Implementing the Factory Method Pattern

***Document.java***

public interface Document {

void open();

}

***DocumentFactory.java***

public abstract class DocumentFactory {

public abstract Document createDocument();

}

***DocumentFactoryTest.java***

public class DocumentFactoryTest {

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();

}

}

***ExcelDocument.java***

public class ExcelDocument implements Document {

@Override

public void open() {

System.out.println("Opening an Excel document."); } }

***ExcelDocumentFactory.java***

public class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

***PdfDocument.java***

public class PdfDocument implements Document {

@Override

public void open() {

System.out.println("Opening a PDF document.");

}

}

***PdfDocumentFactory.java***

public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

***WordDocument.java***

public class WordDocument implements Document {

@Override

public void open() {

System.out.println("Opening a Word document.");

}

}

***WordDocumentFactory.java***

public class WordDocumentFactory extends DocumentFactory {

@Override

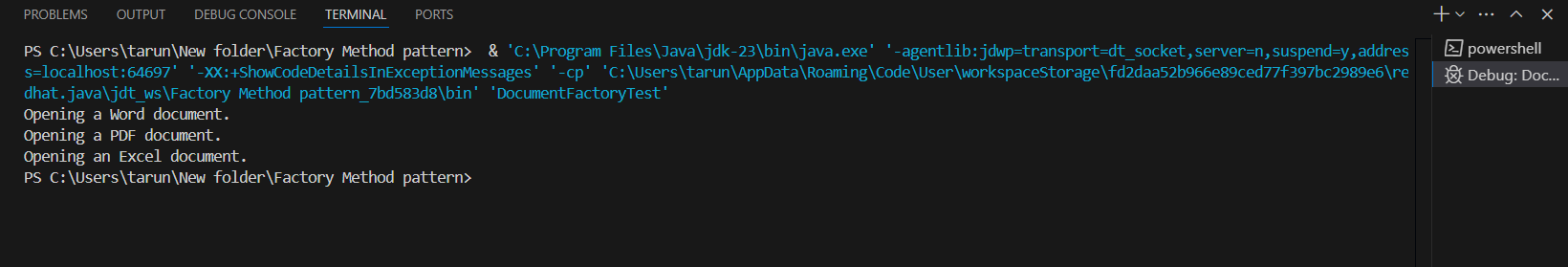
public Document createDocument() {

return new WordDocument();

}

}

**Output**

****