Java FSE Week-1

(Design Patterns and Principals)

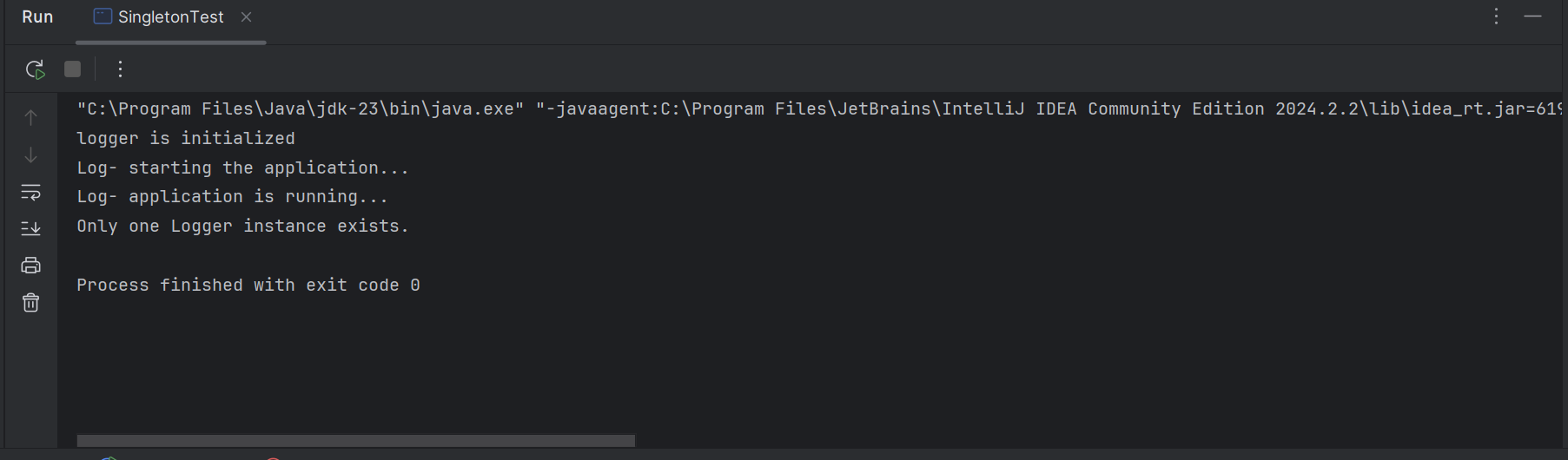
**Q1 Implementing Singleton pattern-**

**//Logger Class**

public class Logger {  
 private static Logger *singleInst*;  
  
 private Logger() {  
 System.*out*.println("logger is initialized");  
 }  
  
 public static Logger getLogger() {  
 if (*singleInst* == null) *singleInst* = new Logger();  
 return *singleInst*;  
 }  
  
 public void writeLog(String message) {  
 System.*out*.println("Log- " + message);  
 }  
}

**//SingletonTest class**

public class SingletonTest {  
 public static void main(String[] args) {  
 Logger A = Logger.*getLogger*();  
 A.writeLog("starting the application...");  
  
 Logger B = Logger.*getLogger*();  
 B.writeLog("application is running...");  
  
 if (A == B) System.*out*.println("Only one Logger instance exists.");  
 else System.*out*.println("Multiple Logger instances found.");  
  
 }  
}



**Q1 Factory Method Pattern-**

**//Document class**

public interface Document {  
 void open();  
}

**// DocumentFactory class**

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

**// ExcelDocument class**

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

**// ExcelFactory class**

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

**// PdfFactory class**

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

**// PdfDocument class**

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

**// WordFactory class**

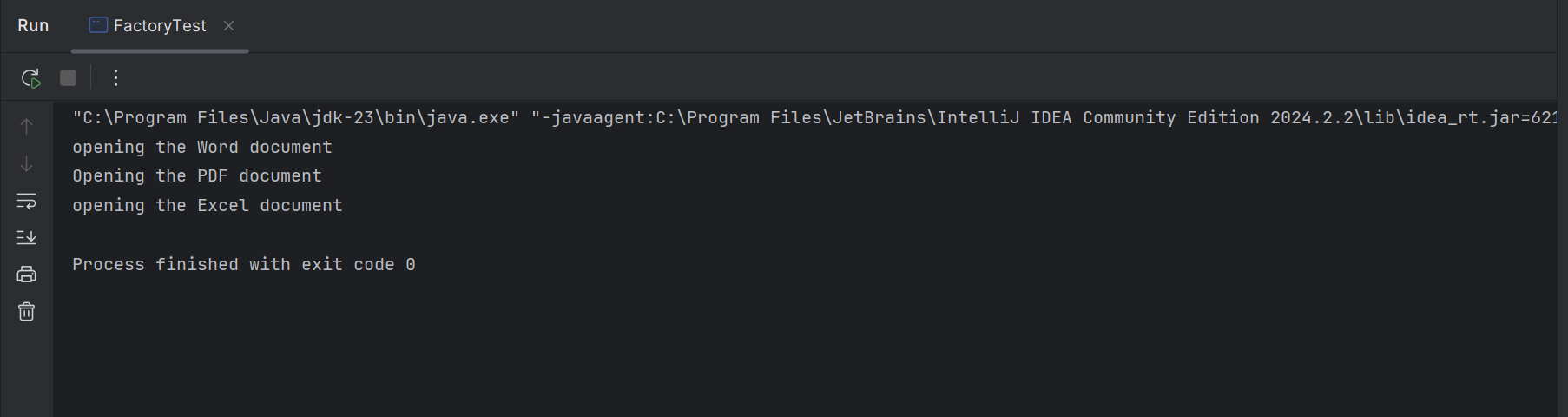
public class WordFactory extends DocumentFactory {  
 public Document createDocument() {  
 return new WordDocument();  
 }  
}

**// WordDocument**

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

**// FactoryTest class**

public class FactoryTest {  
 public static void main(String[] args) {  
 DocumentFactory wordFactory = new WordFactory();  
 Document word = wordFactory.createDocument();  
 word.open();  
  
 DocumentFactory pdfFactory = new PdfFactory();  
 Document pdf = pdfFactory.createDocument();  
 pdf.open();  
  
 DocumentFactory excelFactory = new ExcelFactory();  
 Document excel = excelFactory.createDocument();  
 excel.open();  
 }  
}

****