



# TEORİK ÖDEV SORULARI-1

Hazırlayan:

Rana YILMAZ



## CEVAPLAR

### 1. IOC ve DI nedir?

Yazılımı oluşturan yapıların birbiriyle ilişkisinin bağ veya sınırlandırma içermemesi tercih edilir ve buna Loosely Couple (Gevşek Bağlılık) denir.

IOC yani Inversion of Control birbirine bağımlılığı az(loose coupling) olan nesneler olan nesneler oluşturmayı amaçlayan yazılım prensibidir. Nesnelerin yaşam döngüsünden sorumludur, yönetimini sağlar.

Bu prensiple bir framework kullandığımızda, yazdığımız kod bloğu çalışmadığı tüm zamanlarda, bütün kaynakları bizim yerimize kendisi ayarlar ve yönetir.

DI yani Dependency Injection sistem içerisindeki bağımlılıkları minimize etmek için bağımlılık oluşturacak parçaların ayrılıp bunların, sisteme dışarıdan verilmesi işlemidir.

### 2. Spring Beans Scopes?

Singleton: Tekli nesne oluşturmak için kullanılır. Varsayılan scope'tur.

Prototype: Her nesne oluşturma talebi için yeni nesne oluşturur.

Request: Web uygulamalarında http istekleri geçerli olduğu sürece bir nesne oluşturur.

Session: Web uygulamalarında oturum geçerli olduğu sürece bir nesne oluşturur.

Application: Web uygulamalarında uygulamaya ait tek nesne oluşturur.

Websocket: Web tabanlı websocket uygulamalarında sınıfa ait sadece tek bir nesne oluşturur.

Custom: Geliştirinin tasarladığı şekilde nesne oluşumu devam eder.

### 3. @SpringBootApplication ne yapar?

@Configuration, @EnableAutoConfiguration, @ComponentScan anotasyonlarını içeren temel bir anotasyondur.

@SpringBootApplication: Java tabanlı yapılandırma işlemi yapar, sınıfın Bean sınıfı olduğunu belirtir.

@EnableAutoConfiguration: Varsayılan yapılandırmaların otomatik taranmasını sağlar.

@ComponentScan: Projeye dahil edilen componentlerin otomatik olarak taranmasını sağlar.

#### 4. Neden Spring Boot Spring üstünde?

Spring Boot, Spring kütüphanesini kullanır. Spring Framework kullanılırken uğraşılması gereken prosedürlerin aradan kalkarak sıfırdan proje başlanmasının ve zaman kaybının önlenmesi için Spring Boot, Spring'in üstündedir.

#### 5. Singleton nedir ve nerede kullanılır?

Bean sınıfında default scopetür. Singleton scope ile tanımlanan bir bean'den bir tane initialize edilir. Bu bean ile yapılacak tüm request'ler, cache'lenmiş aynı nesne üzerinden yapılır.

#### 6. Spring Boot'taki @RestController anotasyonunu açıklayınız.

Temel amacı bitiş noktaları üretmek olan bir anotasyondur. @Controller ve @ResponseBody anotasyonlarını birlikte sağlamış olur.

#### 7. Spring ve Spring Boot arasındaki temel farklar nedir?

Spring esneklik uygulamaya odaklanır, Spring Boot kod uzunluğunu kısaltmaya ve bir web uygulaması geliştirmenin en kolay yoluna. Neredeyse hiçbir yapılandırma ayarlaması yapmadan bir uygulama geliştirmemize olanak sağlar. Çünkü Spring'i içerisinde barındırdığından otomatik olarak yapılandırır ve Spring bağımlılıklarını kolaylaştırır, versiyon çakışmalarını önler. Tomcat, Jetty, Undertow direkt gömülü olarak gelir. Maven yapılandırmasını kolaylaştırmak için başlangıç POM'larını sağlar. XML yapılandırması için kod yazmaya gerek yoktur. Container olmadan, komut satırından uygulama çalıştırılabilir.

#### 8. Neden Versiyon Kontrol Sistemi kullanırız?

Temelde projelerimizin daha rahat takibi ve kaydı için kullanırız. Bunun yanında; birden fazla kişinin çalıştığı bir projede gelişimin hızlanmasını, eski ve yeni

kodlarımız arasında gezinmemizi, aynı kod üzerinde eşzamanlı çalışabilmemizi, sorun giderme ve sürüm değişikliklerini daha rahat takip edilmemizi sağlar.

## 9.SOLID prensipleri nelerdir? Java’da basit bir örnekle açıklayınız.

S – Single-responsibility: Bir sınıf sadece bir yapmalıdır.

O – Open closed: Bir sınıfın temel özellikleri sabit olmalı, üzerine ekleme yapılabilir olmalıdır.

L – Liskov substitution: Kodlarımızda değişiklik yapmadan alt sınıfları, ana sınıfların yerine kullanabilmeliyiz.

I – Interface segregation: Bir arayüzde her şeyi yapmak yerine, özelleştirilmiş farklı arayüzler oluşturmaliyiz.

D – Dependency Inversion: Sınıflar arası bağımlıklar az olmalıdır.

## 10.RAD modeli nedir?

Rapid Application Devolopment yani Hızlı Uygulama Geliştirme uygulama fikrinin gelişmesi, belirsizliklerin netleşmesiyle ihtiyaçların daha rahat belirlenmesi için kullanılan, bir prototip oluşturan yazılım geliştirme yöntemidir.

## 11.Spring Boot Starter nedir? Nasıl kullanışlıdır?

Starterlar uygulamamıza ekleyebileceğimiz bir dizi bağımlılık tanımlayıcısıdır. Eklemek istediğimiz teknolojileri tek tek bulup ekleme zahmetinden kurtarır.

Springte starter eklemek için spring-boot-starter-eklenmek istenen teknoloji ismi kalıbı kullanılır. Kişisel bir staterinsa spring-boot ile başlamaması gerekir.

Resmi Spring Boot Staterları;

spring-boot-starter: Otomatik yapılandırma, loglama, YAML desteği

spring-boot-stater-data-jdbc: Spring Data JDBC kullanma

spring-boot-starter-data-jpa: Hibernate ile Spring Data JPA kullanma

spring-boot-starter-security: Spring Security kullanma

spring-boot-starter-web: Tomcat gömülü, Spring Mvc

## 12.Spring Boot anatasyonları nelerdir?

@Bean

@Service

@Repository

@Configuration  
@Controller  
@RequestMapping  
@Autowired  
@Component  
@SpringBootApplication  
@EnableAutoConfiguration  
@ComponentScan  
@Required  
@Qualifier  
@CookieValue  
@Lazy

### **13.Spring Boot Bağımlılık Yönetimi nedir?**

Spring Boot bağımlılıkları otomatik olarak yönetir.

### **14.Spring Boot Actuator nedir?**

Uygulamaların production ortamına hazır özellikleri (health check, disk usage, heap dump vs.) otomatik aktifleştirir ve farklı http endpoint'ler ile etkileşimde bulunmayı sağlar. Spring Boot Actuator'ün projede aktif olması için Maven dependency bloğunun POM dosyasına eklenmesi gerekir.