**Week-1**

**Design Pattern and Principles**

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

**Main.java**

public class Main {  
 public static void main(String[] args){  
 DocumentFactory wordFactory = new wordDocumentFactory();  
 IDocument wordDox = wordFactory.createDocument();  
 wordDox.open();  
   
 DocumentFactory pdfFactory = new pdfDocumentFactory();  
 IDocument pdfDoc = pdfFactory.createDocument();  
 pdfDoc.open();  
   
 DocumentFactory excelFactory = new excelDocumentFactory();  
 IDocument excelDoc = excelFactory.createDocument();  
 excelDoc.open();  
 }  
}

**wordDocumentFactory.java**

public class wordDocumentFactory extends DocumentFactory{  
 @Override  
 public IDocument createDocument() {  
 return new IDocument.wordDocument();  
 }  
}

**pdfDocumentFatory.java**

public class pdfDocumentFactory extends DocumentFactory {  
 @Override  
 public IDocument createDocument() {  
 return new IDocument.pdfDocument();  
 }  
}

**excelDocumentFactory.java**

public class excelDocumentFactory extends DocumentFactory {  
 @Override  
 public IDocument createDocument() {  
 return new IDocument.excelDocument();  
 }  
}

**IDocument.java**

public interface IDocument {  
 void open ();  
  
 class pdfDocument implements IDocument {  
 public void open(){  
 System.*out*.println("Opening Pdf");  
 }  
 }  
  
 class wordDocument implements IDocument {  
  
 public void open(){  
 System.*out*.println("Opening WordDox..");  
 }  
 }  
  
 class excelDocument implements IDocument {  
 public void open(){  
 System.*out*.println("Opening Execel..");  
 }  
  
 }  
}

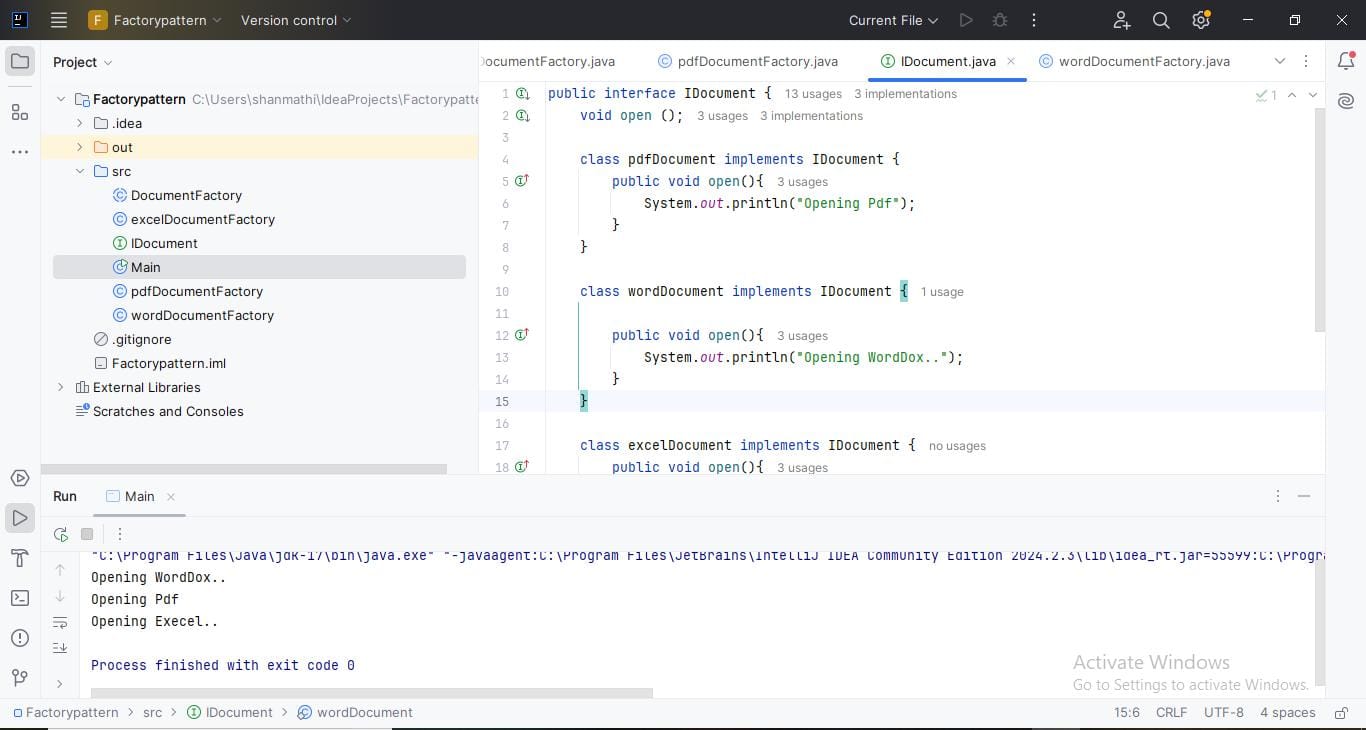
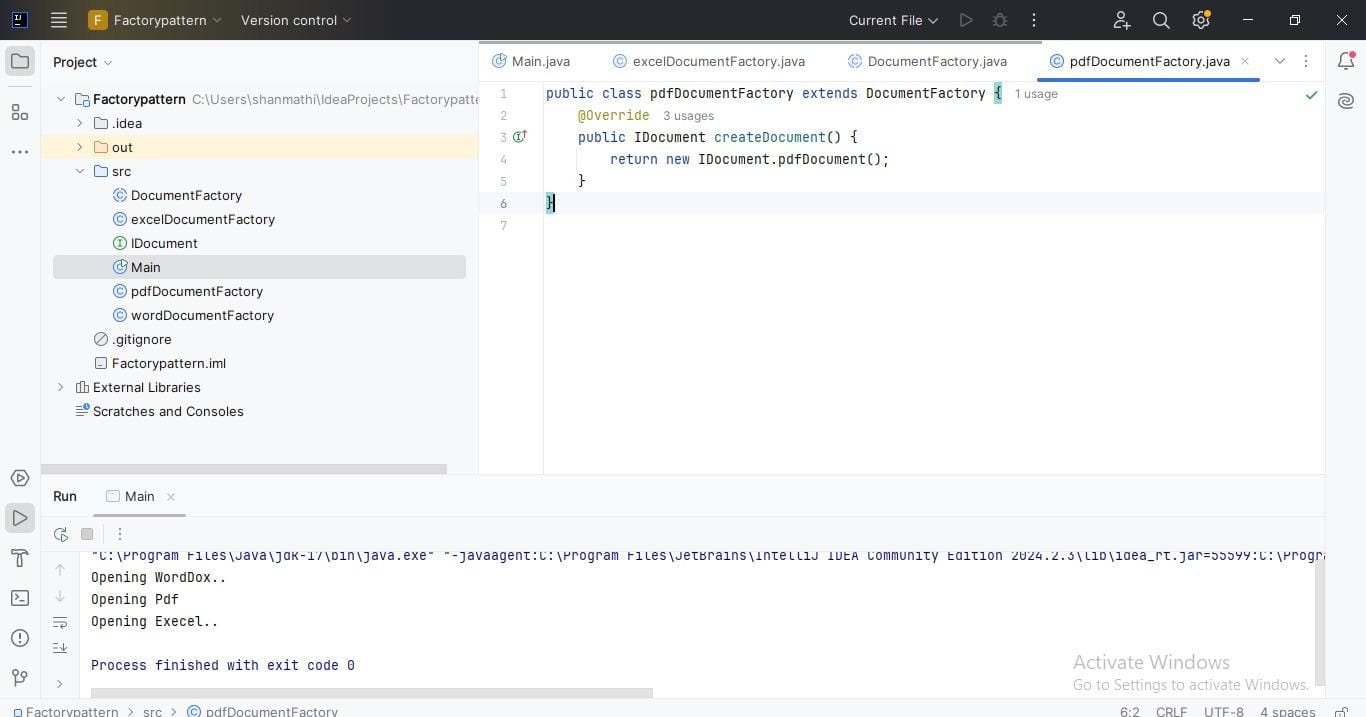
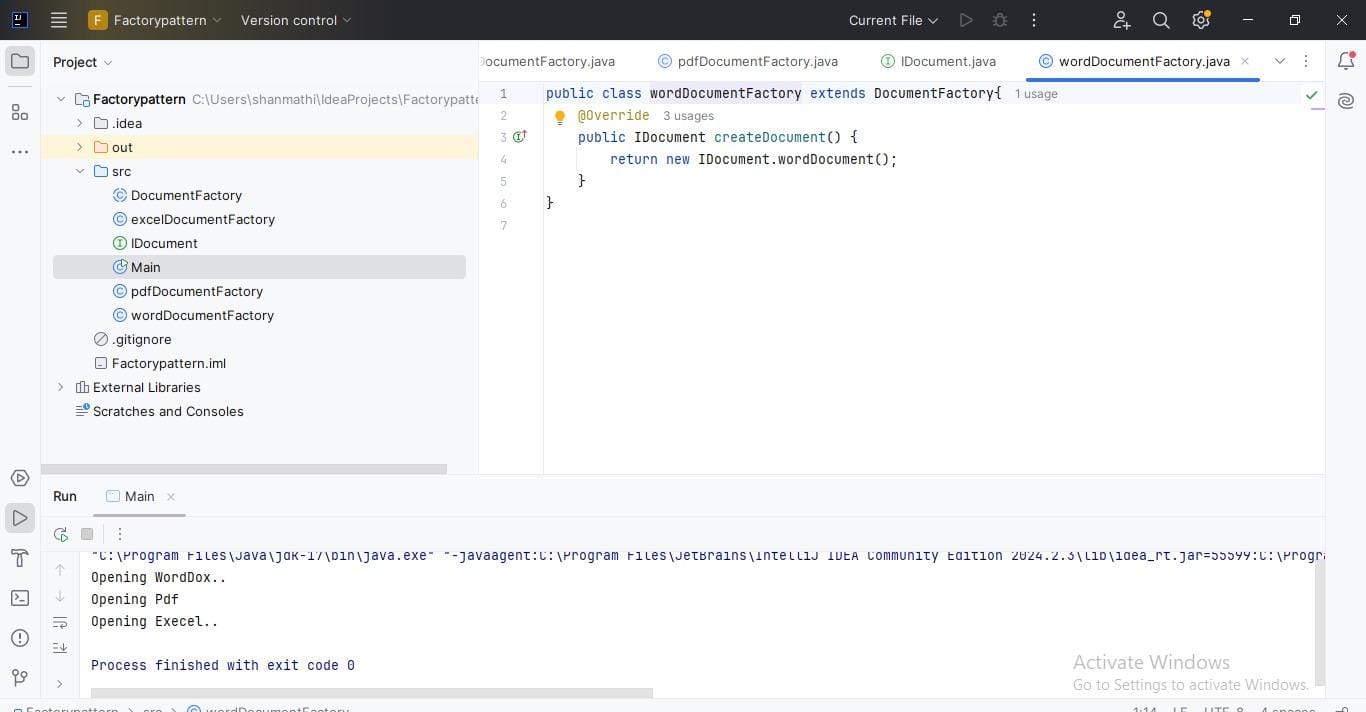
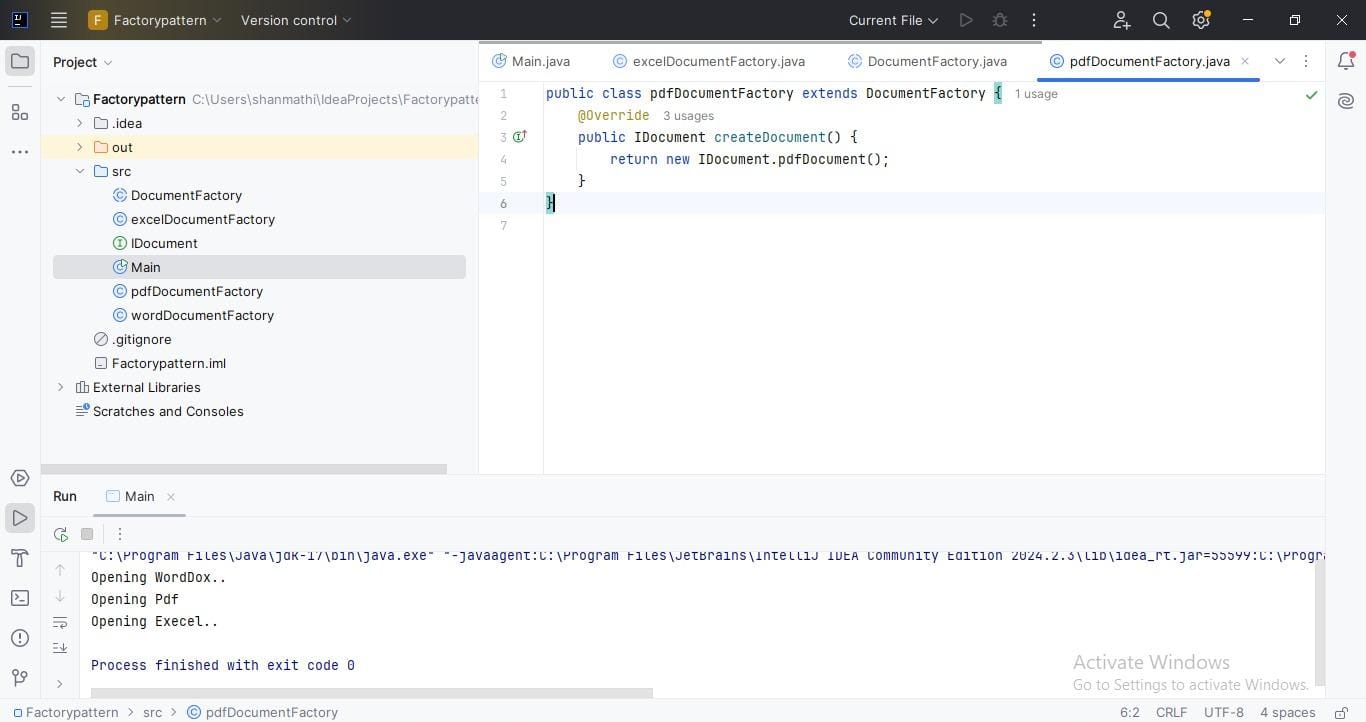
**DocumentFactory.java**

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

}

**Factory Method Implementation**

**A screenshot of a computer

Description automatically generated**