

CS230 – **Distribuirani sistemi –** JunB 2021

**Teorijski deo ispita**:

1. Instrukcije koje imaju implikacije na celokupnu strukturu servlet klase su:
2. JSP direktive
3. JSP akcije
4. JSP biblioteke tagova
5. Klasa Servlet redefiniše metode:
6. get()
7. doGet()
8. post()
9. doPost()
10. Filter je Java klasa koja implementira interfejs javax.servlet.Filter od kojeg dobija sledeće metode:
11. doGet()
12. doPost()
13. doFilter()
14. init()
15. destroy()
16. CDI zrno mora da bude obeleženo:
17. Anotacijom @Named
18. Anotacijom @Bean
19. Anotacijom oblasti zrna, npr. Anotacijom @RequestScoped
20. CDI zrno nije anotirano
21. U aplikaciji može da postoji samo jedna instanca zrna:
22. Stateless
23. Stefull
24. Singletone
25. EJB Module projekat sadrži samo EJB zrna dok Enterprise Application projekat sadrži EJB zrna zajedno sa njihovim klijentima, u formi Java Web aplikacije ili "standalone" Java aplikacije?
26. Tačno
27. Netačno
28. Šta je od navedenog tačno?
29. point-to-point (PTP) messaging - U ovom slučaju JMS poruka se obrađuje od strane samo jednog JMS primaoca;
30. publish/subscribe (pub/sub) messaging - U ovom slučaju svi primaoci poruka koji su registrovani za specifičnu temu (topic) primaju i obrađuju sve poruke koje su u vezi sa datom temom.
31. publish/subscribe (pub/sub) messaging - U ovom slučaju JMS poruka se obrađuje od strane samo jednog JMS primaoca;
32. point-to-point (PTP) messaging - U ovom slučaju svi primaoci poruka koji su registrovani za specifičnu temu (topic) primaju i obrađuju sve poruke koje su u vezi sa datom temom
33. Šta je neophodno obaviti pre svakog započinjanja procesa razmene JMS poruka?
34. Dodati JMS destinaciju(JMS destinations) u aplikativni server
35. U Javi EE 7 i 8 je obavezno kreirati JMS produkciju konekcije (JMS connection factory) kao dodatka JMS destinaciji
36. Ništa od navedenog
37. REST (Representational State Transfer) predstavlja arhitekturni stil u kojem su veb servisi reprezentovani kao resursi
38. Tačno
39. Netačno
40. Šta je od navedenog tačno?
41. Java podrška za RESTful veb servise preko novog Java API za RESTful veb servise (Java API for RESTful Web Services) naziva se JAX-RS
42. JAX-RS je samostalni API
43. JAX-RS je integralni deo Java EE specifikacije

**Praktičan deo ispita:**

1. Otvoriti NetBeans IDE i kreirati Java veb aplikaciju pod nazivom **IspitJunB**;
2. Kreirati MySQL bazu podataka pod nazivom **junb**;
3. Kreirati šemu baze (uključiti konektor – Driver za MySQL bazu podataka) izvršavanjem sledećih SQL naredbi:

drop table if exists PRODUCT;

drop table if exists PRODUCT\_CART;

drop table if exists ROLE;

drop table if exists SHOPPING\_CART;

drop table if exists CLIENT;

create table PRODUCT

(

PRODUCT\_ID int not null auto\_increment,

NAME varchar(128),

PRICE float,

primary key (PRODUCT\_ID)

);

create table PRODUCT\_CART

(

PRODUCT\_ID int not null,

CART\_ID int not null,

QUANTITY int,

primary key (PRODUCT\_ID, CART\_ID)

);

create table ROLE

(

ROLE\_ID int not null auto\_increment,

ROLE\_NAME varchar(128),

primary key (ROLE\_ID)

);

create table SHOPPING\_CART

(

CART\_ID int not null auto\_increment,

CLIENT\_ID int,

DATE timestamp,

primary key (CART\_ID)

);

create table CLIENT

(

CLIENT\_ID int not null auto\_increment,

ROLE\_ID int,

USERNAME varchar(128),

PASSWORD varchar(128),

primary key (CLIENT\_ID)

);

alter table PRODUCT\_CART add constraint FK\_PRODUCT\_CART foreign key (PRODUCT\_ID)

references PRODUCT (PRODUCT\_ID) on delete restrict on update restrict;

alter table PRODUCT\_CART add constraint FK\_PRODUCT\_CART2 foreign key (CART\_ID)

references SHOPPING\_CART (CART\_ID) on delete restrict on update restrict;

alter table SHOPPING\_CART add constraint FK\_CLIENT\_CART foreign key (CLIENT\_ID)

references CLIENT (CLIENT\_ID) on delete restrict on update restrict;

alter table CLIENT add constraint FK\_HAS\_ROLE foreign key (ROLE\_ID)

references ROLE (ROLE\_ID) on delete restrict on update restrict;

insert into ROLE (ROLE\_ID, ROLE\_NAME) values (1, 'Admin');

insert into ROLE (ROLE\_ID, ROLE\_NAME) values (2, 'User');

insert into PRODUCT (PRODUCT\_ID, NAME, PRICE) values (1, 'Product 1', 100);

insert into PRODUCT (PRODUCT\_ID, NAME, PRICE) values (2, 'Product 2', 200);

insert into PRODUCT (PRODUCT\_ID, NAME, PRICE) values (3, 'Product 3', 300);

insert into CLIENT (CLIENT\_ID, ROLE\_ID, USERNAME, PASSWORD) values (1, 1, 'admin', 'admin');

insert into CLIENT (CLIENT\_ID, ROLE\_ID, USERNAME, PASSWORD) values (2, 2, 'user', 'user');

insert into CLIENT (CLIENT\_ID, ROLE\_ID, USERNAME, PASSWORD) values (3, 2, 'test', 'test');

insert into SHOPPING\_CART (CART\_ID, CLIENT\_ID, DATE) values (1, 3, '2020-3-13 4:13:18');

insert into SHOPPING\_CART (CART\_ID, CLIENT\_ID, DATE) values (2, 2, '2020-3-13 4:13:18');

insert into SHOPPING\_CART (CART\_ID, CLIENT\_ID, DATE) values (3, 2, '2020-4-17 8:21:56');

insert into PRODUCT\_CART (PRODUCT\_ID, CART\_ID, QUANTITY) values (1, 2, 2);

insert into PRODUCT\_CART (PRODUCT\_ID, CART\_ID, QUANTITY) values (2, 1, 4);

insert into PRODUCT\_CART (PRODUCT\_ID, CART\_ID, QUANTITY) values (3, 3, 3);

1. Nad kreiranom bazom podataka kreirati JPA entitete za svaku od tabela iz baze podataka;

(kreirati i odgovarajući Data Source za MySQL bazu podataka sa JNDI nazivom **jdbc/ispitJunB**);

1. Prihvatiti sva podešavanja koja NetBeans čarobnjak nudi i definisati naziv paketa kao **ispitjunb.jpa.entiteti**;
2. Kreirati kontrolere za entitetske klase u paketu **ispitjunb.jpa.kontroleri**;
3. Kreirati JSF stranice za CRUD operacije nad entitetima, odnosno bazom podataka;
4. Kreirati RESTFul web servis nad entitetom **PRODUCT**;
5. Kreirati konzolnu aplikaciju kao klijent za kreirani RESTFul web servis i demonstrirati operacije GET i POST
6. Kreirati stranu koja će sadržati polja za unos korisničkog imena i pasvorda za logovanje klijenta pod nazivom **login.xhtml**
7. Ukoliko je korisnik uspešno prijavljen preusmeriti aplikaciju na **index.xhtml**. Napraviti odgovarajuće Java klase koje će ovo omogućiti.
8. Za login koristiti JSF, CDI: model i kontroler.

**BODOVANJE:**

Teorijski deo 10 poena (10 x 1 poen)

Praktičan deo 20 poena (Koraci 1-4 nose 3 poena; koraci 5-7 nose 5 poena; koraci 8-9 nose 5 poena; koraci 10-12 nose 7 poena).

**NAPOMENA:** Student je položio ispit ako u oba dela osvoji najmanje 50% poena tj. minimalno 5 na teorijskom delu i 10 na praktičnom.

Predmetni nastavnik

Dr Vladimir Milićević, docent