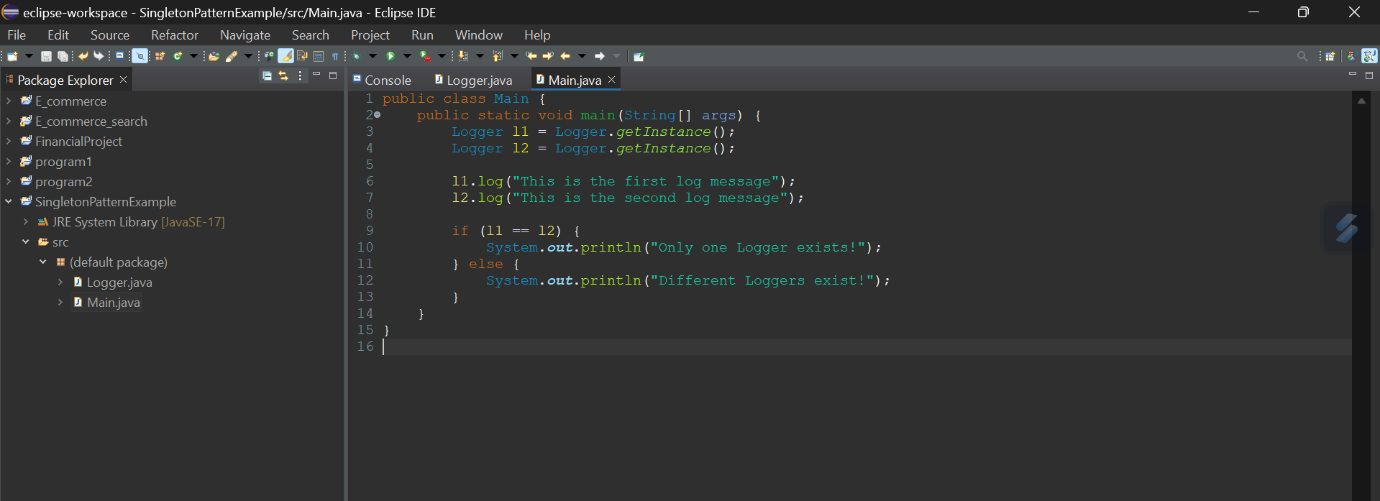
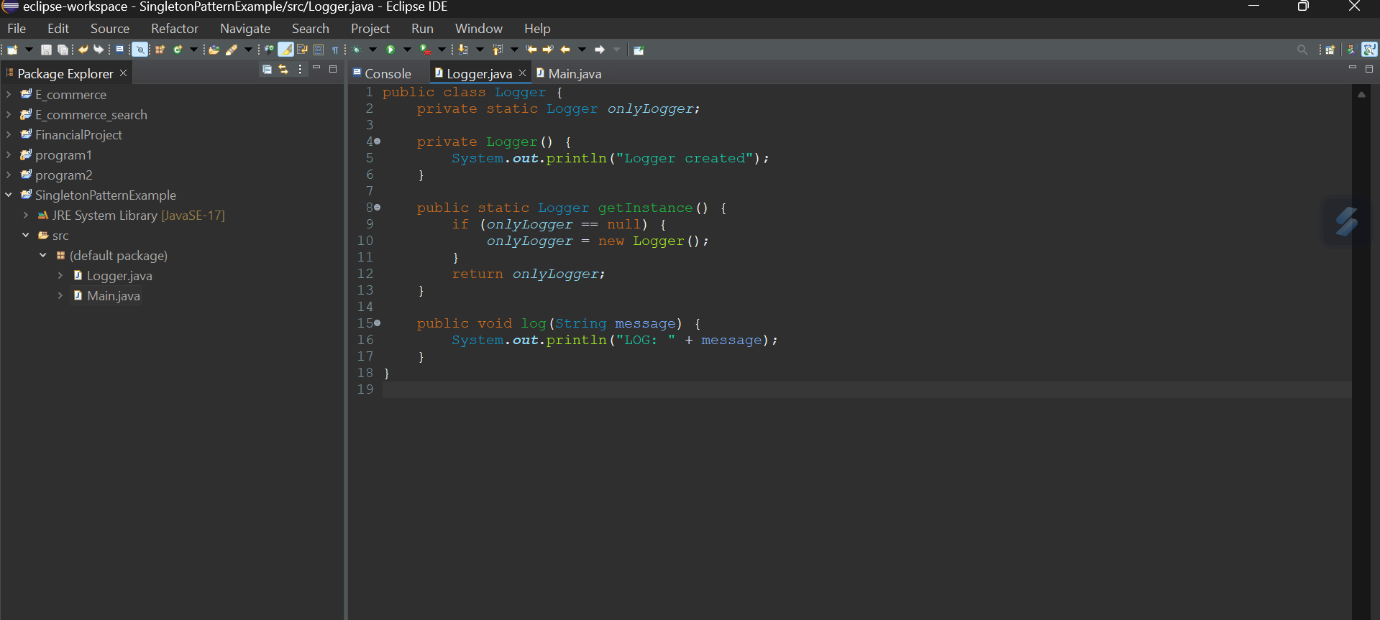
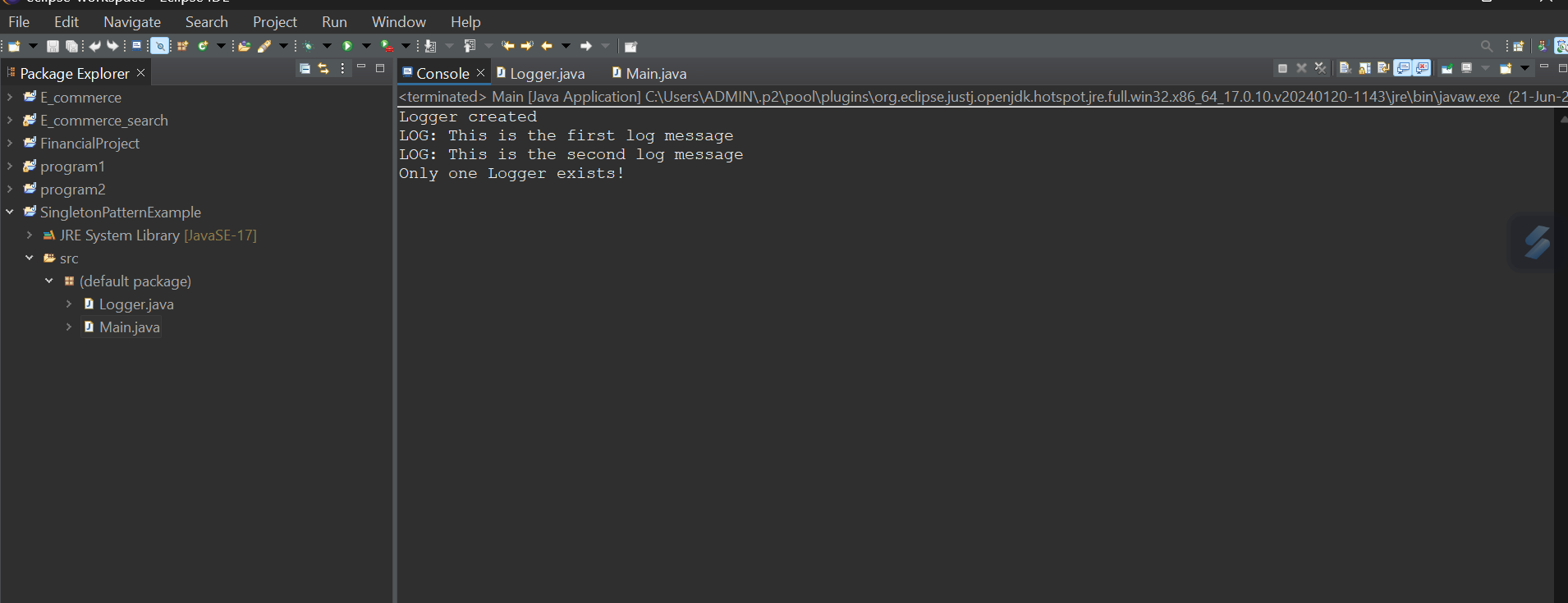
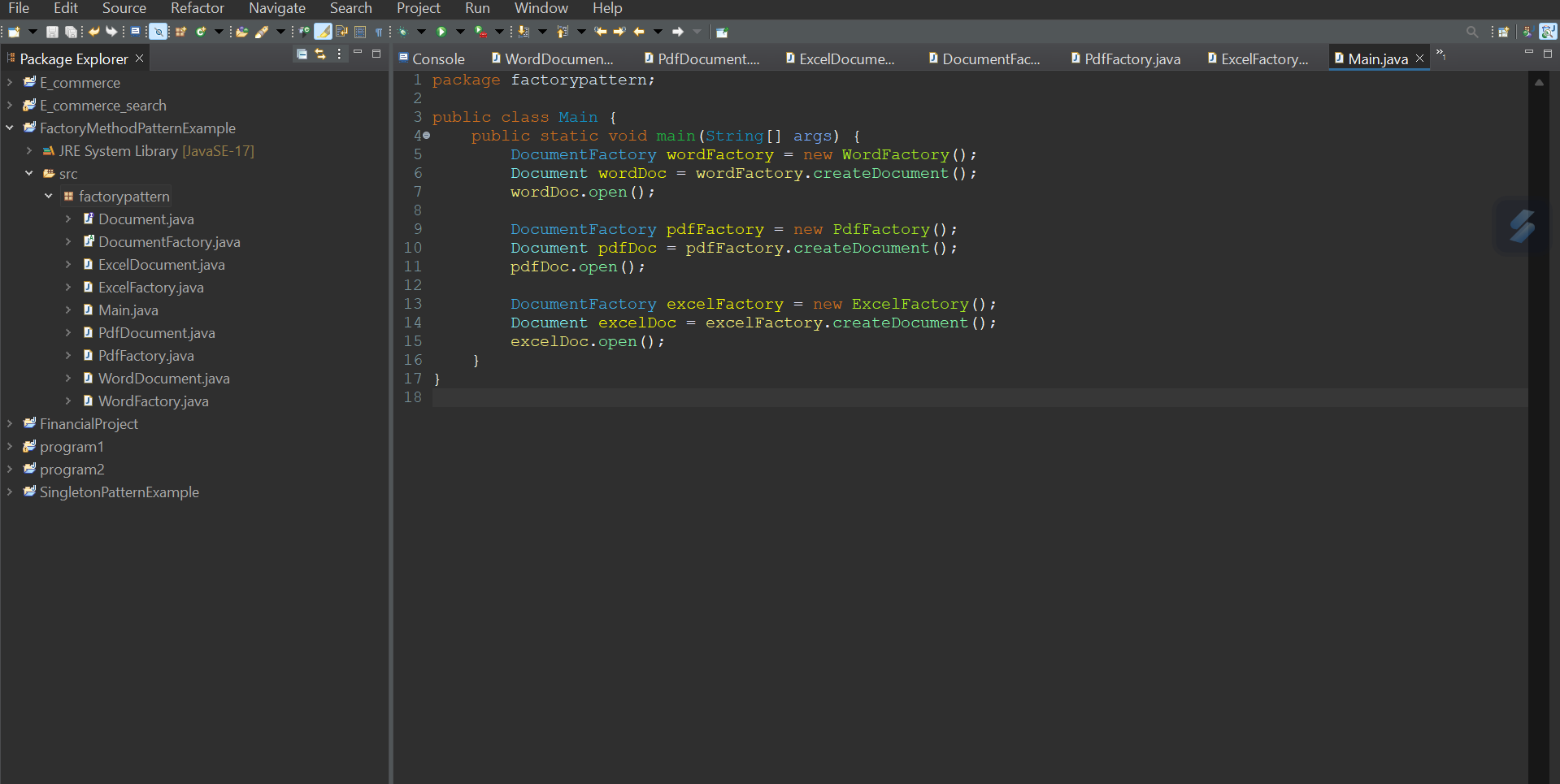
*DESIGN PATTERNS AND PRINCIPLES*

***********Exercise 01:* *Implementing the Singleton Pattern***



***Exercise 02: Implementing the Factory Method Pattern***

**

**Document.java**

package factorypattern;

public interface Document {

void open();

}

**WordDocument.java**

package factorypattern;

public class WordDocument implements Document {

public void open() {

System.out.println("Opening Word Document");

}

}

**PdfDocument.java**

package factorypattern;

public class PdfDocument implements Document {

public void open() {

System.out.println("Opening PDF Document");

}

}

**ExcelDocument.java**

package factorypattern;

public class ExcelDocument implements Document {

public void open() {

System.out.println("Opening Excel Document");

}

}

**DocumentFactory.java**

package factorypattern;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**WordFactory.java**

package factorypattern;

public class WordFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

**PdfFactory.java**

package factorypattern;

public class PdfFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

**ExcelFactory.java**

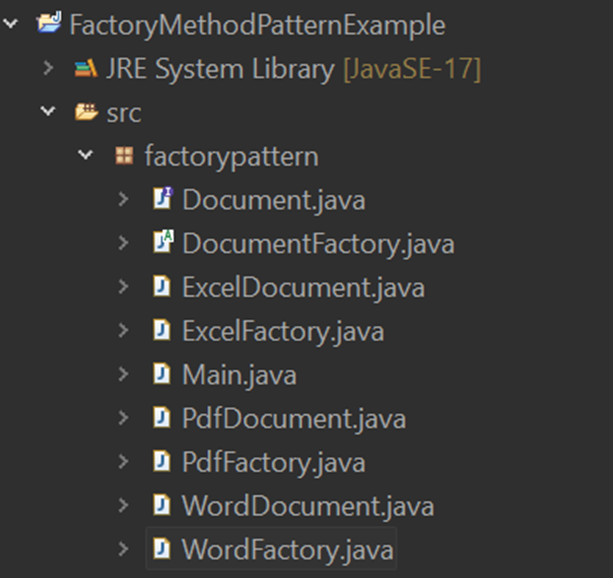
package factorypattern;

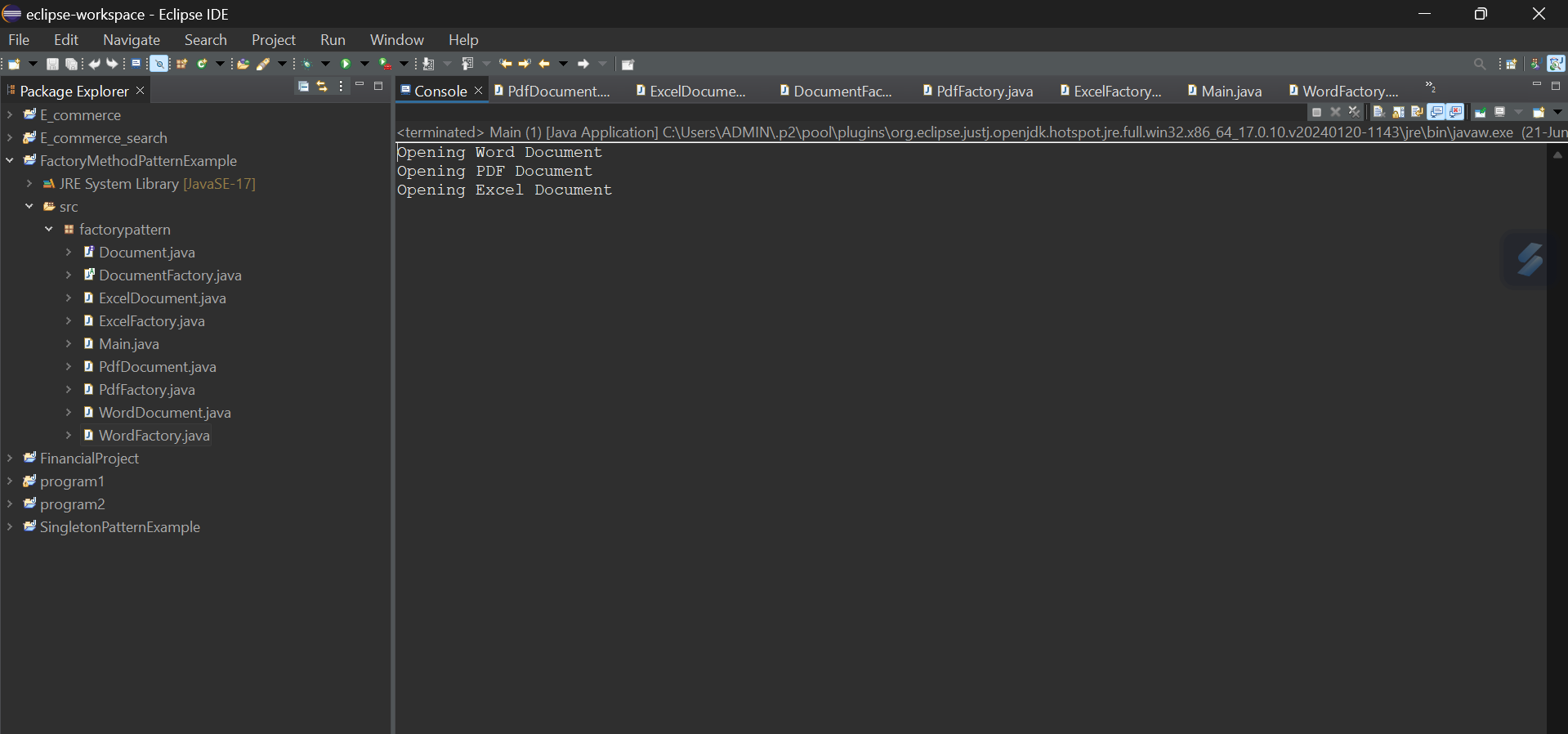
public class ExcelFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}}





*This project implements the* ***Factory Method Design Pattern*** *for a Document Management System. The goal is to create different types of documents—****Word, PDF, and Excel****—without tightly coupling the code to specific classes. We define a common interface Document and provide concrete implementations for each document type. An abstract class DocumentFactory declares the factory method createDocument(), and concrete factory classes (WordFactory, PdfFactory, ExcelFactory) override it to return specific document instances. The Main class demonstrates how each factory creates its respective document and calls the open() method. This structure promotes* ***flexibility, reusability, and ease of future extension****, fulfilling the requirements of the Factory Method pattern effectively.*