

# Domain Nesnelerine Dependency Injection



# Domain Nesnelerine Dependency Injection

- Spring ekibi tarafından yazılmış hazır **AspectJ aspect'leri** yardımı ile Spring tarafından yönetilmeyen **domain nesnelerine dependency injection** yapılabilir
- Bu aspect'ler **spring-aspects.jar** içerisinde
- DI yapılacak sınıflar **@Configurable** ile işaretlenmelidir

# Domain Nesnelerine Dependency Injection

**@Configurable**

```
public class Account {  
    @Autowired  
    private AccountService accountService;  
    // ...  
}
```

- Bağımlılıkları enjekte etmek için domain sınıfında **@Autowired** anotasyonu kullanılabilir

# Domain Nesnelerine Dependency Injection

- **@Configurable** anotasyonunun aktive olması için `<context:spring-configured/>` elemanı XML'de tanımlanmalıdır
- Java tabanlı konfigürasyonda **@EnableSpringConfigured** kullanılmalıdır
- DI normalde **constructor'lardan sonra** gerçekleşir
- **@Configurable(preConstruction=true)** ile constructor öncesi DI yapması sağlanabilir

# Domain Nesnelerine Dependency Injection

- Domain nesnelerine DI işleminin gerçekleşebilmesi için ilgili domain sınıflarının **AspectJ aspectleri** ile **weave edilmesi** gerekir
- Bunun için ya compile time build taskları , ya da **load time weaver** kullanılabilir

# Domain Nesnelerine Dependency Injection

- **@Configurable** anotasyonunun aktive olması için `<context:spring-configured/>` elemanı XML'de tanımlanmalıdır
- Java tabanlı konfigürasyonda **@EnableSpringConfigured** kullanılmalıdır
- DI normalde **constructor'lardan sonra** gerçekleşir
- **@Configurable(preConstruction=true)** ile constructor öncesi DI yapması sağlanabilir

# İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

