Exercise 1: Implementing the Singleton Pattern

Code:

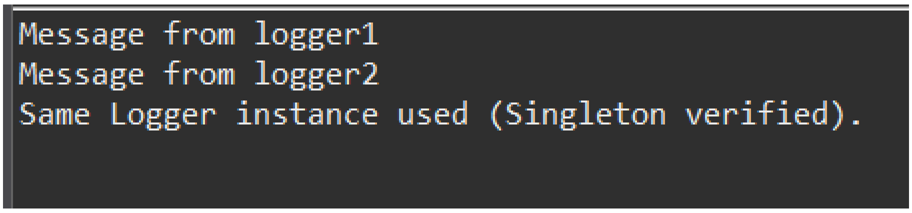
SingletonPattern.java

```
public class SingletonPattern {

        public static void main(String[] args) {
                Logger logger1 = Logger.getInstance();
    logger1.display("Message from logger1");

    Logger logger2 = Logger.getInstance();
    logger2.display("Message from logger2");

    if (logger1 == logger2) {
      System.out.println("Same Logger instance used (Singleton verified).");
    } else {
      System.out.println("Different Logger instances (Singleton failed).");
    }

        }

}
```

Logger.java

```
public class Logger {
 private static Logger logger=null;
 private Logger() {

 }
 public static Logger getInstance() {
  if(logger==null) {
   logger=new Logger();
  }
  return logger;
 }
 public void display(String str) {
  System.out.println(str);
 }

}
```

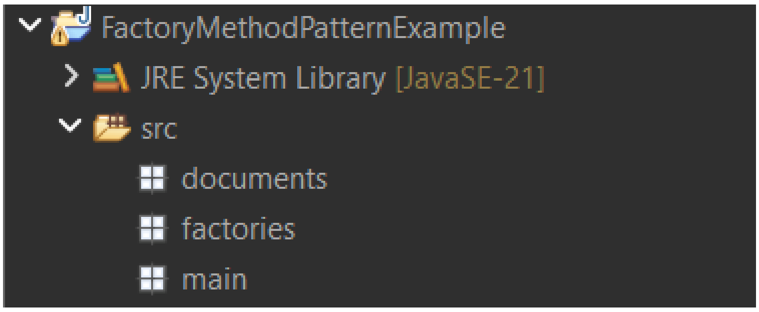
OUTPUT:



```
Message from logger1
Message from logger2
Same Logger instance used (Singleton verified).
```

Exercise 2: Implementing the Factory Method Pattern

File Structure:



```
FactoryMethodPatternExample
  JRE System Library [JavaSE-21]
  src
    documents
    factories
    main
```

Code:

Documents Package:

Document.java

```
package documents;

public interface Document {
 public void open();

}
```

ExcelDocument.java

```
package documents;

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

}
```

PdfDocument.java

```
package documents;

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

}
```

WordDocument.java

```
package documents;

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

}
```

Factories Package:

DocumentFactory.java

```
package factories;

import documents.Document;

public abstract class DocumentFactory {
  public abstract Document createDocument();

}
```

PdfDocumentFactory.java

```
package factories;

import documents.Document;
import documents.PdfDocument;

public class PdfDocumentFactory  extends DocumentFactory{
    @Override
    public Document createDocument() {
        return new PdfDocument();
    }

}
```

ExcelDocumentFactory.java

```
package factories;

import documents.Document;
import documents.ExcelDocument;

public class ExcelDocumentFactory extends DocumentFactory{
    @Override
    public Document createDocument() {
        return new ExcelDocument();
    }

}
```

WordDocumentFactory.java

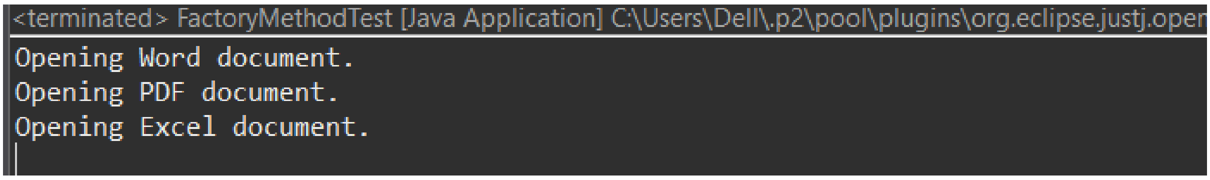
```
package factories;

import documents.Document;
import documents.WordDocument;

public class WordDocumentFactory extends DocumentFactory{
  @Override
  public Document createDocument() {
    return new WordDocument();
  }

}
```

Main Package:

FactoryMethodTest.java

```
package main;

import documents.Document;
import factories.DocumentFactory;
import factories.ExcelDocumentFactory;
import factories.PdfDocumentFactory;
import factories.WordDocumentFactory;

public 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();
     pdf.open();

     DocumentFactory excelFactory = new ExcelDocumentFactory();
     Document excel = excelFactory.createDocument();
     excel.open();
 }

}
```

OUTPUT:



```
<terminated> FactoryMethodTest [Java Application] C:\Users\Dell\.p2\pool\plugins\org.eclipse.justj.open
Opening Word document.
Opening PDF document.
Opening Excel document.
```