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

**Program:**

**1. FactoryMethodTest.java**

class FactoryMethodTest{

    public static void main(String args[]){

        DocumentFactory wordFactory = new WordDocumentFactory();

        Document word=wordFactory.createDocument();

        word.open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Document pdf=pdfFactory.createDocument();

        word.open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document excel=excelFactory.createDocument();

        word.open();

    }

}

interface Document{

    void open();

}

class WordDocument implements Document{

    public void open(){

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

    }}

class PdfDocument implements Document{

    public void open(){

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

    }

}

class ExcelDocument implements Document{

    public void open(){

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

    }

}

abstract class DocumentFactory   {

    public abstract Document createDocument();

}

class WordDocumentFactory extends DocumentFactory{

    public Document createDocument(){

        return new WordDocument();

    }

}

class PdfDocumentFactory extends DocumentFactory{

    public Document createDocument(){

        return new PdfDocument();

    }

}

class ExcelDocumentFactory extends DocumentFactory{

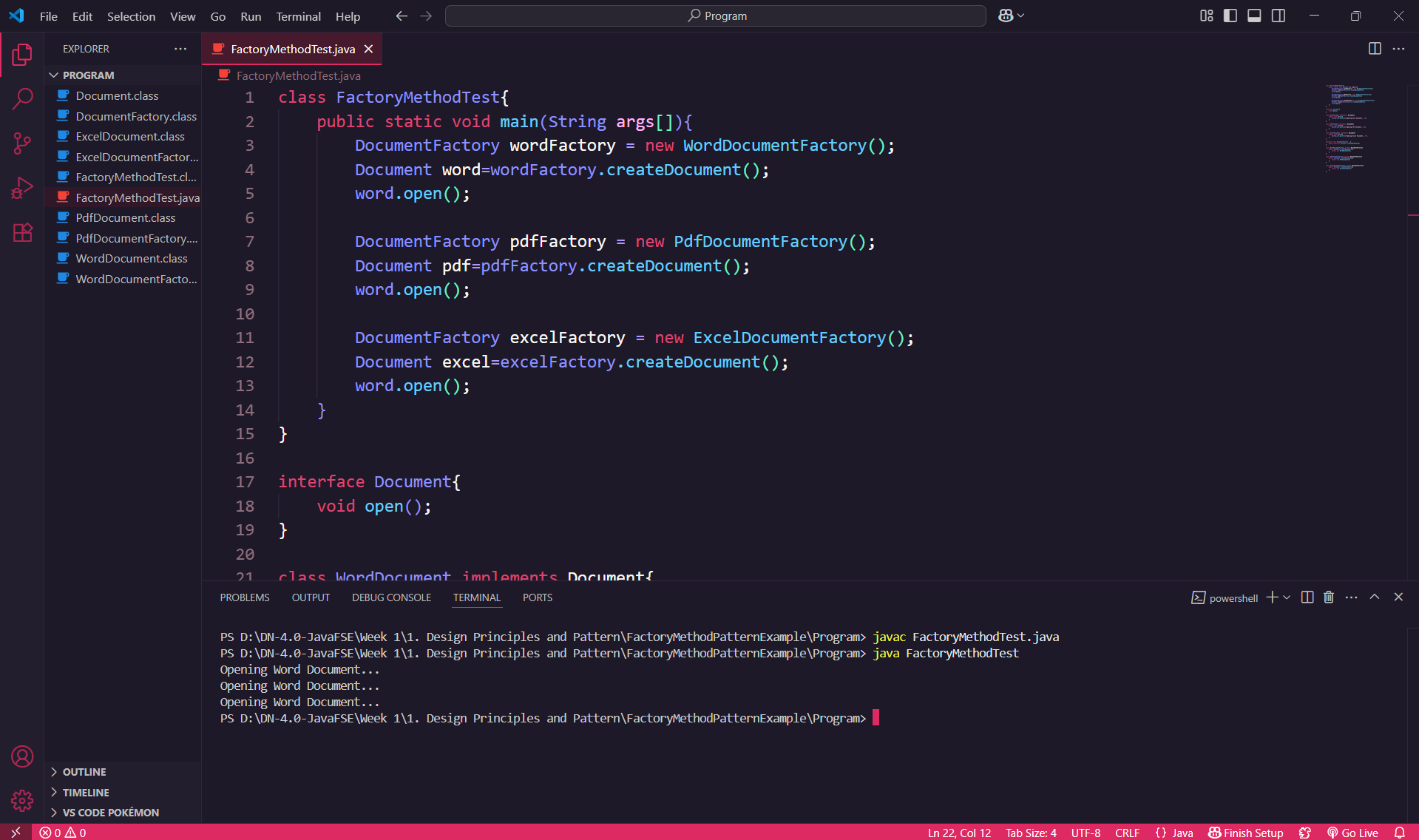
    public Document createDocument(){

        return new ExcelDocument();

    }

}

**Output:**



**Explanation:**

This program is based on the Factory Method Design Pattern. I created a Document interface and different classes like WordDocument, PdfDocument, and ExcelDocument that implement this interface. Then, I created an abstract class called DocumentFactory with a method createDocument(). I extended this factory class to create specific factories like WordDocumentFactory, PdfDocumentFactory, and ExcelDocumentFactory. In the main() method, I called createDocument() through each factory and used it to open the respective document. This allows me to create objects without directly using the new keyword for each document type, making the program more flexible and scalable.