Proxy Pattern

The **proxy pattern** is used to create a class that represents the functionality of another class. It is a **structural pattern**. Using the proxy pattern allows us to reduce the memory footprint of our program.

public interface *Image* {  
 void display();  
}  
  
public class RealImage implements *Image* {  
 private String fileName;  
  
 public RealImage(String fileName) {  
 this.fileName = fileName;  
 loadFromDisk(fileName);  
 }  
  
 @Override  
 public void display() { System.*out*.println("Displaying " + fileName); }  
  
 private void loadFromDisk(String fileName) {  
 System.*out*.println("Loading " + fileName);  
 }  
}  
  
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);  
 realImage.display();  
 }  
}

JAVA

