

TEST 2

1. Kako dobijate objekat klase InetAddress?

- Objekat klase InetAddress dobijamo pomoću iskaza:

InetAddress inetAddress = socket.getInetAddress();

2. Koje metode koristite da bi dobili IP adrese i naziv host računara iz InetAddress?

- IP adresu dobijamo pomoću: **getHostAddress()** metode.
Naziv računara dobijamo pomoću: **getHostName()** metode.

3. Šta će se desiti ako nije u liniji 227 llistinga klase TicTacToeClient definisana veličina ćelije?

- Ne znam ?????????????????????????????

4. Ako igrač nije na potezu a klikne na praznom polju, šta će klijentski program dat klasom TicTacToeClient uraditi?

- Neće uraditi ništa, već će čekati na potez pravog igrača, zato što po uspostavljanju sesije, server naizmenično prima poteze od igrača i po prijemu poteza od igrača server određuje status igre. Ako igra nije završena, server šalje status continue i šalje potez igrača drugom igraču.

5. Šta od navedenog je tačno? ***

- Kada je veza uspostavljena, klijent i server komuniciraju preko softverskih utičnica, tj. objekata klase ServerSocket i Socket.
- Utičnice su krajnje tačke logičkih veza između dva računara i koriste se za slanje i primanje podataka.

6. Kada server prihvati vezu, komunikacija između servera i klijenta se vodi na isti način kao i kod U/I tokova podataka (strimova)? ***

- True

7. Server koristi ime definisano promenljivom serverName, čija vrednost je ili Internet ime domen servera ili njegova IP adresa, da ili ne? ***

- True

8. Šta od navedenog je tačno? ***

- TCP omogućava da dva servera uspostave vezu i da razmene tokove podataka.
- IP adresa je sačinjena od četiri broja od 0 do 255 između kojih se nalaze tačke.
- IP je adresa koja na jedinstveni način utvrđuje računar na internetu.

9. Da li je moguće da se klijent i server izvršavaju na različitim računarima a da to ne ometa njihovu komunikaciju? ***

- Da

TEST 3

1. Opišite unapred pripremljene iskaze. Kako kreirate primerak klase PreparedStatement? Kako izvršavate PreparedStatement? Kako unosite vrednosti parametara u PreparedStatement objekat?

- PreparedStatement omogućava kreiranje parametrizovane SQL izjave. Kreiramo ga sa:
PreparedStatement preparedStatement = connection.prepareStatement("insert into Student (firstName, mi, lastName)" + "values(?,?,?)");
- Izvršenje se vrši pozivom **executeQuery()**.
- Unosimo ih pomoću metode **setX(int parameterIndex, X value)**; Na primer:
`preparedStatement.setString(1, "A");`

2. Koje su povoljnosti koje pruža primena unapred pripremljenih iskaza?

- Prednost je u tome što PreparedStatement omogućava predkompajliranje SQL iskaza sa ili bez parametara, čime su ovi iskazi efikasniji za ponovna izvršenja.

3. Za šta služi DatabaseMetaData? Opišite metode DatabaseMetaData. Kako dobijate primerak od DatabaseMetaData?

- DatabaseMetaData je interfejs koji suži za dobijanje informacija o bazi podataka. Za dobijanje primerka interfejsa DatabaseMetaData za bazu podataka koristi se metoda `getMetaData()`.

4. Za šta služi ResultSetMetaData? Opišite metode u ResultSetMetaData? Kako dobijate primerak ResultSetMetaData?

- ResultSetMetaData je interfejs koji služi za dobijanje informacija u specificiranom ResultSet. Za dobijanje primerka ResultSetMetaData, koristi se metoda **getMetaData()** na skupu rezultata.

5. Kako nalazite broj kolona u skupu rezultata? Kako nalazite nazive kolone u skupu rezultata?

- Broj kolona u skupu rezultata nalazimo pomoću metode **getColumnCount()**.
- Nazive kolona u skupu rezultata nalazimo pomoću metode **getColumnName()**.

6. Koje su prednosti razvoja aplikacije sa bazom podataka upotrebom Jave?

- Prednost je u tome što ove aplikacije mogu da koriste više sistema baza podataka koji su u mreži računara koje koriste.

7. Opišite sledeće JDBC interfejs: Driver, Connection, Statement i ResultSet?

- **Driver** – je uslužni program
- **Connection** – omogućava povezivanje sa bazom podataka.
- **Statement** – kreira SQL iskaz.
- **ResultSet** – obrađuje rezultate.

8. Kako instalirate JDBC drajver? Koje su drajver klase za MySQL, Access i Oracle?

- JDBC drajver instaliramo pomoću: **Class.forName(“com.mysql.jdbc.Driver”);**
Drajver klase su:
Access: **sun.jdbc.odbc.jdbcOdbcDriver**
MySQL: **com.mysql.jdbc.Driver**
Oracle: **oracle.jdbc.driver.OracleDriver**

9. Kako kreirate vezu sa bazom podataka? Koje se URL koriste za MySQL, Access i Oracle?

- Povezivanje baze podataka se vrši pozivanjem statičkog metoda **getConnection()** u klasi DriverManager.
- URL za razne DBMS su:**
MySQL: **jdbc:mysql://hostname/dbname**
Access: **jdbc:odbc:datasource**
Oracle: **jdbc:oracle:thin:@hostname:port#:oracleDBSID**

10. Kako kreirate Statement i izvršavate SQL iskaz?

- Statement objekat definiše SQL islake koje DBMS treba da izvrši i da vrati dobijeni rezultat. On izgleda ovako: **Statement statement = connection.createStatement();**
SQL upit se može izvršiti upotrebom **executeQuery(String sql)**, a rezultat se vraća u obliku SQL objekta.

11. Kako dobijate vrednosti u ResultSet?

- Za dobijanje vrednosti u ResultSet koristimo metod **getMetaData()**.

12. Šta su superključevi, ključevi kandidati i primarni ključevi?

- **Superključ** je atribut ili skup atributa koji na jedinstveni način identifikuje jednu relaciju. Skup svih atributa u jednoj relaciji je superključ.
- **Ključ-kandidat** je podskup superključa koji može da ima više ključa-kandidata.
- **Primarni ključ** je jedan od ključeva-kandidata određen od strane projektanta baze i često se koristi za identifikaciju slogova table.

13. Šta je sekundarni (strani) ključ?

- Sekundarni (strani) ključ ograničava veze između relacija.

14. Da li neka relacija ima više od jednog primarnog ključa u istoj relaciji?

- Ne

15. Da li strani ključ mora da ima isto ime kao što je pozvani primarni ključ?

- Ne

16. Da li strani ključ može da ima vrednost null?

- Ne

17. Izbaciti pojmove koji ne pripadaju skupu? ***

- Set i HashSet.

18. JDBC je Java API koji se koristi za razvoj aplikacija koje koriste relacione baze podataka? ***

- True

19. Čemu služi interfejs Connection? ***

- Za povezivanje sa bazom podataka.

20. Kako se naziva univerzalni jezik za pristupanje relacionim sistemima baza podataka? ***

- SQL

21. Šta je RDBMS? ***

- Sistem za upravljanje bazom podataka.

TEST 4

1. Šta su DAO objekti?

- DAO objekti su objekti koji omogućavaju pristup bazi podataka unutar DAO klase.

2. Zašto primena DAO šablona predstavlja dobru praksu za pisanje koda koji se obraća bazi podataka?

- Zato što se njihovom primenom postiže jasno razdvajanje modula programa pomoću kojeg ostali slojevi aplikacije, poput logike korisničkog interfejsa ili poslovne logike ne zavise od logike perzistencije.

3. Šta je obeleženi upit?

- To je poseban specificirani upit koji sadrži listu od više upita.

4. Šta je JPQL?

- JPQL je JPA specifični jezik upita, čija je sintaksa veoma slična SQL jeziku.

5. Navedite i objasnite anotacije za proveru podataka?

- **@NotNull** – ne dozvoljava polju da prihvati null vrednost.
- **@Size** – ukazuje na maksimalan broj karaktera kojih osobina zrna može da prihvati.
- **@Pattern** – obezbeđuje da se obeleženo polje podudara sa regularnim iskazom.

6. Navedite i objasnite kardinalnosti relacija JPA entiteta?

- **@OneToMany** ← **Ovde jos napisi objasnjenje za svaki**
- **@JoinColumn**
- **@ManyToMany**
- **@ManyToMany**
- **@OneToOne**

7. Pokazati na primeru kardinalnosti JPA entiteta?

- **@ManyToMany**, ovde je primer slučaj sa porudžbinama, gde jedna porudžbina može da sadrži više proizvoda i jedan proizvod može da bude sadržan u više porudžbina.

8. Objasnite proceduru automatskog generisanja JSF stranica iz JPA entiteta?

- Koraci su:

1) Klik na File → New File → odaberemo kategoriju datoteke (Java Server Faces ili Persistence) → odaberemo tip datoteke (JSF Pages from Entity Clases).

2) Klik na Next → markiraju se klase → klik na Add → klik na Next.

Prolaskom kroz zadati Wizard, kreiraju se JSF stranice za sve klase entiteta, koje su bile izabrane.

9. Objasnite proceduru kreiranja JPA entiteta iz tabela baze podataka?

- ?????????????????????????????

10. Objasnite anotacije koje koriste JPA entiteti?

- Java klasa koja se odnosi na JPA entitet je obeležena anotacijom @Entity ?????????????

11. Šta su JPA entiteti?

- JPA entiteti su Java klase čija se polja čuvaju (perzistiraju) u bazama podataka pomoću JPA API.

12. Kojom anotacijom se obeležavaju JPA entiteti?

- @Entity

13. Šta je connection pool i koje su mu prednosti?

- Informacije kao što su: naziv servera, porta, kredencijala i slično, koje omogućavaju konektovanje na bazu podataka, nazivaju se connection pool.

Pomoću connection pool-a se povećavaju performanse aplikacije zahvaljujući tome što se izbegavaju zahtevne akcije otvaranja i zatvaranja konekcije.

14. JPQL je JPA specifični jezik upita čija je sintaksa veoma slična SQL jeziku? ***

- True

15. Izbaci pojmove koji ne pripadaju navedenim? ***

- JDBC (a pripadaju: EclipseLink, Hibernate, TopLink, Essentials, iOpenJPA, KODO)

16. Da li JPA dozvoljava automatsko generisanje primarnih ključeva? ***

- Da primenom anotacije @GeneratedValue u kombinaciji sa anotacijom @Id.

17. EclipseLink je podrazumevana JPA implementacija za GlassFish aplikativni server, da ili ne? ***

- True

18. DAO šabloni dizajna čuvaju sve funkcionalnosti pristupa bazi podataka unutar DAO klase? ***

- True

TEST 5

1. Šta je HQL?

- **HQL** (Hibernate Query Language) je objektno-orjentisan jezik za rad sa bazom podataka u okviru Hibernate ORM. On radi sa trajnim objektima i njihovim svojstvima. HQL upiti se prevode u SQL upite koji izvršavaju potrebne akcije na bazi podataka.

2. Koji značaj ima objekat Criteria?

- Objekat **Criteria** omogućava da definišemo kriterijume za programirani odabir objekata primenom pravila filtriranja i logičkih uslova.

3. Za šta je moguće iskoristiti standardne SQL upite pored HQL upita?

- Moguće je iskoristiti za dobijanje specifičnih svojstava baze podataka kao što je korišćenje **CONNECT** ključne reči.
Hibernate 3.X dozvoljava primenu SQL upita, memorisanih procedura, operacija create, update, delete i load za sve slogove.
Aplikacija će kreirati originalne SQL upite iz Session objekta primenom metoda **createSQLQuery()**.
Moguće je povezati dobiveni SQL rezultat sa postojećim HQL rezultatom, primenom metoda: **addEntity()**, **addJoin()**, **addScalar()**.

4. Koje su prednosti Hibernate anotacija?

- Hibernate Annotations je moćan način obezbeđivanja meta podataka za mapiranje objekata u relacione tabele. Svi meta podaci se stavljaju zajedno u POJO java fajl zajedno sa programskim kodom, pa korisnik odatle razume strukturu tabela i simultano vreme razvoja.

5. Objasnite najčešće korišćene anotacije?

- Najčešće korišćene anotacije su:
 - @Entity** – njome se obeležavaju JPA entiteti
 - @Table** – dozvoljava specificiranje detalja tabele koja će se koristiti za trajno memorisanje u bazi podataka.
 - @Id** – omogućava pristup svojstvima nekog objekta za vreme izvršenja programa.
 - @GeneratedValue** – omogućava promenu vrednosti primarnog ključa. Ima 2 parametra: Strategy i Generator.Ove anotacije sve zajedno daju odgovarajuće instrukcije mapiranja programu tokom njegovog izvršavanja.

6.