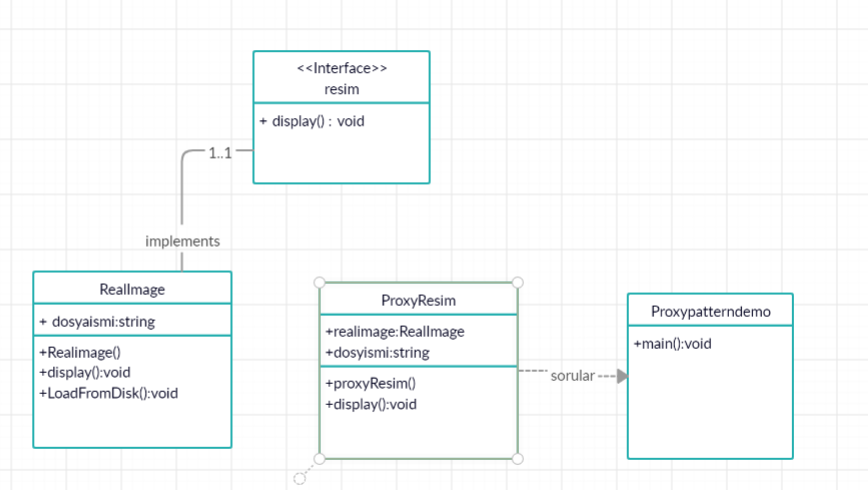
**PROXY TASARIM DESENİ**

Proxy modelinde, sınıf başka bir sınıfın işlevselliğini temsil eder. Bu tip tasarım deseni yapısal desen altındadır. Proxy modelinde, işlevselliğini dış dünyaya arayüzlemek için orijinal nesneye sahip nesne yaratırız.

ÖRNEK UYGULAMA

Bir Image arayüzü ve Image arayüzünü uygulayan somut sınıflar oluşturacağız. ProxyImage, RealImage nesnesi yüklemesinin bellek kapladığı alanı azaltmak için kullanılan bir proxy sınıfıdır.Demo sınıfımız olan ProxyPatternDemo, bir Image nesnesini yüklemek ve gerektiğinde görüntülemek için ProxyImage kullanacaktır.

CLASS DİYAGRAMI :



1.ADIM :

Yeni bir interface oluşturuyoruz.

Dosya adı : image.java

Public interface Image {

Void display();

}

2.ADIM :

Aynı arayüzü uygulayan somut sınıflar oluşturuyoruz.

Dosya adı : RealImage.java

Public class RealImage implements Image {

Private String dosyaismi;

Public RealImage(String dosyaismi){

This.dosyaismi = dosyaismi;

LoadFromDisk(dosyaismi);

}

@Override

Public void display(){

System.out.println(“durdur” + dosyaismi);

}

Private void loadFromDisk(String dosyaismi){

System.out.println(“devam” + dosyaismi);

}

}

Dosya adı : proxyImage.java

Public class ProxyImage implements Image{

Private RealImage realImage;

Private String dosyaismi;

Public ProxyImage(String dosyaismi){

This.dosyaismi = dosyaismi;

}

@Override

Public void display(){

If()realImage == null){

RealImage = new RealImage(dosyaismi);

}

RealImage.display();

}

}

3.ADIM :

Gerrektiğinde RealImage sınıfının nesnesini almak için ProxyImage kullanır.

Dosya ismi : ProxyPatternDemo.java

Public class ProxyPatternDemo {

Public static void main(String[] args) {

Image image = new ProxyImage(“deneme\_10mb.jpg”);

// resimi diskten yüklemek için

image.display();

System.out.println(“”);

//resim diskten yüklenmeyecekse

image.display();

}

}

4.ADIM :

Bu adımda çıktıyı doğrularız.

Devam deneme\_10mb.jpg

Durdur deneme\_10mb.jpg

.

Durdur deneme\_10mb.jpg