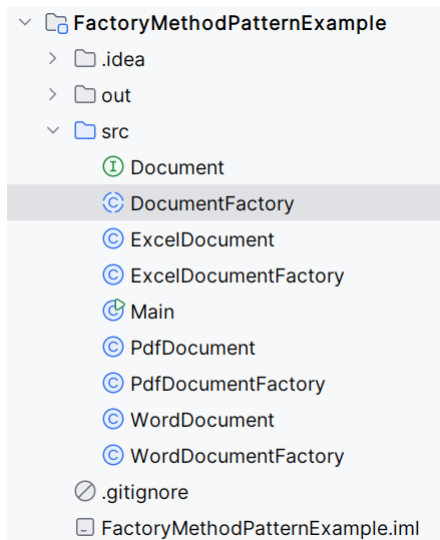


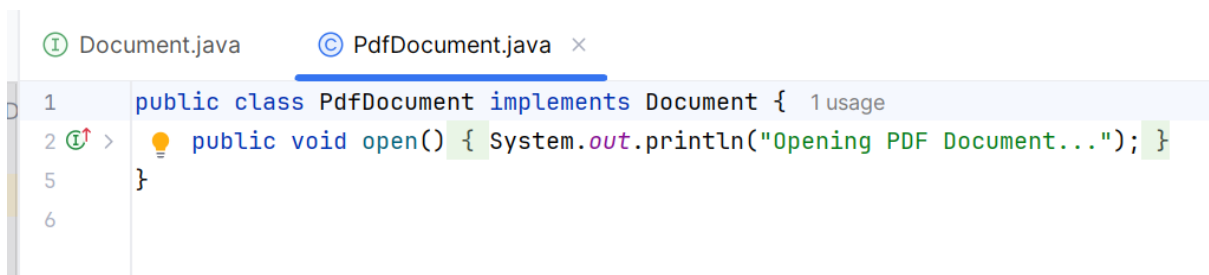
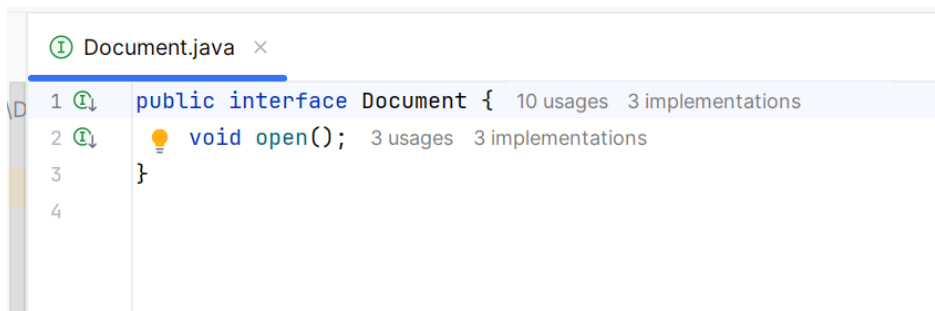
WEEK 1 : Design patterns and principles Handson

Exercise 2 : Factory pattern :

Folder Srtucture :



CODE :



```
Document.java PdfDocument.java WordDocument.java x
1 public class WordDocument implements Document { 1 usage
2   public void open() { System.out.println("Opening Word Document..."); }
5   }
6
```

```
Document.java PdfDocument.java WordDocument.java ExcelDocument.java x
1 public class ExcelDocument implements Document { 1 usage
2   public void open() { System.out.println("Opening Excel Document..."); }
5   }
6
```

```
Document.java x WordDocument.java ExcelDocument.java DocumentFactory.java x
1 public abstract class DocumentFactory { 6 usages 3 inheritors
2   public abstract Document createDocument(); 3 usages 3 implementations
3   }
4
```

```
PdfDocumentFactory.java x
1 public class PdfDocumentFactory extends DocumentFactory { 1 usage
2   public Document createDocument() { return new PdfDocument(); }
5   }
6
```

n control ▾ Current File ▾ ▶ ⚙️ ⋮ 🔍 👤 ⚙️

© PdfDocumentFactory.java © WordDocumentFactory.java ×

```
1 public class WordDocumentFactory extends DocumentFactory { 1 usage
2     public Document createDocument() { return new WordDocument(); }
5
6 }
```

© PdfDocumentFactory.java × © WordDocumentFactory.java © ExcelDocumentFactory.java ×

```
1 public class ExcelDocumentFactory extends DocumentFactory { 1 usage
2     public Document createDocument() { return new ExcelDocument(); }
5
6 }
```

🔍 Main.java ×

```
1 ▶ public class Main {
2 ▶     ⚠ public static void main(String[] args) {
3         DocumentFactory wordFactory = new WordDocumentFactory();
4         Document word = wordFactory.createDocument();
5         word.open();
6
7         DocumentFactory pdfFactory = new PdfDocumentFactory();
8         Document pdf = pdfFactory.createDocument();
9         pdf.open();
10
11        DocumentFactory excelFactory = new ExcelDocumentFactory();
12        Document excel = excelFactory.createDocument();
13        excel.open();
14    }
15 }
16
```

OUTPUT :



The screenshot shows the 'Run' window in IntelliJ IDEA. The title bar indicates the current configuration is 'Main'. The console output displays the command used to run the application, followed by three lines of output indicating the opening of Word, PDF, and Excel documents. The process concludes with the message 'Process finished with exit code 0'. On the left side of the console, there is a vertical toolbar with icons for running, debugging, and other actions.

```
Run Main x
"\"C:\\Program Files\\Java\\jdk-23\\bin\\java.exe\" \"-javaagent:C:\\Program Files\\JetBrains\\IntelliJ IDEA Community Edition 2025.1.2\\lib\\idea_rt.jar\"
Opening Word Document...
Opening PDF Document...
Opening Excel Document...
Process finished with exit code 0
```