

## Exercise 2: Implementing the Factory Method Pattern

### Scenario:

You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

### Steps:

#### 1. Create a New Java Project:

- Create a new Java project named **FactoryMethodPatternExample**.

Created a Folder named FactoryMethodPatternExample.

#### 2. Define Document Classes:

- Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.

Code:

```
public interface Document
{
    void Open();
}
```

#### 3. Create Concrete Document Classes:

- Implement concrete classes for each document type that implements or extends the above interfaces or abstract classes.

Code for WordDocument:

```
public class WordDocument : Document
{
    public override void Open()
    {
        Console.WriteLine("Opening Word Document");
    }
}
```

Code for PdfDocument:

```
public class PdfDocument : Document
{
    public override void Open()
```

```

    {
        Console.WriteLine("Opening PDF Document");
    }
}

```

Code for ExcelDocument:

```

public class ExcelDocument : Document
{
    public override void Open()
    {
        Console.WriteLine("Opening Excel Document");
    }
}

```

#### 4. Implement the Factory Method:

- Create an abstract class **DocumentFactory** with a method **createDocument()**.

Code for DocumentFactory:

```

namespace Factory
{
    public abstract class DocumentFactory
    {
        public abstract Document CreateDocument();
    }
}

```

- Create concrete factory classes for each document type that extends DocumentFactory and implements the **createDocument()** method.

Code for WordDocumentFactory:

```

namespace Factory
{
    public class WordDocumentFactory : DocumentFactory
    {
        public override Document CreateDocument()
    }
}

```

```

        {
            return new WordDocument();
        }
    }
}

```

Code for PdfDocumentFactory:

```

namespace Factory
{
    public class PdfDocumentFactory : DocumentFactory
    {
        public override Document CreateDocument()
        {
            return new PdfDocument();
        }
    }
}

```

Code for ExcelDocumentFactory:

```

namespace Factory
{
    public class ExcelDocumentFactory : DocumentFactory
    {
        public override Document CreateDocument()
        {
            return new ExcelDocument();
        }
    }
}

```

## 5. Test the Factory Method Implementation:

- Create a test class to demonstrate the creation of different document types using the factory method.

Code for Test Class or the main method:

```
using System;
```

```
namespace Factory
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            // Create Word Document
```

```
            DocumentFactory wordFactory = new WordDocumentFactory();
```

```
            Document wordDoc = wordFactory.CreateDocument();
```

```
            wordDoc.Open();
```

```
            // Create PDF Document
```

```
            DocumentFactory pdfFactory = new PdfDocumentFactory();
```

```
            Document pdfDoc = pdfFactory.CreateDocument();
```

```
            pdfDoc.Open();
```

```
            // Create Excel Document
```

```
            DocumentFactory excelFactory = new ExcelDocumentFactory();
```

```
            Document excelDoc = excelFactory.CreateDocument();
```

```
            excelDoc.Open();
```

```
        }
```

```
    }
```

```
}
```

Output:

```
c:\Users\KIIT\Desktop\Personal For  
mplementingFactorypattern\Impleme  
Opening Word Document  
Opening PDF Document  
Opening Excel Document
```