

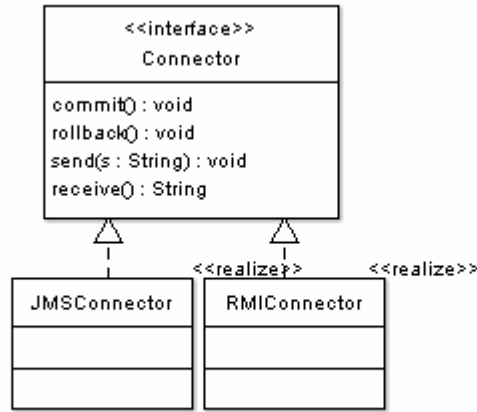
Interface Segregation Principle (ISP)

Arayüz Ayırma Prensibi

KurumsalJava.com

Özcan Acar
Bilgisayar Mühendisi
<http://www.ozcanacar.com>

Birbiriyle ilişkili olmayan birçok metodu ihtiva eden büyük bir interface sınıfı yerine, birbiriyle ilişkili (cohesive) metotların yer aldığı birden fazla interface sınıfı daha makbuldür. ISP uygulanmadığı takdirde, birden fazla sorumluluğu olan interface sınıflar oluşur. Zaman içinde yüklenen yeni sorumluluklarla bu interface sınıflar daha da büyür ve kontrol edilemez bir hale gelebilir. Bunun bir örneğini resim 1 de görmekteyiz.



Resim 1 Connector interface sınıfı bünyesinde gereğinden fazla metot barındırmaktadır

Resim 1 de yer alan *Connector* interface sınıfı bünyesinde JMS¹ (Java Message Service) için gerekli iki metot bulunmaktadır: *commit* ve *rollback*. Bu metodların RMIConnector implementasyonunda implemente edilmeleri anlamsız olur. Büyük bir ihtimalle RMIConnector sınıfında bu metodların implementasyonu kod 1 de yer aldığı şekilde olacaktır.

Kod 1 RMIConnector.java

```
package shop;

public class RMIConnector implements Connector
{
    public void commit()
    {
        throw new RuntimeException("not implemented");
    }

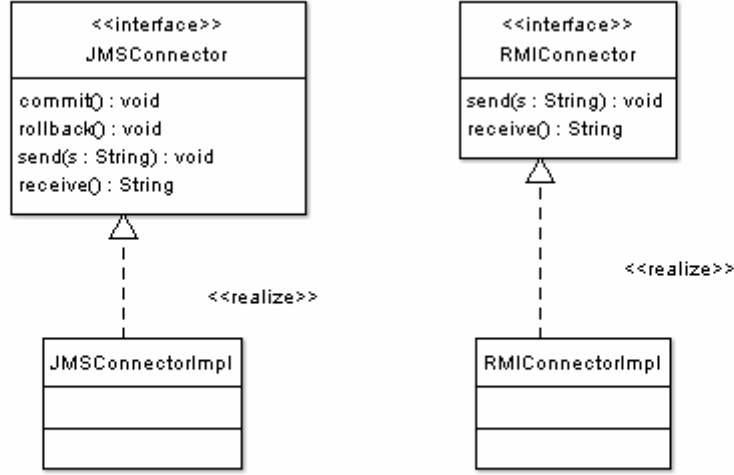
    public void rollback()
    {
        throw new RuntimeException("not implemented");
    }
}
```

Kod 1 de yer alan yapı LSP² ile uyumlu değildir. *Connector* sınıfını kullananlar *RuntimeException* oluşabileceğini göz önünde bulundurmak zorunda bırakılmaktadırlar.

¹ Bakınız: <http://java.sun.com/products/jms/>

² Bakınız <http://www.kurumsaljava.com/2009/10/29/liskov-substitution-principle-lsp-%e2%80%93-liskovun-yerine-gecme-prensibi/>

ISP uygulandıđı taktirde resim 1 de yer alan *Connector* interface sınıfını yok ederek, yerine sorumluluk alanları belli iki yeni interface sınıf oluşturabiliriz. Resim 2 de bu çözüm yer almaktadır.



Resim 2 Connector interface sınıfı bölünerek ISP'ye uygun iki yeni interface sınıf oluşturuldu

Programcı olarak bir interface sınıfa birden fazla sorumluluk yüklememek için, interface sınıfın sistemdeki görevini iyi anlamamız gerekmektedir. Bu sadece bir sorumluluk alanı olan bir interface sınıf oluşturmamızı kolaylaştıracaktır.

EOF (End Of Fun)
Özcan Acar