**QUESTION:**

**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**.
2. **Define Document Classes:**
   * Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.
3. **Create Concrete Document Classes:**
   * Implement concrete classes for each document type that implements or extends the above interfaces or abstract classes.
4. **Implement the Factory Method:**
   * Create an abstract class **DocumentFactory** with a method **createDocument()**.
   * Create concrete factory classes for each document type that extends DocumentFactory and implements the **createDocument()** method.
5. **Test the Factory Method Implementation:**
   * Create a test class to demonstrate the creation of different document types using the factory method.

**CODE:**

//Product section

interface Document{

    void create();

}

class PdfDocument implements Document{

    public void create(){

        System.out.println("A PDF document has been created.");

    }

}

class WordDocument implements Document{

    public void create(){

        System.out.println("A Word document has been created.");

    }

}

class ExcelDocument implements Document{

    public void create(){

        System.out.println("An Excel document has been created.");

    }

}

//Creator section: Factory

abstract class DocumentFactory {

    abstract Document createDocument();

}

class PdfDocumentFactory extends DocumentFactory{

    public Document createDocument(){

        System.out.println("DocumentFactory chilld returns pdf object required to create PDF document.");

        return new PdfDocument();

    }

}

class WordDocumentFactory extends DocumentFactory{

    public Document createDocument(){

        System.out.println("DocumentFactory child returns word object required to create Word document.");

        return new WordDocument();

    }

}

class ExcelDocumentFactory extends DocumentFactory{

    public Document createDocument(){

        System.out.println("DocumentFactory child returns excel object required to create Excel document.");

        return new ExcelDocument();

    }

}

//Main class to test the Factory Method Pattern

public class FactoryMethodPatternExample{

    public static void main(String[] args) {

        DocumentFactory pdffactory = new PdfDocumentFactory();

        Document pdf = pdffactory.createDocument();

        pdf.create();

        DocumentFactory wordfactory = new WordDocumentFactory();

        Document word = wordfactory.createDocument();

        word.create();

        DocumentFactory excelfactory = new ExcelDocumentFactory();

        Document excel = excelfactory.createDocument();

        excel.create();

    }

}

**OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.