

Serverske veb tehnologije

- Inverzija kontrole i aspekt-orijentisano programiranje

Dragan Ivanović

Katedra za informatiku, Fakultet Tehničkih Nauka, Novi Sad

2022.

Inverzija kontrole

- Poznato i pod nazivom Dependency Injection
- Spring kontejner upravlja životnim ciklusom objekata
- Programer piše programski kod u kojem samo koristi objekte

Spring kontejner

- Spring kontejner je zadužen za
 - kreiranje,
 - inicijalizaciju,
 - konfigurisanje i
 - obezbeđivanje objekata dostupnim
- Objekti kojima kontejner upravlja se nazivaju *Beans*

Spring konfigurisanje IoC

- Konfigurisanje kontejnera se vrši putem
 - XML fajlova - tradicionalna varijanta
 - Anotacija – novija varijanta koja se preporučuje

Spring boot IoC

- Spring Primer 11

Spring anotacije IoC

- Anotacija *@Autowired* iznad objekta
 - Označava da objekat treba da bude "injektovan" od strane kontejnera
 - Programer radi sa objektom i ne mora da vodi računa o tome kako je objekat kreiran i dobavljen
 - Može da se koristi i Java anotacija *@Inject* za istu svrhu

Session bean i dependency injection

- Pored JNDI, drugi način da jedan SB dobije referencu na drugi je pomoću **dependency injection** mehanizma
- Referencu na drugi SB upisuje **kontejner** u atribut prvog SB
- Atribut je potrebno označiti anotacijom **@EJB**
- Primer 11

Session bean i dependency injection

- Dependency injection je moguć u sledećim slučajevima (bean A \rightarrow (poziva) bean B)
 - stateless \rightarrow stateless
 - stateful \rightarrow stateless
 - stateful \rightarrow stateful
- A zabranjen je u slučaju
 - stateless \rightarrow stateful

Session bean i dependency injection

- Anotacijom **@Resource** mogu da se označe atributi sledećeg tipa
 - javax.ejb.SessionContext
 - javax.sql.DataSource
 - javax.transaction.UserTransaction
 - javax.jms.Queue, javax.jms.Topic
 - ...
- Anotacijom **@PersistenceContext** označava se atribut tipa
 - javax.persistence.EntityManager
- Anotacijom **@PersistenceContextFactory** označava se atribut tipa
 - javax.persistence.EntityManagerFactory

Aspekt-orientisano programiranje (AOP)

- Sredstvo za izražavanje određenih pravila/procedura koja se mogu primeniti na više mesta u programu
- Npr. logovanje, kontrola pristupa, ... – aspekti programa koji nisu direktno vezani za poslovnu logiku i često izgledaju isto za različite poslovne procedure
- **Aspekt** je parče koda koji se može vezati za neku metodu tako da se izvrši
 - pre poziva metode
 - posle poziva metode
 - oko poziva metode (obuhvata poziv)

Spring i AOP

- MethodBeforeAdvice
- AfterReturningAdvice
- ThrowsAdvice
- MethodInterceptor (Around advice) - može sve sa ovim, ali je lepše ako se koristi neka od prethodnih specifikacija prilagođena potrebi
- <https://www.mkyong.com/spring/spring-aop-examples-advice/>

Spring boot AOP

- Spring Primer 12

Session beans i AOP

- U EJB 3.0 aspekti mogu da **obuhvate** poziv metode
- Metodu je potrebno označiti anotacijom **@Interceptor**
- Ako ima više aspekata onda **@Interceptors**
- Parametar ove anotacije je klasa koja sadrži aspekt
- Aspekt je metoda u klasi označena anotacijom **@AroundInvoke**
- Primer 12

Interceptori i životni ciklus EJB binova

- Klasa navedena kao `@Interceptor` može sadržati i metode za obaveštavanje o događajima u životnom ciklusu
- U primeru 9 te metode su bile deo bean klase
- Možemo ih premestiti u interceptor klasu
- Koristimo iste anotacije: `@PostConstruct`, `@PrePassivate`, `@PostActivate`, `@PreDestroy`
- Primer 13