**ProxyPatternExample**

**Image.java**

package com.proxy;

public interface Image {

    void display();

}

**Main.java**

package com.proxy;

public class Main {

    public static void main(String[] args) {

        Image img1 = new ProxyImage("nature.jpg");

        Image img2 = new ProxyImage("space.png");

        // First time loading

        img1.display();

        System.out.println();

        // Cached loading

        img1.display();

        System.out.println();

        // First time loading another image

        img2.display();

    }

}

**ProxyImage.java**

package com.proxy;

public class ProxyImage implements Image {

    private String filename;

    private RealImage realImage;

    public ProxyImage(String filename) {

        this.filename = filename;

    }

    @Override

    public void display() {

        if (realImage == null) {

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

        } else {

            System.out.println("Image retrieved from cache: " + filename);

        }

        realImage.display();

    }

}

**RealImage.java**

package com.proxy;

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 image: " + filename);

    }

}

