Design Patterns - Proxy Pattern

设计模式-代理模式

In proxy pattern, a class represents functionality of another class. This type of design pattern comes under structural pattern.

In proxy pattern, we create object having original object to interface its functionality to outer world.

在代理模式中,我们使用一个类来代表另一个类的功能。这种类型的设计模式属于结构型设计模式的一种。

在代理模式中，我们将创建一个对象，该对象在在接口中持有原始对象，以对外部提供它的功能。

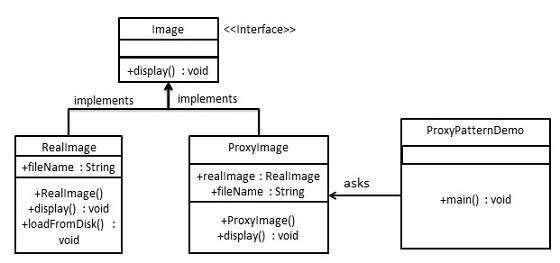
Implementation

We are going to create an *Image* interface and concrete classes implementing the *Image* interface. *ProxyImage* is a a proxy class to reduce memory footprint of *RealImage* object loading.

*ProxyPatternDemo*, our demo class, will use *ProxyImage* to get an *Image*object to load and display as it needs.

我们将创建一个Image接口并且创建一个具体类实现Image接口。ProxyImage是一个代理类，使用它可以减少RealImage对象加载带来的内存占用。

ProxyPatternDemo，我们的demo类将使用ProxyImage类去加载一个Image对象并且如果需要可以展示它。



Step 1

Create an interface.

创建一个接口

*Image.java*

public interface Image {

void display();

}

Step 2

Create concrete classes implementing the same interface.

创建具体类实现Image接口

*RealImage.java*

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);

}

}

*ProxyImage.java*

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();

}

}

Step 3

Use the *ProxyImage* to get object of *RealImage* class when required.

使用ProxyImage在需要的时候获得RealImage类的对象

*ProxyPatternDemo.java*

public class ProxyPatternDemo {

public static void main(String[] args) {

Image image = new ProxyImage("test\_10mb.jpg");

//image will be loaded from disk

image.display();

System.out.println("");

//image will not be loaded from disk

image.display();

}

}

Step 4

Verify the output.

校验输出

Loading test\_10mb.jpg

Displaying test\_10mb.jpg

Displaying test\_10mb.jpg

http://www.tutorialspoint.com/design\_pattern/proxy\_pattern.htm