

Hazırlayan: Barış Dere  
www.barisdere.com @BarisDere  
baris.dere@gmail.com

# Spring Framework Eđitimi

Ders 1: Genel bir bakış



[Http://www.youtube.com/barisdere](http://www.youtube.com/barisdere)

# Ajanda

- Eğitim konuları
- Spring Framework nedir?
- Inversion Of Control (Container)
- Modüller
- Neden Spring kullanmalıyız?
- Tavsiye linkler

# Eğitim konuları

- Spring Framework
- Spring ORM
- Spring JDBC
- Spring AOP
- Spring Web
- Spring WebFlow
- Spring Roo

# Spring Framework nedir?

- Kurumsal Java uygulamalarını geliştirmeyi kolaylaştıran ve hızlandıran uygulama geliştirme çatısı (Application framework)
- Açık kaynak kodlu
- POJO – nesne tabanlı

# Spring Framework nedir? (2)

- Mimarı Rod Johnson
- Ekim 2002 - Expert One-on-One J2EE Design and Development kitabı çıktı
- İlk versiyonu (1.0) 2003 tarihinde Rod Johnson tarafından çıkarıldı
- Apache License 2.0
- En son versiyonu 3.1.0 Aralık 2011 itibariyle

# Spring Framework nedir? (3)

- Spring bir çok problem alanına çözümler sunuyor
  - Modern Web (REST, HTML5, AJAX)
  - Data Access (RDBMS, NoSQL, Map Reduce, Cloud)
  - Integration (Enterprise Orchestration, Messaging, Batch applications)
  - Mobile (Android, Iphone)
  - Sosyal ağlarla entegrasyon (Facebook, Twitter, LinkedIn)

# Spring Framework nedir? (4)

- (Devam...) Spring bir çok problem alanına çözümler sunuyor
  - Security (Authorization ve Authentication entegrasyonu)
  - Cloud Ready (Google App Engine, Amazon EC2, Cloud Foundry)

# Inversion of Control (Dependency Injection)

- En kısa anlatımıyla 'Bağımlılıkları yok etmek'
- Ne yapılacağını ne zaman yapılacağından ayırmak
- Kontrolün yer değiştirmesi
- İlk 1988 Martin Fowler tarafından ortaya atıldı
- Dependency Injection olarak yeniden tanımlandı



# Inversion of Control (Dependency Injection) (2)

## Kontrol sizde

```
#ruby
puts 'What is your name?'
name = gets
process_name(name)
puts 'What is your address?'
address = gets
process_address(address)
```

## Kontrol sistemde

```
require 'tk'
root = TkRoot.new()
name_label = TkLabel.new() {text "What is Your Name?"}
name_label.pack
name = TkEntry.new(root).pack
name.bind("FocusOut") {process_name(name)}
addr_label = TkLabel.new() {text "What is Your Address?"}
addr_label.pack
address = TkEntry.new(root).pack
address.bind("FocusOut") {process_address(address)}
Tk.mainloop()
```

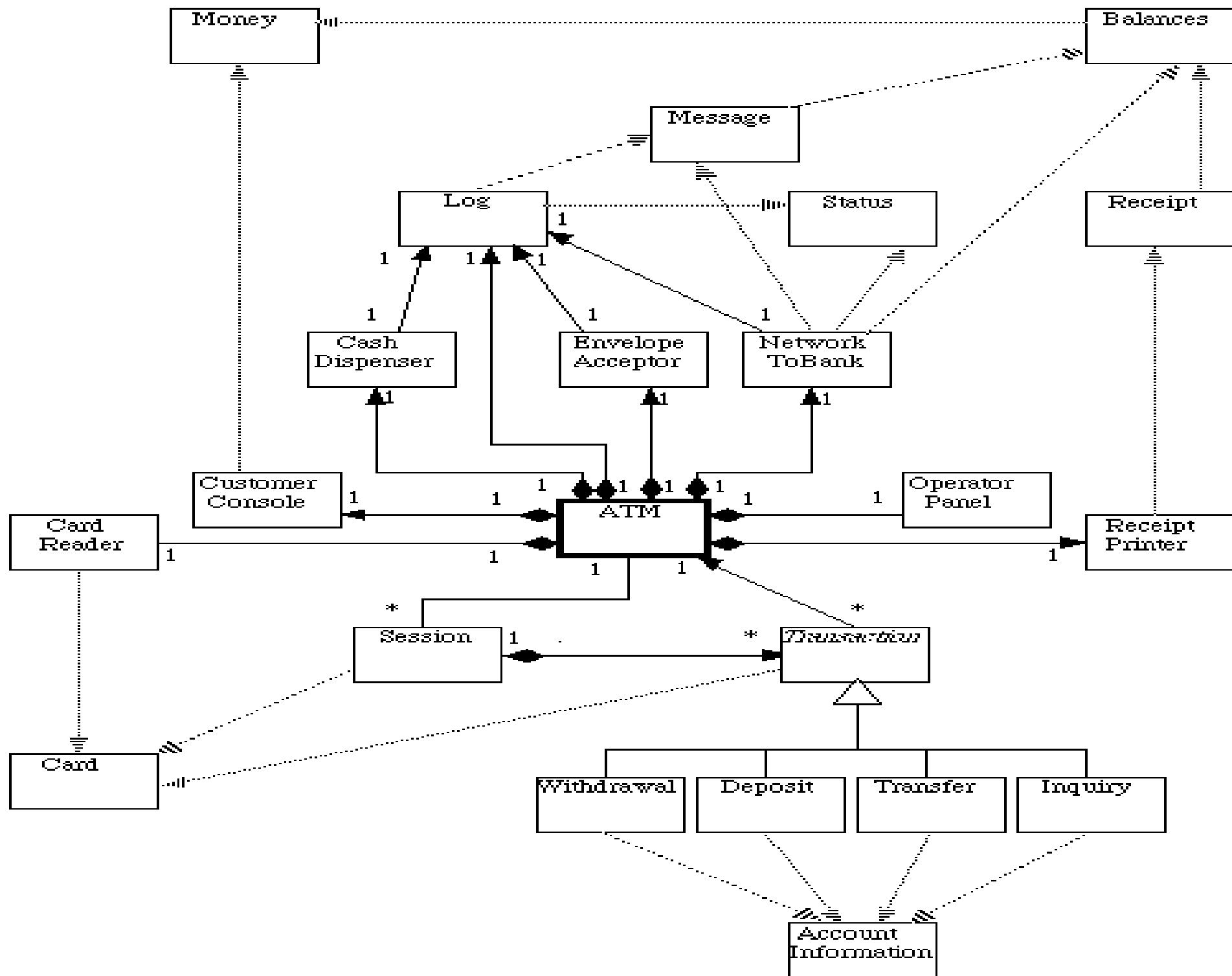
# Inversion of Control (Dependency Injection) (3)

- Inversion of Control uygulamaları:
  - using a factory pattern
  - using a service locator pattern
  - using a constructor injection
  - using a setter injection
  - using an interface injection
  - using a contextualized lookup

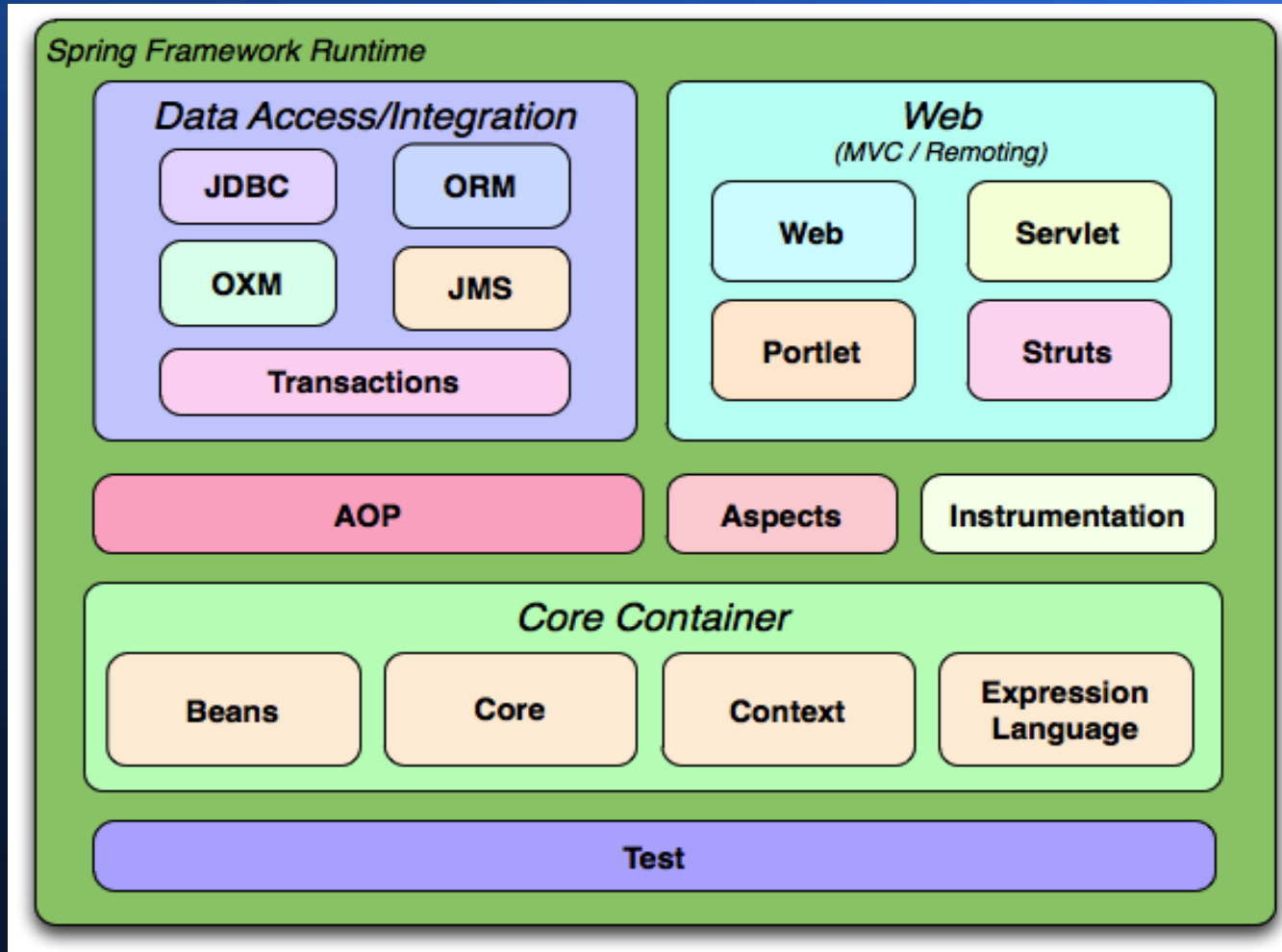
# Inversion of Control container

## Nesnelerin

- hayat döngüsünü yönetmek
- bağımlılıklarını yönetmek
- konfigürasyonunu sağlamak
- beraber bir bütün halinde çalışmasını sağlamak
- tasarım kalıplarının kullanımını kolaylaştırmak



# Modüller



# Neden Spring Framework?

- Kendini ispatlamış (Dünya çapında milyonlarca kullanıcısı var)
- Inversion Of Control gücü
- Kolay test edilebilir
- Gereksiz Exceptions sınıfları gizlenmiş
- Diğer API veya frameworklar için işimizi kolaylaştıran araçlar

# Neden Spring Framework? (2)

- AOP entegrasyonu ve bununla gelen kolaylaştırılmış:
  - Auditing
  - Declarative Transaction Management
  - Caching
  - Exception Handling
  - Recovery (Retry)
  - Security

# Tavsiye linkler

- <http://martinfowler.com/bliki/InversionOfControl.html>
- <http://www.springsource.org/>
- <http://martinfowler.com/articles/injection.html>



# Ders notları

- Presentasyon dosyalarını download etmek için:  
[www.barisdere.com](http://www.barisdere.com)
- Tüm Full HD videolar için:  
<http://www.youtube.com/barisdere>
- Takip etmek için: @BarisDere
- Bana ulaşmak için: [baris.dere@gmail.com](mailto:baris.dere@gmail.com)