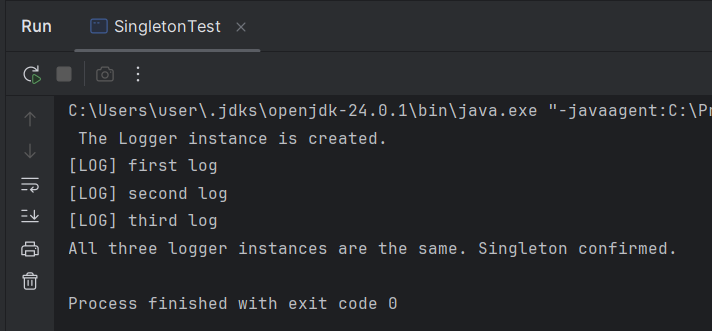
**WEEK-1**

**DESIGN ANALYSIS AND PATTERNS**

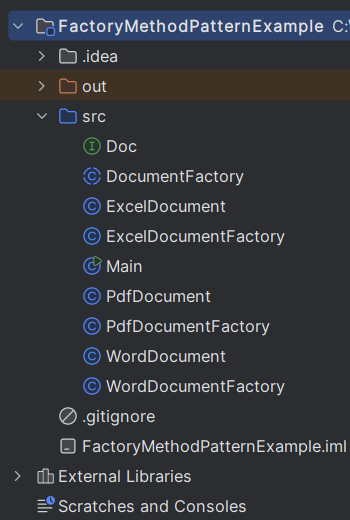
Exercise 1: Implementing the Singleton Pattern

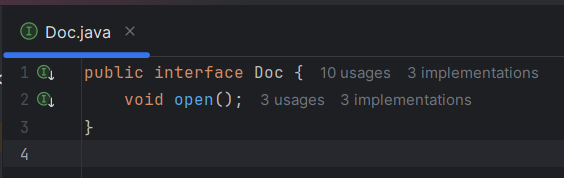
public class Logger {  
 private static Logger *instance*;  
 public Logger() {  
 System.*out*.println(" The Logger instance is created.");  
 }  
 public static Logger getInstance() {  
 if (*instance* == null) {  
 *instance* = new Logger();  
 }  
 return *instance*;  
 }  
 public void log(String message) {  
 System.*out*.println("[LOG] " + message);  
 }  
 }

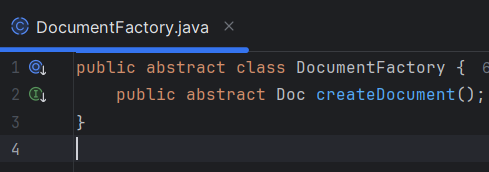
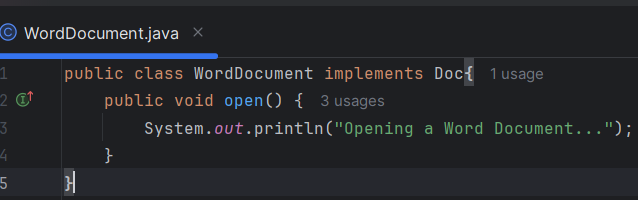
public class SingletonTest {  
 public static void main(String[] args) {  
 Logger logger1 = Logger.*getInstance*();  
 logger1.log("first log");  
  
 Logger logger2 = Logger.*getInstance*();  
 logger2.log("second log");  
  
 Logger logger3 = Logger.*getInstance*();  
 logger3.log("third log");  
  
 if (logger1 == logger2 && logger1 == logger3) {  
 System.*out*.println("All three logger instances are the same. Singleton confirmed.");  
 } else {  
 System.*out*.println("Different logger instances. Singleton failed.");  
 }  
 }  
 }

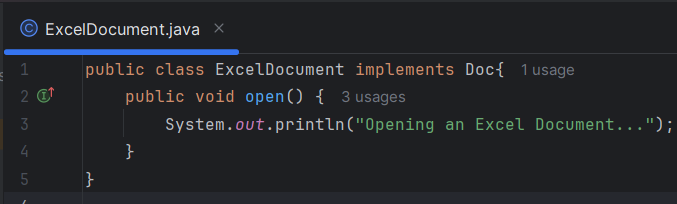
OUTPUT  


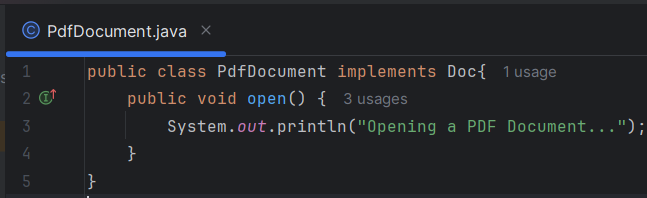
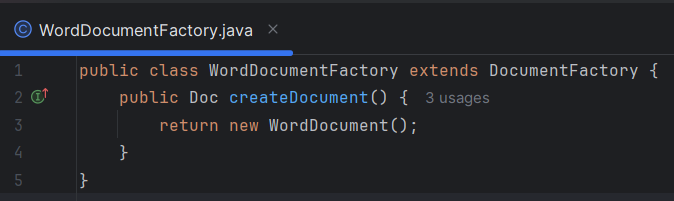
Exercise 2: Implementing the Factory Method Pattern

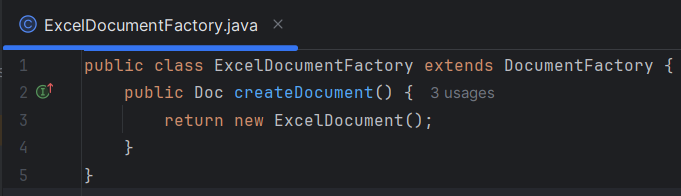
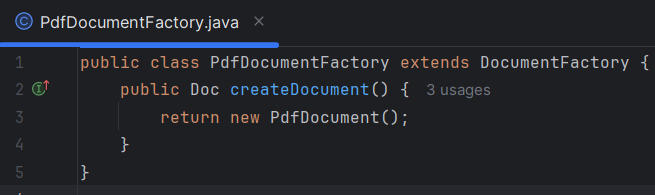








OUTPUT  
