**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:**
   * Create a new Java project named **ProxyPatternExample**.
2. **Define Subject Interface:**
   * Create an interface Image with a method **display()**.
3. **Implement Real Subject Class:**
   * Create a class **RealImage** that implements Image and loads an image from a remote server.
4. **Implement Proxy Class:**
   * Create a class **ProxyImage** that implements Image and holds a reference to RealImage.
   * Implement lazy initialization and caching in **ProxyImage**.
5. **Test the Proxy Implementation:**
   * Create a test class to demonstrate the use of **ProxyImage** to load and display images.

**Solution:**

using System;

public interface IImage

{

void Display();

}

public class RealImage : IImage

{

private string filename;

public RealImage(string filename)

{

this.filename = filename;

LoadFromServer();

}

private void LoadFromServer()

{

Console.WriteLine($"Loading image from server: {filename}");

}

public void Display()

{

Console.WriteLine($"Displaying image: {filename}");

}

}

public class ProxyImage : IImage

{

private string filename;

private RealImage realImage;

public ProxyImage(string filename)

{

this.filename = filename;

}

public void Display()

{

if (realImage == null)

{

realImage = new RealImage(filename);

}

realImage.Display();

}

}

public class Program

{

public static void Main()

{

IImage image1 = new ProxyImage("pic1.jpg");

image1.Display();

image1.Display();

IImage image2 = new ProxyImage("pic2.jpg");

image2.Display();

}

}

