

1. Navedite osnovne korake (6) u postupku razvoja programskih rešenja.

1. kodiranje pojedinačnih modula,
2. unos programskog koda
3. testiranje i verifikacija,
4. dokumentovanje,
5. integrisanje programskih modula,
6. verifikacija i validacija programskog rešenja.

2. Navedite četiri osnovne grupe dokumentacije u sklopu programskih rešenja.

- Instalaciona,
- deinstalaciona,
- korisnicka,
- programska dokumentacija.

3. Navedite tri osnovne podgrupe u sklopu programske dokumentacije.

- In-line,
- on-line,
- operativna dokumentacija.

4. Šta se podrazumeva pod pojmom IN-LINE programska dokumentacija?

komentari u kodu.

5. Šta se podrazumeva pod pojmom ON-LINE programska dokumentacija?

Pod on-line se podrazumeva help, kontekstna pomoc.

6. Šta se podrazumeva pod pojmom OPERATIVNA programska dokumentacija?

Tutorijali : asistencija pri ovladavanju softverskim proizvodom.

[off-line : uputstva, prirucnici]

7. Koja tri koraka obuhvata UVOĐENJE U EKSPLOATACIJU programskog rešenja?

1. instaliranje programskog rešenja,
2. obuka korisnika,
3. operativna verifikacija.

8. Objasnite potrebu postojanja koraka IZVOĐENJA IZ EKSPLOATACIJE PROGRAMSKOG REŠENJA.

Da se softver automatski ukloni iz sistema i pocisti za sobom, a ne da korisnik mora sam da brise pa zatim da trazi dodatne ostatke.

9. Navedite četiri osnovna principa objektne orijentisanosti!

Apstrakcije, enkapsulacija, modularnost, hijerarhija.

10. Šta se podrazumeva pod pojmom APSTRAKCIJE u kontekstu principa objektne orijentacije?

Apstrakcije omogućavaju rukovanje složenosti koncentracijom na esencijalne karakteristike po kojima se jedan entitet razlikuje od drugih.

11. Šta se podrazumeva pod pojmom ENKAPSULACIJE u kontekstu principa objektne orijentacije?

Enkapsulacija razdvaja implementaciju ponašanja objekta od njegovog javnog interfejsa.

Skrivanje informacija omogućava da se ponašanje objekta koristi bez poznavanja na koji način je to ponašanje implementirano.

Obezbeđuje:

zastitu unutrašnjeg stanja objekta od spoljasnih promena

promenu u implementaciji ponašanja bez uticaja na druge objekte.

12. Šta se podrazumeva pod pojmom MODULARIZACIJE u kontekstu principa objektne orijentacije?

Dekompozicija neke složenosti na manje delove kojima se može lakše rukovati.

13. Šta se podrazumeva pod pojmom HIJERARHIJE u kontekstu principa objektne orijentacije?

Predstavlja uređenje ili rangiranje apstrakcija u formi drveta. Elementi na istom nivou hijerarhije trebali bi biti na istom nivou apstrakcije.

14. Navedite bar pet, od devet, elemenata koji čine objektu paradigmu.

Objekat, klasa, atribut, operacija, interface, komponenta, paket, podsistem, relacije.

1. Navedite osnovne elemente koji su, prema Wasserman-u, izmenili softversko inženjerstvo.

- Desktop računari,
- objektna tehnologija,

- životni ciklus softvera u obliku vodopada,
- time-to-market,
- razvoj računarskih mreža,
- korisnicki interfejsi,
- ekonomski rast.

2. Koja dva elementa definišu svaku inženjersku aktivnost?

proces i proizvod.

3. Šta se, sa aspekta inženjerske aktivnosti, podrazumeva pod pojmom proces?

niz koraka koji uključuju: aktivnosti, ograničenja i resurse.

4. Šta se, sa aspekta inženjerske aktivnosti, podrazumeva pod pojmom proizvod?

Klase softverskih sistema i opšti zahtevi prema softverskom proizvodu.

5. Navedite osnovne karakteristike svakog inženjerskog procesa. (ima ih 6)

Propisuje sve glavne aktivnosti,

Poseđuje skup vodećih principa koji definišu ciljeve pojedinačnih aktivnosti,

Aktivnosti su organizovane u nizove (sekvence),

Svaka aktivnost u procesu ima kriterijum za pokretanje i završetak,

Koristi resurse u skladu sa postavljenim ograničenjima i daje prelazne i konačnu formu proizvoda,

Može posedovati unutrašnju strukturu.

6. Koje dve dimenzije karakterišu pristup softverskom inženjerstvu sa aspekta procesa?

aspekti i aktivnosti.

7. Koji elementi čine dimenziju AKTIVNOSTI u sklopu modela softverskog procesa?

- razvoj (dizajn),
- kontrola,
- upravljanje,
- rad.

8. Šta, prema osnovnom modelu softverskog procesa, obuhvata aktivnost RAZVOJ?

zahtevi, analiza, specifikacija dizajna, implementacija, testiranje.

9. Šta, prema osnovnom modelu softverskog procesa, obuhvata aktivnost KONTROLA?

- Potvrda kvaliteta,
- konfigurisanje,

- validacija i verifikacija.

10. Šta, prema osnovnom modelu softverskog procesa, obuhvata aktivnost UPRAVLJANJE?

- Planiranje projekta,
- alokacija resursa,
- procena troškova,
- ugovaranje.

11. Šta, prema osnovnom modelu softverskog procesa, obuhvata aktivnost RAD?

Obuka, prelazak, rad, gašenje.

12. Koji elementi čine dimenziju ASPEKATA u sklopu modela softverskog procesa?

Apstrakcije, predstavljanje, metode, alati, dejstva, komunikacija.

13. Šta, prema osnovnom modelu softverskog procesa, obuhvata aspekt APSTRAKCIJA?

Obuhvata fundamentalne principe i formalne modele (modeli procesa razvoja softvera, konacni automati i petrijeve mreze, cocomo, modularnost, skrivanje informacija).

14. Šta, prema osnovnom modelu softverskog procesa, obuhvata aspekt PREDSTAVLJANJE?

Obuhvata notacione aspekte i jezike (java prog jez, uml use-case, tabele odlucivanja i dijagrami toka podataka, pert-dijagrami).

15. Šta, prema osnovnom modelu softverskog procesa, obuhvata aspekt METODI?

Obuhvataju formalne metode, metodologije i tekucu praksu (dokaz korektnosti programa - metod verifikacije, OO dizajn - metodologija projektovanja, OOP - tekuca implementaciona praksa).

16. Šta, prema osnovnom modelu softverskog procesa, obuhvata aspekt ALATI?

Obuhvataju pojedinačne softverske alate kao i integrisane skupove alata (opšte namene, namenjeni dizajnu i implemtaciji, za podršku upravljanju projektom)

17. Šta, prema osnovnom modelu softverskog procesa, obuhvata aspekt DEJSTVA?

Obuhvata merenja, analizu i evaluaciju softverskih procesa, uticaj softvera na organizacione sisteme, mere i standarde.

18. Šta, prema osnovnom modelu softverskog procesa, obuhvata aspekt KOMUNIKACIJA?

Pisanu i govornu komunikacija, kao i dokumentaciju (on-line[help, kontekstna pomoc], off-line[uputstva, prirucnici], in-line[komentari u programu], tutorial).

19. Šta se podrazumeva pod pojmom životni ciklus softvera?

Životni ciklus softvera obuhvata put od njegovog konceptualnog specificiranja, preko implementacije, isporuke, koriscenja, odrzavanja i migracija do njegovog izvodjenja iz eksploatacije.

20. Na koje dve osnovne grupe se dele modeli životnog ciklusa softvera?

- modele oslonjene na jednu verziju softverskog proizvoda I
- modele oslonjene na više verzija softverskog proizvoda.

21. U koju grupu modela životnog ciklusa softvera spada tzv. big-bang model?

Modeli oslonjeni na jednu verziju.

22. Navedite osam faza u procesu razvoja softvera po modelu vodopada?

1. Analiza zahteva,
2. Dizajn sistema,
3. Dizajn programa,
4. Kodiranje,
5. Testiranje i integracija,
6. Testiranje sistema,
7. Prihvatanje,
8. Rad i održavanje.

23. Koji su osnovni nedostaci modela vodopada?

U procesu izrade ne prelazi se na sledeći korak dok se ne završi prethodni, te je sam postupak vremenski zahtevan i nije zgodan za unapredjenja proizvoda.

24. Kakva je uloga izrade PROTOTIPA u sklopu modifikovanog modela vodopada?

Izrada prototipa je način da se izborimo sa neodređenostima prisutnim u ranim fazama razvoja (sa zahtevima korisnika u pogledu dizajna sistema i programa).

25. Koja je poslednja faza u sklopu modifikovanog modela vodopada sa prototipovima iz koje je moguće uticati na softverske zahteve?

Dizajn sistema.

26. U čemu je razlika između inkrementalnog i iterativnog razvoja softvera?

Inkrementalno - dopunjava proizvod u svakoj fazi.

Iterativno - prerađuje proizvod u svakoj fazi.

27. Navedite dva osnovna predstavnika iterativnih modela životnog ciklusa softvera.

Spiralni model i Scrum model.

28. Kakva je razlika između spiralnog i cikličkog modela životnog ciklusa softvera?

29. Koje dve dimenzije karakterišu pristup softverskom inženjerstvu sa aspekta proizvoda?

30. Šta se podrazumeva pod pojmom klase softverskih sistema?

Odnos prema okruženju,
interne karakteristike,
genericke oblasti primene.

31. Šta se podrazumeva pod pojmom opštih zahteva prema softverskom proizvodu?

dostupnost - odzivnost u svakom momentu,

adaptivost - laka prilagodivost drugim recimo uslovima rada, zahtevima rada,

raspolozivost - slobodan za rad (???),

kompatibilnost - operativna i funkcionalna uskladjenost sa recimo, OS-om,

korektnost - funkcionalna i operativna tacnost pri radu,

otpornost na greske - mogucnost da odgovori na sve iznenadne situacije ne remeteci ni jedan od ostalih 19 zahteva,

integritet - posedovanje jake koncepcijske funkcionalne i operativne povezanosti,

operativnost - zeljeni stepen odzivnosti i izvrsnosti zadatka, operaija, itd,

mogucnost odrzavanja - lako dogradjivanje novih f-ja,

performansa - brz i korektan rad ili sta vec (prica je ista kao za pitanje sta je optimalno - mora se znati kriterijum koji nas interesuje),

prenosivost - laka prilagodivost drugim platformama,

zastita - nesto kao hack-proofness,

pouzdanost - verovatnoca greske pri normalnim uslovima rada - idealno nula,

robusnost - sposobnost da odrzi ostale performanse na zavidnom nivou u uslovima u kojima je za ocekivati da vidino opadnu,

bezbednost - dostupan i raspoloziv samo, jedino i iskljucivo onome kome je namenjen,

sigurnost - pouzdanost, ali pri svim, a narocito nezelenim, uslovima rada,

mogucnost testiranja - modularna prilagodjenost najsirem mogucem spektru testova,

mogucnost koriscenja - laka i raznovrsna,

efikasnost - sa sto je moguće manje utrosenih resursa, izvršenje sto je moguće više zadataka.

32. Kako se mogu klasifikovati softverski proizvodi sa aspekta odnosa prema okruženju?

1. Batch,
2. interaktivni,
3. reaktivni,

4. real time,
5. sadržani (embedded),
6. distribuirani,
7. konkurentni,
8. mrežni.

33. Kako se mogu klasifikovati softverski proizvodi sa aspekta internih karakteristika?

Table driven,
Process driven,
Knowledge based.

34. Kako se mogu klasifikovati softverski proizvodi sa aspekta oblasti primene?

Avio saobraćaj,
komunikacioni sistem,
Operativni sistem,
Sistemi za rukovanje bazama podataka,
CASE CAD CAM alati.

35. Kakva je razlika između interaktivnih i reaktivnih sistema?

Interaktivni su oni koji imaju sposobnost da uspostave komunikaciju između korisnika i okruženja. Reaktivni sistemi su oni koji imaju sposobnost da odgovore na sve zahteve korisnika.

36. Šta se podrazumeva pod pojmom integriteta softverskog proizvoda?

Sprecavanje neovlasćenje promene tog proizvoda (konzistentnost i kontrola kvaliteta).

37. Koje tri apstrakcije čine opšti model softverskog proizvoda?

Specifikacija zahteva,
specifikacija dizajna,
implementacija.

1. Šta se podrazumeva pod pojmom sistem?

Organizovan skup činilaca, sa jasno definisanim međusobnim vezama, koji deluju radi ispunjenja nekog postavljenog cilja.

2. Koje elemente poseduje svaki sistem?

- Misiju,
- organizacionu strukturu,

- funkcionalnu strukturu,
- informacionu strukturu,
- upravljacku strukturu.

3. Šta se podrazumeva pod misijom sistema?

cilj ili skup ciljeva zbog kojih postoji.

4. Šta se podrazumeva pod organizacionom strukturom sistema?

skup činilaca i veze medju njima koje formiraju topologiju sistema.

5. Šta se podrazumeva pod funkcionalnom strukturom sistema?

aktivnosti koje sprovodi.

6. Šta se podrazumeva pod informacionom strukturom sistema?

Semanticke predstave za obavljanje funkcionalnih transformacija.

7. Šta se podrazumeva pod upravljačkom strukturom sistema?

Prinuda koja uskladjuje ukupno ponašanje sistema.

8. Zbog čega se smatra da sistemi nastali delovanjem čoveka spadaju u tzv. nedeterminističke?

zato što su podložni evolucionim i revolucionarnim promenama.

kako je u biti sistema organizovanih od strane čoveka SAM ČOVEK čije ponašanje je, u opštem slučaju, spontano nepredvidivo, ovakvi sistemi spadaju u klasu nedeterminističkih.

9. U čemu se sastoji paradoks osnovne teorije informacija?

Paradoks teorije informacija leži u činjenici da OČEKIVANA PORUKA ili ponašanje nosi MALU KOLIČINU INFORMACIJA (budući da je njen uticaj na buduća ponašanja mali) u odnosu na NEOČEKIVANU PORUKU ili ponašanje koje, u opštem slučaju, uzrokuje dramatične promene u budućem ponašanju.

10. Kako se, na osnovu opšte teorije informacija, može definisati INFORMACIJA?

$I = (O, P, S, K)$

O - objekat (predmet) informacije,

P - prenosnik informacije,

S - semantička interpretacija (značenje),

K - količina informacije.

11. Kako se, sa aspekta opšte teorije informacija, definiše KOLIČINA INFORMACIJE?

Odredjuje smanjenje stepena neodređenosti (entropije).

12. Objasnite ulogu OBJEKTA informacije u sklopu definicije informacije sa aspekta opšte teorije informacija.

Objekat predstavlja predmet informacije.

13. Objasnite ulogu PRENOSNIKA informacije u sklopu definicije informacije sa aspekta opšte teorije informacija.

Posrednik pri prenosu informacija od jedne tacke do druge.

14. Objasnite ulogu SEMANTIKE informacije u sklopu definicije informacije sa aspekta opšte teorije informacija.

Predstavlja značenje informacije.

15. Šta se podrazumeva pod pojmom informacioni sistem?

Deo razmatranog sistema koji snabdeva informacijama sve nivoe upravljanja i odlucivanja u sklopu razmatranog organizacionog sistema.

16. Šta se podrazumeva pod pojmom automatizovani informacioni sistem?

Informacioni sistem za ciju implementaciju se koriste automatizovana tehnicka sredstva.

17. Šta predstavlja POSIX - model na računarima baziranih sistema?

U današnje vreme su jedino od interesa automatizovani informacioni sistemi oslonjeni na računarske sisteme kao tehnička sredstva za implementaciju. Na osnovu toga možemo zaključiti da savremeni informacioni sistemi spadaju u kategoriju na računarima baziranih sistema (Computer-based systems - CBS), pri čemu se kao referentni model koristi distribuirani model obradnih jedinica poznat pod nazivom Posix-distribuirani model.

Portable Operating System Interface - familija standarda za održavanje kompatibilnosti između operativnih sistema.

18. Kakva je uloga OBJEKATA ZA RUKOVANJE INFORMACIJAMA u sklopu Posix-distribuiranog modela na računarima baziranih sistema?

Omogućuju rukovanje informacijama??

19. Kakva je uloga KOMUNIKACIONIH OBJEKATA u sklopu Posix-distribuiranog modela na računarima baziranih sistema?

Omogućuje razmenu podataka među obradnim jedinicama.

20. Kakva je uloga OBRADNE JEDINICE u sklopu Posix-distribuiranog modela na računarima baziranih sistema?

Obradjuje podatke dobijene od korisnika i komunikacionih objekata.

21. Navedite osnovne faze u evoluciji INFORMACIONIH SISTEMA.

sistemi za obradu podataka,
klasicni upravljacki informacioni sistemi,

interaktivni (on-line) upravljački informacioni sistemi,
informacioni sistemi za podršku odlučivanju,
inteligentni informacioni sistemi.

22. Šta se podrazumeva pod pojmom SISTEMI ZA OBRADU PODATAKA?

Sistemi za obradu podataka predstavljaju, u današnje vreme, samo istorijsku fazu transformacije informacionih sistema, budući da ih, sa aspekta savremene naučne misli u domenu razvoja informacionih tehnologija, gotovo i ne možemo smatrati informacionim sistemima.

23. Šta se podrazumeva pod pojmom KLASIČNI UPRAVLJAČKI INFORMACIONI SISTEMI?

Osnovna karakteristika klasičnih upravljačkih informacionih sistema je grupisanje oko "proizvodnje" informacija neophodnih upravljačkom delu složenih sistema kao osnov za planiranje, sprovođenje planova i kontrolu izvršenja planskih zadataka. U sklopu ovih informacionih sistema izvršena je integracija funkcija i obezbeđeno racionalno korišćenje podataka u svim segmentima organizacione strukture sistema, za čije potrebe su projektovani, ali kao oslonac u procesu upravljanja koriste produkte "istorijskih događanja" (upravljanje po posledici). Ovaj vid informacionih sistema, iako poseduje niz nedostataka sa aspekta savremene teorije upravljanja, predstavlja u današnje vreme, posebno u našim uslovima, najzastupljeniju formu informacionih sistema.

24. Šta se podrazumeva pod pojmom INTERAKTIVNI ON-LINE UPRAVLJAČKI INFORMACIONI SISTEMI?

Interaktivni (on-line) upravljački informacioni sistemi poseduju sve dobre karakteristike klasičnih upravljačkih informacionih sistema ali se, u podršci vecini funkcionalnih transformacija, oslanjaju na direktnu interakciju čovek-računar pri čemu se podaci, neophodni za formiranje odgovora na proizvoljan informacioni zahtev, direktno unose, dok se rezultati njihovog procesiranja distribuiraju odmah po obradi i ostaju na raspolaganju ovlašćenim korisnicima u režimu direktnih upita posredstvom radnih stanica računarskog sistema koji služi kao tehnička podrška.

25. Šta se podrazumeva pod pojmom INFORMACIONI SISTEMI ZA PODRŠKU ODLUČIVANJU?

Informacioni sistemi za podršku odlučivanju predstavljaju savremenu formu informacionih sistema razvijenu kao logično unapređenje koncepta interaktivnih upravljačkih informacionih sistema. Ogromna masa analitičkih podataka, i iz njih obradom izvedenih informacija, koji su prisutni u sklopu savremenih informacionih sistema, nije pogodna za BRZO DONOŠENJE ODLUKA i vođenje "politike" sistema u skladu sa njegovom osnovnom misijom, ako nije udružena sa funkcijama AGREGACIJE, SEMANTIČKE INTERPRETACIJE i automatizovane podrške generisanju različitih scenarija (varijanti) mogućeg ponašanja upravljanog sistema.

26. Šta se podrazumeva pod pojmom INTELIGENTNI INFORMACIONI SISTEMI?

predstavljaju objedinjavanje brojnih tehnologija koje su u ranijim fazama korišćene izolovano jedna od druge ili nisu uopšte korišćene.

27. U čemu se sastoji razlika između SISTEMA ZA OBRADU PODATAKA i KLASIČNIH UPRAVLJAČKIH INFORMACIONIH SISTEMA?

Osnovna karakteristika klasičnih upravljačkih informacionih sistema je grupisanje oko "proizvodnje" informacija neophodnih upravljačkom delu složenih sistema kao osnov za planiranje, sprovođenje planova i kontrolu izvršenja planskih zadataka. U sklopu ovih informacionih sistema izvršena je integracija funkcija i obezbeđeno racionalno korišćenje podataka u svim segmentima organizacione strukture objektnog sistema, za čije potrebe su projektovani, ali kao oslonac u procesu upravljanja koriste produkte "istorijskih događanja" (upravljanje po posledici). Ovaj vid informacionih sistema, iako poseduje niz nedostataka sa aspekta savremene teorije upravljanja, predstavlja u današnje vreme, posebno u našim uslovima, najzastupljeniju formu informacionih sistema

28. U čemu se sastoji razlika između KLASIČNIH UPRAVLJAČKIH INFORMACIONIH SISTEMA i INTERAKTIVNIH ON-LINE UPRAVLJAČKIH INFORMACIONIH SISTEMA?

Interaktivni (on-line) upravljački informacioni sistemi poseduju sve dobre karakteristike klasičnih upravljačkih informacionih sistema ali se, u podršci većini funkcionalnih transformacija, oslanjaju na direktnu interakciju čovek-računar pri čemu se podaci, neophodni za formiranje odgovora na proizvoljan informacioni zahtev, direktno unose, dok se rezultati njihovog procesiranja distribuiraju odmah po obradi i ostaju na raspolaganju ovlašćenim korisnicima u režimu direktnih upita posredstvom radnih stanica računarskog sistema koji služi kao tehnička podrška

29. Navedite osnovne grupe alata za implementaciju inteligentnih informacionih sistema.

ALATI ZA OTKRIVANJE ZNANJA,
ALATI ZA OČUVANJE INTEGRITETA I KONTROLU KVALITETA,
ALATI ZA RUKOVANJE HIPERMEDIJOM (MULTIMEDIJOM),
ALATI ZA PODRŠKU PREZENTACIJI I PRIKAZIVANJU SEMANTIČKIH FORMI ZNANJA,
ALATI ZA PODRŠKU DONOŠENJU ODLUKA I ANALIZI SCENARIJA,
ALATI ZA PODRŠKU FORMATIZACIJI PODATAKA (preformatizacija i sl.),
ALATI ZA PROJEKTOVANJE INTELIGENTNIH SISTEMA.

30. Šta obuhvataju ALATI ZA OTKRIVANJE ZNANJA u sklopu inteligentnih informacionih sistema?

Obuhvataju alate za analizu podataka, mašinsko učenje i statističke analize, a postavljaju potpuno nove granice u domenu semantičkih analiza kroz podršku ekstrakciji ZNANJA često skrivenog iza PODATAKA odnosno INFORMACIJA, kao i skrivenih veza koje mogu postojati između pojedinačnih primeraka strukture podataka nad kojom je izgrađen razmatrani informacioni sistem.

31. Šta obuhvataju ALATI ZA OČUVANJE INTEGRITETA i KONTROLU KVALITETA u sklopu inteligentnih informacionih sistema?

Alati za očuvanje integriteta i kontrolu kvaliteta u sklopu složenih informacionih sistema predstavljaju esencijalne alate budući da se, povećanjem globalno raspoložive količine informacija i uvođenjem savremenih informacionih sistema, organizacioni sistem u potpunosti oslanja na informacionu infrastrukturu koja ga podržava, pa samim tim postaje izuzetno "ranjiv" upravo preko

te infrastrukture.

S obzirom da se niti u jednom realnom sistemu verovatnoća otkaza ne može svesti na nulu kao i što se, u bilo kom realnom sistemu čije je ishodište ČOVEK, ne mogu eliminisati greške (bez obzira na motive i uslove pod kojima one nastaju) jasno je da egzistencija pouzdanih alata koji mogu obezbediti konzistentnost informacionog sistema i kontrolu kvaliteta pripadnika njegove infrastrukture ESENCIJALNA.

32. Šta obuhvataju ALATI ZA PODRŠKU HIPERMEDIJI/MULTIMEDIJI u sklopu inteligentnih informacionih sistema?

Hipermedija/multimedija predstavlja savremeno sretstvo integrisanja više raznorodnih nosilaca informacije (slika, zvuk, tekst, slobodno pojmovno povezane tekstualne celine, asocijativne veze među objektima razmatranja i sl.) i formi njenog predstavljanja, koja omogućava izgradnju individualnih "pogleda" različitih korisnika ili grupa korisnika na isti skup baznih informacija. Ovi alati se, u današnje vreme, ne mogu odvojiti od kategorije (4), alata za prezentaciju i prikazivanje različitih semantičkih formi znanja i/ili informacija, i/ili podataka.

33. Šta obuhvataju ALATI ZA PODRŠKU PREZENTACIJI u sklopu inteligentnih informacionih sistema?

Vidi 32.

34. Navedite osnovne izazove savremenih tehnologija i koncepata u domenu projektovanja informacionih sistema.

DataBase/OLTP (OnLineTransactionProcessing),
Data-Warehouse tehnologije,
Data Mining tehnologije,
Prostorni (Spatial) IS,
WEB i E-Poslovanje,
SOA - Servis Orijentisane Arhitekture.

35. Na čemu se bazira izazov DataBase/OLTP tehnologija u sklopu projektovanja informacionih sistema?

Pod pojmom Baza Podataka (DataBase) obično podrazumevamo skup podataka pridružen nekoj organizaciji.

Za razliku od jednostavnog seta Baza Podataka se obično posmatra preko pridružene STRUKTURE (Šeme Baze).

Za razliku od datoteka Baza Podataka je nezavisna od metode fizičkog skladištenja i aplikacija koje joj pristupaju.

DBMS (DataBaseManagementSystem) je softver koji obezbeđuje upravljanje bazama podataka uz oslonac na DDL, DML, SQL i HOST jezike

36. Na čemu se bazira izazov DataWarehouse tehnologija u sklopu projektovanja informacionih sistema?

n/a

37. Na čemu se bazira izazov Data Mining tehnologija u sklopu projektovanja informacionih sistema?

Data Mining tehnologije - ("rudarenje") često se definišu kao tehnologije za traženje skrivenih informacija u podacima. Razlike u odnosu na tradicionalne DB:

Upiti - ne moraju biti dobro formirani ili precizno iskazani!

Podatci - pristupa se rafiniranim i postprocesiranim podacima

Izlazi - ne predstavljaju podskupove podataka u bazi već posledicu neke analize podataka (OLAP - OnLineAnalyticalProcessing).

38. Na čemu se bazira izazov WEB i e-poslovanja tehnologija u sklopu projektovanja informacionih sistema?

WEB - možemo posmatrati kao najveću globalno dostupnu "bazu podataka" koja ne poseduje "šemu baze".

WEB - pretraživanje (WEB MINING) još uvek predstavlja osnovnu aktivnost na globalnoj mreži.

e-poslovanje (e-business) - predstavlja danas širi pojam od web-pristupa i, u opštem slučaju, obuhvata razvoj i eksploataciju poslovnih informacionih sistema.

39. Na čemu se bazira izazov prostornih i geopozicionih tehnologija u sklopu projektovanja informacionih sistema?

Baziraju se na prostornim podacima (spatial data). Prostorni podaci se mogu posmatrati kao podaci o objektima koji su, zajedno sa tim objektima, prostorno raspoređeni.

Geoinformacioni sistemi,

GPS - sistemi opsteg pozicioniranja,

GIS - u sklopu prostorno raspoređenih tehnoloskih i poslovnih sistema,

GIS - u sklopu informacionih sistema sa radom u realnom vremenu,

Mobilne tehnologije.

40. Navedite dva osnovna modela u sklopu Data Mining tehnologija.

Prediktivni i deskriptivni.

41. Navedite tri osnovne kategorije pretraživanja u sklopu www tehnologije.

pretrazivanje sadržaja,

pretrazivanje strukture,

pretrazivanje podataka o koriscenju.

1. U sklopu JAVA programskog jezika KLASA ima dve fundamentalno različite uloge. Navedite ih.

kao obicna klasa - grupise promenljive i metode koje se u njoj nalaze,

i kao klasa koja sadrzi main metodu koja definise program.

2. Šta se podrazumeva pod blokom (block statement) u sklopu JAVA programskog jezika? Koja je njegova namena?

niz od 0 ili vise naredbi izmedju { i }. telo funkcije je blok.

koristi se da grupise nekoliko naredbi npr u while/for petlji.

3. Pretpostavite da "crna kutija" poseduje interfejs i implementaciju. U čemu se, u tom slučaju, sastoji osnovna razlika između pojmova interfejs i implementacija?

interfejs predstavlja vezu sa ostatkom sveta (imena i parametri metoda), treba da bude jednostavan, dobro definisan i lak za razumevanje. Implementacija se odnosi na unutrašnje funkcionisanje crne kutije. Da bi koristili crnu kutiju potrebno je razumeti interfejs, a ne mora se znati nista od implementaciji.

interfejs definise samo metode koje implementacija mora da implementira, dok implementacija moze da ima svoje sopstvene metode i attribute.

4. Navedite dva osnovna razloga korišćenja imenovanih konstanti (deklaracija final) u sklopu JAVA programskog jezika.

1. ako je final onda je nepromenljiva - konstanta vrednost - program je stiti od promene.
2. ako se pojavljuje na vise mesta u programu, lakse je promeniti vrednost samo na mestu definisanje a ne da se trazi gde se sve pojavljuje u programu pa da se menja.

5. Šta u sklopu JAVA deklaracija predstavlja null i kakva je uloga ove deklaracije?

rezervisanu konstantu koja predstavlja praznu referencu, tj pokazivac na nista.

6. Šta je konstruktor i kakva je njegova uloga u sklopu JAVA klase?

metoda klase bez povratnog tipa, nosi isto ime kao ime klase, implicitno se poziva pri instanciranju objekta klase.

7. Šta osnovna uloga polimorfizma u objektnom programiranju?

odnosi se na cinjenicu da razliciti objekti mogu odgovore na istu metodu na razlicite nacine, u zavisnosti od stvarnog tipa objekta.
omogucava ponovno koriscenje koda - reusability.

8. JAVA programski jezik koristi "garbage collection" za rukovanje memorijom. Šta je njegova uloga?

da oslobadja memoriju na heap-u koji su zauzeli objekti koji se vise ne koriste.

9. Šta je alternativa za korišćenje "garbage collection" i koje posledice po JAVA programe ona može imati?

programer onda mora pratiti koji se objekti jos uvek koriste a koji ne, kako bi ih unistio. posledica je da moze doci do toga da se popuni memorija i da program pukne.

10. GUI programi moraju rukovati "dogadjima". Šta se podrazumeva pod događajima? Navedite bar dva primera događaja u sklopu GUI programa.

događaj je bilo sta sto moze da se javi asinhrono a ne pod kontrolom programa, na sta ce program mozda zeleti da odgovori.

uglavnom je rezultat korisnicke akcije, npr pritisak na dugme misa ili tastature.

11. JAVA poseduje standardnu klasu JPanel. Navedite dva različita načina na koje se ona može koristiti u sklopu GUI aplikacija.

JPanel je komponenta - vidljiv element, prazan pravougaoni region na ekranu, takodje i "container" što znaci da se ostale komponente mogu dodati na njega.

12. JAVA poseduje standardnu klasu MouseEvent. Koja je njena uloga? Šta radi objekat klase MouseEvent?

Kada se dogadjaj desi, sistem smesta informacije o dogadjaju u objekat. Taj objekat se prenosi kao parametar rutini za rukovanje dogadjajima. Razliciti tipovi dogadjaja su predstavljeni preko razlicitih klasa objekata. Objekat tipa MouseEvent predstavlja misa ili dogadjaj pomeranja misa. On sadrzi informacije o lokaciji kursora misa i bilo kog pritisnutog tastera misa.

13. JAVA Swing poseduje JComponent klasu. Šta se, u tom smislu, podrazumeva pod terminom komponenta? Navedite bar tri primera SWING komponenti.

Predstavlja graficki entitet odredjene vrste koji se moze prikazati na ekranu (dugme, meni, toolbar...). U programskom jeziku Java, komponenta je objekat (instanca) bilo koje klase koja je potklasa (naslednik) klase JComponent.

Predstavlja vizuelnu komponentu grafickog korisnickog interfejsa racunara. JComponent nije potpuno nezavisan. Mora se dodati u "kontejner" kao sto su applet ili frame. Primeri su: JButton, JTextField i JPanel.

14. Kakva je uloga LayoutManager-a u JAVI?

Realizuje neke smernice za polaganje svih vizuelnih komponenti koje se dodaju u kontejner, kao sto su JPanel ili content pane JApplet-a, tj. on odredjuje velicine i pozicije komponenti. Razlicite vrste LayoutManager-a imaju razlicita pravila o tome kako komponente treba da budu uredjene. Neke standardne layout manager klase su BorderLayout i GridLayout.

15. Šta, po vašem mišljenju, znači konstatacija da je program robustan?

Robustan program je onaj koji moze da rukuje greskama i drugim neocekivanim uslovima na neki razuman nacin. To znaci da program mora da predvidi moguće greske i odgovori na njih, ako do njih dodje.

16. Šta, u objektnom programiranju, predstavljaju preduslovi (preconditions)? Navedite jedan primer preduslova.

Preduslov je uslov koji mora da bude ispunjen u odredjenom trenutku da bi se nastavilo sa izvršavanjem programa. Na primer, izjava "x=A[i];" ima dva preduslova: da A nije null, i da je $0 \leq i < A.length$. Ako je bilo koji od uslova prekršen, onda ce izvršavanje iskaza generisati gresku. Takodje, preduslov potprograma je uslov koji mora biti ispunjen kada se potprogram zove, kako bi potprogram ispravno radio.

17. JAVA poseduje predefinisanu klasu Throwable. Šta ona predstavlja? Koja je njena uloga?

Throwable je natklasa svih gresaka i izuzetaka u Java jeziku. Samo objekti koji su instanca te klase

(ili jedne od njenih potklasa) mogu biti "baceni" od strane Java Virtual Machine. Slicno tome, samo ova klasa ili jedna od njenih potklasa moze biti argument u catch bloku.

18. Neke grupe izuzetaka (exceptions) zahtevaju obavezno rukovanje (mandatory exception handling). Šta to znači?

Potklase klase Exception koje nisu potklase klase RuntimeException zahtevaju obavezno rukovanje izuzecima. To ima dve posledice:

ako metoda moze da baci takav izuzetak onda moramo da oznacimo tu cinjenicu tako sto cemo dodati throws klauzulu,

ako rutina ukljucuje kod koji moze da generise takav izuzetak onda rutina mora rukovati izuzecima.

19. Neke grupe izuzetaka (exceptions) zahtevaju obavezno rukovanje (mandatory exception handling). Koje su to klase u sklopu JAVA SWING biblioteke?

Potklase klase Exceptions koje nisu potklase klase RuntimeException.

Ti izuzeci se takodje nazivaju "checked" exceptions.

20. Zbog čega metode treba da "bace izuzetak" u slučaju njegovog nastanka? Zašto samo nebi prekinuli dalje izvršenje programa?

Prestanak rada programa je suvise drasticna mera, i ova taktika sigurno ne vodi do robusnog programa. Verovatno je da metoda ne zna sta da radi sa gresko, ali to ne znaci da treba prekinuti ceo program. Kada metoda baci izuzetak, ona je prekinuta, ali program koji je pozvao metodu i dalje ima sansu da uhvati i obradi izuzetak.