Design Patterns and Principles

Exercise 1 – Singleton Pattern

Code:

//Logger.java

package SingletonPattern;

public class Logger{

private static Logger *instance*;

String str;

private Logger(String str)

{

this.str=str;

}

public static Logger getInstance(String str)

{

if(*instance*== null)

{

*instance* = new Logger(str);

}

return *instance*;

}

public void display()

{

System.***out***.println(str);

}

}

//Main.java

package SingletonPattern;

public class Main{

public static void main(String [] args)

{

Logger l1=Logger.*getInstance*("Created an instance for logger class, cannot create another anymore");

Logger l2=Logger.*getInstance*("Trying to create a new Instance");

l1.display();

l2.display();

} }

Output:

A computer screen shot of a program

AI-generated content may be incorrect.