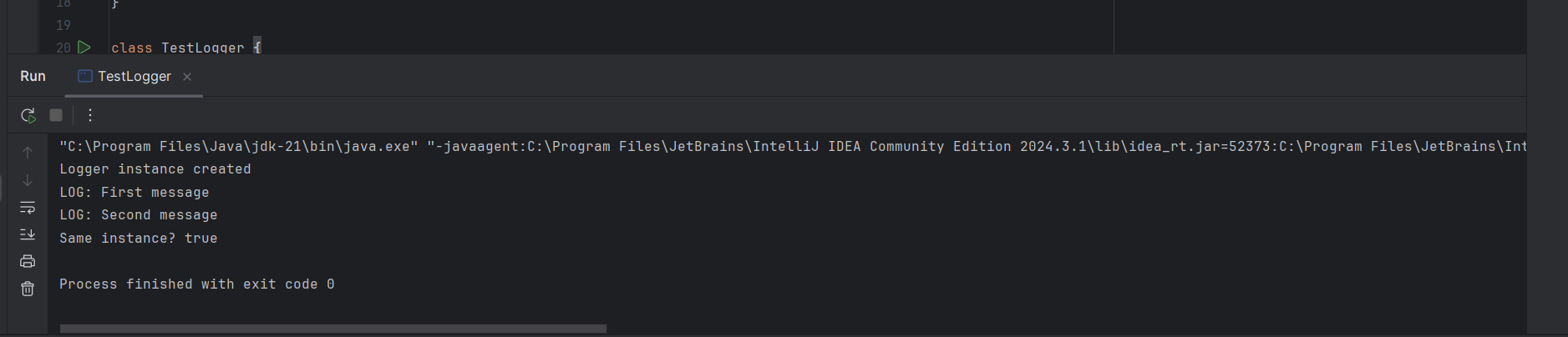
**Design Patterns and Principles**

**1.Implementing the Singleton Pattern:**

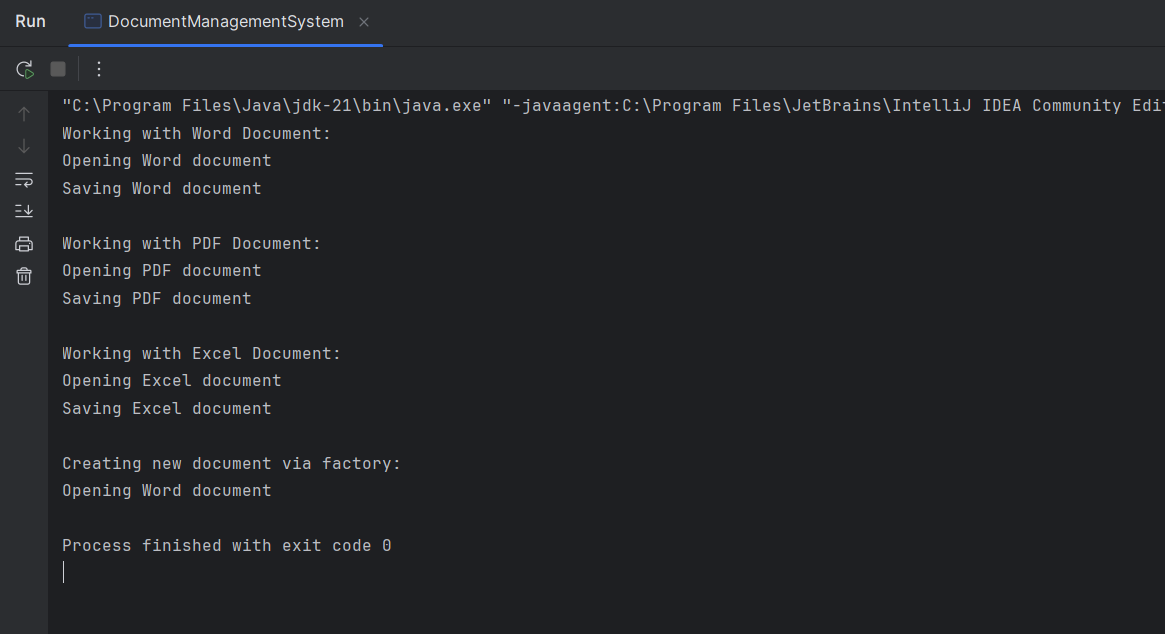
**Code:**

public class Logger {  
  
 private static final Logger *instance* = new Logger();  
  
 private Logger() {  
 System.*out*.println("Logger instance created");  
 }  
  
 public static Logger getInstance() {  
 return *instance*;  
 }  
  
 public void log(String message) {  
 System.*out*.println("LOG: " + message);  
 }  
}  
  
class TestLogger {  
 public static void main(String[] args) {  
  
 Logger logger1 = Logger.*getInstance*();  
 logger1.log("First message");  
  
 Logger logger2 = Logger.*getInstance*();  
 logger2.log("Second message");  
  
 System.*out*.println("Same instance? " + (logger1 == logger2));  
 }  
}

**Output:**  
  


**2. Implementing the Factory Method Pattern:  
  
Code:**public interface Document {  
 void open();  
 void save();  
 void close();  
}  
  
class WordDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Opening Word document");  
 }  
 @Override  
 public void save() {  
 System.*out*.println("Saving Word document");  
 }  
 @Override  
 public void close() {  
 System.*out*.println("Closing Word document");  
 }  
}  
  
class PdfDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Opening PDF document");  
 }  
 @Override  
 public void save() {  
 System.*out*.println("Saving PDF document");  
 }  
 @Override  
 public void close() {  
 System.*out*.println("Closing PDF document");  
 }  
}  
  
class ExcelDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Opening Excel document");  
 }  
 @Override  
 public void save() {  
 System.*out*.println("Saving Excel document");  
 }  
 @Override  
 public void close() {  
 System.*out*.println("Closing Excel document");  
 }  
}  
  
abstract class DocumentFactory {  
 public abstract Document createDocument();  
 public void newDocument() {  
 Document doc = createDocument();  
 doc.open();  
 }  
}  
  
class WordDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new WordDocument();  
 }  
}  
  
class PdfDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new PdfDocument();  
 }  
}  
  
class ExcelDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new ExcelDocument();  
 }  
}  
  
// Test Class  
class DocumentManagementSystem {  
 public static void main(String[] args) {  
 DocumentFactory wordFactory = new WordDocumentFactory();  
 DocumentFactory pdfFactory = new PdfDocumentFactory();  
 DocumentFactory excelFactory = new ExcelDocumentFactory();  
  
 Document wordDoc = wordFactory.createDocument();  
 Document pdfDoc = pdfFactory.createDocument();  
 Document excelDoc = excelFactory.createDocument();  
  
 System.*out*.println("Working with Word Document:");  
 wordDoc.open();  
 wordDoc.save();  
  
 System.*out*.println("\nWorking with PDF Document:");  
 pdfDoc.open();  
 pdfDoc.save();  
  
 System.*out*.println("\nWorking with Excel Document:");  
 excelDoc.open();  
 excelDoc.save();  
  
 System.*out*.println("\nCreating new document via factory:");  
 wordFactory.newDocument();  
 }  
}

**Output:**

****