**MANDATORY HANDS-ON**

**DESIGN PATTERNS AND PRINCIPLES**

**Exercise 1: Implementing the Singleton Pattern**

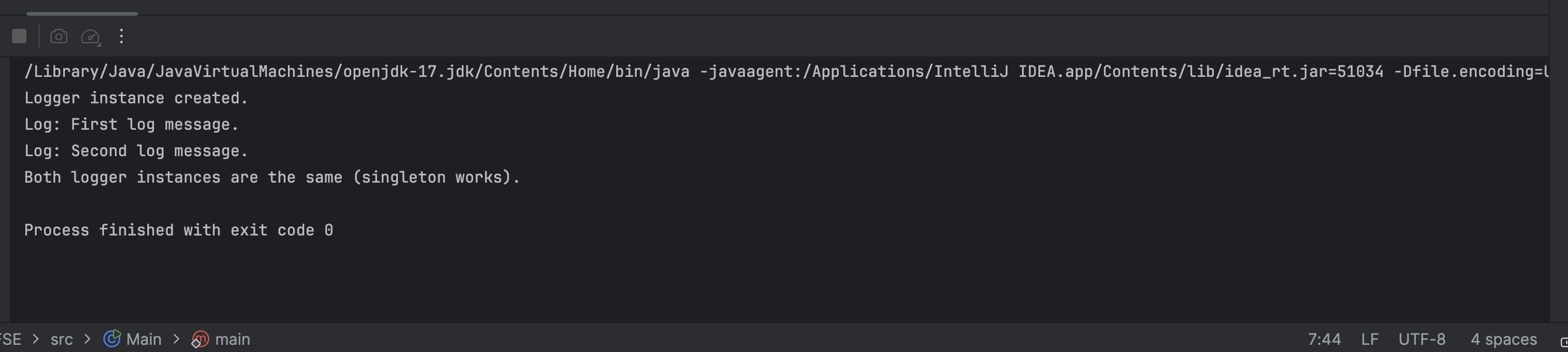
Main.java

public class Main {  
 public static void main(String[] args) {  
 Logger logger1 = Logger.*getInstance*();  
 logger1.log("First log message.");  
  
 Logger logger2 = Logger.*getInstance*();  
 logger2.log("Second log message.");  
  
  
 if (logger1 == logger2) {  
 System.*out*.println("Both logger instances are the same (singleton works).");  
 } else {  
 System.*out*.println("Different logger instances (singleton failed).");  
 }  
 }  
}

Logger.java

public class Main {  
 public static void main(String[] args) {  
 Logger logger1 = Logger.*getInstance*();  
 logger1.log("First log message.");  
  
 Logger logger2 = Logger.*getInstance*();  
 logger2.log("Second log message.");  
  
  
 if (logger1 == logger2) {  
 System.*out*.println("Both logger instances are the same (singleton works).");  
 } else {  
 System.*out*.println("Different logger instances (singleton failed).");  
 }  
 }  
}

OUTPUT:



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

Document.java

public interface Document {  
 void open();  
}

DocumentFactory.java

public abstract class DocumentFactory {  
 public abstract Document createDocument();  
}

ExcelDocument.java

public class ExcelDocument implements Document {  
 public void open() {  
 System.*out*.println("Opening an Excel document.");  
 }  
}

ExcelDocumentFactory.java

public class ExcelDocumentFactory extends DocumentFactory {  
 public Document createDocument() {  
 return new ExcelDocument();  
 }  
}

Main.java

public class Main {  
 public static void main(String[] args) {  
 DocumentFactory wordFactory = new WordDocumentFactory();  
 Document wordDoc = wordFactory.createDocument();  
 wordDoc.open();  
  
 DocumentFactory pdfFactory = new PdfDocumentFactory();  
 Document pdfDoc = pdfFactory.createDocument();  
 pdfDoc.open();  
  
 DocumentFactory excelFactory = new ExcelDocumentFactory();  
 Document excelDoc = excelFactory.createDocument();  
 excelDoc.open();  
 }  
}

PdfDocument.java

public class PdfDocument implements Document {  
 public void open() {  
 System.*out*.println("Opening a PDF document.");  
 }  
}

PdfDocumentFactory.java

public class PdfDocumentFactory extends DocumentFactory {  
 public Document createDocument() {  
 return new PdfDocument();  
 }  
}

WordDocument.java

public class WordDocument implements Document {  
 public void open() {  
 System.*out*.println("Opening a Word document.");  
 }  
}

WordDocumentFactory.java

public class WordDocumentFactory extends DocumentFactory {  
 public Document createDocument() {  
 return new WordDocument();  
 }  
}

OUTPUT:

