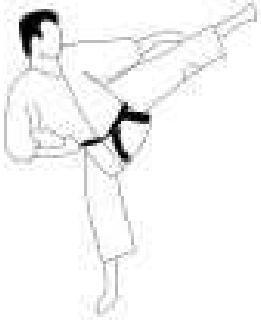


Kurumsal Java **Akademisi**

Java Enterprise Architecture



Bilge Programcı

işini bilir... kendini devamlı geliştirir...

Nasıl Uzman Yazılımcı Olunur?

Özcan Acar

acar@unitedinter.net

<http://www.ozcanacar.com>

<http://www.kurumsaljava.com>

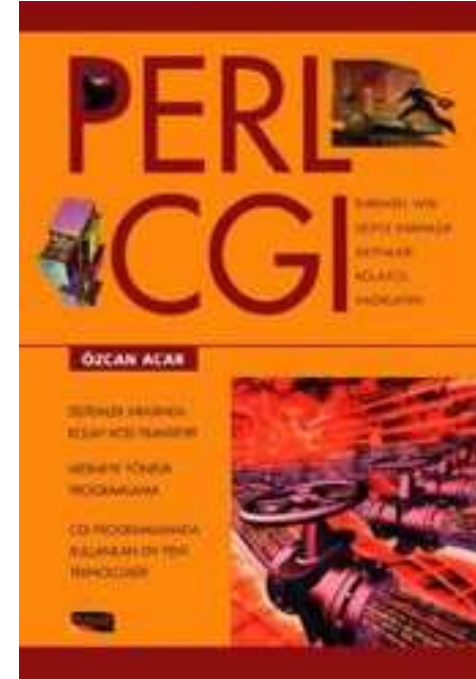
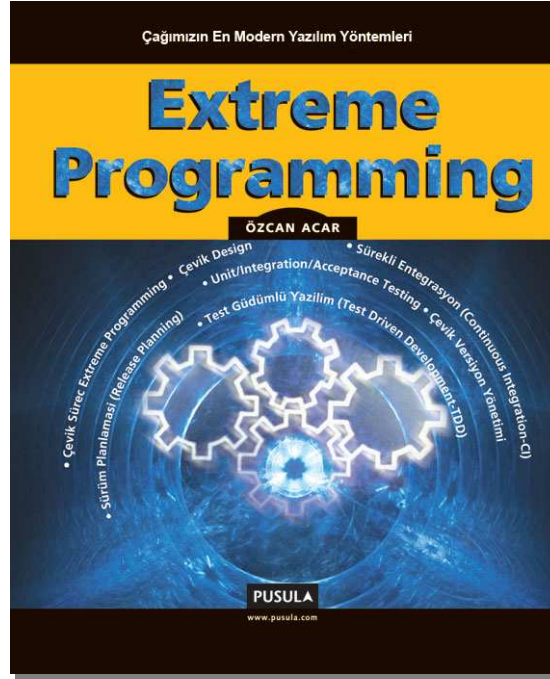
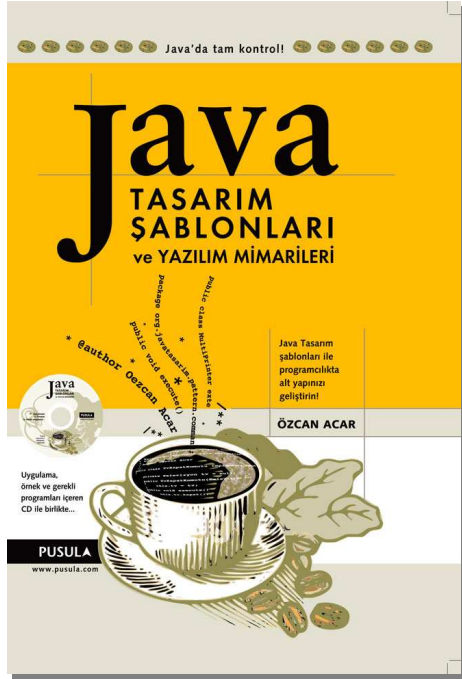
Özcan Acar Hakkında

- **1974 İzmir doğumlu**
- **Bilgisayar Mühendisi**
- **Sun Certified Enterprise Architect (SCEA 5)**
- **Sun Java Champion**
- **KurumsalJava.com ve JugTR.org kurucusu**

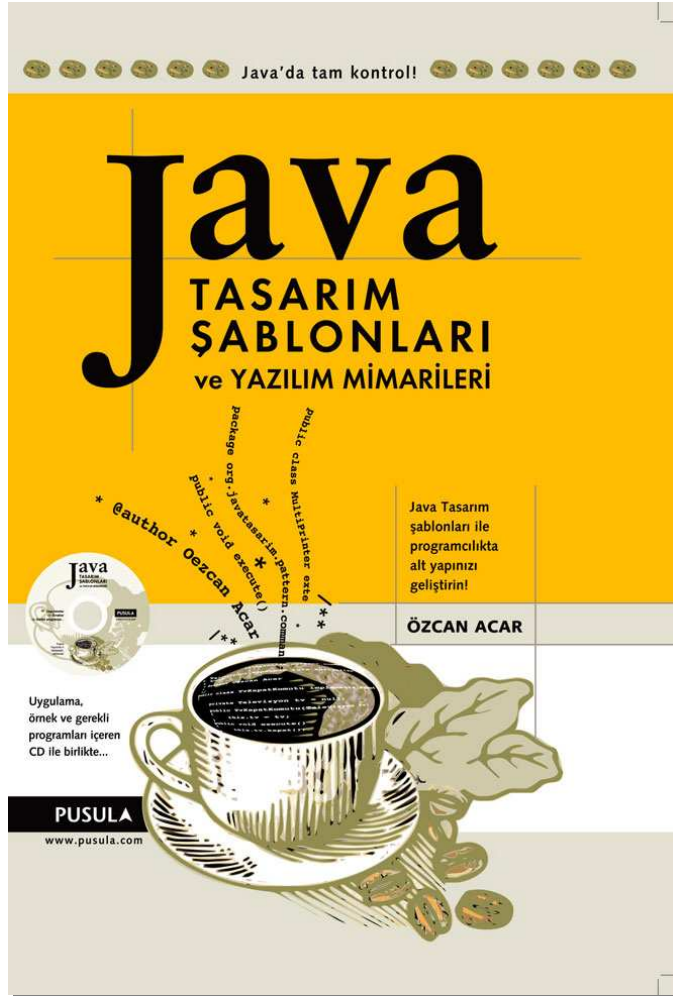
Özcan Acar İletişim

- <http://www.twitter.com/oezcanacar>
- <http://www.friendfeed.com/oezcanacar>
- acar@unitedinter.net

Özcan Acar Hakkında



Java Tasarım Şablonları ve Yazılım Mimarileri



Tasarım şablonu nedir?

Interface / Abstract sınıf nedir?

Oluşturucu Tasarım Şablonları

Factory, Abstract Factory, Builder, Prototype, Singleton

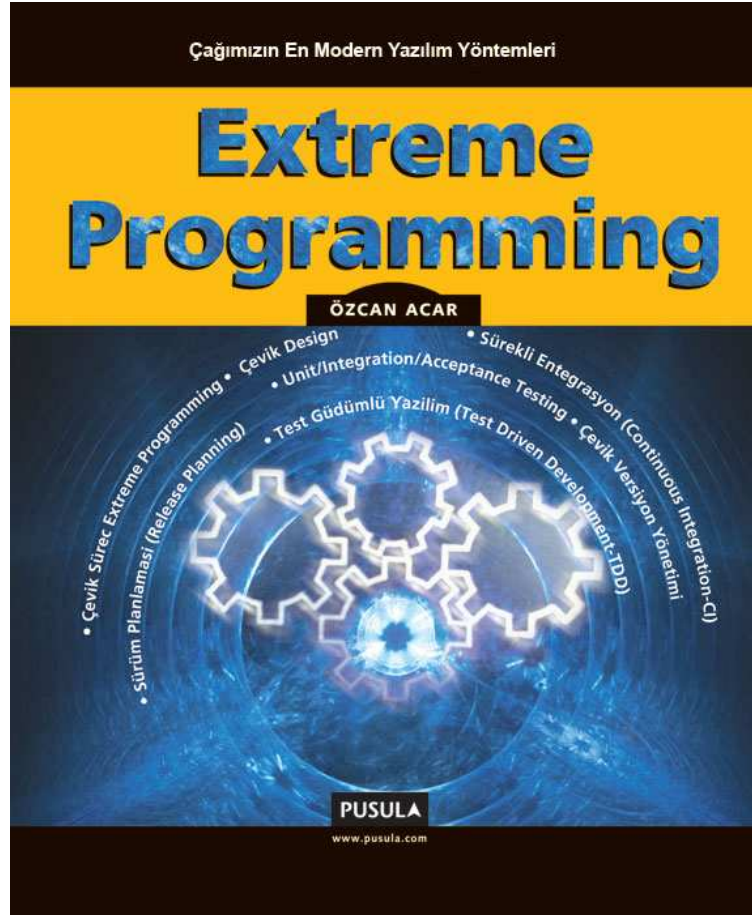
Yapısal Tasarım Şablonları

Adapter, Bridge, Facade, Decorator, Composite, Flyweight, Proxy

Davranışsal Tasarım Şablonları

Command, Memento, Strategy, Iterator, State, Observer, Visitor

Extreme Programming



Çevik süreç nedir?

Çevik manifesto

Extreme Programming nedir?

XP değerleri ve prensipleri

Çevik proje planlaması

Çevik süreçlerde iletişim

Çevik tasarım

Sürekli entegrasyon

Test güdümlü yazılım

Yazılım metrikleri

www.BilgeProgramci.com



Bilge Programcı

işini bilir... kendini devamlı geliştirir...

BP Hakkında

- [BP Nedir?](#)
- [BP Adayları](#)
- [BP Bilekliği](#)
- [BP Değer Sistemi](#)
- [BP Seminerleri](#)

BP Kuşakları

- [Kuşaklar Hakkında](#)
- [Beyaz](#)
- [Sarı](#)
- [Turuncu](#)

Bilge Programcı (BP)

Karate-Do ^[1] da kuşaklar olgunlaşma ve mükemmelleşmenin sembolüdür. Bu mükemmellik sadece Karate-Do tekniklerinde değil, büyüklerine sevgi, saygı, toplum kurallarına uymak ve sahip çıkmak , hocalara sonsuz itaat, zayıfları korumak, hoşgörü, inanç, camiayı sahiplenme ve tüm manevi değerler diye devam eder.



Karate-Do da kullanılan kuşaklar hakkındaki bu kısa açıklamadan, Karate-Do'nun sahip olduğu prensip, pratik ve değer sistemi hak

Bilge Programcı nedir?



Karateci ve Yazılımcının Ortak Yönleri Nelerdir?



Karate-Do Sporu

Karate-Do'da kuşaklar olgunlaşma ve mükemmelleşmenin sembolüdür. Bu mükemmellik sadece Karate-Do tekniklerinde değil, büyüklerine sevgi, saygı, toplum kurallarına uymak ve sahip çıkmak , hocalara sonsuz itaat, zayıfları korumak, hoşgörü, inanç, camiayı sahiplenme ve tüm manevi değerler diye devam eder.



Karate-Do Sporu

Karate-Do da kullanılan kuşaklar hakkındaki bu kısa açıklamadan, Karate-Do'nun sahip olduğu prensip, pratik ve değer sistemi hakkında fikir sahibi olmak mümkündür. Her bir üst kuşak öğrencinin takip ettiği yolda olgunlaşma seviyesinin göstergesidir. Öğrenci, geldiği seviyenin ihtiva ettiği prensip, pratik ve değer sistemini benimseyerek olgunlaşma sürecinde emin adımlarla ilerler.



Kuşak Rengi	Kademe	15 yaş altı Ay adedi	Haftalık Antrenman	15 yaş üstü Ay adedi	Haftalık Antrenman
Beyaz	9. Kyu	2	1x	1	2x
Yarı Sarı		3	1x	-	-
Sarı	8. Kyu	3	1x	2	2x
Yarı Turuncu		4	1x	-	-
Turuncu	7. Kyu	3	1x	4	2x
Yarı Yeşil		3	1x	-	-
Yeşil	6. Kyu	3	2x	5	2x
Yarı Mavi		3	2x	-	-
Mavi	5. Kyu	3	2x	6	2x
Yarı Mor		3	2x	-	-
Mor	4. Kyu	3	2x	6	2-3x
Yarı Kahverengi		3	2x	-	-
Kahverengi I	3. Kyu	6	2-3x	6	2-3x
Kahverengi II	2. Kyu	9	2-3x	9	2-3x
Kahverengi III	1. Kyu	9	2-3x	9	2-3x
Ön Sınav		6	2-3x	11	2-3x
Siyah	1. Dan	6	2-3x	1	2-3x

Total	72	60
--------------	-----------	-----------

**Karate-Do'dan yola çıkarak Bilge
Programcı için bir değer sistemi ve
uzmanlaşma metodolojisi formalize
edebilir miyiz?**



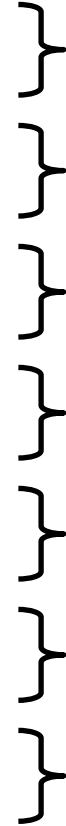
Bilge Programcılığın Reçetesi

- Değer Sistemi
- Prensipler
- Pratikler
- Olgunlaşma Simgesi



Bilge Programcı Kuşakları

Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler



1. Kuşak: Beyaz

2. Kuşak: Sarı

3. Kuşak: Turuncu

4. Kuşak: Yeşil

5. Kuşak: Mavi

6. Kuşak: Kahverengi

7. Kuşak: Siyah

BP Değer Sistemi

Bilge Programcı Bilekliği





Beyaz Kuşak

Prensipier

1. Kendi kendini yetiştirme prensibi
2. Kırık cam prensibi
3. DRY prensibi

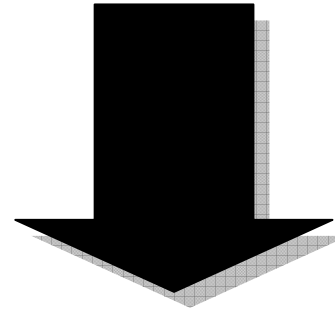
Pratikler

1. İzci kuralı
2. Kodun ifade gücünü artırmak
3. Her gün tefekkür etmek

30 gn sonra...



Beyaz Kuşak



Sarı Kuşak



Sarı Kuşak

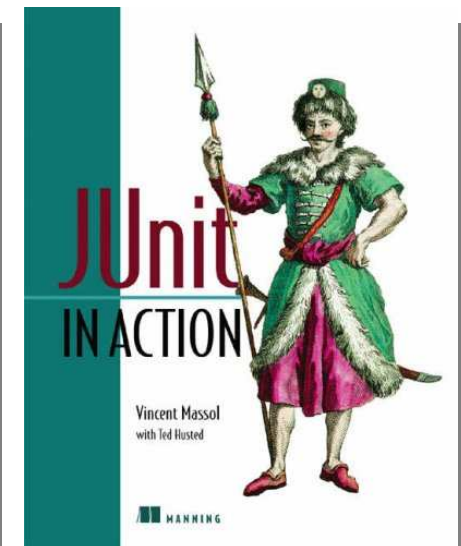
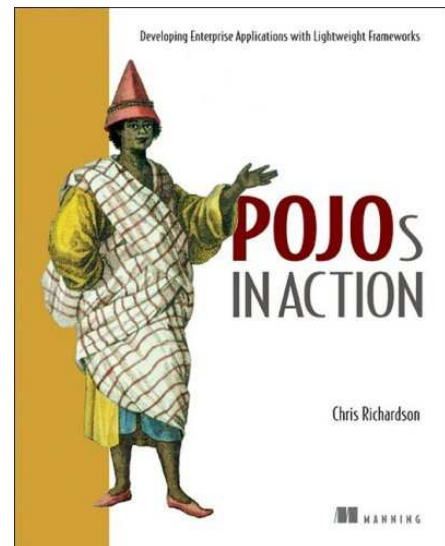
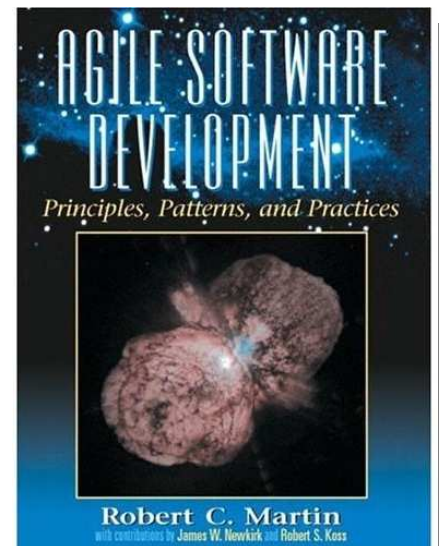
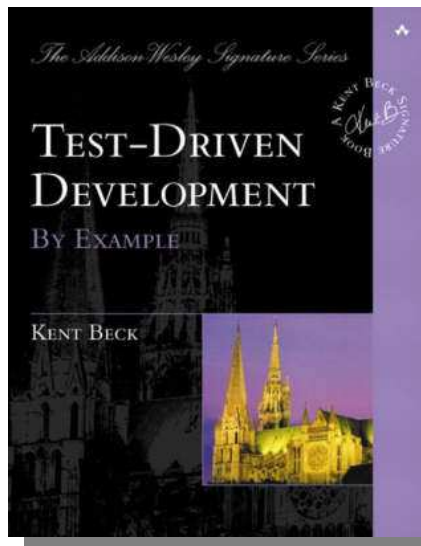
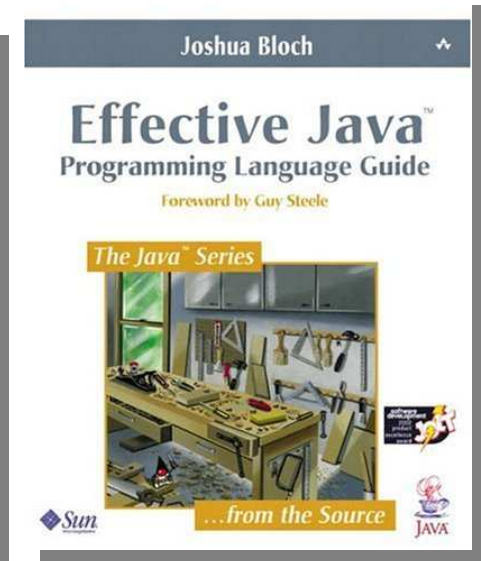
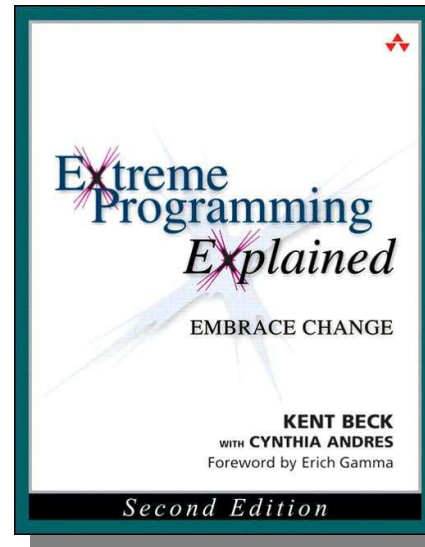
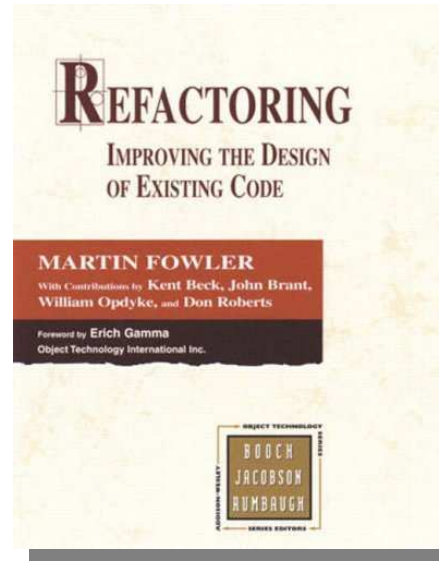
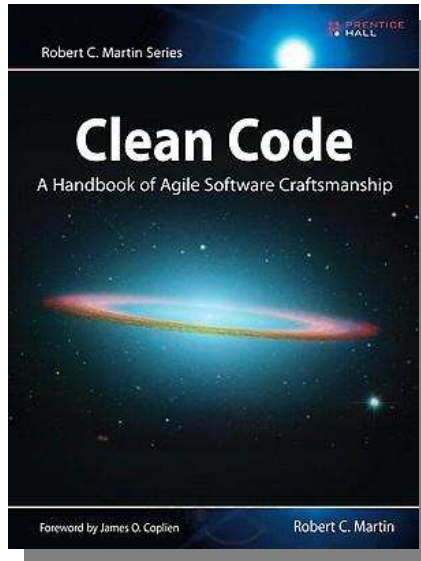
Prensipier

1. **KISS prensibi**
2. **Yazılım araçlarına hakim olma prensibi**
3. **Nesneye yönelik programlama**
4. **Yazılımda test konseptlerinden haberdar olmak**

Pratikler

1. **Interface ve Abstract sınıfları tasarımda kullanmak**
2. **JUnit Testleri oluşturmak**
3. **Kalıtım yerine kompozisyon kullanmak**
4. **Kitap okumak, kitap okumak, kitap okumak**
5. **Kodu gözden geçirmek**

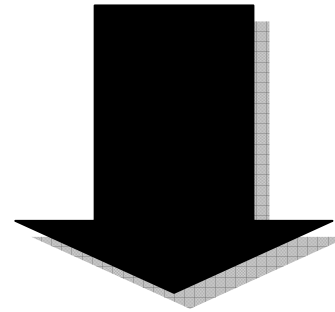
Kitaplar



30 gn sonra...



Sarı Kuşak



Turuncu Kuşak



Turuncu Kuşak

Prensipier

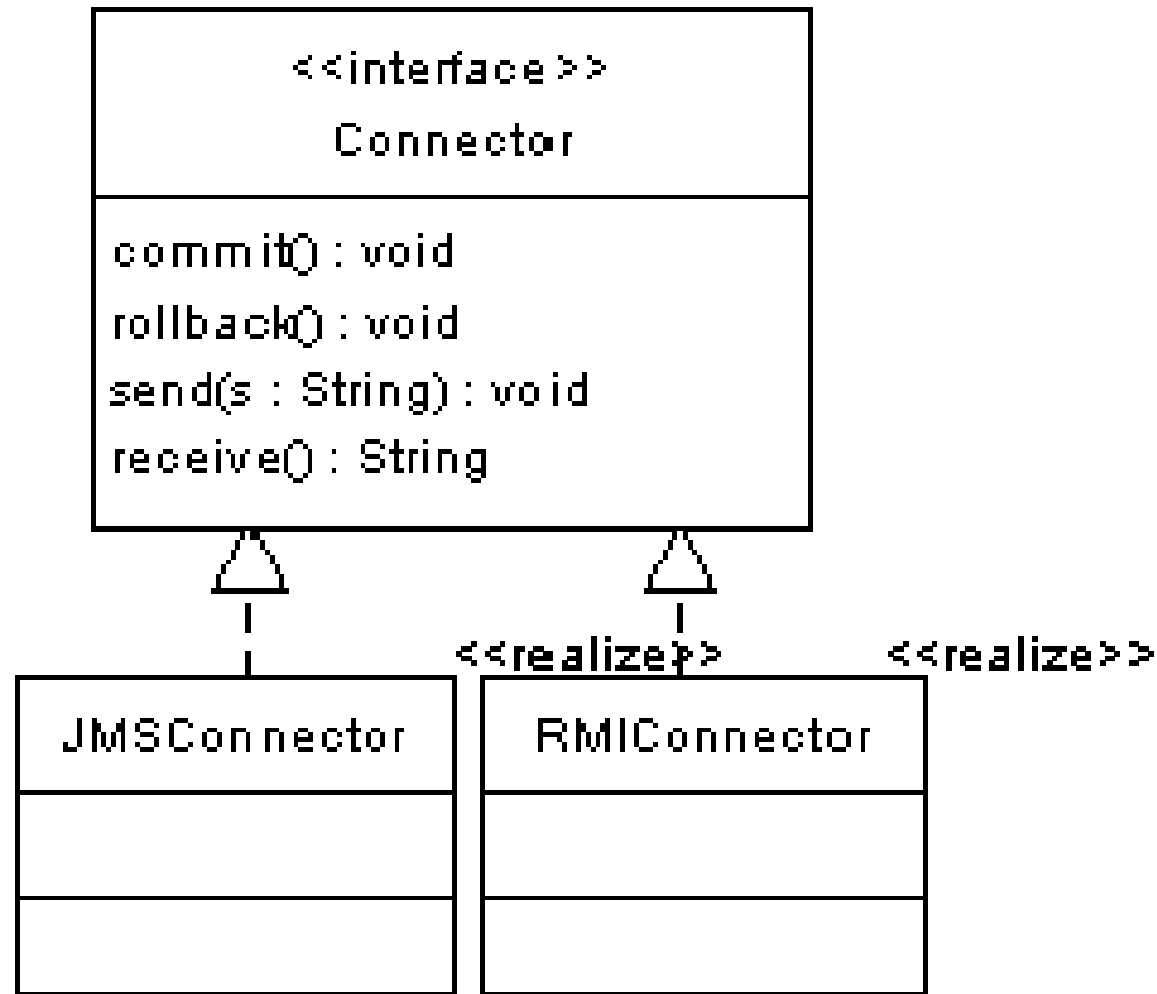
1. Arayüz ayırma prensibi
2. Bağımlılıkların tersine çevrilmesi prensibi
3. Liskov'un yerine geçme prensibi

Pratikler

1. Sektörün öncülerini takip etmek
2. Yazılımla ilgili konferans ve seminerlere katılmak
3. Versiyon kontrol sistemi kullanmak
4. Test güdümlü yazılım yapmak
5. Bir logging framewörkü kullanmak

Arayüz Ayırma Prensipli

Interface Segregation Principle (ISP)



Arayüz Ayırma Prensipli

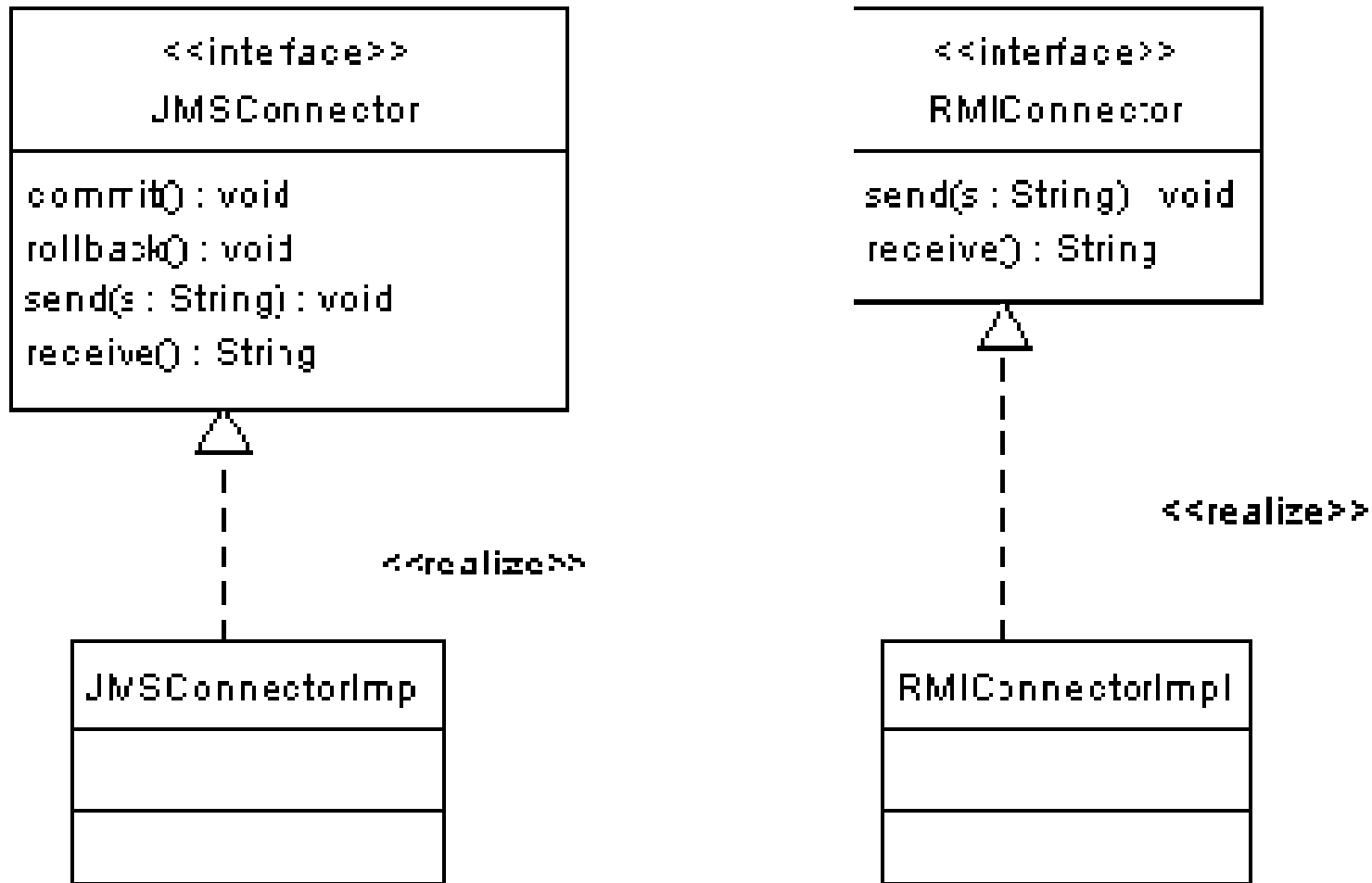
Interface Segregation Principle (ISP)

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

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

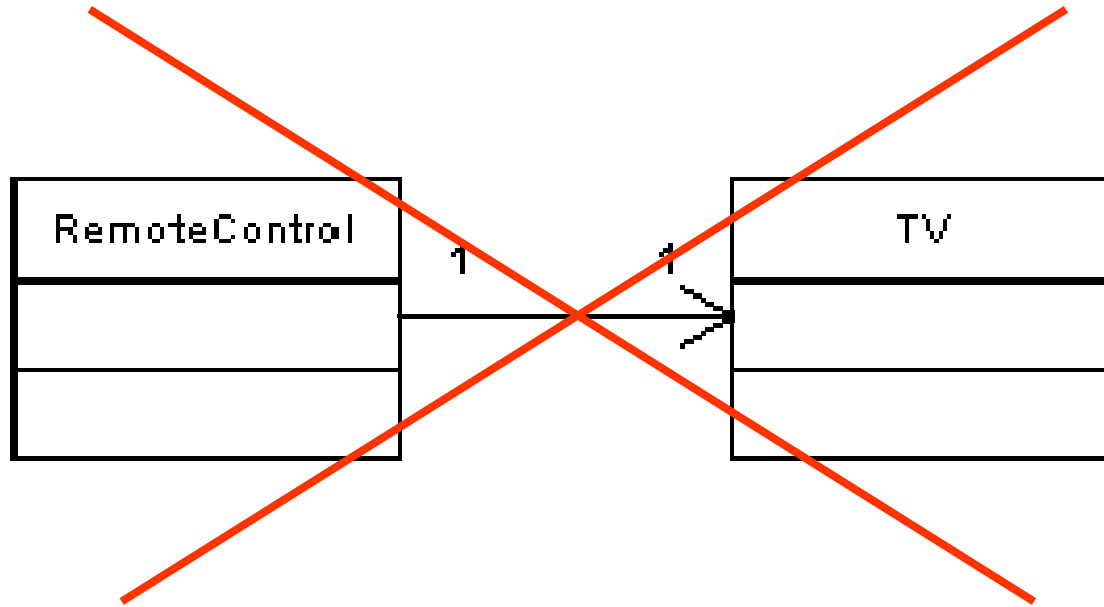
Arayüz Ayırma Prensipleri

Interface Segregation Principle (ISP)



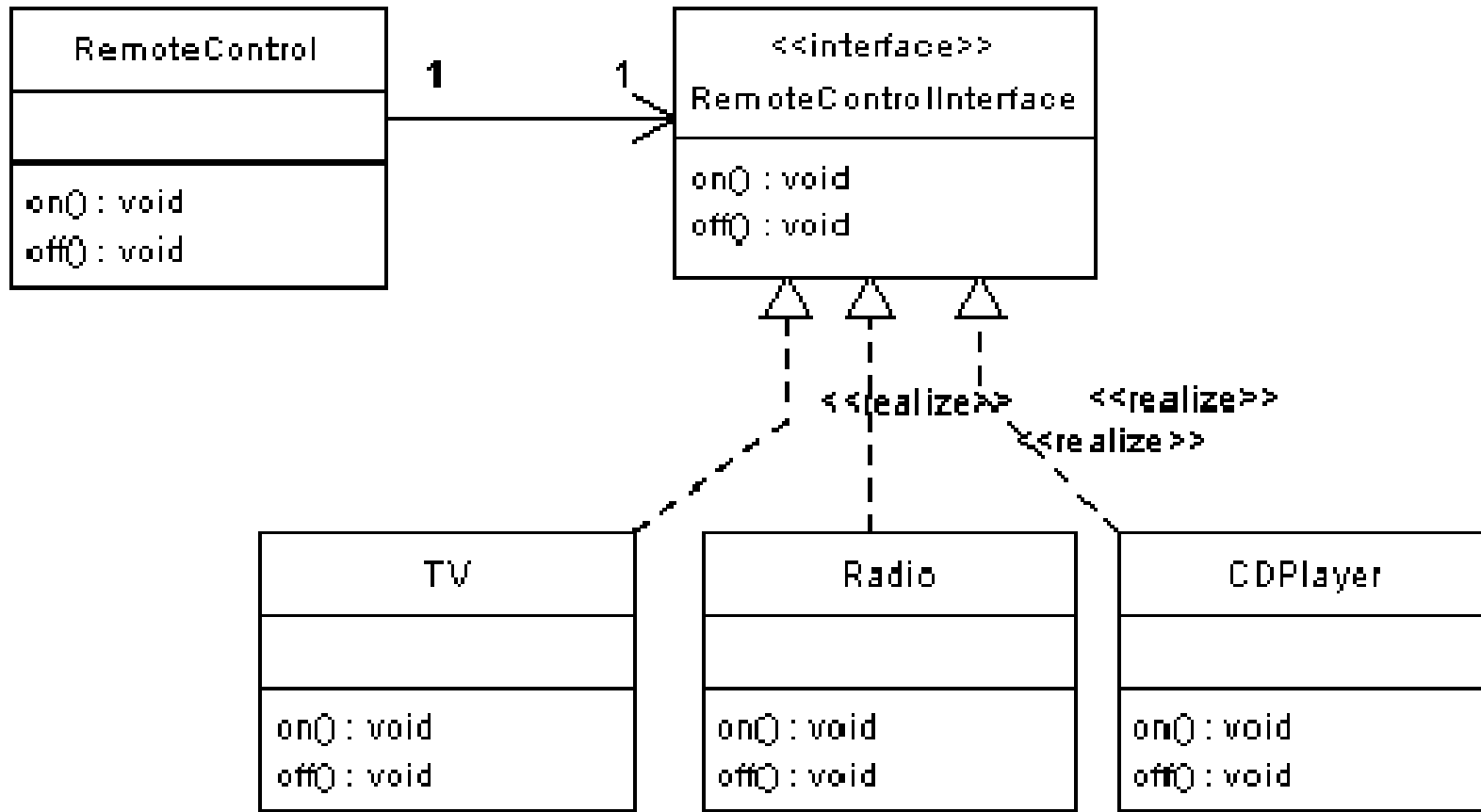
Bağımlılıkların Tersine Çevrilmesi Prensipleri

Dependency Inversion Principle (DIP)



Bağımlılıkların Tersine Çevrilmesi Prensipleri

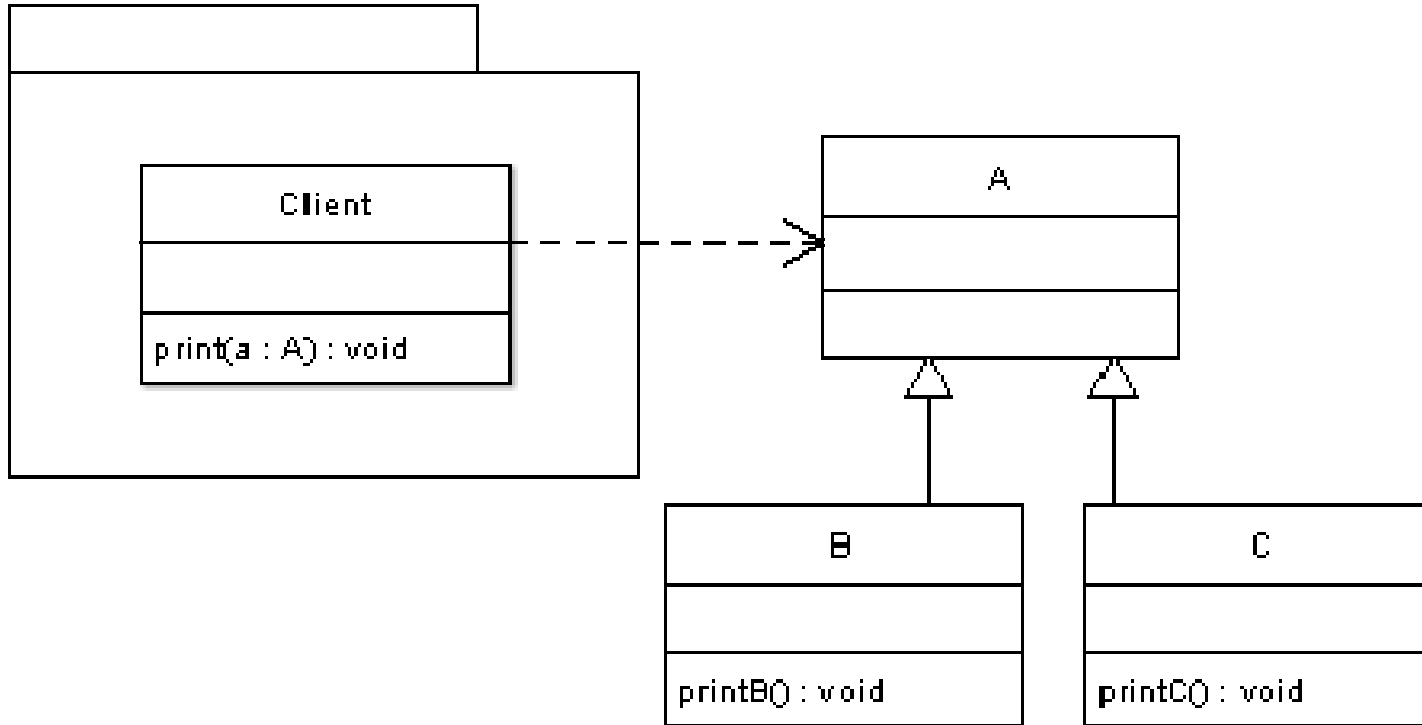
Dependency Inversion Principle (DIP)



Liskov Yerine Geçme Prensipli

Liskov Substitution Principle (LSP)

Alt sınıflardan oluşturulan nesnelere üst sınıfların nesneleriyle yer değiştirdiklerinde aynı davranışı göstermek zorundadırlar.



Liskov Yerine Geçme Prensipleri

Liskov Substitution Principle (LSP)

```
public void print(A a)
{
    if(a instanceof B)
    {
        ((B)a).printB();
    }
    else if(a instanceof C)
    {
        ((C)a).printC();
    }
}
```



Turuncu Kuşak

Prensipier

1. Arayüz ayırma prensibi
2. Bağımlılıkların tersine çevrilmesi prensibi
3. Liskov'un yerine geçme prensibi

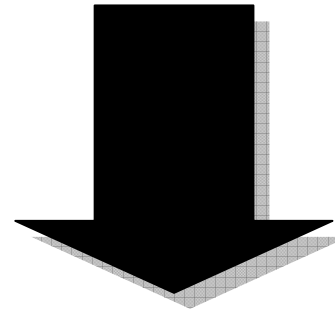
Pratikler

1. Sektörün öncülerini takip etmek
2. Yazılımla ilgili konferans ve seminerlere katılmak
3. Versiyon kontrol sistemi kullanmak
4. Test güdümlü yazılım yapmak
5. Bir logging framewörkü kullanmak

60 gn sonra...



Turuncu Kuşak



Yeşil Kuşak



Yeşil Kuşak

Prensipier

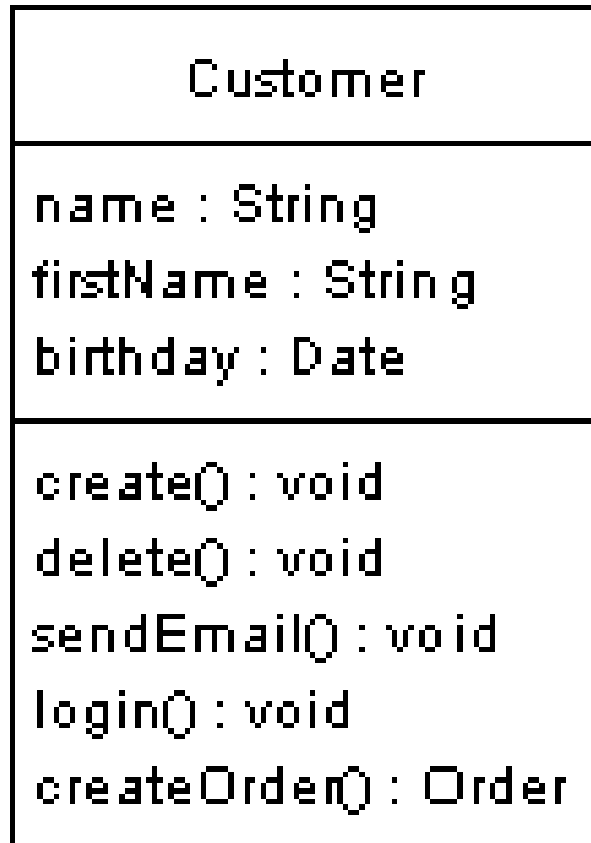
1. Tek sorumluluk prensibi
2. Açık-Kapalı prensibi
3. Esnek bağ prensibi

Pratikler

1. Basit refactoring işlemleri yapmak
2. Tasarım şablonlarını kullanmak
3. Sürekli entegre etmek
4. MVC framewörklerle haşır, neşir olmak

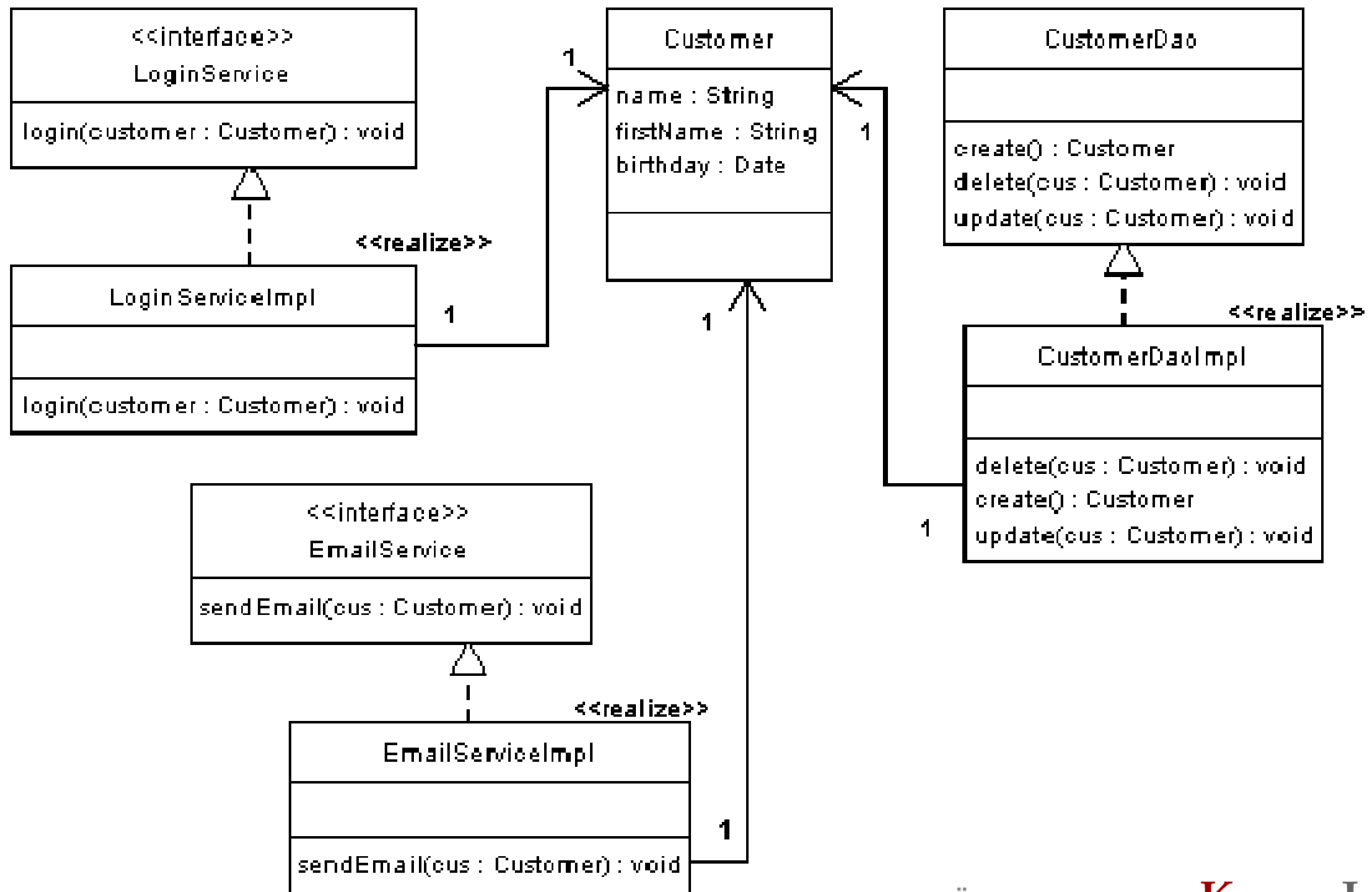
Tek Sorumluk Prensipli

Single Responsibility Principle (SRP)



Tek Sorumluk Prensipli

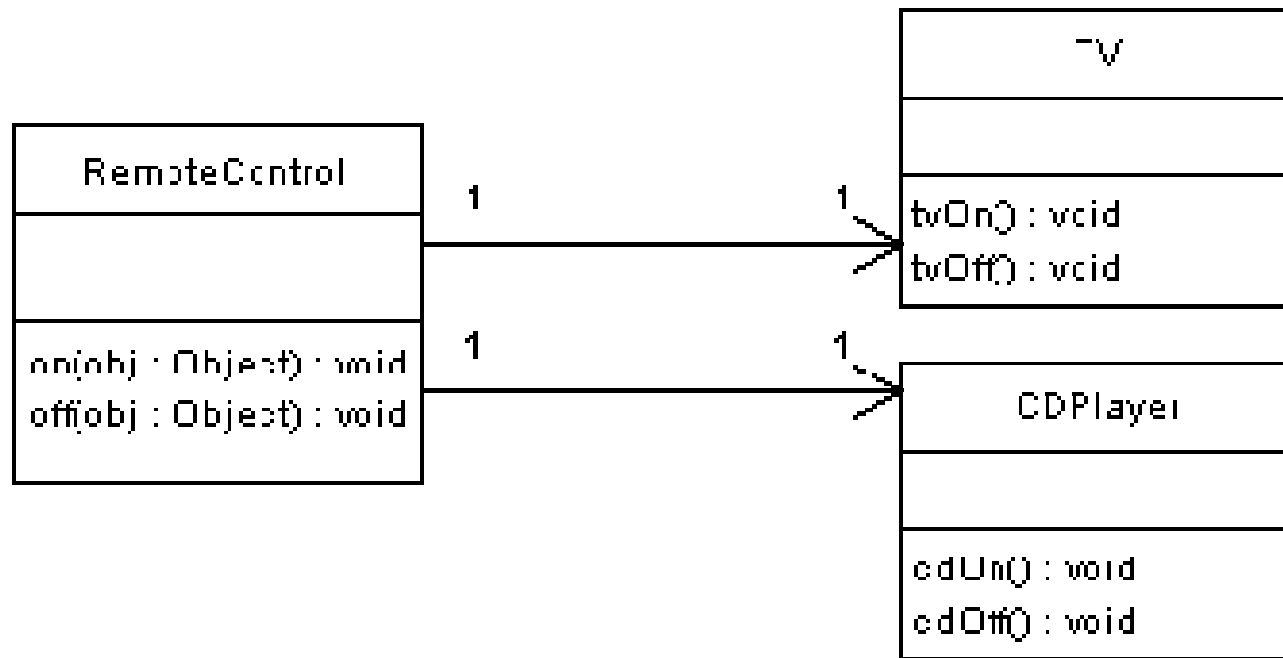
Single Responsibility Principle (SRP)



Açık Kapalı Prensibi

Open Closed Principle (OCP)

Programlar geliştirilmeye açık ama değiştirilmeye kapalı olmalıdır.



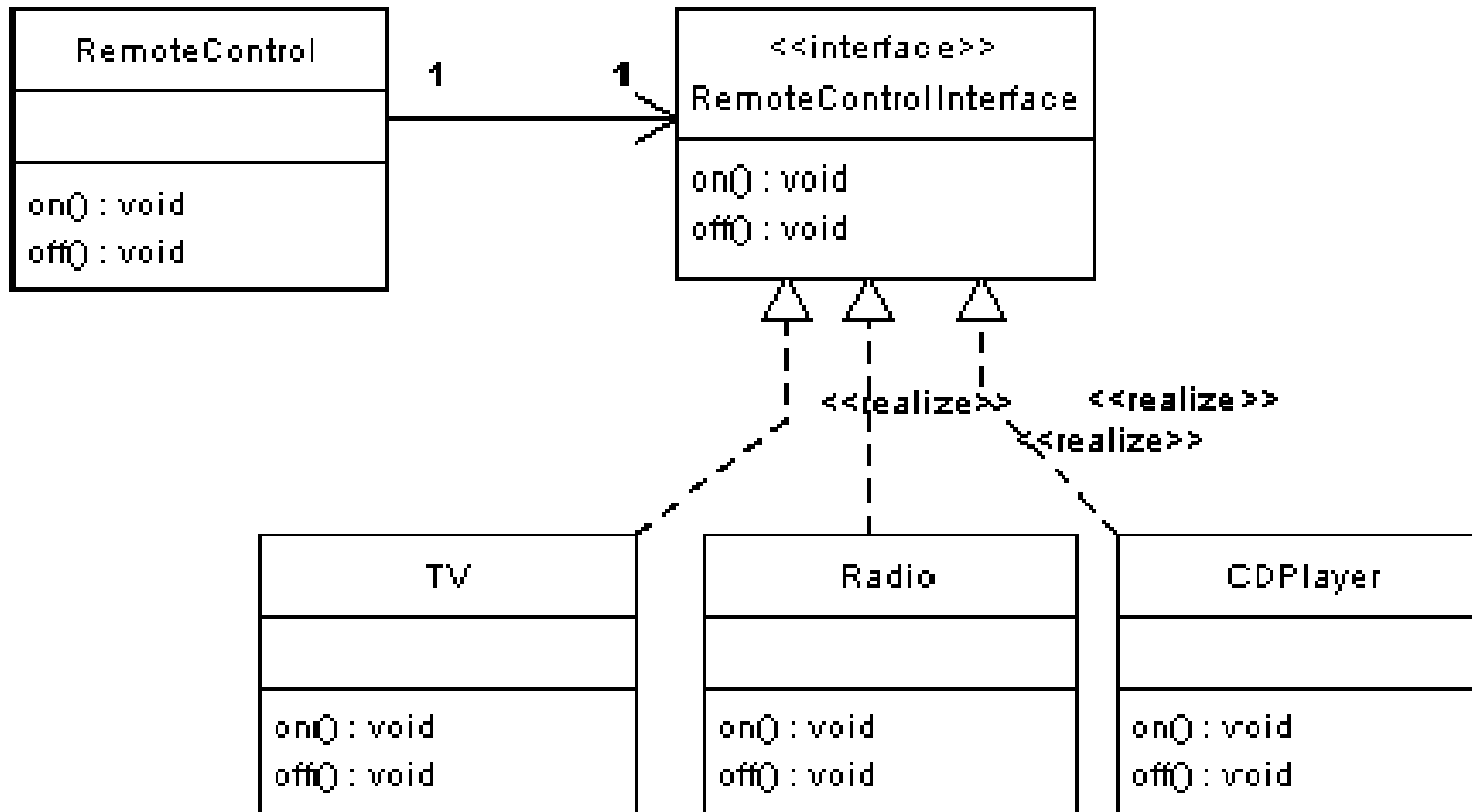
Açık Kapalı Prensibi

Open Closed Principle (OCP)

```
public void on(Object obj)
{
    if(obj instanceof TV)
    {
        ((TV)obj).tvOn();
    }
    else if(obj instanceof CDPlayer)
    {
        ((CDPlayer)obj).cdOn();
    }
}
```

Açık Kapalı Prensibi

Open Closed Principle (OCP)



Açık Kapalı Prensibi

Open Closed Principle (OCP)

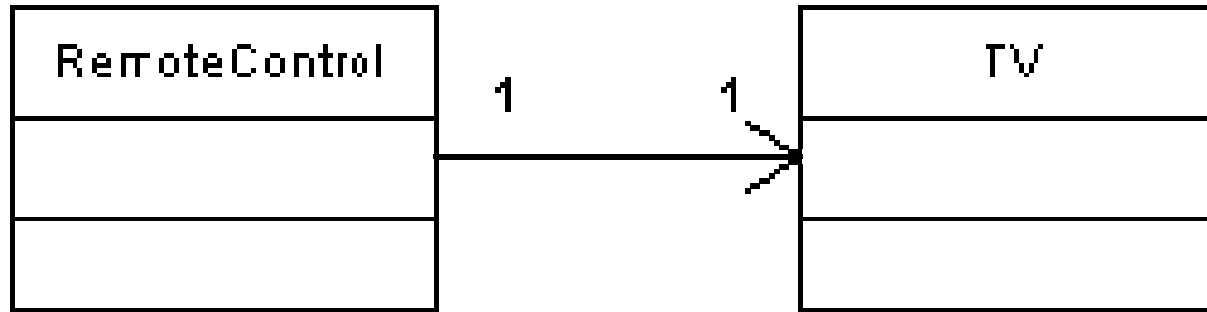
```
private RemoteControlInterface remote;  
  
public RemoteControl(RemoteControlInterface _remote)  
{  
    this.remote = _remote;  
}  
  
public void on()  
{  
    remote.on();  
}  
  
public void off()  
{  
    remote.off();  
}
```

Stratejik Kapama

- Eğer kapama tam sağlanamıyorsa, kapamanın stratejik olarak implemente edilmesi gerekir.
- Programcı implementasyon öncesi meydana gelebilecek değişiklikleri kestirerek, implemente ettiği metotların kapalılık oranını yükseltmelidir. Bu tecrübe gerektiren stratejik bir karardır.
- Programcı her zaman ne gibi değişikliklerin olabileceğini kestiremeyebilir. Bu durumda konu hakkında araştırma yaparak, oluşabilecek değişiklikleri tespit edebilir. Eğer olabilecek değişikliklerin tespiti mümkün değilse, beklenen değişiklikler meydana gelene kadar beklenir ve implementasyon yeni değişiklikleri de yansıtacak şekilde OCP uyumlu hale getirilir.

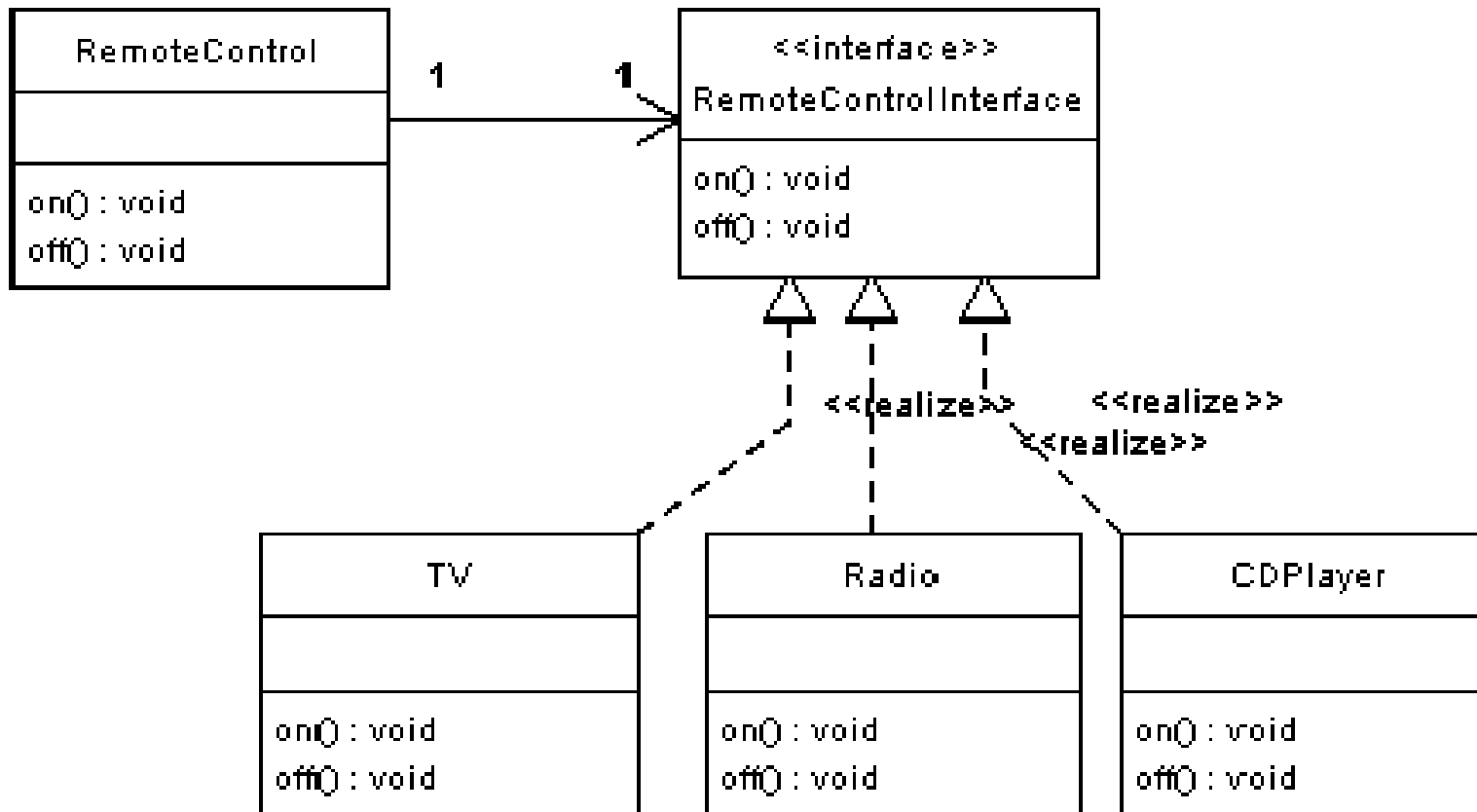
Esnek Bağ

Loose Coupling (LC)



Esnek Bağ

Loose Coupling (LC)





Yeşil Kuşak

Prensipier

1. Tek sorumluluk prensibi
2. Açık-Kapalı prensibi
3. Esnek bağ prensibi

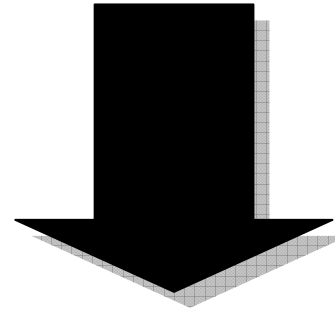
Pratikler

1. Basit refactoring işlemleri yapmak
2. Tasarım şablonlarını kullanmak
3. Sürekli entegre etmek
4. MVC framewörklerle hasır, neşir olmak

60 gn sonra...



Yeşil Kuşak



Mavi Kuşak



Mavi Kuşak

Prensipier

1. Paket bazındaki tasarım prensipleri
2. Komponent tabanlı yazılım prensibi
3. Kod konvensiyonları
4. Kök neden analizi

Pratikler

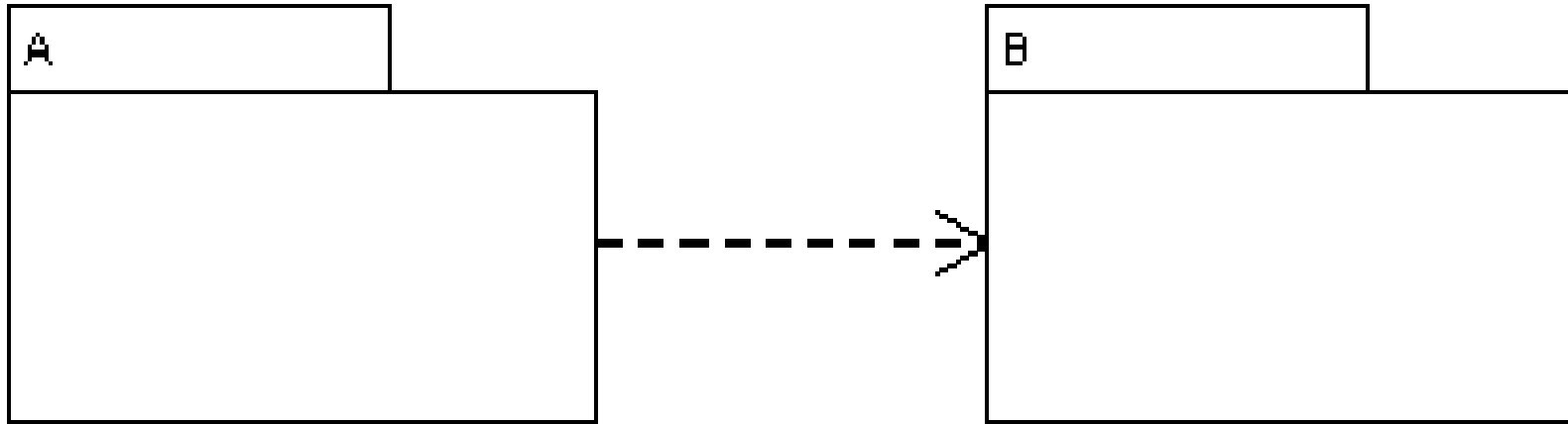
1. Code coverage (kod kapsama alanı) analizi yapmak
2. Bilgiyi paylaşmak
3. Bugtracking ve wiki kullanmak
4. Katmanlı mimarileri oluşturmak
5. Bulunan her bug için bir Unit test oluşturmak

Paket Bazında Tasarım Prensipleri

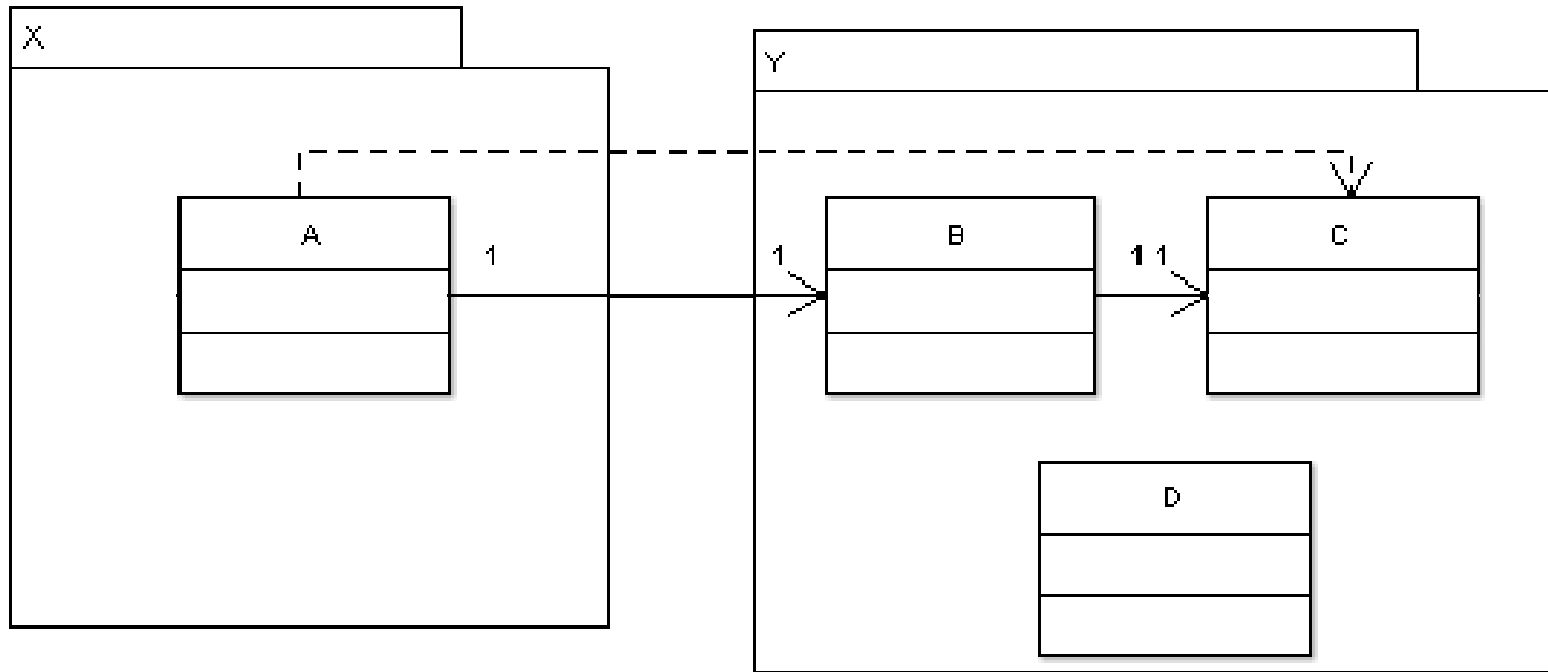
- **Reuse-Release Equivalence Principle (REP)**
- **Common Reuse Principle (CRP)**
- **Common Closure Principle (CCP)**
- **Acyclic Dependency Principle (ADP)**
- **Stable Dependencies Principle (SDP)**
- **Stable Abstractions Principle (SAP)**

Tekrar Kullanım ve Sürüm Eşitliği

Reuse-Release Equivalence Principle (REP)



Yeniden Ortak Kullanım Prensipli Common Reuse Principle (CRP)

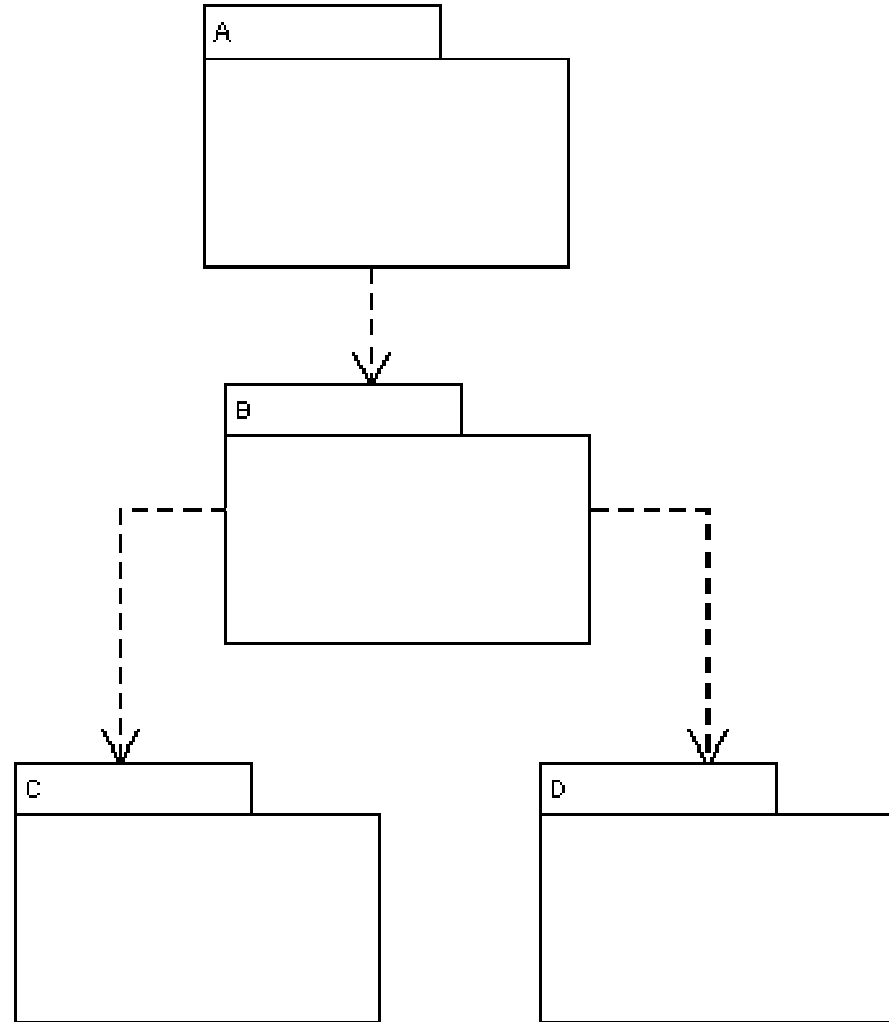


Ortak Kapama Prensipleri

Common Closure Principle (CCP)

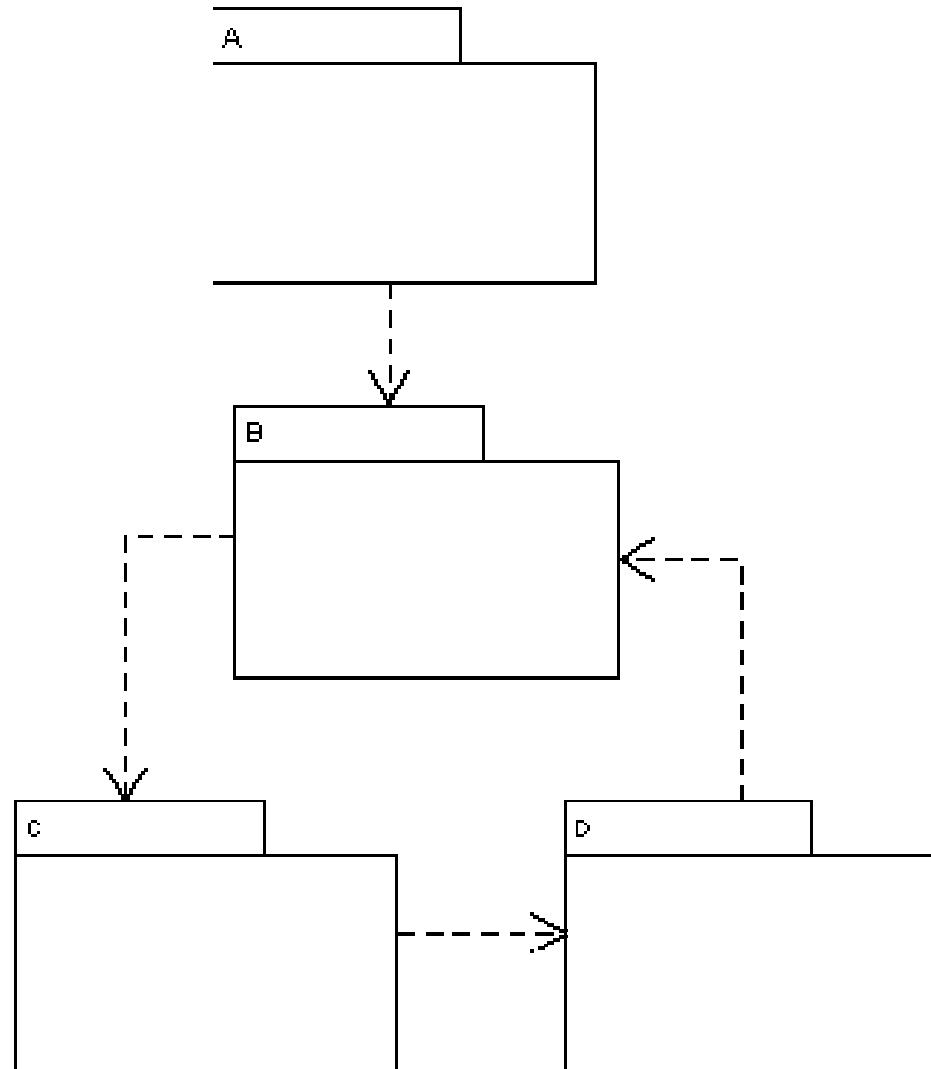
- Aynı sebepten dolayı değişikliğe uğrayabilecek sınıfların aynı paket içinde yer alması gerekir.
- CCP daha önce incelediğimiz, sınıflar için uygulanan Single Responsibility (SRP) prensibinin paketler için uygulanan halidir.
- Her paketin değişmek için sadece bir sebebi olmalıdır.
- CCP uygulandığı takdirde sistemin bakılabilirliği artırılır ve test ve yeni sürüm için harcanan zaman ve emek azaltılır.

Çevrimsiz Bağımlılık Prensipli Acyclic Dependency Principle (ADP)

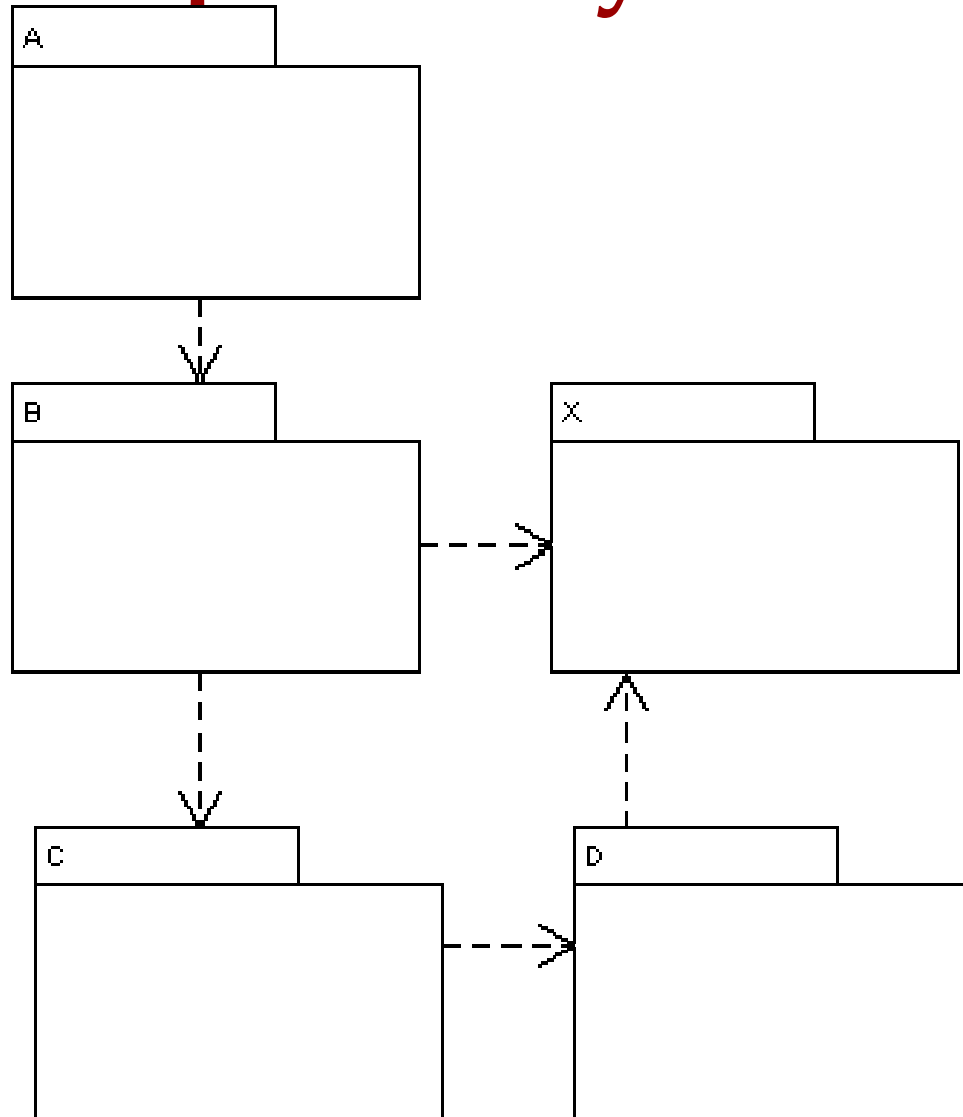


Çevrimsiz Bağımlılık Prensipli

Acyclic Dependency Principle (ADP)



Çevrimsiz Bağımlılık Prensipli Acyclic Dependency Principle (ADP)





Mavi Kuşak

Prensipier

1. Paket bazındaki tasarım prensipleri
2. Komponent tabanlı yazılım prensibi
3. Kod konvensiyonları
4. Kök neden analizi

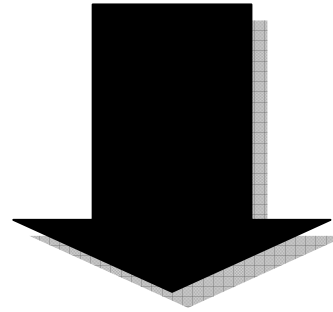
Pratikler

1. Code coverage (kod kapsama alanı) analizi yapmak
2. Bilgiyi paylaşmak
3. Bugtracking ve wiki kullanmak
4. Katmanlı mimariler oluşturmak
5. Bulunan her bug için bir Unit test oluşturmak

90 gn sonra...



Mavi Kuşak



Kahregengi Kuşak



Kahregengi Kuřak

Premsipler

1. Kiřiselleřtirmeme prensibi
2. Anlayana kadar sorma prensibi
3. Kodu kırmama prensibi

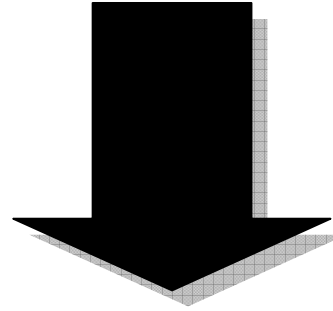
Pratikler

1. Inversion of Control
2. Mockup nesne kullanmak
3. Sürüm yönetimi yapmak
4. Kompleks refactoring işlemleri yapmak
5. Oluřan tüm hataları raporlamak

90 gn sonra...



Kahregengi Kuşak



1. Dan Siyah Kuşak



1. Dan Siyah Kuşak

Prensipier

1. YAGNI prensibi
2. Müşteriye kulak verme prensibi
3. Hafif bavulla yola çıkma prensibi
4. Tell, don't ask prensibi

Pratikler

1. Sertifikasyon
2. Çevik süreç kullanmak
3. Yeni yazılım dilleri öğrenmek
4. Davranış güdümlü yazılım yapmak
5. Soft skilleri geliştirmek



1. Dan Siyah Kuşak

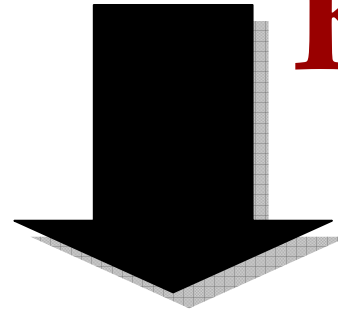
Soft Skill

1. Takım oyunculuğu
2. İletişim becerisi
3. Sorumluluk üstlenebilme
4. Yönetim kompetansı
5. Kreatiflik
6. Zor şartlarda çalışabilme

1 yıl sonra...



1. Dan Siyah
Kuşak



2. Dan Siyah
Kuşak



2. Dan Siyah Kuşak

Prensipier

1. Kendi prensiplerini geliştirme prensibi

Pratikler

1. Konferanslara sunucu olarak katılmak
2. Açık kaynaklı bir projeye katkıda bulunmak
3. Açık kaynaklı bir proje geliştirmek

Bilge Programcı

Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler
Prensip & Pratikler

1. Kuşak: Beyaz (30 gün)

2. Kuşak: Sarı (30 gün)

3. Kuşak: Turuncu (60 gün)

4. Kuşak: Yeşil (60 gün)

5. Kuşak: Mavi (90 gün)

6. Kuşak: Kahverengi (90 gün)

1. Dan: Siyah

2. Dan: Siyah (360 gün)

Toplam:
720 iş
günü

Detaylar için bakınız:

BilgeProgramci.com

Son

İlginiz için teşekkür ederim.

Java Dergisi

Mayıs / Haziran 2010

Java ♦ Mimari ♦ Çevik Süreçler

BT Press

CD-ROM

www.javadergisi.com

Java Dergisi

Java ♦ Mimari ♦ Çevik Süreçler

JDK 1.6 Update 18

Glassfish V3 App Server

Eclipse 3.5 Getirer

GlassFish Tools Bundle
For Eclipse

Spring 3.0.1

5.10

Mayıs / Haziran 2010

RESTJAX-RS 1.1 ile
REST Mimarileri**Java ME**Java ME ile mobile
dünya**Spring**Java EE 6 ve Spring
entegrasyonu**Scrum**Scrum ile çevik proje
yönetimi

Java EE 6

Yeni Java Enterprise
Edition 6 teknolojisi

EJB 3.1

EJB 3.1'in getirdiği
yenilikler

Servlet 3.0

Servlet 3.0 ile web programı

JSF 2.0

JSF'de 4. boyut



4 195985 801509

JugTR.org

Java User Group Turkey - Windows Internet Explorer bereitgestellt von GMX

http://jugtr.org/

Google

Suche

Freigeben

AutoFill

acar

Konvertieren

Auswählen

Favoriten

Seminer Takvimi - Kurumsal ...

Google Docs - All Items

Java Dergisi

Java Dergisi Okurlari

Sun Microsystems Java EE 6...

Java EE 6 one-week Online ...

Java User Group Turkey

Home

Diğer Türk JUG'lar

İletişim

JugTR Nedir?

Java User Group Türkiye

RECENT NEWS

İlk JugTR.org Toplantısı 1 Nisan 2010'da

Sun firmasının desteklediği JUG (Java User Group) programına kayıtlı ilk Türk JUG olan JugTR.org 1 Nisan 2010 tarihinde ilk toplantısını İstanbul'da gerçekleştiriyor. JugTR.org'un amacı Java kullanıcılarını düzenli olarak bir araya getirmek ve Türkiye'de Java'nın yayılmasını desteklemektir.



Toplantı Tarihi:
1 Nisan 2010

Toplantı Yeri:
Bahçeşehir Üniversitesi Beşiktaş kampüsü

15 readers
BY FEEDBURNER

HABER ABONELİĞİ LİSTESİ

E-Posta Adresiniz:

Abone Ol

JUGTR YAZIŞMA GRUBU

<http://groups.google.com/group/jugtr>

İLETİŞİM

<http://www.twitter.com/JugTR>
<http://www.friendfeed.com/JugTR>

POPÜLER YAZILAR

İlk JugTR.org Toplantısı 1 Nisan 2010'da

ETİKETLER

KATEGORİLER

100%

Internet | Geschützter Modus: Aktiv