

5. MODULARIZACIJA I OBJEKTNO USMJERENA ARHITEKTURA

1. Što od navedenog predstavlja problem u oblikovanju programske potpore?
 - a) Skupljanje procesnih koraka u module
 - b) Implicitna struktura podsustava
 - c) Disperzija koda koji se odnosi na jednu odluku
 - d) Zatvaranje operacija i podataka, skivanje predstavljanja
2. Ako jedna promjena utječe na mnoge module tada se radi o problemu oblikovanja:
 - a) Nenamjerno otkrivanje interne strukture
 - b) Povezane odluke o oblikovanju
 - c) Prodiranje odluka o oblikovanju
 - d) Disperzija koda koji se odnosi na jednu odluku
3. Kako se naziva problem u oblikovanju kada povezane definicije raspršuju odluke (trebale bi biti lokalizirane - na jednom mjestu)?
 - a) Povezane odluke o oblikovanju
 - b) Prodiranje odluka o oblikovanju
 - c) Disperzija koda koji se odnosi na jednu odluku
 - d) Ranjivost na globalne (ili široko dijeljene) varijable
4. Kako se naziva problem u oblikovanju kada klasični programski jezici kreiraju dijeljenje (blokove strukture, globalne varijable) ?
 - a) Nenamjerno otkrivanje interne strukture
 - b) Disperzija koda koji se odnosi na jednu odluku
 - c) Ranjivost na globalne (ili široko dijeljene) varijable
 - d) Povezane odluke o oblikovanju
5. Rješenje za probleme u oblikovanju programske potpore je:
 - a) Jedna nit izvođenja
 - b) Skrivanju informacija i apstraktnih tipova podataka
 - c) Implicitna struktura podsustava
 - d) Modularizirati sustav
6. Modularizacija omogućuje bolje:
 - a) Upravljanje sustavom
 - b) Evoluciju sustava
 - c) Razumijevanje
 - d) Sve navedeno
7. S obzirom na povijest modularizacije razlikujemo:

- a) Glavni program i subrutine
 - b) Funkcijski moduli
 - c) Metode, operacije
 - d) Pod a) i b)
8. Najkasniji oblik modularizacije je:
- a) Glavni program i subrutine
 - b) Funkcijski moduli
 - c) Objekti i Objektno usmjerena arhitektura
 - d) Komponente i oblikovanje zasnovano na komponentama
9. Koji oblik modularizacije nedostaje u zadatku iznad (8 zadatak)?
- a) Hijerarhijska dekompozicija
 - b) Hijerarhijsko rasuđivanje
 - c) Apstraktni tipovi podataka (ADT Abstract Data Types)
 - d) Svi su tu _
10. Što nije obilježje modularizacije glavnim programom i subrutinama?
- a) Hijerarhijska dekompozicija
 - b) Hijerarhijsko rasuđivanje
 - c) Implicitna struktura podsustava
 - d) Podijeli i vladaj
11. Što znači implicitna struktura podsustava?
- a) Višestruka sučelja, posrednici, binarna kompatibilnost
 - b) Ispravno izvođenje rutine ovisi o ispravnom izvođenju subrutine koja se poziva
 - c) Struktura podsustava temeljena na odnosu definicija – uporaba
 - d) Subrutine su tipično skupljene u (funkcijske) module
12. Najčešći ali vrlo ograničen pogled za modul kaže:
- a) Jedinica posla
 - b) Jedinica kompilacije, koja uključuje deklaracije i sučelje
 - c) Dio koda
 - d) Sve navedeno
13. Organiziranje proceduralne apstrakcije u kontekstu podatkovne apstrakcije označava:
- a) Proceduralnu paradigmu
 - b) Objektno usmjerenu paradigmu
 - c) Instancu klase

d) Apstraktnu paradigmu

14. Dodavanje apstrakcije podacima :

- a) Pomaže u smanjivanju kompleksnosti sustava
- b) Pomaže u skrivanju informacija
- c) Izbjegava se grupiranje djelova podataka
- d) Omogućava apstraktne klase

15. Prema UML notaciji objekt se sastoji od:

- a) Naziva objekta i obilježja
- b) Naziva objekta i klase koja ga predstavlja
- c) Naziva objekta, atributa i metoda
- d) Obilježja, ponašanja i identiteta

16. Jedinicom apstrakcije u objektno orijentiranom programu nazivamo?

- a) Objekt
- b) Varijablu
- c) Klasu
- d) Operaciju

17. Instanca – objekt

- a) To su potpuno različite stvari
- b) Odnosi se na istu stvar, jedina razlika je korištenju prirodnog jezika
- c) Objekt je nadklasa instance
- d) Svaki objekt sastoji se od nekoliko instanci

18. Koja su osnovna svojstva objekta?

- a) Obilježje, ponašanje i identitet
- b) Obilježje, ponašanje i operacija
- c) Obilježje i metoda
- d) Obilježje operacija

19. Koja tvrdnja nije istinita?

- a) Varijable definirane unutar klase odgovaraju podacima prisutnim u svakoj instanci
- b) Svaki objekt je instanca jedne klase
- c) Klasa sadrži metode za implementaciju ponašanja instanci
- d) Klasa opisuje skup objekata sa potpuno različitim svojstvima

20. Jedan objekt može biti referenciran _____ u isto vrijeme:

- a) Samo sa jednom varijablom

- b) Sa više različitih varijabli
- c) Nijednom varijablom
- d) Samo jednom instancom

21. Varijabla klase je 1) _____ varijabla sa nekom vrijednošću 2) _____.

- a) 1) statička, 2) koju dijele sve instance klase
- b) 1) primitivna, 2) koju dijele sve instance klase
- c) 1) statička, 2) koju nikad ne dijele sve instance klase
- d) 1) primitivna, 2) koju vidi samo jedna instanca klase

22. Procedure (metode) će pohraniti svoje privremeno stanje u _____.

1. (lokalne varijable)

23. Što je točno? (može biti više točnih odgovora)

- a) Lokalna varijabla dostupna je čitavoj klasi unutar koje je deklarirana
- b) Lokalna varijabla nije dostupna čitavoj klasi, ali ju vide sve metode te klase
- c) Lokalna varijabla vidljiva jedino metodi unutar koje je deklarirana
- d) Ako jedna instanca promijeni vrijednost lokalne varijable tada tu promijenu vide sve instance te klase

24. Parametri su uvijek klasificirani kao varijable?

2. Točno Netočno

25. Najviša razina proceduralne apstrakcije koja specificira tip ponašanja je _____.

3. (operacija)

26. Koja tvrdnja nije točna?

- a) Metode implementiraju iste apstraktne operacije na način prikladan svakoj klasi
- b) Metoda služi za implementaciju ponašanja klase
- c) Metoda je viša razina apstrakcije od operacije
- d) Metode istog imena možemo imati u više različitih klasa

27. Polimorfizam zahtjeva:

- a) Da se objekt koji donosi odluku o metodi nalazi isključivo u lokalnoj varijabli
- b) Postojanja više metoda istog imena
- c) Liskov pristup zamjene
- d) Enkapsulaciju

28. Polimorfizam je svojstvo objektno orijentiranog programa da

se_____.

- a) Apstraktna operacija može izvesti na isti način u različitim klasama

- b) Apstraktna operacija može izvesti na različit način u istim klasama
- c) Apstraktna operacija ne može izvesti na različit način u istim klasama
- d) Apstraktna operacija može izvesti na različit način u različitim klasama

29. U UML notaciji trokut prikazuje:

- a) Agregaciju
- b) Kompoziciju
- c) Propagaciju
- d) Generalizaciju

30. Koja je tvrdnja točna (može biti više njih) vezana uz nasljeđivanje?

- a) Nastaje podklasa koja nasljeđuje attribute, relacije, operacije od temeljne klase
- b) Sve podklase implicitno posjeduju sva obilježja definirana u nadklasi
- c) Sve što vrijedi za nadklasu vrijedi i za podklasu ali ne i obrnuto
- d) Sve navedeno

31. Gdje se koristi „Is a “ pravilo?

- a) Za određivanje implementacijskih metoda koje se mogu nanovo definirati
- b) Kod dinamičkog povezivanja
- c) Prilikom određivanja konkretnih klasa
- d) Prilikom izgradnje hijerarhije objekata

32. Što od navedenog opisuje Liskov pristup zamjene?

- a) Ako imamo varijablu tipa nadklase i ako u nju stavimo instancu nadklase ili bilo koju instancu njezinih podklasa program neće raditi ispravno.
- b) Ako imamo varijablu tipa nadklase i ako u nju stavimo instancu nadklase ili bilo koju instancu njezinih podklasa program bi trebao raditi ispravno.
- c) Ako imamo varijablu tipa podklase i ako u nju stavimo instancu nadklase ili bilo koju instancu njezinih podklasa program bi trebao raditi ispravno.
- d) Ništa od navedenog

33. Ako je klasa apstraktna?

- a) Tada je beskorisna, možemo je izbrisati
- b) Ne može se instancirati
- c) Ona automatski znamo da se radi o krajnjoj klasi u hijerarhiji
- d) Znači da sadrži implementacijske metode

34. Suprotno od apstraktne klase je _____.

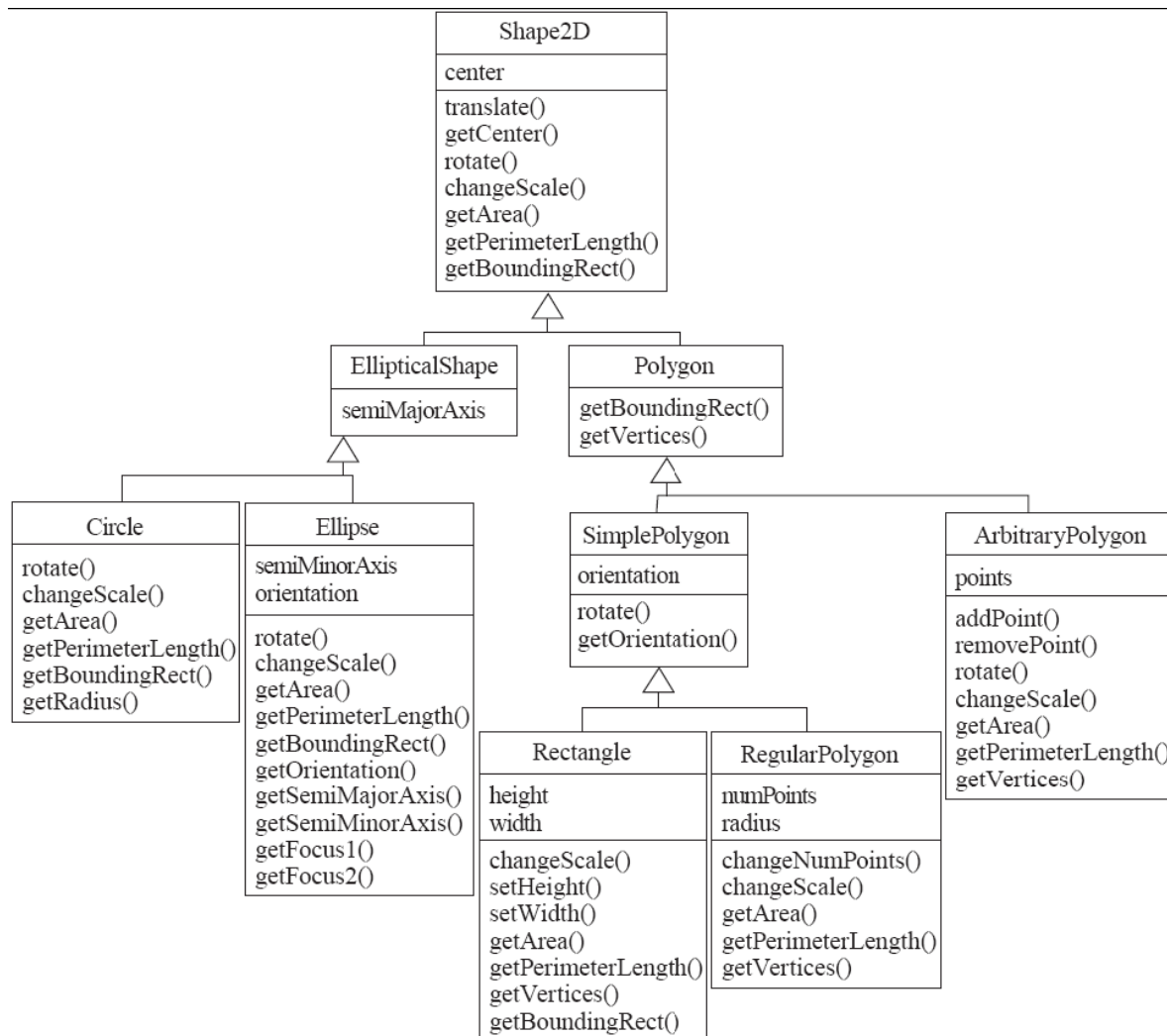
4. (konkretna klasa)

35. Ako nadklasa sadrži apstraktnu operaciju tada njezine podklase moraju imati _____
(konkretnu metodu za tu operaciju)

36. Što je točno za krajnju klasu u hijerarhihi (više odg)?

- a) Mora biti konkretna
- b) Može imati apstraktne operacije
- c) Mora imati ili nasljediti konkretne metode za sve operacije
- d) Može se instancirati

5. Zadaci 39 - 43 odnose se na sljedeću sliku:



37. Ako neka varijabla imena aShape sadrži instancu klase SimplePolygon to znači da:

- a) Varijabla aShape može biti tipa podklase
- b) Varijabla aShape može biti tipa nadklase

- c) Varijable aShape može biti bilo kojeg tipa klase u stablu nasljeđivanja klase Shape2D
 - d) Ne može se deklarirati takva varijabla
38. Ako se u varijablu aShape (koja je tipa Shape2D) nalazi instanca klase RegularPolygon i poziva se operacija getCenter:
- a) Program će prvo tražiti konkretnu metodu za tu operaciju u RegularPolygon a zatim gore u hijerarhiji.
 - b) Program će javiti grešku
 - c) Ovdje se primjenjuje mehanizam nasljeđivanja
 - d) Ovdje se primjenjuje mehanizam dinamičkog povezivanja
39. Ako se u varijablu aShape (koja je tipa Shape2D) nalazi instanca klase RegularPolygon i poziva se operacija changeScale:
- a) Konkretnu metodu u Shape2D je poništila konkretna metoda u RegularPolygon
 - b) Ovdje se primjenjuje mehanizam nasljeđivanja
 - c) Korišten je Liskov pristup zamjene
 - d) Ovdje se primjenjuje mehanizam dinamičkog povezivanja
40. Ako imamo varijablu myRect (koja je tipa Rectangle) i u njoj se nalazi instanca klase Rectangle:
- a) Primjenjuje se mehanizam dinamičkog povezivanja
 - b) Primjenjuje mehanizam statičkog povezivanja jer nema podklasa
 - c) Varijabla ne može biti tipa Rectangle, uvijek mora biti tipa nadklase
 - d) Korišten je Liskov pristup zamjene
41. U danom dijagramu klasa metoda getArea primjer je _____ jer se u svim krajnjim klasama implementira na različite načine.

6. (polimorfizma)

42. Više apstrakcije od metoda su:

- a) Klase
- b) Operacije
- c) Objekti
- d) Instance

43. Razlikuju li se dva objekta istog tipa iako sadrže iste podatke? _____

7. (da)

44. Za private metode vrijedi:

- a) Podklase klase mogu pristupati podacima i funkcijama nadređene klase
- b) Podaci i funkcije deklarirani u klasi dostupni su i izvan klase
- c) Samo funkcije deklarirane unutar klase imaju pristup podacima i funkcijama klase
- d) Ne znam

45. Za public metode vrijedi:

- a) Podklase klase mogu pristupati podacima i funkcijama nadređene klase
- b) Podaci i funkcije deklarirani u klasi dostupni su i izvan klase
- c) Samo funkcije deklarirane unutar klase imaju pristup podacima i funkcijama klase
- d) Jedna ili više vanjskih funkcija ili cijela klasa može imati pristup private podacima ili funkcijama klase s kojom je prijatelj

46. Što se od navedenog povezuje uz pojam enkapsulacija ?

- a) Kod se može konstruirati isključivo od klasa
- b) Programeri ne trebaju znati sve detalje klase
- c) Provodi se automatsko oslobađanje resursa
- d) Kod se ne smije sastojati bez komentara

47. Što nije točno, a vezano je uz stil programiranja

- a) Komentari bi trebali zauzimati 25-50% koda
- b) Potrebno je izbjegavati dupliciranje koda
- c) Koristi public metode što je više moguće
- d) Koristi private metode što je više moguće

48. Koja tvrdnja nije točna u vezi UML-a?

- a) Podržava cjelokupni životni razvoj softwara
- b) Podržan od mnogih alata
- c) Temeljen na iskustvu i potrebama korisnika
- d) Ipak su se sve točne

49. Koliko ima statičkih dijagrama

- a) 3
- b) 4
- c) 5
- d) ∞

50. Statički dijagrami su:

- a) use case, sequence, collaboration, statechart, activity

- b) use case, sequence, class, object, activity
- c) use case, class, object, statechart, deployment
- d) use case, class, object, component, deployment

51. Use case dijagram gradi se u:

- a) Kasnim fazama razvoja
- b) Ranim fazama razvoja
- c) To je zadnja faza razvoja u kojoj se vrši testiranje
- d) Na prijelazima iz faze u fazu

52. Statički dijagram koji prikazuje funkcionalnost sustava sa stajališta korisnika je

_____.

8. Use case

53. U Use case statičkom dijagramu aktor specificira 1) _____, a ne 2)

_____.

- a) 1) KAKO, 2) ŠTO
- b) 1) KADA, 2) ŠTO
- c) 1) ŠTO, 2) KAKO
- d) 1) ŠTO, 2) KADA

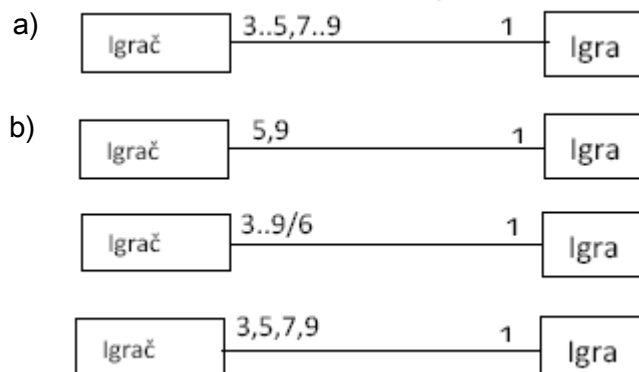
54. Koja je svrha use case dijagrama? (više točnih odgovora)

- a) Validacija arhitekture sustava
- b) Model kontrole toka
- c) Prikazuje zahtjeve sustava
- d) Određivanje odgovornosti

55. Sekvencijski dijagram prikazuje 1) _____ ponašanje (2) _____ orijentiran).

- a) 1) statičko, 2) strukturno
- b) 1) dinamički, 2) strukturno
- c) 1) statičko, 2) vremenski
- d) 1) dinamički, 2) vremenski

56. Ako jednu igru istovremeno mogu igrati između 3 i 5 ili između 7 i 9 igrača, koja je višestrukost od navedenih ispravna?



c)

d)

57. Koje asocijacije je potrebno izbjegavati?

- a) Nepotrebne više - prema - više asocijacije
- b) Nepotrebne više - prema - jedan asocijacije
- c) Nepotrebne jedan - prema - više asocijacije
- d) Nepotrebne jedan - prema – jedan asocijacije

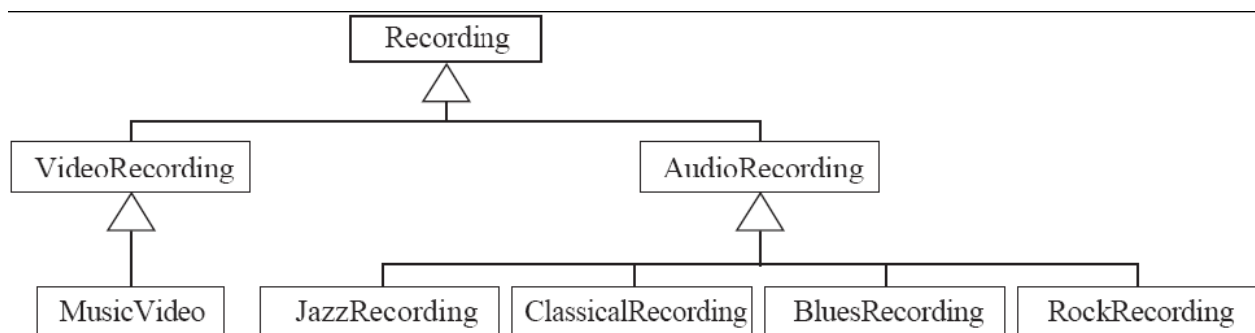
58. Asocijacijske klase (pridružene klase) nastaju:

- a) Kad imamo asocijacije jedan - prema - više
- b) Kad imamo asocijacije više - prema – jedan
- c) Kad imamo asocijacije više - prema – više
- d) Kad imamo asocijacije jedan - prema – jedan

59. Kad se atributi koje se odnose na dvije pridružene klase ne mogu smjestiti ni u jednu od njih stvara se _____.

9. (nova klasa)

60. Koja je tvrdanja točna za sljedeću hijerarhiju:



- a) Potpuno ispravna hijerarhija
- b) Nepotrebna hijerarhija klasa. Klase bi trebale biti instance
- c) Bolja solucija su dvije odvojene hijerarhije koje dijele istu nadklasu
- d) Generalizacija nije ispravno označena

61. Instance nikad ne bi smjele mijenjati klasu.

10. Točno Netočno

62. Dijagram objekata nastaje iz _____.

11. (dijagrama klasa)

63. Koji dijagram prikazuje instance i veze u vrijeme izvođenja, odnosno opisuje sustav u jednom vremenskom trenutku?

- a) Sekvencijski dijagram
- b) Dijagram klasa
- c) Dijagram objekata
- d) Use case

64. U dijagramu objekata izvedenom iz dijagrama klasa:

- a) Pojavljuje se generalizacija i za nju vrijedi sve isto kao i u dijagramu klasa
- b) Uopće se ne pojavljuje generalizacija
- c) Pojavljuje se i ponaša se kao agregacija u dijagramu klasa
- d) Ništa od navedenog

65. Kad se asocijacija može pretvoriti u agregaciju? (više odg točno)

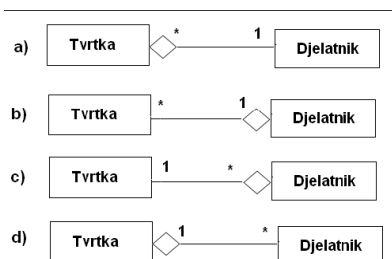
- a) Ako nešto posjeduje ili kontrolira cjelinu
- b) Bilo kad
- c) Ako je cjelina sastavljena od dijelova
- d) Kada postoje alternativne veze

66. Simbol koji predstavlja agregaciju naziva se:

- a) „is a“
- b) isPartOf
- c) nema neki poseban naziv
- d) ne znam

67. Tvrtka se sastoji od više djelatnika, a jedan djelatnik može raditi samo u jednoj tvrtki.

Ispravno je:



68. Za kompoziciju vrijedi:

- a) Dio može postojati i nezavisno od cjeline

- b) Uništavanjem cjeline također se uništavaju i njezini dijelovi
- c) Uništavanje cjeline ne uništava njene dijelove
- d) Podklasa nasljeđuje sva obilježja nadklase

69. Što opisuje dio vidljivog ponašanja skupa objekat. Slično je klasi i nema implementirane metode?

- a) Objekt
- b) Apstraktna klasa
- c) Sučelje
- d) Propagacija

70. Exploratory domain model (Istraživački), System domain model, System model:

- a) UML modeli u procesu razvoja dijagrama klasa koji se razlikuju s obzirom na razinu detalja
- b) UML modeli koji opisuju kategorije odgovornosti
- c) Modeli koji opisuju klase
- d) Ne znam

71. Najmanje detalja ima?

- a) Exploratory domain model (Istraživački)
- b) System domain model
- c) System model
- d) Ništa od navedenog

72. Koji model(i) imam(ju) najviše detalja i uključuje(ju) klase koje se koriste za izgradnju korisničkog sučelja i arhitekture sustava?

- a) System model
- b) Exploratory domain model (Istraživački)
- c) System domain model
- d) System model i Exploratory domain model (Istraživački)

73. Što je točno za System domain model?

- a) Nastaje tijekom analize domene
- b) Prikazuje kompletni model sustava
- c) Može sadržavati manje od pola klasa sustava
- d) Sadrži i Utility klase

74. Što ne spada u preporučeni niz aktivnosti oblikovanja dijagrama klasa?

- a) Pronaći generalizaciju

- b) Navesti glavne odgovornosti za svaku klasu
- c) Odlučiti se na određene operacije
- d) Raspodijeliti zahtjeve svake klase

75. Za otkrivanje domenskih (osnovnih) klasa promatramo:

- a) Pridjeve
- b) Imenice
- c) Glagole
- d) Imenice koje predstavljaju instance

76. Prikazivanje akcija kao veza je:

- a) Dobro, zašto ne...
- b) Dobro, iako postoji bolja solucija
- c) Pogreška
- d) Ovisno o situaciji

77. Imenice koje smo odbacili prilikom traženja osnovnih klasa mogu biti?

- a) Asocijacije
- b) Operacije
- c) Metode
- d) Atributi

78. Za identifikaciju generalizacije postoje:

- a) 3 načina
- b) Samo jedan način
- c) 2 načina
- d) Svaki odnos između klasa ima svoj način

79. Umjesto klasa stvaramo sučelja ako:

- a) Su klase potpuno iste, imaju iste attribute, iste operacije
- b) Jedna ili više klasa već imaju svoj podklasu
- c) Jedna ili više klasa već imaju svoj nadklasu
- d) Uvijek kad se grupiraju slične klase

80. Viša razina apstrakcije od operacije je?

- a) Ne postoji viša od nje
- b) Metoda
- c) Klasa
- d) Odgovornost

81. Glavne operacije koje ostvaruju odgovornost deklarirane su kao?

- a) Private
- b) Public
- c) Friends
- d) Nista od navedenog

Software Architecture and the UML

82. Interakcijski dijagrami su _____ i _____.

12. (sequence i collaboration diagrams)

83. Kad se žele prikazati objekte sustava i njihovo međusobno djelovanje koristimo dijagram?

- a) Use case
- b) Bilo koji od 9 dijagrama
- c) Bilo koji dijagram interakcije
- d) Bilo koji statički dijagram

84. Ako imamo kolaboracijski dijagram (dijagram suradnje) onda iz njega možemo konstruirati drugi dijagram, koji?

- a) use case
- b) dijagram klasa
- c) dijagram objekata
- d) sekvencijski dijagram

85. Što sve spada u savjete o modeliranju interakcija? (više odg točno)

- a) postaviti kontekst za interakcije
- b) izraziti tok slijeva na desno i odozgo prema dolje
- c) stavljaj aktivne instance na lijevu stranu/vrh, a one pasivne na desnu stranu/dno
- d) sve od navedenog

86. Ne pokazuje vrijeme kao posebnu dimenziju, pa slijed interakcija mora biti označen rednim brojevima poruka obilježje je :

- a) Sekvencijskog dijagrama
- b) Bilo kojeg statičkog dijagrama
- c) Kolaboracijskog dijagrama
- d) Dijagrama komponente

87. Svrha kolaboracijskog dijagrama je:

- a) Validacija arhitekture sustava
- b) Prikazuje zahtjeve na sustav
- c) Predstavlja tipične scenarije
- d) Model kontrole toka

88. Za razliku od kolaboracijskog dijagrama naglasak na izmjeni poruka u vremenu (vremenski orijentiran) obilježje je:

- a) Dijagrama komponente
- b) Dijagrama objekata
- c) Sekvencijskog dijagrama
- d) Bilo kojeg dinamičkog dijagrama

89. Stroj stanja koji prilikom prijelaza iz jednog stanja u drugo generira izlaze je

_____.

13. (Mealy automation)

90. Prošireni stroj stanja nastaje dodavanjem _____ i akcije nad njima.

- a) Novih stanja
- b) Podstanja
- c) Podaktivnosti
- d) Varijabli

91. U dijagramu stanja  predstavlja?

- a) Početno stanje
- b) Konačno stanje
- c) Akciju
- d) Aktivnost

92. Koje su vrste događaja u stroju stanja?

- a) Interakcije (sinkrono pozivanje operacija i asinkroni odziv signala)
- b) Vremenske pojave i promjene vrijednosti entiteta
- c) Interakcije (sinkrono pozivanje operacija i asinkroni odziv signala) i promjene vrijednosti entiteta
- d) Pod a i b

93. Izlaz iz automata stanja posljedica je:

- a) Trenutnog stanja
- b) Trenutnog stanja i stanja u kojem se nalazio

- c) Samo stanja u kojem se nalazio
- d) Ne znam

94. U stanju stroja razlikujemo _____ i _____ uvjetno grananje.

14. (statičko i dinamičko)

95. Regije komuniciraju putem:

- a) Konačnog stanja
- b) Zajedničkih varijabli
- c) Objekta
- d) Ništa od navedenog

96. Sustav je modeliran kao mreža automata stanja koji međusobno komuniciraju u dijagramu:

- a) Stanja
- b) Klasa
- c) Objekata
- d) Ništa od navedenog

97. Dijagram stanja je:

- a) Interakcijski dijagram
- b) Statički dijagram
- c) Dinamički dijagram
- d) Dinamički i interakcijski dijagram


98. Dijagram stanja opisuje:

- a) redoslijed aktivnosti koje mogu biti slijedne ili paralelne
- b) opisuje komponente sustava i njihovu ovisnost
- c) opisuje fizičke veze softvera i hardvera u isporučenom sustavu
- d) mogući slijed djelovanja i stanja kroz koje element može proći za vrijeme svojeg postojanja kao rezultat reakcije na događaj

99. Zaokruži točnu tvrdnju:

- a) U dijagramu stanja akcija je pridružena stanju i može se prekinuti
- b) U dijagramu stanja akcija je pridružena stanju i ne može se prekinuti
- c) U dijagramu stanja akcija je vezana uz prijelaz stanja i ne može se prekinuti (pojavljuje se brzo)
- d) U dijagramu stanja akcija je vezana uz prijelaz stanja i može se prekinuti (pojavljuje se sporo)

100. Zaokruži točnu tvrdnju:
- a) U dijagramu stanja aktivnost je pridružena stanju i može se prekinuti jer traje dulje
 - b) U dijagramu stanja aktivnost je vezana uz prijelaz stanja i ne može se prekinuti (pojavljuje se brzo)
 - c) U dijagramu stanja aktivnost je vezana uz prijelaz stanja i ne može se prekinuti iako traje dulje
 - d) U dijagramu stanja aktivnost je isto što i akcija
101. U dijagramu stanja 1) _____ uzrokuju prijelaz iz jednog stanja u drugo i 2) _____ uvijek promijeniti stanje objekta.
- a) 1) akcije, 2) moraju
 - b) 1) aktivnosti, 2) ne moraju
 - c) 1) događaji, 2) moraju
 - d) 1) događaji, 2) ne moraju
102. Ovako se: događaj (argumenti) [uvjet]/akcija opisuje:
- a) Prijelaz
 - b) Događaj
 - c) Stanje
 - d) Aktivnost
103. Za dijagram aktivnosti vrijedi:
- a) Prikazuju organizaciju i ovisnost između komponenata
 - b) Opisuje redosljed aktivnosti koje mogu biti slijedne ili paralelne
 - c) Opisuje mogući slijed djelovanja i stanja kroz koje element može proći za vrijeme svojeg postojanja kao rezultat reakcije na događaj
 - d) opisuje fizičke veze softvera i hardvera u isporučenom sustavu
104. Što nije točno za dijagram aktivnosti:
- a) Namijenjen aplikacijama koje koriste model upravljačkog toka ili pak toka objekata/podataka
 - b) Svi grafovi aktivnosti su metode za operacije
 - c) Svaki slijedeći korak započinje se na vanjski poticaj
 - d) Koristi se za modeliranje poslovnog procesa
105. Podaktivnost :
- a) Ne započinje novi graf aktivnosti, npr. pozivanje operacije nad nekim objektom
 - b) Započine novi graf aktivnosti bez pozivanja operacije

- c) Započine novi graf aktivnosti pozivanjem operacije
 - d) Ništa od navedenog
106. Pozvani aktivacijski graf u dijagramu aktivnosti mogu biti korišteni za _____ stanje(a) podaktivnosti:
- a) Više
 - b) Jedno
 - c) Samo za ono koje koje poziva aktivacijski graf
 - d) Ne znam
107. Aplikacija je kompletno OO kada:
- a) Samo jedno stanje aktivnosti poziva operacije
 - b) Sva stanja aktivnosti pozivaju operacije
 - c) Sva stanja aktivnosti ne pozivaju operacije
 - d) Postoje stanja koje ne pozivaju operacije
108. Posebno stanje u dijagramu aktivnosti koje ističe raspoloživost određene vrste objekata u određenom trenutku zove se:
- a) Akcija
 - b) Aktivnost
 - c) Podaktivnost
 - d) Tok objekata (object flow)
109. Isprekidane linije kod toka objekata:
- a) Razlikuju se od prijelaza u ostalim dijagramima
 - b) Imaju isto značenje kao i prijelaz u ostalim dijagramima
 - c) Služe za povezivanje komentara
 - d) Predstavljaju uvjetno spajanje
110. U dijagramu aktivnosti  predstavlja:
- a) Konačno stanje
 - b) Početno stanje
 - c) Točku odluke i spajanja
 - d) Sinkronizaciju
111. U dijagramu aktivnosti predstavlja 1) _____, a u dijagramu stanja 2) _____.

- a) 1)pseudostanje dinamičkog izbora, 2) sinkronizaciju
 - b) 1)pseudostanje statičkog izbora, 2) sinkronizaciju
 - c) 1) sinkronizaciju, 2) pseudostanje dinamičkog izbora
 - d) 1) sinkronizaciju, 2) pseudostanje statičkog izbora
112. Sinkronizacija stanja u dijagramu aktivnosti:
- a) Pruža mogućnost komunikacije između paralelnih procesa
 - b) Da slijedi spajanje
 - c) Pokazuje tko je odgovoran za pojedinu aktivnost u procesu
 - d) Prikazuje statičnu točku u dijagramu aktivnosti
113. Ako se nalazimo u fazi gdje smo više koncentrirani na aktivnosti, a ne na objekte koji su odgovorni za njih, konstruirat ćemo:
- a) Dijagram stanja
 - b) Kolaboracijski dijagram
 - c) Dijagram aktivnosti
 - d) Sekvencijski dijagram

OBJEKTNO USMJERENA ARHITEKTURA – 2

114. Sustav u kojem se proračunavanja izvode u zasebnim programmima koji rade na odvojenim djelovima arhitekture i surađuju kako bi izvršili zadatak sustava naziva se:
- a) Prošireni sustav
 - b) Raspodijeljeni sustav
 - c) Usmjereni sustav
 - d) Ne znam
115. Program koji pruža uslugu drugim programima koju su spojeni komunikacijskim kanalom naziva se _____.

15. (poslužitelj)

116. Poslužitelj može biti istovremeno pozvan od _____klijenta(ata).
- a) Točno jednog
 - b) Nula ili jednog
 - c) Više
 - d) Najviše dva
117. P2P mreža može također značiti:
- a) Računalno inženjerstvo
 - b) Komunikacija od točke do točke u mreži (Point to Point)

- c) Suradivačko računarstvo (Collaborative Computing)
 - d) Ne znam
118. Metoda oblikovanja arhitekture klijent-poslužitelj temeljena je na _____
16. (ponovnoj i višestrukoj uporabi komponenata)
119. Nakon što se pokrene, poslužitelj:
- a) Šalje zahtjev klijentima da se spoje
 - b) Ako niti jedan klijent ne čeka da se spoji, poslužitelj se automatski isključuje
 - c) Čeka da se spoji klijent (osluškuje)
 - d) Šalje e-mail klijentima
120. Što ne spada u alternative klijent-poslužitelj arhitekture:
- a) Jedan program na jednom računalu koji izvodi sve operacije
 - b) Nema komunikacije (svako računalo obavlja posao zasebno)
 - c) Prisluškavanje klijent-poslužitelj komunikacije
 - d) Postoji drugi mehanizam za razmjenu informacija (npr. jedan program piše u bazu podataka; drugi čita iz baze podataka)
121. Što nije prednost klijent-poslužitelj sustava?
- a) Svi podaci mogu se držati na poslužitelju
 - b) Klijent i poslužitelj mogu biti dizajnirani zasebno
 - c) Poslužitelju istovremeno može pristupati više klijenata
 - d) Uslugama poslužitelja jedino mogu pristupati klijenti koji se nalaze na malim udaljenostima od poslužitelja
122. Što je od navedenog zajedničko aktivnosti klijenta i aktivnosti poslužitelja?
- a) Prihvatanje povezivanja
 - b) Oslušivanje
 - c) Prihvata povezivanje
 - d) Mora „jasno“ završiti (cleanly)
123. Za mršavog klijenta vrijedi:
- a) Što je moguće više posla dodjeljuje se klijentima
 - b) Većinu posla obavlja poslužitelj
 - c) Klijent je što je moguće manji
 - d) Pod a) i c)
124. Dva jezika oblikovana od strane klijenta i poslužitelja zajedno sa pravilima komunikacije naziva se:

- a) Komunikacijska dretva
 - b) Radni okvir (framework)
 - c) Protokol
 - d) Router
125. Vrste ponovne uporabe (reuse) su:
- a) Ponovna uporaba: vještina, standarnog dizajna i algoritama
 - b) Ponovna uporaba: kompletnih aplikacija, biblioteke klasa ili procedura, snažnih naredbi ugrađenih u jezike i operacijski sustav
 - c) Ponovna uporaba: zajedničkog jezika oblikovanog od strane klijenta i poslužitelja, pravila komunikacije
 - d) Pod a i b
126. Softver koji implementira općenito rješenje za generalizirani problem naziva se:
- a) Protokol
 - b) Uvođenje među-sloja (middleware)
 - c) Uslužno usmjereni program
 - d) Radni okvir (framework)
127. U objektno orijentiranoj paradigmi radni okvir (framework) sastoji se od 1) _____, a API je definiran skupom 2) _____.
- a) 1) biblioteke klasa, 2) javnih (public) metoda tih klasa
 - b) 1) biblioteke samo apstraktnih klasa, 2) javnih (public) metoda tih klasa
 - c) 1) biblioteke samo apstraktnih klasa, 2) zajedničkih javnih (public) metoda tih klasa
 - d) 1) biblioteke klasa, 2) varijabli tih klasa
128. Skup proizvoda izgrađenih na zajedničkoj temeljnoj tehnologiji su _____.

17. (proizvodne linije)

129. Postoje dva tipa radnog okvira (framework) i to su _____ i _____.

18. (horizontalni i vertikalni)

130. Što je OCSF?
- a) Object Client-Server Framework
 - b) Object Client-Server Functionality
 - c) Object Client-Server Foundation
 - d) Our County's Science Fair

131. Softverski inženjeri koristeći OCSF _____ njegove tri klase.
- a) Uvijek modificiraju
 - b) Nikad ne modificiraju
 - c) Brišu
 - d) Ponekad modificiraju
132. Softverski inženjeri koristeći OCSF osim što nikad ne modificiraju njegove tri klase, oni:
- a) Stvaraju podklase apstraktnih klasa u radnom okviru i nanovo definiraju određene metode kopči
 - b) Stvaraju nove klase i njihove podklase
 - c) Pozivaju javne metode osigurane radnim okvirom
 - d) Pod a i c
133. Bilo koje podklase klase AbstractClient moraju omogućiti implementaciju za metodu:
- a) openConnection
 - b) getHost
 - c) connectionException
 - d) handleMessageFromServer
134. Kako se zovu funkcije koje se nikad ne pozivaju direktno, ali su pozvane od nekog drugog?
- a) Controlling functions
 - b) Accessing functions
 - c) Call functions
 - d) Callback functions
135. Koja je callback metoda klase AbstractClient?
- a) closeConnection
 - b) getPort
 - c) handleMessageFromServer
 - d) sendToServer
136. Metode klase AbstractClient koje se mogu nanovo definirati (overridden) su:
- a) connectionEstablished
 - b) connectionClosed
 - c) connectionException

- d) sve navedene
137. Koja metoda klase AbstractClient je callback, mora se nanovo definirati (overridden) da bi se mogla implementirati ?
- a) getPort
 - b) handleMessageFromServer
 - c) setHost
 - d) isConnected
138. Privatna metoda AbstractClient unutar koje se nalazi petlja koja čeka poruke od poslužitelja je:
- a) Get
 - b) Set
 - c) Run
 - d) Start
139. Na strani poslužitelja nalaze se dvije klase, koje?
- a) AbstractServer i ConnectionToClient
 - b) serverSocket i getClientConnections
 - c) AbstractServer i getClientConnections
 - d) serverStarted i ConnectionToClient
140. Koja od navedenih nije callback metoda klase AbstractServer ?
- a) clientException
 - b) serverStarted
 - c) serverClosed
 - d) getInfo
141. Koji su rizici usvajanja klijent-poslužitelj pristupa?
- a) Sigurnost i visoki zahtjevi
 - b) Potreba za prilagodljivim održavanjem i dokumentacija
 - c) Potreba za prilagodljivim održavanjem i visoki zahtjevi
 - d) Sigurnost i potreba za prilagodljivim održavanjem
142. Klijenti i poslužitelji su organizirani u _____.

19. (slojeve/razine (tiers))

143. Unutar trofazinske klijent-poslužitelj arhitekture imamo tri razine, koje?
- a) Korisnička, kontrolna i upravljačka
 - b) Korisnička, kontrolna i logička

- c) Korisnička , logička i podatkovna
 - d) Korisnička, kontrolna i podatkovna
144. Zašto trebamo middleware?
- a) Lakše je oblikovati i razvijati raspodijeljene primijenjene sustave
 - b) Preuzima detalje mreže
 - c) Omogućuje osobama koji razvijaju sustav da se fokusiraju na primjenski dio
 - d) Sve točno
145. Svaka komponenta je dodijeljena točno jednoj razini i ona:
- a) Može koristiti usluge bilo koje komponente niže razine
 - b) Može koristiti usluge bilo koje komponente više razine
 - c) Ne može koristiti usluge komponente niže razine
 - d) Ništa od navedenog
146. Standard za implementaciju middlewarea je _____.

20. (COBRA)

147. Zadaća brokera je?
- a) Prosljeđivati zahtjeve
 - b) Implementirati "middleware" u objektno usmjerenom stilu arhitekture
 - c) Podijeliti zahtjeve
 - d) Ne znam
148. COBRA je kratica za _____.

21. (CORBA - Common Object Request Broker Architecture)

149. _____ arhitektura organizira aplikacije kao skup usluga koje
22. komuniciraju koristeći dobro definirana sučelja.

23. (Uslužno usmjerena)

150. Što predstavlja problem objektnog pristupa?
- a) Sigurnost
 - b) Upravljanje sa više objekata
 - c) Potreba za prilagodljivim održavanjem
 - d) Ništa od navedenog