

PIS – odgovori za 2. kolokvij

1. Što je model podataka informacijskog sustava i koji se dijagram koristi za prikazivanje modela podataka?

Model podataka je formalni način predstavljanja podataka koji se koriste, pohranjuju ili stvaraju u poslovnom procesu. Najčešće se prikazuje ERD-om (Entity-Relationship dijagramom).

2. Koji su osnovni elementi ERD dijagrama? Opišite ih.

Entiteti – predstavlja fizičke ili logičke pojmove (osobe, mjesta, predmeti, ...)

Atributi – svojstva entiteta

Veze – prikazuju povezanost dvaju entiteta

3. Koja su dva svojstva ERD relacija? Što ta svojstva određuju u relaciji.

Kardinalnost – određuje koliko je instanci entiteta asocirano s koliko instanci entiteta djeteta

Modalnost – određuje može li instanca entiteta djeteta postojati bez instance entiteta roditelja

4. Da li svaki entitet ERD-a mora imati identifikator? Koje smo tipove identifikatora naveli?

Svaki entitet treba imati identifikator kako bi se instance mogle identificirati. Postoje spojni identifikator koji se sastoji od nekoliko atributa i umjetni identifikator.

5. Koja su tri tipa ERD entiteta? Opišite ih.

Zavisni entitet – zahtjeva attribute entiteta roditelja da bi se identificirala i njegova instanca

Nezavisni entitet – može se identificirati bez pomoći drugog entiteta

Presijecajući entitet – postoji u svrhu prikupljanja informacija o relaciji između dva entiteta

6. Što je CRUD matrica, čemu služi, koje podatke sadrži?

CRUD matrica je alat za prikazivanje relacija između procesnog modela i modela podataka i koja opisuje kako procesi koriste podatke unutar sustava. Matrica sadrži procese u prvom retku, a entitete i njihove attribute u lijevom stupcu, a u ćelijama se opisuju interakcije pojedinog procesa s pojedinim entitetom (Create, Read, Update, Delete).

7. Kako se radi validacija modela podataka? Opišite svaki korak u validaciji.

Validacija se obavlja tehnikom normalizacije, kojom se iz postojećih podataka nastoje ukloniti redundantni podaci. Provodi se u tri koraka (normalne forme):

1NF – uklanjaju se višestruki stupci i kreiraju posebne tablice za svaki skup povezanih podataka

2NF – atributi koji ovise samo o dijelu spojenog identifikatora odvajaju se u zasebni entitet

3NF – iz tablice se izdvajaju stupci koji nisu potpuno ovisni o primarnom ključu

8. Koji je prvi korak u fazi dizajna? Koje su tri mogućnosti u prvoj fazi dizajna te koje se prednosti, a koji nedostaci svake od tri mogućnosti?

Prvi korak faze dizajna je odabir strategije izrade sustava između tri moguće strategije. Postoje tri strategije:

Razvoj vlastite aplikacije – prednosti su potpuni nadzor nad aplikacijom, lako povezivanje s drugim vlastitim aplikacijama i razvoj vlastitog kadra, a nedostaci su dugo trajanje i veliki trud za razvoj, nepoznavanje novih tehnologija, teže ograničavanje dodatnih korisničkih zahtjeva i veća cijena i rizik.

Kupovina gotove aplikacije i prilagodba klijentovim zahtjevima – prednosti su dostupnost, pouzdanost, brza isporuka i manja cijena, a nedostatak je što rijetko u potpunosti odgovara zahtjevima.

Razvoj aplikacije izvan kompanije – mogući rizici su gubitak povjerljivih informacija, gubitak kontrole nad razvojem i gubitak mogućnosti učenja.

9. Opišite prve tri normalizacijske forme u modelu podataka?

1NF – uklanjaju se višestruki stupci i kreiraju posebne tablice za svaki skup povezanih podataka

2NF – mora biti zadovoljena 1NF, dok se atributi koji ovise samo o dijelu spojenog identifikatora odvajaju u zasebni entitet

3NF – uvjet je da se zadovoljene 1NF i 2NF te se iz tablice izdvajaju stupci koji nisu potpuno ovisni o primarnom ključu

10. Što se uzima u obzir prilikom odabira strategije dizajna?

Poslovne potrebe – postoji li već gotova aplikacija

„In-house“ iskustvo – postoji li unutar kompanije iskustvo potrebno za razvoj sustava

Znanje – postoji li u kompaniji znanje potrebno za razvoj sustava

Upravljanje projektima – postoji li znanje i iskustvo u upravljanju projektima

Vremenski okvir – koliko brzo sustav treba biti isporučen

11. Što je alternativna matrica pri odabiru strategije dizajna? Kako se matrica razvija (na temelju kojih informacija)?

Alternativna matrica daje kombinaciju nekoliko analiza izvedivosti u jednoj matrici za svaku od mogućih strategija dizajna. Matrica se razvija na temelju kriterija tehničke, financijske i organizacijske izvedivosti

12. Što je arhitektura softverskog sustava (definicija)?

Arhitektura softverskog sustava je struktura koja objedinjuje softverske elemente, vidljiva svojstva tih elemenata i veze među njima.

13. Što su vanjska vidljiva svojstva elemenata u arhitekturi softverskog sustava?

Vanjska vidljiva svojstva podrazumijevaju servise koje elementi nude drugim elementima, performanse elemenata, dijeljenje resursa s drugim elementima i sl.

14. Što su ADL (architecture description languages) jezici i čemu služe? Ako znate neke nabrojite ih.

ADL jezici su jezici koji pružaju formalan način predstavljanja arhitekture te služe za analizu arhitekture kroz analizu performansi, upravljivosti, sigurnosti i sl. Primjeri ADL-a su ACME, Rapide, Wright, Unicon itd.

15. Navedite barem 5 arhitekturnih predložaka koje smo naveli na predavanjima.

Klijent-server arhitektura, višeslojna arhitektura, objektno-orijentirana arhitektura, servisno-orijentirana arhitektura, front-end/back-end arhitektura, monolitna aplikacija, ...

16. Na temelju čega se radi izbor arhitekture softverskog sustava?

Na temelju nefunkcionalnih zahtjeva, cijene infrastrukture, cijene razvoja i skalabilnosti aplikacija.

17. Na koja 4 osnovna tipa se dijele nefunkcionalni zahtjevi?

Operativni zahtjevi, zahtjevi za performanse, sigurnosni zahtjevi, kulturološki/politički zahtjevi.

18. Što su operativni nefunkcionalni zahtjevi softvera i kako ih dijelimo?

To su zahtjevi koji specificiraju okolinu u kojoj sustav treba raditi i kako se ta okolina može mijenjati tijekom vremena. Dijelev se na tehničke operacijske zahtjeve, sistemske integracijske zahtjeve, prijenosne zahtjeve i zahtjeve prema održavanju.

19. Što su zahtjevi za performanse softvera i kako ih dijelimo?

To su zahtjevi koji se odnose na performanse sustava kao što su pouzdanost, kapacitet, vrijeme odziva i sl. Dijelimo ih na zahtjeve za brzinom, zahtjeve za kapacitetom i zahtjeve za dostupnošću i pouzdanošću.

20. Koja su dva tipa zahtjeva prema brzini softvera? Objasnite ih.

Vrijeme odgovora – vrijeme potrebno da sustav odgovori na korisnički zahtjev

Brzina propagacije promjena – odnosi se na to koliko brzo promjene u sustavu propagiraju kroz cijeli sustav

21. Koja su dva tipa zahtjeva prema kapacitetu softvera? Objasnite ih.

Ukupni broj korisnika softvera – bitno za određivanje veličine baza podataka

Maksimalni broj korisnika koji istovremeno mogu koristiti sustav – bitno za određivanje procesorske snage i propusnosti mreže

22. Što su sigurnosni nefunkcionalni zahtjevi softvera i kako ih dijelimo?

Oni se odnose na stupanj zaštite podataka od oštećenja ili gubitka uzrokovanog namjernim postupkom ili slučajnim događajem. Dijelev se na procjenu vrijednosti sustava, zahtjeve za kontrolom pristupa, zahtjeve za enkripcijom i autentikacijom i kontrolom virusa.

23. Što su kulturološki/politički (lokalizacijski) nefunkcionalni zahtjevi softvera i kako ih dijelimo?

To su zahtjevi specifični za pojedine države ili područja u kojima se sustav implementira. Dijelev se na zahtjeve za višezjezičnost, zahtjeve za prilagodbom, eksplicitno postavljanje „podrazumijevanih“ normi i pravne zahtjeve.

24. Što obuhvaća specifikacija softvera i hardvera?

Sve softverske i hardverske komponente (OS, posebne softverske potrebe, obuka, garancija, održavanje, kupnja licenci, serveri, klijenti, periferni uređaji, uređaji za pohranu)

25. Što je korisničko sučelje, a što sistemsko sučelje softverskog sustava?

Korisničko sučelje je dio sustava preko kojeg korisnik komunicira sa sustavom, a sistemsko sučelje je dio sustava preko kojeg sustav komunicira s drugim sustavima.

26. Koja tri osnovna dijela uključuje korisničko sučelje? Ukratko opišite čemu služe ta tri dijela korisničkog sučelja.

Mehanizam navigacije – određuje na koji način korisnik postavlja zahtjeve prema sustavu

Mehanizam ulaza – definira način na koji korisnik unosi informacije u sustav

Mehanizam izlaza – definira način na koji sustav isporučuje informacije korisniku

27. Kojih 6 osnovnih principa se koristi kod dizajna korisničkog sučelja?

Razmještaj, svijest o sadržaju, estetika, iskustvo korisnika, konzistentnost, minimiziranje korisnikovog truda.

28. Što je princip razmještaja kod dizajna korisničkog sučelja? Koji se standardni razmještaj danas najčešće koristi? Nacrtajte primjer horizontalnog i vertikalnog razmještaja.

Razmještaj određuje organiziranje područja određenih funkcionalnosti na ekranu, formi ili izvještaju. Danas se većinom koristi Windows/Macintosh pristup razmještaju.

VERTIKALNI		HORIZONTALNI	
Ime: <input type="text"/>	Ime: <input type="text"/>	Prezime: <input type="text"/>	Email: <input type="text"/>
Prezime: <input type="text"/>			
Email: <input type="text"/>			

29. Što je princip svijesti o sadržaju kod dizajna korisničkog sučelja i na koji način se podiže svijest o sadržaju?

Svijest o sadržaju je svojstvo sučelja da korisnik uz minimalan trud percipira informacije koje sučelje sadrži. Svijest o sadržaju podiže se naslovima svakog ekrana, forme i izvještaja i izbornicima koji pokazuju gdje se korisnik nalazi i kako je tu došao, pažljivo odabranim oznakama polja i područja, točno određenim formatima podataka, ...

30. Koja se estetska pravila koriste za slova kod dizajna korisničkog sučelja?

Sav tekst treba biti u istom tipu slova i slične veličine, preporučena minimalna veličina je 10 piksela, promjena veličine slova trebala bi ukazivati na neku promjenu u informaciji, serif fontovi najvidljiviji su na tiskanim izvještajima, dok su sans serif slova vidljivija na ekranu, izbjegavati korištenje velikih slova...

31. Što je princip iskustva korisnika kod dizajna korisničkog sučelja i koja je razlika između početnika i iskusnih korisnika aplikacije?

Iskustvo korisnika odnosi se na dizajniranje sučelja u skladu s korisnikovim iskustvom. Korisnicima početnicima bitno je koliko je vremena potrebno da se nauče služiti programom, dok su iskusniji korisnici zainteresiraniji za jednostavnost uporabe programa.

32. Što je princip konzistentnosti kod dizajna korisničkog sučelja?

Konzistentnost omogućava korisnicima da predvide što će se dogoditi. Implementira se na različitim razinama sučelja – navigaciji, terminologiji, izvještajima, formama.

33. Što je princip minimiziranja korisnikova truda kod dizajna korisničkog sučelja?

Minimiziranje korisnikova truda odnosi se na minimiziranje truda koji korisnik mora uložiti za obavljanje neke akcije (npr. smanjenje klikova mišem ili pritisaka tipki da bi se došlo do neke funkcionalnosti).

34. Dizajn korisničkog sučelja radi se u pet koraka. Koji su to koraci? Opišite ih ukratko.

Razvoj scenarija korištenja – razvoj općih pregleda koraka koje korisnik napravi da bi izvršio određeni zadatak

Dizajn strukture sučelja – definira osnovnu strukturu sučelja tako što definira sve osnovne komponente ili elemente sučelja i način na koji su te komponente međusobno povezane.

Dizajn standarda sučelja – standardi sučelja su temeljni dizajnerski predlošci elemenata sučelja zajednički pojedinim ekranima, formama i izvještajima unutar aplikacije.

Prototipiranje dizajna sučelja – izrada prototipa, tj. nacрта izgleda ekrana ili simulacija ekrana, formi ili izvještaja sučelja.

Evaluacija sučelja – identifikacija mogućih poboljšanja predloženog dizajna sučelja.

35. Opišite dijagram strukture sučelja (interface structure diagram) koji se koristi za dizajn strukture korisničkog sučelja. Kako se dobiva i što prikazuje?

Dijagram strukture sučelja (ISD) prikazuje sve ekrane, forme i izvještaje, njihovu povezanost te način na koji se korisnik kreće između njih. Svaka komponenta sučelja predstavlja se kvadratom s jedinstvenim identifikatorom i imenom. U ISD kvadratima obično se navodi i oznaka procesa iz DFD-a koji je vezan uz određeni element sučelja.

36. Koje se metode koriste za prototipiranje korisničkog sučelja? Opišite ih.

Storyboard – elementi sučelja se nacrtaju i slažu kako bi se prikazali rezultati korisnikovih aktivnosti.

HTML prototip – sučelje se prikazuje kao HTML stranica, što korisniku omogućuje stvarno testiranje.

Jezični prototip – prototip se izrađuje kao i HTML prototip, ali u onom jeziku u kojem će se razviti i stvarno sučelje.

37. Koje se metode koriste za evaluaciju korisničkog sučelja? Opišite ih.

Heuristička evaluacija – članovi tima uspoređuju razvijeni dizajn s principima dizajna.

Walk-through evaluacija – tim korisniku simulira kretanje kroz sučelje te on identificira moguća poboljšanja.

Interaktivna evaluacija – korisnici sami testiraju sustav.

Formalno testiranje uporabljivosti – provodi se u uvjetima specijalnog laboratorijskog testiranja.

38. Navedite i opišite tipove korisničkih navigacijskih kontrola.

Jezici – korisnik unosi naredbe definirane posebnim naredbenim jezikom

Izbornici – izbornik korisniku pruža listu mogućih komandi koje korisnik može izabrati

Direktna manipulacija – omogućava korisniku direktno pozivanje naredbe na objektu sučelja

39. Što je svrha dizajna ulaza korisničkog sučelja i koji su osnovni principi dizajna ulaza?

Cilj dizajna ulaza je jednostavno uhvatiti točnu informaciju u sustav. Osnovni principi su korištenje online i pozadinskog procesiranja u skladu sa situacijom, prikupljanje podataka što bliže izvoru i minimiziranje korištenja tipkovnice.

40. Što je svrha dizajna izlaza korisničkog sučelja i koje su osnovni principi dizajna izlaza?

Cilj mehanizma izlaza je prezentirati informacije korisniku na način da ih korisnik razumije uz što manje truda. Osnovni principi dizajna izlaza su razumijevanje upotrebe izvještaja, racionalno korištenje informacija i minimiziranje pristranosti.

41. Što određuje dizajn programa?

Dizajn programa određuje koji se programi trebaju napisati, na koji se način pojedini dijelovi kôda povezuju kako bi se dobila integralno funkcionalna aplikacija te na koji način programeri trebaju pisati te programe.

42. Koji su koraci prilikom dizajna programa? Opišite ih ukratko.

Definiranje fizičkog dijagrama toka podataka – izvodi se iz dijagrama toka podataka dobivenih u fazi analize programa.

Definiranje dijagrama strukture – povezivanje svih modula koji se izvode u sustavu.

Izrada programske specifikacije – izrada detaljnih instrukcija za pisanje programa.

43. Koja je razlika između logičkog i fizičkog DFD-a? Koja četiri koraka se provode prilikom prebacivanja logičkog DFD-a u fizički DFD?

Fizički DFD sadrži implementacijske tehničke detalje o izradi sustava, a logički ne. Tranzicija se provodi u četiri koraka:

- Dodavanje implementacijskih referenci
- Postavljanje granice između čovjeka i uređaja
- Dodavanje sistemskih procesa, podataka i tokova
- Ažuriranje podataka u tokovima

44. Što prikazuje dijagram strukture programa (structure chart)? Opišite dijelove dijagrama strukture programa.

Dijagram strukture programa prikazuje sve module koji trebaju biti uključeni u program u hijerarhijskom formatu tijeka izvršavanja programa. Modul se u dijagramu označava pravokutnikom. Linije koje povezuju module označavaju prenošenje kontrole izvršavanja aplikacije između modula. Kontrolni modul je modul više razine i sadrži logiku pozivanja modula niže razine (podređenih modula). Iteracija se označava zakrivljenom strelicom, a selekcija rombom. Modul koji se ponovno koristi (biblioteka) se označava dodatnim linijama na stranama pravokutnika. Konektor omogućava povezivanje više dijagrama strukture. Sprega modula definira povezanost modula.

45. U koja četiri koraka se odvija izgradnja strukturnog dijagrama programa?

Identifikacija modula i razina

Identifikacija selekcija i iteracija

Dodavanje sprege

Provjera strukturnog dijagrama

46. Opišite transakcijsku strukturu strukturnog dijagrama programa.

Transakcijska struktura koristi se kad svaki modul izvodi zasebnu transakciju. Obično ima malo ulaznih i puno izlaznih modula te se obično nalazi na višim razinama dijagrama strukture.

47. Opišite transformacijsku strukturu strukturnog dijagrama programa.

Koristi se kad su moduli povezani i zajedno čine neki proces koji transformira ulaz u izlaz. Obično ima puno ulaznih, a malo izlaznih procesa i obično se nalazi na nižim razinama dijagrama strukture.

48. Što uključuju mjere dobrog dizajna modula strukturnog dijagrama programa? Opišite ih kratko.

Maksimalna kohezija modula – važno je da pojedinačni modul ne uključuje različite funkcionalnosti jer to otežava izmjene tog modula

Minimalna sprega modula – cilj dizajna je postići što veću neovisnost modula

Prikladne razine *fan-in* i *fan-out* – *fan-in* opisuje broj kontrolnih modula koji komuniciraju s podređenim modulom, a *fan-out* broj podređenih modula s kojima komunicira kontrolni modul

49. Što je kohezija programskih modula i koja dva tipa kohezije smo definirali?

Kohezija se odnosi na količinu funkcionalnosti pojedinačnog modula. Slabiji oblik kohezije naziva se logička kohezija, a jači funkcionalna kohezija.

50. Što je implicitna sprega programskih modula i zašto je treba izbjegavati? Koji su tipični primjeri implicitne programske sprege?

To je sprega koja nije očigledna i često uzrokuje greške u softveru i upravo zbog toga je treba izbjegavati. Tipičan primjer su globalni podaci.

51. Što je fan-in i fan-out razina programskih modula strukturnog dijagrama i koje su prikladne razine fan-in i fan-out modula i zašto?

Fan-in opisuje broj kontrolnih modula koji komuniciraju s podređenim modulom i poželjno je da bude što veći, a *fan-out* broj podređenih modula s kojima komunicira kontrolni modul i poželjno je da bude što manji.

52. Što je programska specifikacija i koja četiri dijela (koje podatke) obično sadrži?

Programska specifikacija je pisani dokument koji sadrži eksplicitne instrukcije za pisanje dijelova kôda. Sadrži osnovne informacije o modulu, događaje koji okidaju promatrani modul, ulaze i izlaze modula, pseudokôd.

53. Što uključuje dizajn pohrane podataka informacijskog sustava?

Uključuje definiranje formata pohrane podataka, formata podataka u pohrani i optimizaciju pohrane podataka.

54. Koja su dva osnovna tipa formata pohrane podataka?

Datoteke i baze podataka.

55. Opišite kratko četiri različita tipa baza podataka koje smo definirali na predavanjima.

Hijerarhijske – koriste hijerarhiju (stablastu strukturu) za definiranje odnosa među podacima.

Relacijske – podaci su grupirani po tablicama koje sadrže slogove jednoznačno određene primarnim ključevima.

Objektne – informaciju pohranjuju kao objekte u objektno-orijentiranom programiranju. Objekt ima podatke i procese koji se primjenjuju na tim podacima.

Višedimenzionalne – optimizirane za podatkovna spremišta i aplikacije za online analitičko procesiranje i DSS sustave za podršku odlučivanju.

56. Opišite hijerarhijske baze podataka, svojstva, prednosti i nedostatke.

Hijerarhijske baze koriste strukturu stabla za definiranje relacija među podacima. Podržavaju 1:1 i 1:N relacije, dok se M:N relacije ne mogu prikazati. Prednost im je efikasno pohranjivanje i brzo pretraživanje, a nedostatak složenost aplikacija za rad s njima.

57. Opišite objektno-orijentirane baze podataka, svojstva, prednosti i nedostatke.

Objektne baze informacije pohranjuju kao objekte u objektno-orijentiranom programiranju. Objekt ima podatke i procese koji se primjenjuju na tim podacima. Objekt je definiran klasom i postoji mogućnost nasljeđivanja. Prednost je u upotrebi za multimedijalne aplikacije i sustave s kompleksnim podacima. Nedostatak su nepostojanje standardnog jezika za upite i standardiziranog pristupa objektnim bazama i slične funkcionalnosti.

58. Opišite višedimenzionalne baze podataka.

Višedimenzionalne baze podataka (MDB) usmjerene su olakšavanju dohvata podataka iz baze. MDB je tip baze optimiziran za podatkovna spremišta i aplikacije za online analitičko procesiranje i DSS sustave za podršku odlučivanju. Podaci se korisniku predstavljaju u obliku višedimenzionalnog niza, pri čemu je svaki podatak sadržan u jednoj ćeliji niza kojoj se može pristupiti preko višestrukih indeksa.

59. Od kojih pet koraka se sastoji postupak prelaska sa logičkog na fizički model podataka?

Pretvaranje entiteta u tablice ili datoteke

Pretvaranje atributa entiteta u polja

Dodavanje primarnih ključeva

Dodavanje sekundarnih ključeva

Dodavanje sistemskih podataka

60. Koje se tehnike koriste za optimizaciju brzine pristupa bazi podataka? Opišite ih.

Denormalizacija – namjerno se unose redundancije u podatke koje ubrzavaju dohvat podataka

Uskupljavanje – reduciranje broja pristupa bazi fizičkim postavljanjem sličnih zapisa jednih blizu drugih

Indeksiranje – dodatni podaci koji se spremaju za promatranu tablicu, ali višestruko ubrzavaju rad s tablicom