

INFORMATIKA

1. Kaj je *informatika*? Kaj zajema?

Informatika je znanstvena disciplina, ki raziskuje zgradbo, funkcije, zasnovu, organiziranje in delovanje informacijskih sistemov.

INFORMATIKA = INFORMACIJA + AVTOMATIKA

- Nekaj področij, bistvenih za prihodnje obdobje informacijske družbe:
 - Delo in učenje na daljavo
 - Telematske storitve za srednja in mala podjetja
 - Upravljanje cestnega prometa
 - Nadzor zračnega prometa
 - Omrežja zdravstvenega varstva
 - Evropsko omrežje javne uprave

2. Katere so glavne znanstvene discipline, ki se ukvarjajo z informatiko? Kako jih delimo?

Tehnični pristop: Računalništvo, Upravljanje, Operacijske raziskave

Psihološki pristop: Sociologija, Politologija, Psihologija

Prve tri veje se osredotočajo na tehnični vidik IS (tehnični pristop), spodnje tri pa IS proučujejo bolj iz vidika njihovega vpliva na družbo in posameznika (vedenjski pristop). Tehnični pristop proučuje IS na osnovi matematičnih modelov, prav tako pa proučuje fizično tehnologijo, ki omogoča njihovo delovanje. **Računalništvo** se ukvarja predvsem s principi programiranja in reševanja problemov z računalnikom, s teorijo izračunljivosti, podatkovnimi strukturami ter načini učinkovitega shranjevanja in dostopa do podatkov, **upravljanje** proučuje razvoj modelov za podporo odločanju in učinkovitemu upravljanju, **operacijske raziskave** pa se usmerjajo v matematične metode za optimizacijo parametrov, vezanih na *transport, skladiščenje* in druge procese organizacij.

Močno napredujejo tudi vede, ki se ukvarjajo s proučevanjem problemov in zadev, povezanih z vplivom IS na posameznika, organizacijo ali celotno družbo. Tehnični pristop se pri tovrstnem proučevanju ne izkaže, saj navadno temelji na formalnih metodah, ki se za proučevanje socioloških, psiholoških in političnih vplivov ne izkažejo primerne. Med tovrstne vede sodijo **psihologija, sociologija** ter **politologija**.

PODATEK IN INFORMACIJA

3. Kaj je *podatek*?

- Podatek je predstavitev informacije na *formaliziran način*, ki je primeren za komunikacijo, interpretacijo ali obdelavo (s strani človeka ali stroja).
- Podatek je poljubna predstavitev s pomočjo simbolov ali analognih veličin, ki ji je pripisan, ali se ji lahko pripiše nek pomen.

4. Kaj je *informacija*?

- Informacija je znanje, ki se nanaša na objekte, kot so dejstva, dogodki, stvari, procesi ali ideje, vključno s koncepti, ki imajo v okviru nekega konteksta določen pomen (ISO).

5. Kakšna je razlika oziroma povezava med podatkom in informacijo?

- Podatki niso informacija
- Podatki ne vsebujejo informacije
- Podatki posredujejo informacijo prejemniku, katerega znanje je konsistentno z izbrano predstavitvijo podatkov in modelom sveta, na katerega se nanašajo.

6. Kdaj podatki ne posredujejo nobene informacije?

- Če je količina podatkov tako velika, da se jih v času, ki je na voljo za ukrepanje na njihovi osnovi, ne da interpretirati, se lahko zgodi, da s podatki ni posredovana nobena informacija.

7. Kako informacijo opredelimo v informacijski teoriji?

V informacijski teoriji je informacija opredeljena kot:

- znanje, ki zmanjša negotovost, povezano s pojavom določenega dogodka iz končne množice možnih dogodkov

8. Kako lahko merimo količino informacije, ki jo posredujejo izbrani podatki?

$$I = -\log_2 p(x) \text{ [bit]}$$

Podatek je v zgornji definiciji sporočilo, da se je zgodil dogodek x_i , znanje je prepoznavanje verjetnosti nastopa posameznih dogodkov $p(x_i)$, $i = 1..n$.

INFORMACIJSKA DRUŽBA

9. Kakšne so značilnosti današnje družbe?

- Izpostavljeni smo neprestanim znanstvenim in tehnično-tehnološkim odkritjem.
- Človeško znanje se v dveh do treh letih podvoji.
- Potrebna je dinamičnost – sposobnost prilagajanja ter spremljanja in izkoriščanja potencialov, ki so na voljo.
- Informacije kot potencial presegajo vse ostale naravne potenciale (surovine, energija).
- Gospodarstvo vseh razvitih držav temelji na učinkovitem zbiranju, hranjenju, obdelavi in posredovanju podatkov.
- Razvoj informatike je nujen.
- Informacijske in telekomunikacijske tehnologije so ključnega pomena.

10. Kako informatika vpliva na gospodarski in družbeni razvoj?

Nekaj področij, bistvenih za prihodnje obdobje informacijske družbe:

- Delo in učenje na daljavo
- *Telematske* storitve za srednja in mala podjetja
- Upravljanje cestnega prometa
- Nadzor zračnega prometa
- Omrežja zdravstvenega varstva
- Evropsko omrežje javne uprave

11. Kakšne so značilne faze gospodarskega in družbenega razvoja in kakšni so bili oziroma so poglaviti viri, značilni za posamezne faze?

- Agrarna družba (kmetijstvo) do leta 1906 – sami kmetje
- Industrijska družba (industrija) do 1954 – 40% zaposlenih v ind.
- Informacijska družba (storitve) - *Večinoma storitve vezane na Zbiranje, hranjenje, obdelavo in distribucijo podatkov. Informacijske storitve, izobraževanje,...*

12. Kako opredelimo informacijsko družbo?

Informacijska družba je sinonim za novo nastajajočo družbo, ki ne temelji zgolj na izkoriščanju surovin in energije, temveč kot najpomembnejši vir jemlje informacije in znanje.

13. Kaj so informacijske avtoceste?

Informacijske avtoceste (*Information Highways*) bodo omogočale prenos informacij (podatkov) v obliki teksta, slik, govora itd. Povzročile (omogočile) bodo popolno reorganizacijo številnih obstoječih dejavnosti ter razvoj novih.

SPLOŠNO O SISTEMIH

14. Kaj je sistem?

- Sistem je vse, kar izvira iz narave ali pa je ustvaril človek, in deluje kot urejena celota, ki se ravna po določenih zakonitostih.
- Sistem je skupina medsebojno povezanih elementov za doseganje določenega cilja oziroma za opravljanje določene funkcije.
- Sistem je skupina objektov, ki sodelujejo po pravilih medsebojne interakcije ali komunikacije.

15. Narišite shemo splošnega sistema.

<Slide 20>

16. Kakšne so značilnosti splošnega sistema?

- Vsaka komponenta je za sistem pomembna – obstoj in funkcija komponente vplivata na obstoj in funkcijo celotnega sistema.
- Nobena komponenta ni *izolirana*.
- Sistem s svojo funkcijo vpliva na funkcijo komponente.

17. Kaj so komponente splošnega sistema?

- OKOLJE sistema
- Okolje sistema je množica komponent, ki so v interakciji s sistemom, vendar niso del sistema.
- VHOD in IZHOD sistema
- Sistem deluje z določenim namenom oz. za doseg določenega cilja. K cilju strmi z izvajanjem procesa, s pomočjo katerega pretvarja *vhod* v *izhod*.

18. Kaj nas pri obravnavi sistemov zanima?

- **Namen sistema:** namen sistema opredeljuje razlog za obstoj sistema. Namen sistema za *vpis kandidatov na visokošolski študij* je na primer podpora Prijavnemu postopku, ki skuša v čim večji meri ugoditi kandidatovim željam in sicer na osnovi njihovih prijav ter točk, ki jih zberejo glede na razpisne pogoje. Namen informacijskega sistema je zagotavljanje podatkov oziroma informacij, ki so potrebne za delovanje organizacije.
- **Meje sistema:** meje sistema določajo, kaj je znotraj in kaj zunaj sistema. Na primer, *izvedba mature* ne spada v sistem za *vpis kandidatov na visokošolski študij*, ampak gre za samostojen sistem. Sistema bi seveda lahko delovala združeno, njune skupne meje pa bi tedaj bile drugačne.
- **Okolje sistema:** okolje sistema je vse, kar je izven meja sistema in s sistemom sodeluje, oziroma je za sistem pomembno.
- **Vhod in izhod sistema:** vhod in izhod sistema tvorijo fizični objekti (material, blago ipd) in podatki, ki bodisi prihajajo v sistem (vhod) ali iz sistema izhajajo v okolje (izhod).

POSLOVNI SISTEM IN POSLOVNI PROCESI

19. Kako opredelimo *poslovni sistem*?

- Poslovni sistem opredelimo kot: *mikroekonomski* sistem (ali asociacijo mikroekonomskih sistemov), katerega temeljni proces je *reprodukcijski proces*, temeljni cilj pa *ekonomski*. Je celota medsebojno smotno povezanih komponent, ki omogočajo poslovni proces z določenimi cilji in ki pri njem v večji ali manjši meri vplivajo druga na drugo. Pri tem obstajajo tudi medsebojni vplivi med poslovnim sistemom kot zaokroženo celoto in njenim okoljem.

20. Iz kakšnih podsistemov je sestavljen poslovni sistem?

Poslovni sistem je razčlenjen na **izvajalni, organizacijski in informacijski podsistem**, od katerih vsak zase spet nastopa kot delni sistem.

21. Kako opredelimo odnos med poslovnim in informacijskim sistemom?

Informacijski sistem je del poslovnega sistema. Poslovni sistem ima lahko več informacijskih sistemov (enega ali več za en podsistem), a naceloma ima en informacijski sistem, ki zagotavlja informacije za celotno organizacijo.

22. Kaj je *poslovni proces*?

- povezana skupina korakov oziroma aktivnosti, ki se izvajajo v poslovnem sistemu in posredno ali neposredno vplivajo na dodano vrednost pri uresničevanju skupnega cilja poslovnega sistema.
- Izvajanje poslovnega procesa vključuje ljudi, informacijske vire in vsa druga sredstva, potrebna za izvedbo posameznih aktivnosti. Aktivnosti so časovno in prostorsko povezane, imajo začetek in konec ter vhodne in izhodne elemente. Izhodi poslovnega procesa lahko pomenijo dodano vrednost za notranjo ali zunanjo stranko.

23. Kaj je *aktivnost poslovnega procesa*?

Aktivnost je majhna naloga, korak ali operacija znotraj procesa in je navadno najmanjša enota, ki jo določimo pri obravnavi poslovnega procesa.

24. Kdaj govorimo o *podprocesih*?

Poslovni procesi so lahko manj ali bolj obsežni. V slednjem primeru jih delimo na podprocese, ki so pravzaprav sami zase procesi. Tako bi lahko, na primer, proces *Prodaja* razdelili na *Sprejem in obdelavo naročila, Izdajo naloga v proizvodnjo, Pripravo dobavnic in izdajnih nalogov* ipd. Na določeni ravni podrobnosti ne govorimo več o poslovnem procesu, temveč o aktivnostih.

25. Kaj je *obseg poslovnega procesa*?

Če je **obseg procesa** velik (proces je sestavljen iz velikega števila podprocesov in aktivnosti), potem je težje razumljiv in obvladljiv, in ga je bolje razbiti na podprocese, ki se obravnavajo posebej. Premajhen obseg pa lahko pomeni majhno dodano vrednost.

26. Kako opredelimo *dodano vrednost poslovnega procesa*?

Dodana vrednost procesa je tista pridobitev, ki jo rezultat procesa predstavlja, bodisi za notranjo ali zunanjo stranko.

27. Kaj je namen *prenove poslovnih procesov*?

Vodstvo s tem želi poskrbeti, da je vsaka aktivnost vsakega poslovnega procesa dejansko potrebna ter optimalna iz vidika časa in sredstev, potrebnih za njegovo izvedbo.

28. Kaj so ključni in kaj podporni poslovni procesi?

Procesom, ki neposredno dodajajo vrednost napram stranki pravimo **ključni poslovni procesi** ali kar **ključni procesi**. Njihova značilnost je, da vplivajo na konkurenčnost organizacije, da potekajo čez več funkcij ter da na vhodu in izhodu komunicirajo z zunanjim subjektom, kot je na primer stranka. Druga vrsta poslovnih procesov so **Podporni poslovni procesi** ali **Podporni procesi**. Ti za razliko od ključnih procesov neposredno ne vplivajo na dodano vrednost pri uresničevanju skupnih ciljev poslovnega sistema (med njimi je tudi zadovoljitev stranke) temveč predstavljajo le podporo za izvajanje ključnih procesov. Njihova vloga v sklopu poslovnega sistema je podpora poslovanju, njihov vhod in izhod pa sta tipično subjekta znotraj poslovnega sistema.

29. Kaj je vrednostna veriga?

V povezavi s poslovnimi procesi, ki v poslovnem sistemu skrbijo za zadovoljevanje strank, pogosto nastopa tako imenovana **Vrednostna veriga**. Vrednostna veriga zajema vse procese, ki neposredno ali posredno zadovoljujejo stranke.

30. Kakšen je odnos med funkcionalnimi področji in poslovnimi procesi? Kako lahko poteka poslovni proces?

Poslovni sistemi se tradicionalno organizirajo okrog funkcionalnih področij, ki sama zase predstavljajo majhna podjetja. Takšna področja so, na primer:

- Prodaja
- Nabava
- Računovodstvo in knjigovodstvo

Procesi, ki prečkajo funkcionalna področja, procesi, povezani s posameznim funkcionalnim področjem, Aktivnosti in podprocesi, ki se tipično izvajajo na vseh funkcionalnih področjih.

31. Kako razgradimo funkcionalno področje?

- Funkcije organizacijskega sistema se sestavljajo v *funkcionalna področja*. Funkcionalno področje sestavlja notranje povezana skupina *funkcij*, ki zajema glavna področja dejavnosti nekega organizacijskega sistema.
- Razčlenitev funkcij nas na določenem nivoju pripelje na nivo *elementarnih funkcij*, kjer že ugotavljamo, *kaj* je rezultat elementarnih funkcij. Vsaka elementarna funkcija ima točno določen vhod in točno določen izhod. Temeljna razlika med funkcijo in elementarno funkcijo je tudi v stalnosti izvajanja. Pri funkciji ni mogoče opredeliti začetka in konca izvajanja in tudi ne določiti posameznih primerov izvedbe te funkcije, pri elementarni funkciji pa je to mogoče.
- Elementarne funkcije so na najnižjem nivoju sestavljene iz zaporedja *aktivnosti ali korakov*

32. Opredelite pojme poslovno okolje, podjetje, poslovni sistem, informacijski sistem, informacijska tehnologija. Kakšen je njihov odnos!

V poslovnem okolju je podjetje, v njem poslovni sistem, sledi informacijski sistem in na koncu informacijska tehnologija.

33. Zakaj govorimo o večanju preseka med poslovnim in informacijskim sistemom?

Distinkcija med poslovnim in informacijskim sistemom je nekdaj bila večja kot danes. Delo se izvaja v poslovnem sistemu, podatki o tem pa se zajemajo ter hranijo v informacijskem sistemu. To ponekod še danes velja – zamislimo si na primer podjetje, ki goji več hektarjev trt. Osnovna naloga poslovnega sistema je skrb za trte ter ob trgatvi pobiranje grozdja. IS, če sploh obstaja, pa pomaga pri določitvi primernega termina trgatve in podobno. Spet drugje je meja med PS in IS bistveno manj izrazita. Takšen primer je recimo Izbirni postopek za sprejem kandidatov na visokošolski študij univerz ter samostojnih visokošolskih zavodov v Sloveniji. Zaradi velike intenzivnosti zbiranja ter obdelave raznih podatkov je poslovni oziroma delovni sistem v veliki meri kar informacijski sistem.

Prekrivanju med poslovnim in informacijskim sistemom botrujejo predvsem številne novosti ter možnosti, ki jih ponuja informacijska tehnologija. Če primerjamo IS za podporo proizvodnji včasih in danes to lepo vidimo.

SHEMA WCA (Work-Centered Analysis)

34. Kakšen je namen *sheme WCA*?

- WCA – Work Centered Analysis framework
- WCA predstavlja splošno shemo, ki daje začetni okvir za proučevanje poslovnih ter informacijskih sistemov.
- WCA je dobila ime po tem, ker poudarja potrebo poslovnih uporabnikov po dobrem razumevanju poslovnega sistema, da bi se lahko odločali o potrebi po gradnji, izboljšavah, ali prenovitvi informacijskih sistemov.
- WCA združuje ideje različnih disciplin, na primer:
 - Upravljanje kakovosti
 - Prenovitev poslovnih procesov
 - Teorija sistemov ipd.

35. Kateri so osnovni gradniki WCA?

- Notranje in zunanje stranke (uporabniki poslovnega sistema)
- Izdelke (proizvode, produkte) poslovnega sistema
- Aktivnosti (korake) poslovnega sistema
- Udeležence poslovnega sistema
- Podatke (informacije), ki jih poslovni sistem kreira ali uporablja
- Tehnologijo, ki jo poslovni sistem uporablja

36. Narišite shemo WCA ter opredelite njene komponente.

Slide 47.

37. Kaj so *stranke* po WCA shemi? Kaj so *zunanje* in kaj *notranje stranke*?

Stranke poslovnega sistema, to je uporabnike rezultatov (izdelki in/ali storitve), ki jih sistem proizvaja (nudi), delimo na notranje in zunanje. **Notranje stranke** so uporabniki, ki so zaposleni znotraj poslovnega sistema, in uporabljajo rezultate sistema kot delne izdelke, katerim dodajo novo vrednost, preden le-ti pridobijo končno obliko, ustrezno za zunanje stranke.

Zunanje stranke so ljudje, ki niso zaposleni znotraj sistema, temveč so zgolj uporabniki končnih rezultatov poslovnega sistema.

38. Kaj je *izdelek* po WCA shemi? V kakšni obliki lahko izdelek nastopa? Kakšne so značilnosti izdelka?

- Izdelek je rezultat oziroma izhod poslovnega sistema.
- Izdelek je lahko:
 - fizičen objekt
 - storitev
 - podatek
- Karakteristike izdelka:
 - Cena
 - Odzivnost
 - Kakovost
 - Zanesljivost
 - Dostopnost

- o Ustreznost standardom

39. Kdo so *udeleženci* v poslovnem procesu?

- Udeleženci v poslovnem sistemu so posamezniki, ki opravljajo svoje vloge v sklopu aktivnosti posameznih delovnih procesov.
- Še tako avtomatizirani sistemi vključujejo ljudi, ki morajo biti kdaj pa kdaj prisotni.
- Vloge udeležencev so različne. Sodelujejo tako v izvedbenem, poslovnem kot tudi v informacijskem sistemu.

40. Kakšen je pomen znanja udeležencev v poslovne procesu oziroma?

- Obravnava udeleženca v poslovnem sistemu je pomembna tako iz psihološkega in sociološkega kot tudi iz tehničnega vidika.
- Poslovni sistemi so odvisni od znanja, ki ga imajo bodisi posamezniki ali organizacija kot celota.
- Analiza poslovnega sistema razkrije tudi informacije, ki niso nikjer zapisane. So v glavah posameznikov (Knowledge management – upravljanje z znanjem)

41. S čim se ukvarja *upravljanje z znanjem*?

- Znanje je strateška pridobitev vsake organizacije oziroma njeno premoženje.
- Uspešnost organizacije postaja vse bolj odvisna od njene sposobnosti upravljanja z znanjem.
- Področje, ki se ukvarja z upravljanjem znanja, lahko delimo po pomenu obravnave:
 - o Upravljanjem z znanjem kot objektom, ki ga lahko identificiramo in z njim upravljamo v sklopu računalniško podprtih sistemov (računalniška in informacijska znanost).
 - o Upravljanjem z znanjem kot sklopom izkušenj, sposobnosti in *know-how*-a posameznikov ali organizacije, ki je dinamično in se konstantno spreminja (filozofija, sociologija in psihologija).

42. Kaj predstavljajo *podatki* po WCA shemi? Kakšne oblike lahko zavzamejo? Kje nastajajo?

Podatki, s katerimi imamo opravka v poslovnem sistemu, lahko zavzamejo številne oblike: tekst, številke, slike, zvok, video zapis itd. Podatki lahko prihajajo v sistem iz okolja oziroma iz drugih poslovnih sistemov, lahko pa znotraj sistema nastajajo.

43. Kakšen je odnos med *podatkom*, *informacijo* in *znanjem*?

Slide 55.

44. Kakšne vrste znanja ločimo?

- Znanje lahko delimo na:
 - o tacitno ali skrito znanje in
 - o eksplicitno znanje.
- Eksplicitno znanje je formalizirano znanje, ki ga je moč razmeroma enostavno izraziti, običajno v obliki principov, postopkov, dejstev, likov, pravil, formul itd. Sčasoma postane rutinsko in prevzame značaj podatkov
-
- Tacitnega znanja ni enostavno izraziti niti videti. Je precej subjektivno in prepleteno z vedenjem in časom. Obsega izkušnje, ideale, čustva, intuicijo in notranji vpogled. Deli se na *tehnično znanje* (*know how*) in *zaznavno ali kognitivno znanje*.

45. Kaj predstavlja *tehnologija* v WCA shemi?

- Tehnologija so orodja, ki bodisi neposredno izvajajo določene aktivnosti znotraj poslovnega sistema ali pa so v pomoč udeležencem pri izvajanju njihovega dela.
- Posebna vrsta tehnologije, ki je pri obravnavi informacijskih sistemov najpomembnejša, je informacijska tehnologija.
- Informacijska tehnologija označuje programsko (*software*) in strojno (*hardware*) opremo, ki se uporablja za podporo delovanju informacijskega sistema.

INFORMACIJSKI SISTEM, VRSTE INFORMACIJSKIH SISTEMOV

46. Kako opredelimo *informacijski sistem*?

- Informacijski sistem lahko opredelimo kot množico medsebojno odvisnih komponent (strojna oprema, programska oprema, ljudje), ki zbirajo, procesirajo, hranijo in porazdeljujejo podatke in s tem podpirajo delavne procese v organizaciji.
- Formalni/neformalni informacijski sistemi
- Računalniško podprti informacijski sistemi

47. Kakšna je razlika med *formalnim* in *neformalnim* informacijskim sistemom?

Medtem ko ima formalni informacijski sistem jasno določene podatke, s katerimi operira, določene postopke za njihovo obdelavo ter jasno definirana pravila, je neformalni informacijski sistem odvisen od implicitnih dogovorov in nedefiniranih pravil. V neformalnih informacijskih sistemih ni predpisov o tem, kaj so podatki in kaj mehanizmi za njihovo obdelavo.

48. Kakšne vrste informacijskih sistemov poznate?

- Transakcijski informacijski sistemi (*Transaction Processing System - TPS*)
- Upravljalno-ravnateljevalni informacijski sistemi (*Management Information System - MIS*)
- Odločitveni sistemi (*Decision Support System - DSS*)
- –Ekspertni sistemi (*Expert System – ES*)
- Sistemi za upravljanje delovnih procesov (*Workflow System - WS*)

49. Kaj veste o *transakcijskih* informacijskih sistemih?

Transakcijski IS (TPS)

- transakcija = enostaven diskreten dogodek v sklopu delovanja organizacije
- avtomatizirano delo s podatki o poslovnih dogodkih (transakcijah)
- funkcije
 - zajem podatkov o transakcijah
 - verificiranje transakcij
 - hranjenje podatkov o transakcijah
 - obdelava podatkov o transakcijah

50. Kaj veste o *ekspertnih* sistemih?

Ekspertni sistem (ES)

- je sistem, ki se v določenih situacijah obnaša kot ekspert, oz. izurjena oseba.
- najbolj zahteven del je gradnja baze
- lupino lahko uporabljamo večkrat z različnimi bazami, se pravi jo programiramo samo enkrat

51. Kaj so *odločitveni* in kaj *poslovodni* informacijski sistemi?

Odločitveni IS (DSS)

- nudi organizacijam pomoč pri sprejemanju odločitev
- bistveno bolj aktiven od TPS

- sestavni deli:
 - podatkovna baza (zajema podatke iz TPS in MIS)
 - grafični ali matematični del poslovnih aktivnosti)

52. Kaj so *direktorski (upravljalno - ravnateljevalni) informacijski sistemi*?

Upravljalno-ravnateljevalni IS (MIS)

- zajema podatke iz TPS in jih preoblikuje v oblike, ki nudijo informacije vodstvenim delavcem za njihovo delo
- poslovni sistem

53. Kaj so *sistemi za upravljanje poslovnih procesov*?

Sistemi za upravljanje delovnih procesov

(= WS - Workflow Systems) so sistemi, ki podpirajo:

- analizo (analiziramo kakšni poslovni procesi so v podjetju)
- modeliranje (prikažemo poslovne procese)
- izvajanje (ko so modeli sprejeti, jih z določenimi orodji izvajamo)
- nadzor nad izvajanjem (omejitve pri izvajanju)

ŽIVLJENSKI CIKLI RAZVOJA

54. Naštejte življenjske cikle razvoja IS, ki jih poznate.

- Zaporedni ali slapovni model (waterfall model)
- Iterativni model
- Prototipni model
- RAD – Rapid Application Development
- Inkrementalni model
- Kombinirani razvojni model

55. Opišite zaporedni življenjski cikel razvoja programske opreme. Kakšne so njegove največje slabosti?

- Najstarejši razvojni model, značilen za prve oblike strukturnega pristopa
- Faze si sledijo zaporedno (analiza, nacrtovanje, izvedba, upeljava)
- Vračanje nazaj ni mogoče
- **SLABOSTI** zaporednega modela
 - Zahteve nikoli niso statične in se spreminjajo.
 - Posamezne faze ne moremo preprosto zaključiti, potrebno je vračanje nazaj.
 - Zaporedni model ne dopušča vračanja nazaj à razvit sistem lahko ne ustreza dejanskim zahtevam
- –Tveganje, da sistem ne ustreza zahtevam je visoko vse do zadnje faze razvoja

56. Opišite iterativni življenjski cikel razvoja programske opreme. Primerjajte ga z zaporednim ciklom.

- Razvit kot odziv na pomanjkljivosti slapovnega pristopa.
- Pri iterativnem pristopu izvajamo korake slapovnega pristopa v več iteracijah.
- V vsaki iteraciji razvijemo določen del funkcionalnosti celotnega sistema.
- V začetnih iteracijah razvijemo najbolj tvegane dele sistema.
- Sistem se razvija inkrementalno.
- **Prednosti** iterativnega razvoja (proti zaporednemu):
 - Najbolj tvegani deli so razrešeni še preden postane investicija velika
 - Začetne iteracije omogočijo zgodnje povratne informacije s strani uporabnikov
 - Preizkušanje in povezovanje v sistem sta nepretrgana
 - Ciljni mejniki omogočajo kratkoročno osredotočenje
 - Napredek merimo z ocenjevanjem izvedenega dela
 - Možna je predaja izvedenega dela projekta še preden je dokončan celoten projekt

57. Kaj veste o prototipiranju? Kdaj v fazi razvoja IS lahko nastopa? Kaj je prototipni pristop?

S pojavitvijo iterativnih modelov so se prvič pojavili prototipi, ki so iterativnim modelom prinesli dodatne prednosti. Prototipi se danes uporabljajo v več ali manj vseh razvojnih modelih, na njihovi osnovi pa se je razvil tudi poseben prototipni razvojni model.

Prototipni model temelji na izdelavi prototipov, s katerimi označujemo predhodno izdelane in navadno nepopolne verzije sistema. Uporaba prototipov v veliki meri olajša komuniciranje z uporabnikom, zato jih uporabljamo v različnih fazah razvoja. Za izdelavo prototipov so bila razvita posebna razvojna okolja, ki omogočajo vizualno sestavljanje zaslonских mask, izpisov in poizvedb ter vsebujejo mehanizme za avtomatsko generiranje kode. Izdelava prototipov kot tehnika razvoja aplikacij temelji na tesnem sodelovanju med uporabnikom in analitikom. Ko se uskladi glede zahtev sistema, analitik razvije prototip in ga da uporabniku v testiranje oziroma v uporabo. Ta analitiku sporoči, kaj mu je pri prototipu všeč in kaj ne. Povratna informacija služi analitiku za pripravo nove verzije prototipa. Proces traja, dokler uporabnik ni s prototipom zadovoljen.

Prototipi se navadno uporabljajo le kot del specifikacije sistema, za pridobitev jasnejše podobe bodočega sistema in se v nadaljevanju zavržejo. Vendar pa obstajajo tudi metode, ki tako izdelane prototipe izkoristijo kot osnovo za izdelavo produkcijskega sistema (*Rapid Application Development – RAD*).

METODOLOGIJE RAZVOJA IS IN RAZVOJNI SISTEMI

58. Kakšna je razlika med *metodo*, *tehniko* in *metodologijo*?

Metodologija razvoja IS opisuje postopke pri razvoju IS ter definira metode in tehnike, ki se v postopkih uporabljajo. Vsak postopek je sestavljen iz več aktivnosti, katerih izvedba največkrat pomeni izdelavo enega ali več izdelkov. Za izvedbo aktivnosti in s tem izdelavo izdelkov so zadolženi posamezniki ali skupine ljudi, ki sodelujejo pri razvoju. Pri izvajanju aktivnosti uporabljajo predpisane metode in tehnike, ki so sestavni del jezika metodologije.

59. Kaj opredeljuje *metodologija* razvoja IS? Narišite metamodel.

Metodologija razvoja IS navadno sledijo izbranemu pristopu in natanko predpisujejo korake, postopke, tehnike, izdelke in orodja za njihovo izdelavo v posameznih korakih razvoja IS.

60. Naštejte nekaj metodologij razvoja IS.

- *IE - Information Engineering* (Strukturni pristop/ James Martin/ 1981)
- *CASE *Method* (Strukturni pristop/ Richard Barker, Oracle)
- *SSADM – Structured System Analysis and Design Method* (Strukturni pristop/ CCTA-Central Computing and Telecommunications Agency (1981) razvije metodologije za vladne organizacije/ CCTA predlaga SSADM kot standard)
- *OMT – Object Method Technique* (Objektni pristop/ Jim Rumbaugh)
- *RUP – Rational Unified Process* (Objektni pristop/ Rational)

61. Kaj zaznamuje sodobne metodologije razvoja IS?

- Z naglim naraščanjem procesne moči vseh vrst računalnikov,
- Z integracijo poslovnih procesov ter poslovnih IS,
- S porazdeljenim procesiranjem in razvojem računalniških mrež,
- Z bogato ponudbo standardnih aplikativnih rešitev,
- Z razvojem in naglo uveljavljanje računalniških orodij za razvoj in projektiranje IS.

62. Na kaj mora biti pozorna vsaka moderna metodologija razvoja IS?

- Zajemati mora celoten življenjski cikel IS in pripadajoče programske opreme, in ne zgolj posameznih faz.
- Omogočati mora sistematičen prehod v naslednjo fazo.
- Omogočati mora preverjanje pravilnosti procesa skozi vse faze življenjskega cikla IS.
- Podpirati mora skupinsko delo na projektu razvoja IS ter omogočati uporabo sodobnih metod organizacije in vodenja projektov.
- Biti mora uporabna za čim širši spekter računalniških projektov.
- Biti mora dovolj enostavna za priučitev.
- Omogoča mora uporabo čim širšega spektra avtomatiziranih orodij za povečanje produktivnosti posameznikov in celotne skupine.
- Omogočati mora dokumentiranje in spremljanje razvoja IS skozi vso njegovo življenjsko dobo.

63. Kakšna je razlika med *lower* in *upper* CASE orodji?

- *Upper CASE*: orodja, ki podpirajo aktivnosti prvih faz razvoja IS: poslovno modeliranje, vzpostavitev okvirjev projekta, zajem informacij, konceptualno modeliranje, analiza in načrtovanje IS.

- *Lower CASE*: orodja, ki so specializirana za podporo izvedbeni IS ter njegovemu vzdrževanju: generiranje programske kode, podatkovne baze, baznih sprožilcev in baznih procedur.

64. Naštejte nekaj primerov CASE orodij.

Oracle Designer, Power Designer, Delphi, Visual Basic

65. Kaj je SUPB? Kakšne so njegove osnovne značilnosti?

- Sistemi SUPB (Sistemi za upravljanje s podatkovnimi bazami) omogočajo delo s podatki (hranjenje, iskanje, obdelava).
- Poleg omenjenega vsebujejo tudi potrebne funkcije za njihovo vzdrževanje.
- SQL.

66. Naštejte nekaj primerov SUPB sistemov.

Oracle, MS SQL Server, Ingress, DB2, Informix, Sybase itd.

STRUKTURNI PRISTOP

67. Naštejte osnovne značilnosti strukturnega pristopa?

- Eden prvih sistematičnih pristopov k razvoju IS
- Zgleduje se po standardnih postopkih razvoja tehničnih izdelkov: aktivnosti si sledijo zaporedno.
- Izoblikoval se je konec 60 in v začetku 70 let.
 - Razlog: uvedba discipliniranega izvajanja analize in načrtovanja.
 - Cilj: zmanjšanje stroškov izgradnje in uvajanja IS.
- Pristop Top Down
- **Koraki** strukturnega pristopa:
 - Strukturna analiza
 - Strukturno načrtovanje
 - Strukturno programiranje
- Najpomembnejše tehnike:
 - Diagrami podatkovnih tokov
 - Funkcionalna dekompozicija
 - Diagrami entiteta-razmerje (E-R)

68. Kaj veste o metodologiji *Information Engineering* (IE)?

- IE je primer metodologije, ki opisuje razvoj IS po strukturnem pristopu.
- Nastane leta 1981, glavni avtor je James Martin.
- Uveljavitev v sredini 80-tih let, uporablja se še danes.
- IE je zasnovan na teoretičnih in praktičnih dosežkih 80-tih let iz metodološkega in tehnološkega vidika.

69. Opišite osnovne značilnosti IE.

- sloni na povezani množici tehnik za planiranje, analizo, načrtovanje, razvoj in vzdrževanje informacijskega sistema celotne organizacije ali vsaj njenih glavnih delov.
- uporablja pristop od vrha navzdol (*top-down approach*)
- je podatkovno usmerjen
- podpira avtomatizacijo razvoja
- uveljavlja strateško planiranje
- povečuje produktivnost.

70. Kako poteka razvoj po IE? Naštejte osnovne faze.

- Strateško planiranje
- Analiza
- Načrtovanje
- Izvedba

IE posebej obravnava podatke in posebej aktivnosti

71. Kaj je strateško planiranje? *Kdaj ga izvajamo in zakaj?*

Po metodologiji Informacijskega inženiringa se razvoj IS prične z izdelavo strateškega plana. To je temeljni dokument, na podlagi katerega se izvajajo vse nadaljnje aktivnosti pri razvoju.

72. Kakšni so osnovni cilji strateškega planiranja?

- Povezati razvoj IS s poslovno strategijo OS.
- Izboljšati komunikacije med vodstveno strukturo in informatiki.
- Načrtovati pretok informacij in procesov.
- Učinkovito razporediti človeške vire.
- Zmanjšati stroške in skrajšati čas, ki je potreben za razvoj aplikacij.
- Predlagati optimalno zaporedje nadaljnjih korakov pri planiranju in razvoju IS.

- Pripraviti vsa potrebna izhodišča, ki bodo služila kot pomoč pri nadaljnjih korakih informatizacije vse do izdelave aplikativnih sistemov.
- Uporabiti standarde za enotne tehnološke rešitve.

73. Kaj opredeljuje metamodel strateškega planiranja?

Metamodel strateškega planiranja prikazuje elemente, ki jih obravnavamo v sklopu strateškega planiranja, izdelke, ki pri tem nastanejo ter opredeljuje povezave med njimi.

74. Kaj obravnavamo v sklopu strateškega planiranja?

Cilje, usmeritve, probleme in kritične dejavnike uspeha (KDU), organizacijske enote, geografske lokacije, funkcionalna področja, funkcije in postopke, delovne procese, entitete, informacijsko tehnologijo, kadre itd.

75. Kaj so izdelki strateškega planiranja?

- Seznam strateških elementov
- Pregledni model,
- Analiza obstoječega stanja IS,
- Načrt IT in
- Plan razvoja IS.

76. Koliko časa približno traja izdelava strateškega načrta?

Od 3 do 6 mesecev. Strateški plan je potrebno osveževati!

77. Koliko časa predvidevamo za uresničitev strateškega načrta?

Uresničevanje strateškega načrta traja približno od 2 do 5 let.

78. Opišite postopek izdelave strateškega načrta.

- Analiza obstoječega stanja
- Opredelitev poslovnih zahtev
- Opredelitev tehnoloških zahtev
- Planiranje informacijskega sistema
- Dokumentacija

79. Kaj zajema analiza obstoječega stanja?

- Namen analize obstoječega stanja je analizirati strateške elemente organizacije, da bo razvit IS v celoti usklajen z njenimi smernicami in cilji ter podati pregled obstoječega stanja IS.
- Koraki:
 - Analiza strateških elementov
 - Analiza obstoječega stanja IS

80. Kaj zajema opredelitev poslovnih zahtev?

- Namen opredelitve poslovnih zahtev je doseči čim večjo stopnjo razumevanja dogajanja v organizaciji ali delovnem področju.
- Koraki zajemajo izdelavo naslednjih izdelkov:
 - Organizacijske sheme
 - Globalnega funkcionalnega modela
 - Globalnega modela podat. tokov
 - Globalnega modela del. procesov
 - Globalnega podatkovnega modela
 - Povezovalnih matrik

81. Kaj zajema opredelitev tehnoloških zahtev?

- Namen opredelitve tehnoloških zahtev je identificirati tiste potrebne tehnološke značilnosti sistema (kritični moduli sistema, distribuiranje podatkov in programov), ki bodo omogočale delovanje IS.
- Koraki zajemajo opredelitve tehnoloških zahtev zajemajo:
 - Analizo vpliva IT
 - Načrt IT

82. Kaj je namen planiranja IS?

Namen planiranja IS je izdelati termnsko in po sredstvih opredeljen načrt aktivnosti, ki so potrebne za uresničitev strateškega načrta.

83. Kaj je *glavni namen analize po IE*?

- Glavni namen analize je izdelati razumljiv opis realnega sveta oziroma poslovnega okolja, na katerega se nanaša razvoj IS.
- Analiza daje odgovor na vprašanje, KAJ naj IS podpira. Kaj se izvaja v poslovnih funkcijah in kakšne podatke te rabijo?

84. Naštejte *aktivnosti, ki sodijo v fazo analize po IE*.

- Zajem zahtev: zajem zahtev se nanaša na opredelitev funkcionalnosti, ki naj jo sistem podpira. Uporabniki sodelujejo z analitiki.
- Modeliranje sistema: predstavitev zajetih zahtev v razumljivi in nedvoumni obliki. Model analize navadno zajema več vidikov, ki so predstavljeni vsak z svojim modelom.
- **Med ostale aktivnosti analize štejejo tudi:**
 - Izdelava dokumenta o podpori sistemu
 - Izdelava dokumenta o strategiji distribuiranja podatkov in programov
 - Izdelava dokumenta o potrebnih tehničnih značilnostih sistema
 - Prevedba podatkov
 - Izdelava dokumenta o zahtevah in standardih dokumentacije
 - Izdelava strategije uvajanja
 - Izdelava strategije testiranja

85. Kakšen je namen *zajema zahtev*?

- Naloga zajema zahtev je zbrati čim več informacij o lastnostih obstoječega IS ter zahtev za nov IS.
 - Kako zaposleni izvajajo svoje delo?,
 - kateri podatki so potrebni za pravilno delovanje sistema?,
 - Kakšne izpise je potrebno generirati?,
 - Kako uporabniki uporabljajo sistem pri svojem delu?,
 - So planirane spremembe načina dela v novem sistemu?

86. Kakšne *tehnik*e zajema zahtev poznate?

Zajem informacij najlaže izvedemo tako, da neposredno komuniciramo z ljudmi, ki so kakorkoli vključeni v sistem (uporabniki računalniških programov, vodstveni delavci itd.) ter z zbiranjem in preučevanjem dokumentacije o obstoječem sistemu. Poslužimo se lahko mnogih tradicionalnih in modernih metod zajema lastnosti ter zahtev o nekem sistemu.

Tradicionalne tehnike zajema zahtev:

- individualni in skupinski razgovori,
- zajem informacij s pomočjo vprašalnikov,
- opazovanje ljudi ob delu,
- preučevanje obstoječe dokumentacije, itd.,

Moderne tehnike zajema zahtev:

- skupno načrtovanje aplikacij (ang. *JAD – Joint Application Design*)
- uporaba sistemov za skupinsko delo (ang. *Group Support Systems*),
- uporaba CASE orodij,
- uporaba prototipov itd.

87. Kdaj je smiselno uporabljati intervjuje in kdaj vprašalnike?

Vprašalnike uporabimo takrat, ko nam razgovori zaradi časovne ali cenovne zahtevnosti ne ustrezajo, ali takrat, ko želimo zajeti informacije o neki specifični stvari, recimo, zanima nas splošno mnenje o nečem itd. Vprašalniki imajo tip zaprtih vprašanj – bolj za specifične stvari, ko so osnove že jasne (odprta vprašanja – intervjuji).

88. Kakšne so slabosti vprašalnikov?

Ločen si od uporabnikov. Ni pogovora. Ne dobiš dovolj informacij, tako kot pri odprtih vprašanjih.

89. Kakšna so priporočila za izvajalca zajema zahtev?

- Razgovor je potrebno organizirati,
- Na razgovor se je potrebno pripraviti,
- Če je možno, razgovor snemamo,
- Izogibati se moramo vprašanjem, ki nakazujejo odgovor,
- Zapiske ali posnetek razgovora čim prej uredimo.

90. Kaj je JAD?

Moderna metoda zajema zahtev.

- *JAD – Joint Application Design* ali Skupinsko načrtovanje aplikacij
- Pojavi se konec sedemdesetih let
- Avtor IBM
- Osnovna ideja: organizirati sejo in na njega povabiti vse ključne ljudi za analizo sistema.
- Omogoča identifikacijo področij, kjer si uporabniki niso enotni. Daje možnost, da konflikte rešimo ali razjasnimo.
- Sestanek je priporočljivo organizirati izved delovnega okolja – npr. V ustreznem hotelu.
- Izvedba je zelo draga – prisotni so vodstveni delavci, stroški hotela, organizacije,...

91. Zakaj modeliramo sisteme?

Ko enkrat zajamemo lastnosti obstoječega sistema ter zahteve in želje za nov sistem, moramo zajete informacije predstaviti v taki obliki, ki jedrnato, razumljivo in nedvoumno definira problemsko domeno. V ta namen pripravimo različne modele sistema, ki opisujejo sistem vsak s svojega vidika, skupaj pa sestavljajo specifikacijo sistema. Izbira modelov je v veliki meri odvisna od pristopa, ki ga pri razvoju sistema uporabljamo. Zato si pogledjmo najprej nekaj splošnih podatkov o modeliranju.

Modeliranje

Modeliranje je uveljavljena inženirska tehnika na mnogih področjih. Npr. v gradbeništvu skuša gradbeni inženir predstaviti končen produkt tako, da pripravi arhitekturni model stavbe. Različne vplive na stavbo (vetrovi, potresi...) pa preverja s pomočjo matematičnega modela. Podobno se modeliranje uporablja na mnogih področjih (avtomobilska industrija, avio-industrija, filmska industrija, ekonomija, upravljanje, sociologija...), med katere sodi tudi razvoj programske opreme.

92. Kaj je model?

- Model je poenostavitev realnosti, pri čemer je abstrakcija realnosti poljubno natančna.
- Pomembno je, da model prikazuje pomembne elemente in izpušča tiste, ki nas ne zanimajo.

93. Kaj zajema *modeliranje* po IE? Iz česa se sestoji *specifikacija sistema* po IE?

- **Podatkovni model:** diagram entiteta-razmerje
- **Procesni model:** procesni diagram, diagram podatkovnih tokov, funkcionalna razgradnja
- **Model procesne logike:** naravni jezik, strukturiran jezik, odločitvene tabele, odločitveni grafi, diagrami prehajanja stanj

94. Kakšna je razlika med formalnimi in neformalnimi tehnikami modeliranja sistema?

Formalni – uporabljamo diagrame, pri neformalnih uporabljamo naravni jezik in ostale podobne tehnike, kjer je velikokrat v igri dvojni pomen...

95. Katere tehnike uporabljamo za *podatkovno modeliranje*?

Za izdelavo podatkovnega modela uporabljamo diagrame entiteta-razmerje.

96. Kako obravnavamo *podatke* skozi vse faze razvoja po IE?

- Globalni podatkovni model
- Konceptualni podatkovni model
- Logični podatkovni model
- Fizični model
- Podatkovna baza

97. Kako obravnavamo *aktivnosti* skozi vse faze razvoja po IE?

- diagram funkcionalne razgradnje ali funkcionalna dekompozicija in
- diagram podatkovnih tokov.

98. Kaj je E-R diagram? Kdaj ga uporabljamo? Kakšni so njegovi gradniki?

Za izdelavo podatkovnega modela uporabljamo diagrame entiteta-razmerje.

Uporabljamo ga, ko modeliramo podatke, kako se bodo shranjevali.

Gradniki: entiteta, razmerje (ime, vloga, stevnost), atributi (ID ali osnovni, opisni).

99. Kaj je diagram podatkovnih tokov? Kdaj ga uporabljamo? Kakšni so njegovi gradniki?

V sklopu analize sistema uporabimo diagrame podatkovnih tokov za prikaz okolja, v katerem bo sistem deloval ter za prikaz odvisnosti med procesi, ki jih bo sistem podprl.

Diagramska tehnika diagramov podatkovnih tokov (ang. Data Flow Diagrams - DFD) služi za predstavitev podatkovnih tokov skozi organizacijski sistem, predstavitev procesov, ki podatke obdelujejo ter predstavitev izvorov in ponorov podatkov. Diagram podatkovnih tokov združuje podatkovni in procesni pogled na obravnavano področje.

- Osnovni gradniki diagramov podatkovnih tokov:
 - Proces
 - Podatkovni tok
 - Podatkovno skladišče (shramba)
 - Zunanji izvor ali ponor (zunanja entiteta)

100. Kaj je funkcionalna dekompozicija? Kdaj jo uporabljamo? Kakšni so njeni gradniki?

Z diagramom funkcionalne razgradnje ali krajše s funkcionalno dekompozicijo prikažemo hierarhijo funkcij, ki jih želimo s sistemom podpreti.

- Hierarhijo funkcij lahko prikažemo na različne načine:
 - drevesna struktura kot navpična hierarhija pravokotnikov,
 - drevesna struktura kot vodoravna hierarhija pravokotnikov,
 - drevesna struktura kot ugnedeno zamaknjeno besedilo,
 - večnivojska drevesna struktura, kjer je pravokotnik vhod v strukturo na nižjem nivoju in
 - krožni prikaz hierarhije.
- Značilnosti dekompozicije:

- Vsaka hierarhična struktura se začne na vrhu z eno samo vseobsegajočo enoto - koren strukture (root).
- Po najbolj enostavni tehniki so podrejene enote nanizane vodoravno en nivo nižje v hierarhiji in na podoben način do najnižjega nivoja. Elemente na najnižjem nivoju lahko zaradi podobnosti z drevesno strukturo imenujemo listi.

101. Kakšne tehnike modeliranja procesne logike poznate?

Naravni jezik, strukturni jezik, odločitvene tabele, odločitvena drevesa, diagrami prehajanja stanj.

102. Kakšne tehnike poznate za izdelavo procesnega modela?

Diagrami podatkovnih tokov, diagrami funkcionalne razgradnje.

103. Kaj opredeljuje *procesni model*?

Procesni model opredeljuje dinamično plat sistema. Prikazuje hierarhijo funkcij in procesov ter njihovo odvisnost.

104. Opišite postopek gradnje procesnega modela.

Gradnja podrobnega funkcionalnega modela poteka vzporedno z gradnjo podrobnega podatkovnega modela. **Prvi korak** pri izdelavi podrobnega funkcionalnega modela je izdelava dekompozicijskega diagrama funkcij in procesov. **Drugi korak** je opisovanje posameznih funkcij in procesov. Tako kot pri podrobnem podatkovnem modelu je tudi pri podrobnem funkcionalnem modelu opis zelo pomemben, saj je mogoče z njim preveriti pravilnost hierarhične razgradnje funkcij in procesov.

105. Kaj je *glavni namen načrtovanja po IE*?

Glavni namen načrtovanja je izdelati načrt zgradbe sistema glede na specifikacije, ki so bile zbrane v fazi analize.

106. Naštejte *aktivnosti*, ki sodijo v fazo načrtovanja po IE.

- opredelitev tehnoloških zahtev,
- načrtovanje podatkovne baze,
- načrtovanje in izdelava programskih modulov,
- prevedba podatkov,
- dokumentacija,
- testiranje,
- uvajanje in
- prehod na novi sistem.

107. Kaj je cilj faze načrtovanja po IE?

- izdelati načrt IS, ki ustreza specifikacijam, ki smo jih pridobili v fazi analize, in upošteva tehnološke omejitve sistema,
- dokumentirati specifikacije načrta na način, ki bo omogočal vzdrževanje sistema in
- zasnovati strategijo prehoda iz obstoječe na novo aplikacijo.

108. Kaj zajema načrtovanje po podatkovni plati?

Konceptualni model je rezultat analize podatkov. V fazi načrtovanja ga pretvorimo v logični model ter nato v fizični model, ki je osnova za generiranje podatkovne baze.

109. Opišite značilnosti prehoda iz konceptualnega v logični model.

Slide 261.

110. Kako je v podatkovni bazi predstavljena entiteta?

S tabelo.

111. Kaj je funkcionalna odvisnost?

- Funkcionalna odvisnost je posebno razmerje med dvema atributoma oziroma skupinama atributov.
- V relaciji R velja, da je atribut B funkcionalno odvisen od atributa A, če je z vsako veljavno vrednostjo atributa A določena tudi vrednost atributa B.

112. Kaj je relacijska shema?

Račun(Številka računa, Datum, #Šifra kupca, Valuta, Skupna cena)

Postavka(#Številka računa, #Šifra artikla, količina, cena, popust)

Artikel(Šifra artikla, naziv, cena, zaloga, davek)

Kupec(Šifra kupca, naziv, naslov, DŠ, ŽR)

113. Zakaj izvajamo normalizacijo?

- Normalizacija je postopek pretvorbe kompleksnih podatkovnih struktur v enostavne, stabilne podatkovne strukture.
- Nenormalizirane relacije so slabo strukturirane in obsegajo določeno stopnjo redundance, katere posledica so napake in nekonsistentnosti pri vnosu, spreminjanju in brisanju vrstic v tabeli.

114. Opišite postopek normalizacije.

Normalizacija relacije poteka v več fazah, vsaka faza pa ustreza normaliziranju v skladu s pravili, ki jih določena normalna oblika zahteva. Vsako normalno obliko sestavlja eno ali več pravil, ki se nanašajo na odvisnosti atributov oziroma skupina atributov relacije. Določena relacija je v določeni normalni obliki, če odvisnosti med atributi te relacije ustrezajo pravilom, ki jih obsega ta normalna oblika. Pri razvoju relacijske teorije je bilo evidentiranih šest normalnih oblik, pri čemer se najpogosteje uporabljajo prve tri.

115. Kaj so pravila normalizacijskih oblik?

- **Prva normalna oblika (1NO):** Relacija ni v prvi normalni obliki, če obstajajo ponavljajoče vrednosti atributov. Relacijo je potrebno pretvoriti v prvo normalno obliko tako, da je v tabeli na vsakem presečišču vrstice in stolpca ena sama vrednost.
- **Druga normalna oblika (2NO):** Druga normalna oblika predpostavlja, da je relacija v prvi normalni obliki, hkrati pa ne smejo obstajati nepolne funkcionalne odvisnosti, kar pomeni, da noben neosnovni atribut ne sme biti odvisen od podmnožice ključa (neosnovni atributi so tisti, ki niso del ključa).
- **Tretja normalna oblika (3NO):** Tretja normalna oblika predpostavlja, da je relacija v drugi normalni obliki, hkrati pa ne smejo obstajati neosnovni atributi, ki bi bili odvisni od drugega neosnovnega atributa (tranzitivna odvisnost).

116. Kaj zajema načrt strukture modulov?

- Funkcije in procesi, ki jih obravnavamo v fazi analize, predstavljajo logične sklope sistema. V fazi načrtovanja jih pretvorimo v fizične oz. programske sklope ali module.
- Implementacija enega logičnega sklopa je lahko izvedena z več programskimi sklopi. En programski sklop pa lahko implementira tudi več logičnih enot.

117. Opišite lastnosti strukturnih diagramov.

- Strukturni diagram prikazuje, kako je aplikativni sistem organiziran s hierarhijo programskih modulov.
- Iz diagrama je razvidna odvisnost med moduli aplikativnega sistema z vidika podatkov, ki se med moduli prenašajo.
- Iz strukturnega diagrama so razvidni tudi osnovni elementi strukturnega programiranja: zaporedje, izbira in ponavljanje.
- Moduli so organizirani v hierarhijo, podobno kot funkcije v funkcionalni razgradnji.

- Na najvišjem mestu je vseobsegajoč modul ali koren. Na naslednjem nivoju so moduli, ki jih koren lahko kliče (analogno kot izbire v meniju).
- Moduli komunicirajo med seboj s pomočjo parametrov:
 - nosilci podatkov
 - kontrolne zastavice

OBJEKTNI PRISTOP

118. Kaj so osnovni principi objektne usmerjenosti?

- **abstrakcija** - Vsak model, ki vsebuje najpomembnejše, bistvene ali razločevalne vidike nečesa, medtem ko zamolči ali ignorira manj pomembne, nebistvene podrobnosti. Je rezultat odstranitve razlik in poudarjanja skupnih značilnosti
- **ograjevanje** - Skrije posebnosti implementacije pred odjemalcem Odjemalec je odvisen oz. komunicira preko vmesnika
- **modularnost** - Razbitje nečesa kompleksnega na manjše obvladljive kose (module)
- **hierarhija** - Kakršno koli razvrščanje ali urejanje abstrakcij v drevesno strukturo. Vrste: agregacijska hierarhija, razredna hierarhija, hierarhija vsebovanja, hierarhija dedovanja, delitvena hierarhija, specializacijska hierarhija, hierarhija tipov.

119. Kaj je ograjevanje?

Skrije posebnosti implementacije pred odjemalcem Odjemalec je odvisen oz. komunicira preko vmesnika

120. Katere koncepte (gradnike) objektne usmerjenosti poznate?

- Objekt
- Razred
- Atribut
- Operacija
- Vmesnik (Polimorfizem-mnogoličnost)
- Komponenta
- Paket
- Podsistem
- Povezave

121. Kaj je objekt in kaj razred?

- **Objekt** je konkretna manifestacija abstrakcije; entiteta z dobro definiranimi mejami in identiteto, ki ograjuje njegovo stanje in obnašanje; primerek razreda.
- **Razred** je opis skupine objektov z enakimi lastnostmi (atributi), enakim obnašanjem (operacije), povezavami, in semantiko (pomenom)

122. Kaj je paket?

- Paket je splošen mehanizem za organizacijo elementov modela v skupine
- Je element modela, sestavljen iz drugih elementov modela

123. Kaj je vmesnik?

- Vmesnik formalizira polimorfizem
- Vmesniki podpirajo arhitekturo "*plug-and-play*"
- Remember Interface from COM.

124. Kakšna je razlika med podsistemom in paketom?

- Podsistem je kombinacija med paketom (vsebuje več drugih elementov modela) in razredom (ima določeno obnašanje)
- Podsistem realizira enega ali več vmesnikov, ki predstavljajo njegovo obnašanje

125. Kaj je operacija in kaj metoda?

Operacija je lahko ukaz ali vprašanje. Vprašanje ne sme nikoli spremeniti stanja objekta. Samo ukaz lahko spremeni stanje objekta. Rezultat operacije je odvisen od trenutnega stanja objekta.

126. Kakšna je razlika med objektom in entitetnim tipom?

V bistvu je entitetni tip nekaj podobnega kot razred. Entitetni tip združuje entitete istega tipa. Entiteta je primerek določenega tipa.

Objekt je pa instanca nekega razreda, tako kot entiteta nekega entitetnega tipa.

127. Kaj je komponenta?

- Netrivialen, skoraj neodvisen in nadomestljiv del sistema, ki izpolnjuje jasno funkcijo v skladu z dobro definirano arhitekturo
- Komponenta je lahko:
 - Komponenta izvorne kode
 - *Run-time* komponenta ali
 - Izvršljiva komponenta

128. Kaj je UML?

Unified Modeling Language (UML) je jezik za specifikacijo, vizualizacijo, konstrukcijo in dokumentacijo izdelkov programsko-intenzivnega sistema.

129. Naštejte diagramске tehnike UML?

- Diagrami primerov uporabe za predstavitev interakcij med uporabnikom in sistemom.
- Diagrami razredov za predstavitev logične strukture.
- Diagrami objektov za predstavitev objektov in medsebojnih povezav.
- Diagrami stanj za predstavitev obnašanja objekta.
- Diagrami komponent za predstavitev fizične strukture programske opreme.
- Diagrami postavitve za prikaz preslikave med programskimi komponentam in strojno opremo.
- Diagrami interakcije (diagrami sodelovanja in diagrami zaporedja) za prikaz obnašanja sistema.
- Diagrami aktivnosti za prikaz toka dogodkov v primerih uporabe.

130. Čemu so namenjeni diagrami primerov uporabe?

Z diagramom primerov uporabe prikažemo zakaj se sistem uporablja in kakšna je povezava med primeri uporabe ter med primeri uporabe in akterji. Ko akter želi neko storitev od sistema, sproži ustrezen primer uporabe.

131. Kaj je primer uporabe?

Rečemo lahko, da primer uporabe modelira dialog oziroma komunikacijo med akterjem in sistemom.

132. Kaj je akter?

Z akterji označujemo subjekte, ki so izven sistema in za katere obstaja določen vmesnik, da s sistemom lahko komunicirajo.

133. Med akterji in primeri uporabe lahko uporabljamo različne povezave. Kakšne?

Puščica, ki povezuje primere uporabe med seboj ter akterje in primere uporabe, je neobvezna, prikazuje pa smer vzpostavitve interakcije. Na primer, študent *požene* primer uporabe *prijava na izbirni predmet*. Primer uporabe *zaključitev prijave* pa vzpostavi komunikacijo z zunanjim sistemom za pripravo urnika.

134. Kako so lahko povezani razredi v razrednem diagramu?

- **Asociacije** predstavljajo strukturno povezavo med objekti različnih razredov (objekt je trajno povezan z drugim objektom). Imamo dve vrsti asociacije: agregacijo in kompozicijo.
- **Odvisnost** je povezava, ki predstavlja odvisnost med odjemalcem in dobaviteljem, kjer lahko sprememba v dobavitelju povzroči tudi spremembo v odjemalcu.
- **Generalizacija (specializacija) je povezava, kjer so elementi bolj specializiranega elementa (podtipa, otroka) lahko nadomeščeni z bolj generalnim elementom (staršem).**
- **Realizacija** je semantična povezava med dvema elementoma modela. En klasifikator služi kot pogodba, da se drugi klasifikator strinja z izvršitvijo.

135. Kaj pomeni navigacija na povezavi med dvema razredoma?

Študent pozna vse svoje urnike, medtem ko urnik ne ve kateremu študentu pripada (enosmerna povezava, ponazarja jo puščica proti urniku).

Tako omenjene omejitve števnosti kot tudi navigacija predstavljata poslovna pravila in/ali načrtovalske odločitve. Podrobnejše je načrtovanje povezav obravnavano v sklopu Načrtovanje razredov.

136. Kaj je povezava tipa odvisnost?

- Povezava med dvema elementoma modela, kjer lahko sprememba v enem elementu povzroči spremembo v drugem
- Povezava tipa "uporablja"

137. Kaj je povezava tipa realizacija?

- Realizacija je semantična povezava med dvema elementoma modela. En element modela služi kot pogodba, ki jo bo realiziral drugi element.
- Realizacijo lahko modeliramo kot črtkano povezavo s prazno puščico (kanonična oblika) ali v kombinaciji z vmesnikom, ki je predstavljen s krogcem (enostavna oblika).
- Realizacija je kombinacija odvisnosti in generalizacije. Ni prava generalizacija, saj se podeduje le "pogodba" (oznake operacij). Mešanica je v jeziku UML predstavljena kot kombinacija odvisnosti in generalizacije.

138. Čemu so namenjeni interakcijski diagrami?

- Diagrami interakcij opisujejo sodelovanje skupine objektov pri določenem obnašanju
- Interakcijo lahko opazujemo iz dveh perspektiv:
 - časovna perspektiva – diagrami zaporedja
 - prostorska perspektiva – diagrami sodelovanja
- Interakcija je odsev skupnega obnašanja

139. Kaj je *stereotip*?

- **Stereotipi** omogočajo razširitev osnovne UML notacije. Na osnovi obstoječih elementov lahko definiramo svoje nove elemente. Novi element lahko vsebuje dodatno semantiko. S tem načinom je zmanjšano število osnovnih modeliranih simbolov ter je enostavnejša tudi celotna notacija.
- **Stereotip** lahko uporabimo pri vseh modeliranih elementih: razredih, povezavah, komponentah itd. Vsak UML element ima lahko največ en stereotip.

140. Kaj je RUP

RUP je primer procesa, ki:

- Daje smernice za učinkovit razvoj kakovostne programske opreme
- Zmanjšuje tveganje in povečuje predvidljivost
- Zajema in vpeljuje najboljše izkušnje
 - učenje iz izkušenj drugih
 - mentorstvo v elektronski obliki
 - razširitev izobraževalnega gradiva
- Pospešuje skupno vizijo in kulturo

- Vpeljuje načrt za uvedbo orodij
- Omogoča enostaven in hiter dostop do informacij v elektronski obliki (spletna stran,...)

141. Naštejte izkušnje, na katerih je osnovan RUP.

Iterativni razvoj
 , Obvladovanje zahtev
 , Komponentna arhitektura
 , Vizualno modeliranje
 , Preverjanje kakovosti
 , Nadzor sprememb
 .

142. Kakšne postopke določa RUP?

Slide 310.

Za dober opis ali predstavitev sistema v razvoju je potrebnih več modelov. Ti so pridobljeni na osnovi postopkov, ki sestavljajo RUP, in se postopoma dopolnjujejo skozi iteracije.

- S poslovnim modelom prikažemo poslovne procese ter druge lastnosti poslovnega okolja. Poslovni model je lahko pomemben vir pri zajemu zahtev za podporni informacijski sistem.
- Model primerov uporabe zajema podatke o tem, kaj naj bi sistem delal, oziroma kakšna je zahtevana funkcionalnost sistema.
- Model načrta je objektni model, ki opisuje realizacijo primerov uporabe. Služi kot abstrakcija modela izvedbe in njegove izvorne kode.
- Model izvedbe je kolekcija komponent in podsistemov, ki te komponente vsebujejo.
- Model testiranja zajema testne primere uporabe in ostale potrebne podprograme za testiranje sistema.

143. Kaj je značilno za življenjski cikel razvoja, kateremu sledi RUP?

- Začetna faza: določimo okvir projekta: kaj je vključeno in kaj ni. To storimo z opredelitvijo vseh akterjev in vseh primerov uporabe, pri čemer najpomembnejše kratko opišemo (takih je približno 20%). V tej fazi pripravimo tudi projektni načrt, kjer opredelimo sredstva, ki morajo biti dodeljena projektu.
- Zbiranje informacij: zbiranje informacij je druga faza, kjer se osredotočimo na dve stvari: doseči dobro razumevanje zahtev (90%) ter določiti primerno arhitekturo. Z dobrim razumevanjem in primerno arhitekturo lahko močno zmanjšamo stopnjo tveganja, dosežemo pa tudi tisto raven, kjer znamo dobro oceniti časovne in finančne potrebe.
-

- **Konstrukcija:** konstrukcija zajema izgradnjo izdelka. Konstrukcija poteka v več iteracijah in se konča z beta verzijo izdelka.
- **Prevzem:** prevzem je faza, kjer predamo izdelek končnemu uporabniku in se osredotočimo na njegovo inštalacijo, podporo in uvajanje uporabnika.
- Potreben čas za izvedbo posameznih faz je spremenljiv. Pri velikih projektih, kjer obstaja veliko tehničnih nejasnosti ter nejasnosti v samih zahtevah, lahko zajem informacij traja 3-5 iteracij. Pri enostavnih projektih z jasnimi zahtevami in enostavno arhitekturo, pa je ena sama iteracija lahko tudi dovolj.

144. Kaj zajema analiza arhitekture?

Namen:

- Določitev arhitekturnih vzorcev, ključnih mehanizmov in modelirnih konvencij za celotni sistem.
- Določitev ponovno uporabljive strategije.
- Priskrbeti vhode v proces planiranja.

Rezultati:

- Prenovljen dokument programske arhitekture
- Prenovljen model načrta
- Prenovljene smernice načrtovanja
- Realizacije primerov uporabe (identificirane, ne razvite)

145. Kaj zajema analiza primerov uporabe?

V analizi primerov uporabe identificiramo razrede analize in določimo njihove odgovornosti. Zatem opozorimo na uporabo arhitekturnih vzorcev, definiranih v aktivnosti arhitekturne analize. Arhitekturni nivoji in njihove odvisnosti lahko vplivajo na dodelitev odgovornosti definiranim razredom analize. Dodelitev odgovornosti modeliramo v realizacijah primerov uporabe, ki opisujejo, kako razredi analize medsebojno sodelujejo pri izvajanju primerov uporabe. Realizacije primerov uporabe bodo izpopolnjene v aktivnosti načrtovanja primerov uporabe.

146. Kaj je realizacija primera uporabe? Kako jo naredimo?

Realizacija primera uporabe opisuje, kako je posamezen primer uporabe realiziran znotraj modela načrta v smislu medsebojnega sodelovanja objektov. Realizacijo primera uporabe iz modela načrta lahko zasledujemo v primeru uporabe v modelu primerov uporabe. Povezava realizacije poteka od realizacije primera uporabe proti primeru uporabe, ki ga realizira.

147. Koliko interakcijskih diagramov je potrebno izdelati v fazi analize primerov uporabe?

Potrebno je modelirati več tokov dogodkov, da zagotovimo, da so identificirane vse zahteve po operacijah sodelujočih razredov. Začnemo z opisom osnovnega toka, ki je najpogostejši ali najpomembnejši tok dogodkov. Nato opišemo različne alternativne tokove. Ni potrebno opisovati prav vseh tokov dogodkov, zelo trivialne tokove, ki se nanašajo na primer na samo en objekt, lahko izpustimo.

Primeri izjemnih tokov vsebujejo sledeče:

- Obravnavanje napak: Kaj naj sistem stori, če naleti na napako?
- Obravnavanje prekinitev: Če uporabnik v določenem času ne odgovori, se mora primer uporabe na nek način zaključiti.
- Obravnavanje napačnih uporabniških vnosov v objekte, ki sodelujejo v primeru uporabe.

Primeri opcijskih tokov vsebujejo sledeče:

- Akter iz množice možnosti sam izbere, kaj naj sistem izvede v nadaljevanju.
- Naslednji tok dogodkov je odvisen od vrednosti shranjenih atributov ali povezav.
- Naslednji tok dogodkov je odvisen od tipa podatkov, ki jih želimo procesirati.

148. Kaj je vhod v analizo primerov uporabe?

- Slovar

- Dodatne specifikacije
- Smernice za modeliranje primerov uporabe
- Model primerov uporabe
- Realizacije primerov uporabe (le identificirane)
- Dokument o arhitekturi

149. Kaj je rezultat analize primerov uporabe?

- Razredi analize
- Model analize in/ali Model načrta
- Izvedene realizacije primerov uporabe

150. Kaj je razred analize? Poimenujte in opišite tri *stereotipe* razredov, ki jih uporabljamo v analizi.

Razred je abstrakcija ki:

- poudarja važne značilnosti in
- zamolči ostale značilnosti.

Razred je sestavljen iz treh razdelkov:

- Prvi razdelek vsebuje ime razreda.
- Drugi razdelek prikazuje njegovo zgradbo (attribute).
- Tretji razdelek prikazuje obnašanje razreda (operacije).

Tehnika za določitev razredov analize, predstavljena v tem modulu, pri identifikaciji kandidatov za razrede uporablja tri različne perspektive sistema. To so:

- **boundary (mejni)** razredi med akterji in sistemom,
- informacije, ki jih uporablja sistem, so predstavljene z **entity (poslovnimi)** razredi,
- **control (kontrolni)** razredi, s katerimi je predstavljena kontrolna logika.

INFORMACIJSKE TEHNOLOGIJE

151. Kaj so *podatkovna skladišča*?

Podatkovno skladišče (*Data Warehouse*) je enotna zbirka podatkov, ki zajema najpomembnejše podatkovne entitete celotne organizacije ali organizacij. Iz stališča arhitekture predstavlja temelj sistemov za podporo odločanju.

152. Naštete osnovne značilnosti podatkovnih skladišč.

- Predmetno usmerjeno načrtovanje: podatkovno skladišče hrani podatke o najpomembnejših področjih poslovanja, ki so skupni celotni organizaciji.
- Statičnost podatkov: podatki v podatkovnem skladišču so statični. Ko jih enkrat zajamemo, jih v skladišču ne spreminjamo več. Namenjeni so zgolj pregledu oziroma poizvedovanju.
- Časovna dimenzija: v podatkovnem skladišču hranimo zgodovino podatkov in ne zgolj trenutne vrednosti. Podatkovno skladišče predstavlja zaporedje posnetkov stanj v določenem trenutku.
- Združevanje podatkov: pomembna lastnost podatkovnih skladišč je, da so sestavljena iz podatkov, ki lahko pridejo iz številnih virov.

153. V čem se podatkovna skladišča razlikujejo od podatkovnih baz?

Podatkovno skladišče je obsežna zbirka podatkov, zbranih iz podatkovnih baz posameznih aplikativnih področij. S posebno organizacijo podatkov omogoča razne analize ter poizvedovanja, ki sicer v lokalnih bazah niso mogoča. Ker predstavlja popolnoma ločeno strukturo, ne vpliva na izvajanje delovnih postopkov oziroma vsakodnevnih opravil, podprtih s transakcijskimi sistemi.

154. Kaj je *OLAP*?

- Čeprav se izraz OLAP pogosto uporablja v povezavi z večdimenzijskimi podatkovnimi bazami, gre le za koncept realizacije učinkovitega uporabniškega vmesnika, ki vodstvenim delavcem omogoča različne analize.
- V osnovi poznamo dve implementaciji OLAP orodij:
 - o ROLAP (relacijski OLAP), ki ga sestavlja množica uporabniških vmesnikov in aplikacij in temelji na relacijskih bazah, ter
 - o MOLAP (večdimenzijski OLAP), kjer je za razliko relacijskih OLAP-ov močno poudarjena dimenzijska značilnost.
 - o Možna pa je tudi kombinacija obeh rešitev: HOLAP (Hibridni OLAP).

155. Kaj so *področna podatkovna skladišča*?

Posebna oblika podatkovnega skladišča je področno podatkovno skladišče (*Data Mart*). Od navadnega podatkovnega skladišča se razlikuje v tem, da ne obsega podatkovnih entitet celotne organizacije, temveč pokriva le izbrano področje, npr. trženje, finančno poslovanje, materialno poslovanje ipd. Področno podatkovno skladišče zajema manj podatkov, manjše pa je tudi število virov, iz katerih podatke črpa. Kot tako omogoča hitrejši dostop in je lažje za izvedbo. V večjih organizacijah takšne rešitve pogosto srečamo, v mnogih primerih pa je poleg področnih skladišč na voljo tudi enotno podatkovno skladišče.

156. Kaj so portali?

Beseda »portal« izhaja iz latinske besede »porta«, kar pomeni vrata. Na področju informacijskih tehnologij gre za pogosto uporabljen termin, ki označuje koncept enotne vstopne točke, preko katere uporabniki dostopajo do vsebin, ki jih bodisi želijo sami ali pa jim jih portal ponudi, navadno na osnovi poznavanja uporabnikovega profila.

157. Kakšna je razlika med *informacijskim* in *storitvenim* portalom. Naštete nekaj primerov.

- Informacijske portale: namen informacijskih portalov je omogočiti čim lažji dostop do podatkov.
- Storitvene portale: Storitveni portali poleg dostopa do podatkov omogočajo tudi opravljanje posameznih storitev.

158. Kako bi opredelili *elektronsko poslovanje*?

Elektronsko poslovanje je splošen koncept, ki označuje uporabo informacijskih in telekomunikacijskih tehnologij za doseg učinkovitega sodelovanja znotraj organizacije ter zunaj nje.

159. Kakšne oblike elektronskega poslovanja poznate?

?

160. Kakšne *načine integracije sistemov za vzpostavitev elektronskega poslovanja* poznate?

?

161. Kaj so *sistemi za upravljanje, upodabljanje in arhiviranje dokumentov*?

V velikih organizacijskih sistemih imamo opravka z velikim številom raznih dokumentov, zaradi česar se pojavi potreba po orodju za upravljanje dokumentov. Tako orodje omogoča dober nadzor nad pretokom dokumentov, njihovem lastništvu ter življenjski dobi posameznega dokumenta. Elektronsko upravljanje dokumentov je pomemben pripomoček, ki omogoča preprosto in hitro iskanje informacij, nadziranje dostopa do datotek, beleženje različic dokumentov ter urejanje informacij v mape.

162. Opišite osnovne značilnosti *sistemov za podporo skupinskemu delu*.

- Omogočajo distribucijo podatkov, informacij in tudi znanja
- Povezujejo kompleksne organizacijske strukture – *virtualne organizacije*
- Enostaven in za uporabnika *transparenten* dostop do podatkov (dokumentov)
- Enostavna in učinkovita komunikacija ne glede na komunikacijsko opremo (LAN, modem ipd.)
- Enostavno vključevanje novih aplikacij in nadgradnja obstoječih
- Kompleksen sistem varnosti dostopa.

163. Kakšna področja dela pokrivajo *sistemi za podporo projektne vodnji*?

- priprava projektov,
- planiranje projektov,
- vodenje projektov,
- spremljanje projektov,
- koordinacija projektov,
- spremljanje kakovosti projektov,
- opredelitev tveganja projekta,
- spremljanje razvoja izdelkov,
- in druga področja.

164. Na kakšne ravni lahko razdelimo *logično arhitekturo aplikativnega sistema*?

- predstavitevno raven,
- poslovno raven ter
- podatkovno raven.

165. Kaj pokriva *posamezna raven*?

- **Predstavitevna raven** zajema vse, kar vidi uporabnik (uporabniški vmesnik). Občutljiva je na spremembe, ki nastanejo na osnovi uporabniških želja, po drugi strani pa neodvisna od tehnoloških sprememb. Predstavitevna raven vse svoje delo opravi tako, da kliče ustrezne storitve poslovnih ravnih.
- **Poslovno raven** najdemo v vseh delih aplikacije. Težko jo omejimo s paketom oziroma knjižnico podprogramov, saj je navadno integrirana med uporabniškim vmesnikom in podatki. *Duamish model* predlaga delitev poslovne ravni na tri kategorije: poslovni vmesnik (*Business Facade*), delovni tok (*Workflow*) in poslovna pravila (*Business Rules*).
- Poslovni vmesnik predstavlja poslovne storitve, ki so vidne predstavitevni ravni ali drugim aplikacijam. S tem poenostavlja uporabo poslovnih storitev, obenem pa skriva implementacijske podrobnosti. Primer takšnega vmesnika je, na primer, *logika za preverjanje pravilnosti vnosa*.

- Delovni tok predstavlja zaporedje korakov, ki zajemajo spremembo stanja sistema. Primer delovnega toka je logika, ki *dirigira* korake pri obravnavi naročila: izbira artiklov, količin, izračun skupnega zneska, izračun davka, plačilo itn.
- Poslovna pravila vključujejo poslovno logiko, ki zagotavlja izvajanje poslovne politike organizacije. Navadno se nanašajo na pravila implementacije posameznih avtonomnih transakcij kot na primer: pravila dodelitve kredita, pravila, ki določajo zalogo vrednosti določenega podatkovnega atributa, pravila, ki sprožajo akcije itd.
- **Raven dostopa do podatkov** zajema vse potrebne storitve, ki poslovni ravni omogočajo dostop do podatkov. Tudi tu gre za poenostavitve dostopa in skrivanje implementacijskih podrobnosti. S pomočjo ravni, ki skrbi za dostop do podatkov, izoliramo dele aplikacije, ki delajo s podatki, od drugih delov aplikacije.
- **Podatkovna raven** je raven, ki obsega podatke ter programsko opremo za upravljanje s podatki: relacijske podatkovne baze, shrambe elektronske pošte, sporočilne vrste (*Message Queue*) ipd.

166. Kakšen je odnos med *fizično* in *logično* arhitekturo aplikativnega sistema?

Čeprav velja, da sta lahko logična in fizična arhitektura medsebojno neodvisni, pa v večini primerov ni tako. Logična arhitektura vpliva, skupaj z aplikacijskimi zahtevami in izbrano platformo, tudi na izbiro fizične arhitekture in v idealnem primeru bi pričakovali skoraj popolno uravnanost. Vendar pa se pogosto dogaja, da omejitve in zahteve, s katerimi imamo opravka v fazi načrtovanja, kličejo po rešitvah, ki zahtevajo določena odstopanja. Na primer, če poslovna pravila izvedemo kot shranjene programe v podatkovni bazi (*Stored Procedures*), potem bo poslovna logika porazdeljena med COM komponentami in baznimi programi, namesto da bi bila zajeta v posebni komponenti s poslovno logiko. Fizično bo poslovna logika porazdeljena, logična arhitektura pa jo priporoča v enem kosu. S fizično arhitekturo implementiramo logično arhitekturo in če gre za uravnane arhitekture, potem logična raven postane fizična raven.

167. Kakšne fizične arhitekture aplikativnih sistemov so najpogostejše?

V grobem poznamo predvsem dve fizični arhitekturi: dvo-nivojsko ali strežnik-odjemalec (*Client-Server*) ter tri-nivojsko arhitekturo.

168. Naštete osnovne značilnosti arhitekture *strežnik-odjemalec*.

Strežnik, kot že ime samo pove, streže odjemalce z zahtevanimi podatki.

V okviru topologije C/S imamo opraviti z dvema vrstama porazdeljenosti: porazdeljenost podatkov ter porazdeljenost procesov.

169. Kakšne so *slabosti* arhitekture *strežnik-odjemalec*?

- Zaradi porazdeljenosti (lahko gre za večje število odjemalcev) imamo opraviti z visokimi obratovalnimi stroški ter stroški vzdrževanja.
- Stalne zahteve po nadgradnji programske opreme inicira potrebe tudi po nadgrajevanju strojne opreme.
- Debeli in dragi odjemalci (letni stroški debelega odjemalca presegajo \$10.000)
- Slaba izkoriščenost programske opreme

170. Kaj veste o *tronivojski* arhitekturi?

Trinivojska arhitektura pomeni, da so v sistemu odjemalci in strežniki (aplikacijski in podatkovni) medsebojno povezani preko povezovalnega vmesnega sloja, ki omogoča in hkrati skrbi za medsebojno komunikacijo. Bistvo je v umeščanju dodatnega elementa v komunikacijski verigi, ki ga imenujemo aplikacijski strežnik.

Vodilo pri vpeljavi take arhitekture je predvsem zmanjševanje stroškov nakupa in obsega vzdrževanja, zanesljivosti delovanja sistema ter prilagodljivost in odprtost razvitih sistemov za nove tehnologije.

171. Kakšno vlogo ima *aplikacijski strežnik* v *tronivojski* arhitekturi?

- Aplikacijski strežniki predstavljajo vmesni nivo fizične arhitekture aplikacijskih sistemov, ki povezuje:
 - o dele za komunikacijo in integracijo s spodnjim nivojem (podatkovni viri),
 - o dele za komunikacijo z odjemalci (običajno spletni odjemalci).

- Kot skupni nivo predstavljajo tudi ogrodje, na katerega lahko postavimo poslovno logiko.
- Kot rezultat dobimo modularen sistem, ki je prilagodljiv in se zato boljše odziva na želene spremembe.

172. Kakšna je razlika med *odjemalcem* v dvonivojski in tronivojski arhitekturi?

?

173. Kakšne so *prednosti* tronivojske arhitekture?

- odjemalci so ločeni od podatkovnih virov, zato je za dostop do podatkov dovolj zgolj poznavanje abstraktnih operacij in njihovih vhodnih in izhodnih parametrov,
- migracija in popravljanje podatkovne baze sta možna brez prekinjanja ali spreminjanja programov odjemalcev,
- spreminjanje programov odjemalcev je možno brez popravljanja podatkovne baze ali novih dostopnih mehanizmov,
- odjemalci lahko uporabljajo podatke različnih podatkovnih virov (na različnih podatkovnih strežnikih),
- zmanjšanje števila povezav (pri dvonivojski arhitekturi je za povezovanje M odjemalcev z N strežniki potrebnih $M \times N$ povezav, pri trinivojski pa samo $M+N$).

174. Kakšne tehnologije poznate za povezovanje informacijskih sistemov?

- lokalnih omrežjih (LAN),
- javnih regionalnih omrežjih (WAN),
- navideznih privatnih omrežjih (VPN) in
- mednarodnem omrežju (internet).

175. Kaj veste o tehnologiji *CORBA*?

Good sources for more information on mechanisms are the various documents on CORBA and COM. COM is usually only an implementation mechanism, but the CORBA documentation tends to be a bit more general. So you can look at the definitions for the CORBA services and, with a little abstraction, define general mechanisms for messaging, transaction management, persistence, etc.

176. Kako zagotavljamo varnost prenosa podatkov po javnih omrežjih?

Kriptografija

177. Kakšne *kriptografske metode* poznate?

Kriptografija s skritim ključem
(des), Kriptografija z javnim ključem
(mit rsa)

178. Kako deluje tehnologija kodiranja z *javnim ključem*?

?

179. Kaj je *digitalni podpis*?

Narava relacije med javnim in privatnim ključem zagotavlja, da lahko sporočilo kodirano s privatnim ključem pošiljatelja, dekodiramo le z njegovim javnim ključem. Tako lahko sporočilu, ki ga želimo poslati prejemniku, dodamo še nek zapis (recimo svojo telefonsko številko), ga kodiramo s svojim privatnim ključem, vse skupaj kodiramo s prejemnikovim javnim ključem ter pošljemo prejemniku. Prejemnik lahko sporočilo dekodira s svojim privatnim ključem, dodani zapis pa s pošiljateljevim javnim ključem. Tako je prejemnik prepričan, da je sporočilo prišlo res od pravega pošiljatelja (če pozna njegovo telefonsko številko).

180. Kaj je *digitalno potrdilo*?

?

