**Exercise 6: Implementing the Proxy Pattern**

**Scenario:**

You are developing an image viewer application that loads images from a remote server. Use the Proxy Pattern to add lazy initialization and caching.

**Steps:**

**1. Create a New Java Project:**

* **Project Name**: ProxyPatternExample
* Use your preferred IDE (like IntelliJ IDEA, Eclipse, or NetBeans) to create a new Java project.

**2. Define Subject Interface:**

**Image Interface**:

* Create an interface Image with a method display().

public interface Image {

void display();

}

**3. Implement Real Subject Class:**

**RealImage Class**:

* Create a class RealImage that implements Image and simulates loading an image from a remote server.

public class RealImage implements Image {

private String fileName;

public RealImage(String fileName) {

this.fileName = fileName;

loadFromRemoteServer();

}

private void loadFromRemoteServer() {

System.out.println("Loading image from remote server: " + fileName);

}

@Override

public void display() {

System.out.println("Displaying " + fileName);

}

}

**4. Implement Proxy Class:**

**ProxyImage Class**:

* Create a class ProxyImage that implements Image and holds a reference to RealImage.
* Implement lazy initialization and caching in ProxyImage.

public class ProxyImage implements Image {

private RealImage realImage;

private String fileName;

public ProxyImage(String fileName) {

this.fileName = fileName;

}

@Override

public void display() {

if (realImage == null) {

realImage = new RealImage(fileName); // Lazy initialization

}

realImage.display(); // Delegates the display call to RealImage

}

}

**5. Test the Proxy Implementation:**

**TestProxyPattern Class**:

* Create a test class to demonstrate the use of ProxyImage to load and display images.

public class TestProxyPattern {

public static void main(String[] args) {

Image image1 = new ProxyImage("photo1.jpg");

Image image2 = new ProxyImage("photo2.jpg");

// Image will be loaded and displayed

image1.display();

System.out.println("");

// Image will not be loaded again; it will be retrieved from the cache and displayed

image1.display();

System.out.println("");

// Image will be loaded and displayed

image2.display();

}

}