Design patterns and principles

SuperSet ID:6412063

Exercise 1: Implementing the Singleton Pattern

Code:

// Logger.java

public class Logger {

    private static Logger instance;

    private Logger() {

        System.out.println("Logger initialized");

    }

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

    public void log(String message) {

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

    }

}

// LoggerTest.java

public class LoggerTest {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        logger1.log("This is the first log message.");

        System.out.println(logger1.hashCode());

        Logger logger2 = Logger.getInstance();

        logger2.log("This is the second log message.");

        System.out.println(logger2.hashCode());

        if (logger1 == logger2) {

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

        } else {

            System.out.println("Different instances were created.");

        }

    }

}

Output:

A close-up of a text

Description automatically generated