Exercise 2: Implementing the Factory Method Pattern

---

1. Create a New Java Project

I created a java project named `FactoryMethodPatternExample`. The task is to create different types of documents (Word, Pdf, Excel) using the **Factory Method Pattern**.

---

2. Define Document Interface

We create a common interface called `Document` that all document classes will implement.

---

3. Create Concrete Document Classes

We create `WordDocument`, `PdfDocument`, and `ExcelDocument` classes that all implement the `Document` interface.

---

4. Implement the Factory Method Pattern

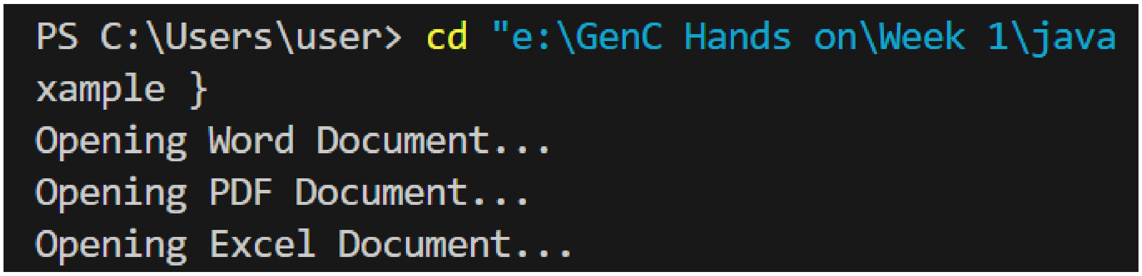
We use an abstract class **DocumentFactory** that has a method **createDocument()**, and then create 3 factory classes that extend it and return the correct document objects.

JAVA CODE

```
// Main class to run the program

public class FactoryMethodExample {

    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();
    }
}

// Document interface
interface Document {
    void open();
}

// WordDocument class
class WordDocument implements Document {
    public void open() {
        System.out.println("Opening Word Document...");
    }
}

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

// ExcelDocument class
class ExcelDocument implements Document {
    public void open() {
        System.out.println("Opening Excel Document...");
    }
}

// Abstract Factory
abstract class DocumentFactory {
    public abstract Document createDocument();
}
```

```
// Word factory
class WordDocumentFactory extends DocumentFactory {
    public Document createDocument() {
        return new WordDocument();
    }
}

// PDF factory
class PdfDocumentFactory extends DocumentFactory {
    public Document createDocument() {
        return new PdfDocument();
    }
}

// Excel factory
class ExcelDocumentFactory extends DocumentFactory {
    public Document createDocument() {
        return new ExcelDocument();
    }
}
```

Output:



```
PS C:\Users\user> cd "e:\GenC Hands on\Week 1\java
xample }
Opening Word Document...
Opening PDF Document...
Opening Excel Document...
```

Conclusion

We used the Factory Method design pattren to create diffrent types of documents. Each document class is separate, and each one has its own factory class to create it. This makes the code more organized and easy to extend in the future.
I wrote all the classes in a single file just for easy testing..