Java Design Patterns - Exercises 1 to 11

## Exercise 1: Singleton Pattern – Logger

// Logger.java

public class Logger {

private static Logger instance;

private Logger() {}

public static Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

public void log(String message) {

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

}

}

// Main.java

public class Main {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

Logger logger2 = Logger.getInstance();

logger1.log("First message");

System.out.println(logger1 == logger2); // true

}

}

## Exercise 2: Factory Method Pattern – Documents

// Document.java

public interface Document {

void open();

}

// WordDocument.java

public class WordDocument implements Document {

public void open() {

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

}

}

// DocumentFactory.java

public abstract class DocumentFactory {

public abstract Document createDocument();

}

// WordFactory.java

public class WordFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

// Main.java

public class Main {

public static void main(String[] args) {

DocumentFactory factory = new WordFactory();

Document doc = factory.createDocument();

doc.open();

}

}

## Exercise 3: Builder Pattern – Computer

// Computer.java

public class Computer {

private String cpu, ram, storage;

private Computer(Builder builder) {

this.cpu = builder.cpu;

this.ram = builder.ram;

this.storage = builder.storage;

}

public static class Builder {

private String cpu, ram, storage;

public Builder setCpu(String cpu) {

this.cpu = cpu; return this;

}

public Builder setRam(String ram) {

this.ram = ram; return this;

}

public Builder setStorage(String storage) {

this.storage = storage; return this;

}

public Computer build() {

return new Computer(this);

}

}

public String toString() {

return "CPU: " + cpu + ", RAM: " + ram + ", Storage: " + storage;

}

}

// Main.java

public class Main {

public static void main(String[] args) {

Computer pc = new Computer.Builder()

.setCpu("Intel i7")

.setRam("16GB")

.setStorage("512GB SSD")

.build();

System.out.println(pc);

}

}