# **EXERCISE 1: IMPLEMENTING THE SINGLETON PATTERN**

**CODE:  
  
Logger.java**

**package** singleton;

**public** **class** Logger {

**private** **static** Logger *instance*;

**private** Logger() {

System.***out***.println("Logger instance created.");

}

**public** **static** Logger getInstance() {

**if** (*instance* == **null**) {

*instance* = **new** Logger();

}

**return** *instance*;

}

**public** **void** log(String message) {

System.***out***.println("[LOG]: " + message);

}

}

**Main.java**  
  
**package** singleton;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Logger logger1 = Logger.*getInstance*();

logger1.log("First log message");

Logger logger2 = Logger.*getInstance*();

logger2.log("Second log message");

**if** (logger1 == logger2) {

System.***out***.println("Both logger1 and logger2 are the same instance.");

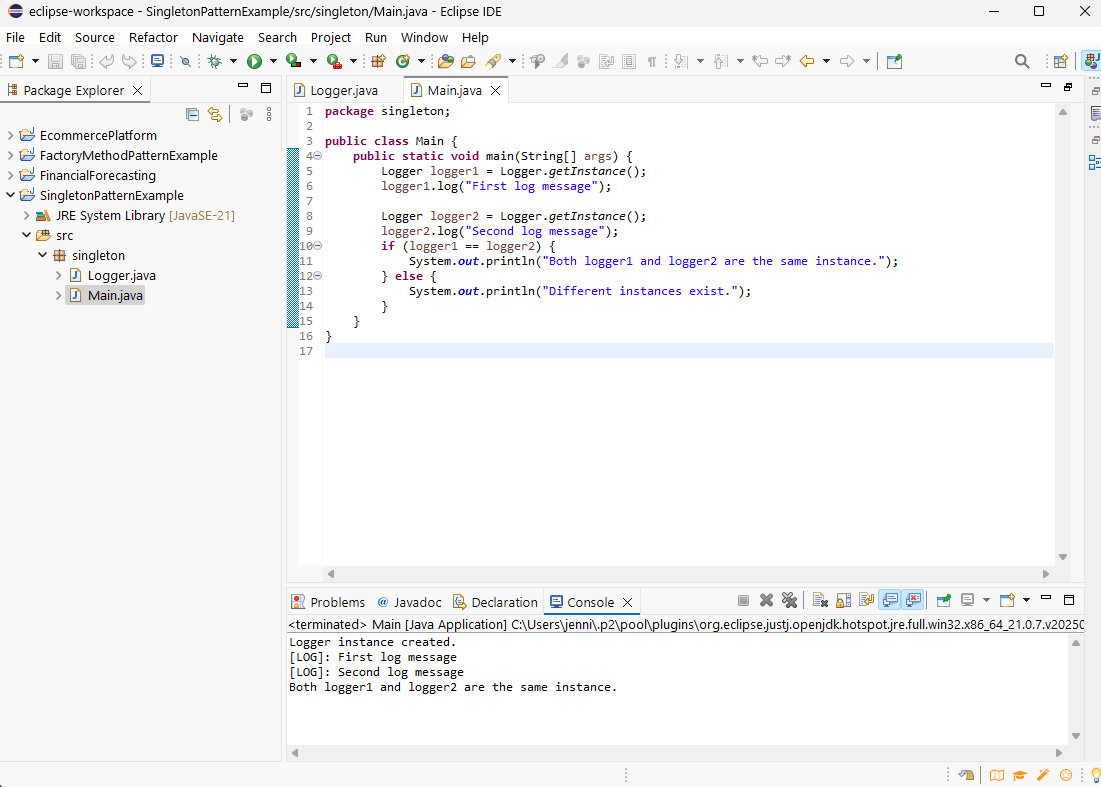
} **else** {

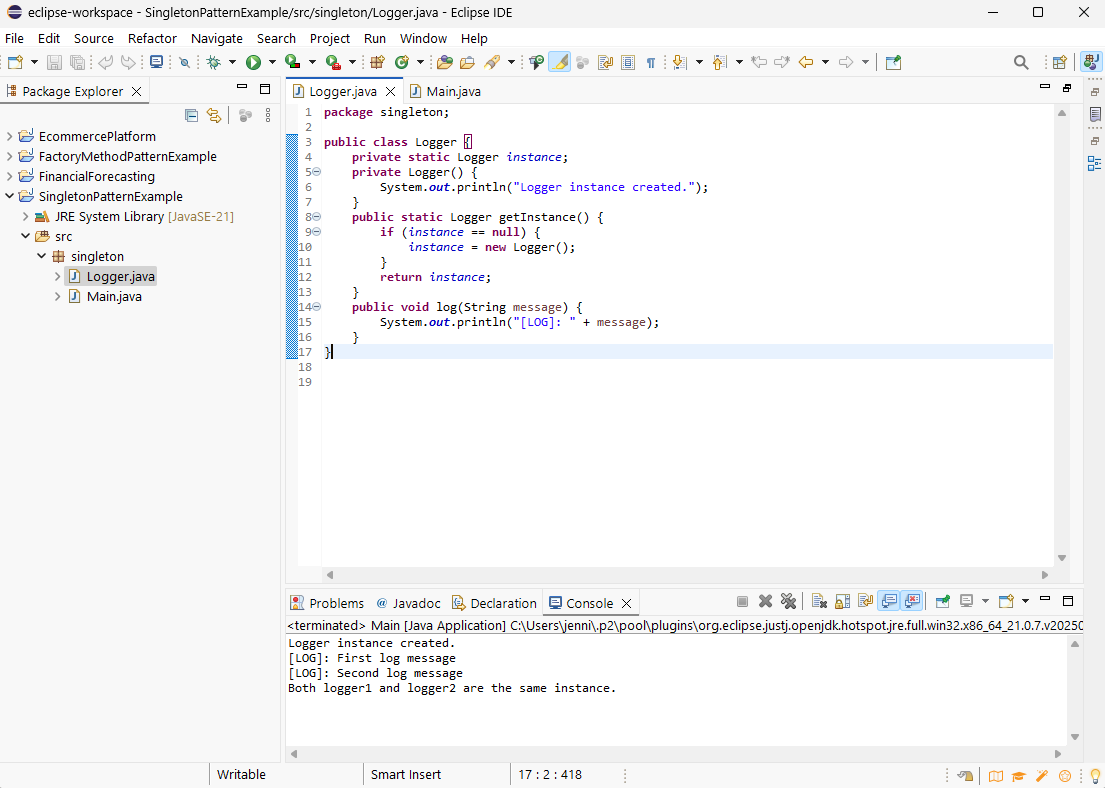
System.***out***.println("Different instances exist.");

}

}

}





# **EXERCISE 2: IMPLEMENTING THE FACTORY METHOD PATTERN**

**CODE:**

**Main.java**  
  
**package** factoryMethod;

**public** **class** Main {

**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.java**

**package** factoryMethod;

**public** **interface** Document {

**void** open();

}

**DocumentFactory.java**

**package** factoryMethod;

**public** **abstract** **class** DocumentFactory {

**public** **abstract** Document createDocument();

}

**ExcelDocument.java**

**package** factoryMethod;

**public** **class** ExcelDocument **implements** Document {

@Override

**public** **void** open() {

System.***out***.println("Opening an Excel document.");

}

**ExcelDocumentFactory.java**

**package** factoryMethod;

**public** **class** ExcelDocumentFactory **extends** DocumentFactory {

@Override

**public** Document createDocument() {

**return** **new** ExcelDocument();

}

}

**PdfDocument.java**

**package** factoryMethod;

**public** **class** PdfDocument **implements** Document {

@Override

**public** **void** open() {

System.***out***.println("Opening a PDF document.");

}

}

**PdfDocumentFactory.java**

**package** factoryMethod;

**public** **class** PdfDocumentFactory **extends** DocumentFactory {

@Override

**public** Document createDocument() {

**return** **new** PdfDocument();

}

}

**WordDocument.java**

**package** factoryMethod;

**public** **class** WordDocument **implements** Document {

@Override

**public** **void** open() {

System.***out***.println("Opening a Word document.");

}

}

**WordDocumentFactory.java**

**package** factoryMethod;

**public class** WordDocumentFactory **extends** DocumentFactory {

@Override

**public** Document createDocument() {

return new WordDocument();

}

}

