**Exercise 6: Implementing the Proxy Pattern**

**Code:**

**Image Interface: -**

public interface Image {  
 void display();  
}

**RealImage Class: -**

public class RealImage implements Image{  
 String filename;  
 public RealImage(String filename){  
 this.filename=filename;  
 loadsFromRemoteServer();  
 }  
 private void loadsFromRemoteServer(){  
 System.*out*.println("Loading image from the remote server: "+filename);  
 }  
 @Override  
 public void display() {  
 System.*out*.println("Displaying the Image: "+filename);  
 }  
}

**ProxyImage Class: -**

import java.util.HashMap;  
import java.util.Map;

public class ProxyImage implements Image{  
 private String filename;  
 private static Map<String,RealImage> *imgCache*=new HashMap<>();  
 public ProxyImage(String filename){  
 this.filename=filename;  
 }

@Override  
 public void display() {  
 RealImage realImage=*imgCache*.get(filename);  
 if(realImage==null){  
 System.*out*.println("Image is not cached.Creating RealImage");  
 realImage=new RealImage(filename);  
 *imgCache*.put(filename,realImage);  
 }  
 else{  
 System.*out*.println("Image cached.Displaying the cached image "+filename);  
 }  
 realImage.display();  
 }  
}

**TestClass (Main class): -**

public class TestClass {  
 public static void main(String[] args){  
 Image img1=new ProxyImage("photo1.jpeg");  
 Image img2=new ProxyImage("photo2.jpeg");  
 Image img3=new ProxyImage("photo1.jpeg");  
 img1.display();  
 img2.display();  
 img3.display();  
 img1.display();  
 }  
}

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**