

PITANJA (I ODGOVORI) S PRVOG BLICA:

1. *U svijetu velikih promjena računalnog sklopolja, računalnih mreža, timskog programiranja, složenosti primjenskih programa (aplikacija) i razvojnih alata, što se najmanje promijenilo?*
(Odgovor u najviše pet riječi)

Odgovor: **apstrakcija u programiranju**

2. *Inženjerski model je: Potpuna reprezentacija sustava gledana iz određene perspektive.*

Odgovor: **Netočno**

Točan odgovor: Apstraktna reprezentacija procesa, predstavlja opis procesa iz određene perspektive

3. *U oblikovanju programske potpore model postaje sustav koji se modelira.*

Odgovor: **Točno**

4. *Formalna verifikacija (FV) je postupak provjere da li izvedeni sustav logički zadovoljava specifikaciju.*

Odgovor: **Netočno**

Točan odgovor: Postupak provjere da li formalni model dijela izведенog sustava (I) odgovara formalnoj specifikaciji (S) sa matematičkom izvjesnošću

5. *Moderan postupak oblikovanja programske potpore razlikuje se od tradicijskog po uvođenju specifikacije temeljem analize zahtjeva na sustav.*

Odgovor: **Netočno**

Točan odgovor:

6. *Teorije u okviru računarske znanosti daju potpunu podlogu programskom inženjerstvu.*

Odgovor: **Netočno**

7. *Dionici u oblikovanju i korištenju programske potpore su: kupac, korisnik, osoba koja razvija i oblikuje programski produkt, rukovoditelj razvoja. Kupcu je najvažnije: Lagano učenje i prihvatanje novog sustava, efikasnost u uporabi, pomaže mu da se posao obavi.*

Odgovor: **Netočno**

Točan odgovor: Kupcu je važno rješavanje problema uz prihvatljivu cijenu, a ovo gore je važno korisniku

8. *Dokumenti koji specificiraju zahtjeve na sustav dijele se prema razini detalja na funkcionalne, nefunkcionalne i zahtjeve domene primjene.*

Odgovor: **Netočno**

Točan odgovor: Prema razini detalja se dijele na Korisničke, Zahtjeve sustava i Specifikaciju programske potpore
Ovo gore je podjela prema sadržaju

9. *Temeljem analize zahtjeva detaljna specifikacija programske potpore radi se za osobe koje su zadužene za pisanje ugovora.*

Odgovor: **Netočno**

Točan odgovor: Detaljna specifikacija PP se radi za Klijente (inženjere), Specijaliste za oblikovanje sustava (arhitekti) i za specijaliste za razvoj PP

10. *Funkcionalni zahtjevi su: Ograničenja u uslugama i funkcijama, kao što su vremenska ograničenja, (ne)usvojeni standardi, ograničenja u procesu razvoja i oblikovanja, itd.*

Odgovor: **Netočno**

Točan odgovor: Funkcionalni zahtjevi su izjave o uslugama koje sustav mora pružati, kako će sustav reagirati na određeni ulazni poticaj, te kao bi se sustav trebao ponašati u određenim situacijama. Ovo gore su nefunkcionalni zahtjevi

11. *Na kojoj se ideji (metodi) temelji izlučivanje i specifikacija zahtjeva UML obrascima upotrebe?*
(Odgovor jedna riječ)

Odgovor: **Scenariju**

12. *Jedine temeljne skupine elemenata u dijagramima obrazaca upotrebe su aktori i obrasci uporabe.*

Odgovor: **Netočno**

Točan odgovor: Ima ih 3, to su: Obrasci uporabe (use cases), aktori i odnosi (relations) među njima

13. *Jedan od najvažnijih modela procesa programskog inženjerstva je iteracijski.*

Odgovor: **Netočno**

Točan odgovor: ?

14. *RUP proces programskog inženjerstva pridružuje poglede pojedinim aktivnostima.*

Odgovor: **Točno**

15. *Ispitivanje (testiranje) sustava je adekvatan način da se pokaže odsustvo pogrešaka.*

Odgovor: **Netočno**

Vježba za 1. blic - pitanja s moodla

1. Za tehnike simulacije i ispitivanja (engl. testing) programa u procesu dokazivanja ispravnost programa vrijede tvrdnje:

- a. Ne može dokazati ni postojanje ni nepostojanje pogreške.
- b. Može dokazati postojanje i nepostojanje pogreške.
- c. Ne može dokazati postojanje, ali može dokazati nepostojanje pogreške.
- d. Može dokazati postojanje, ali ne može dokazati nepostojanje pogreške.**

2. Za sustave za formalnu verifikaciju vrijede tvrdnje:

- a. garantira izvan svake sumnje ispravnu relaciju modela i specifikacije**
- b. mora postojati izgrađen model sustava**
- c. osigurava ispravnu relaciju implementacije i specifikacije
- d. mora postojati definirana formalna specifikacija sustava**
- e. postupak se provodi na izgrađenom sustavu

3. Kao dioniku u ocjeni kakvoće programskog produkta kupcu je najvažnije:

- a. Lakoća ponovne uporabe dijelova
- b. Lakoća održavanja**
- c. Lakoća oblikovanja
- d. Nijedno od navedenog
- e. Sve navedeno

4. Programskom potporom nazivamo:

- a. razvojnju okolinu
- b. program i sklopošku dokumentaciju
- c. program, pripadajuću dokumentaciju, korisničke upute**
- d. skup naredbi i procedura
- e. skup povezanih programa

5. Koja temeljna značajka(e) je svojstvena modernom načinu oblikovanja programske potpore?

- a. evaluacija sustava
- b. sve navedeno
- c. uvođenje analize zahtjeva
- d. uvođenje modela sustava**
- e. specifikacija sustava

6. Etnografija je:

- a. metoda strukturiranja formalnih poslovnih procesa u svrhu izlučivanja zahtjeva
- b. pregled utjecaja svih dionika na zahtjeve sustava
- c. Zahtjevi izvedeni temeljem istraživanja kako ljudi stvarno rade**
- d. metoda brze izrade prototipa
- e. Kvalitativno promatranje i opis ponašanja ljudi pri obavljanju poslova

7. Ako svi zahtjevi obuhvaćeni u dokumentu inženjerstva zahtjeva imaju samo jednu moguću interpretaciju tada možemo tvrditi da je dokument specifikacije zahtjeva:

a. jednoznačan

- b. moguće verificirati
- c. sve navedeno
- d. konzistentan
- e. Ispravan

8. Koje od navedenih aktivnosti spadaju u osnovne aktivnosti inženjerstva zahtjeva?

a. Sve navedeno

- b. Izlučivanje i analiza zahtjeva
- c. Validacija zahtjeva
- d. Studija izvedivosti
- e. Specifikacija zahtjeva

9. Odredite istinite tvrdnje svojstava modela procesa programskog inženjerstva:

- a. Evolucijski model generira najbolju strukturu.

b. Najmanje novog kôda generira se u modelu zasnovanom na komponentama.

- c. Implementacija promjena je najlakša u vodopadnom modelu.
- d. Iteracije su primjenjive samo na RUP model.
- e. RUP (engl. Rational Unified Process) je najstariji model.

10. Opseg rezultirajućih dokumenata inžinjertvu zahtjeva ovisi:

a. opsegu problema

- b. ugovorenoj cijeni
- c. trajanju projekta
- d. zahtjevima programske podrške
- e. uporebljenim metodama

11. Tipične metode u izlučivanju i analizi zahtjeva su:

a. specificiranje zahtjeva obrascima uporabe

b. Izrada scenarija

c. Intervjuiranje

d. Specificiranje dinamičkih interakcija u sustavu

- e. obavezno analizira doprinose sustava postavljenim ciljevima

12. Primarna uloga dokument specifikacije zahtjeva obuhvaća:

- a. osigurava tehničku specifikaciju za razvojni tim

b. definiranje sustava razumljivo korisniku

c. osigurava komunikaciju s klijentima i korisnicima

d. osigurava razumljiv opis zahtjeva

- e. služi za komunikaciju s razvojnim timom

13. U inžinjertvu zahtjeva programske potpore za zahtjeve vrijede tvrdnje:

- a. mogu se izražavati prirodnim jezikom**
- b. to su utvrđena ograničenja implementirane programske podrške
- c. to su ograničenja programske podrške**
- d. izražavaju se isključivo posebnim jezicima za oblikovanje sustava, dijagramima i matematičkom notacijom
- e. to su opisi usluga sustava**
- f. to su opisi ograničenja sklopoljia**

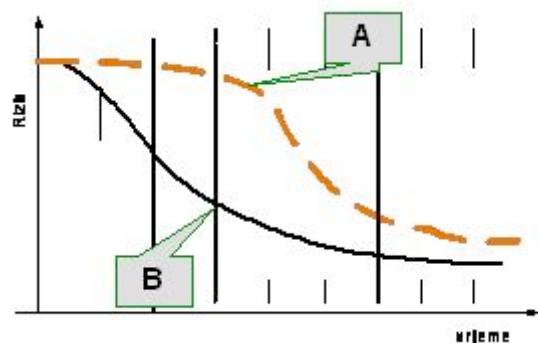
14. Analiza rizika u spiralnom modelu programskog inžinjerstva provodi se:

- a. samo na završnom modelu
- b. nije potrebna
- c. u svakoj petlji**
- d. u prvoj petlji
- e. prije početka modeliranja

15. Što nije sektor u spiralnom modelu iteracija programskog inženjerstva?

- a. Planiranje
- b. Procjena i smanjivanje rizika
- c. Razvoj i validacija
- d. Pridjeljivanje zahtjeva inkrementima**
- e. Postavljanje ciljeva

16. Na slici je prikazana procjena rizika razvoja programske potpore tijekom vremena temeljena na vodopadnom i inkrementalnom modelu.



- a. Krivulja B prikazuje inkrementalni model.**
- b. Krivulja A prikazuje inkrementalni model.

17. Što je potrebno uključiti u dokument oblikovanja arhitekture programske potpore?

Odaberite jedan ili više odgovora:

- a. Prioritete koji su vodili proces oblikovanja**
- b. Alternativna rješenja
- c. Referenciju prema dokumentu zahtjeva**
- d. Opis s najviše razine promatranja**
- e. Dobro komentirani kod

18. U drugu fazu unificiranog procesa (engl. Unified Process) (Faza elaboracije) ne spada:

- a. Oblikovanje produkta.
- b. Definiranje dosega projekta.**
- c. Izrada plana projekta.
- d. Definiranje temelja arhitekture sustava.
- e. Specifikacija značajki.

19. Oznaka 0 ..* na liniji pridruživanja pri ulazu u obrazac uporabe znači:

Odaberite jedan odgovor:

- a. Nula
- b. Nijedan
- c. Nula ili više**
- d. Više ali ne nula
- e. Nula ili jedan

20. Za dobro definirane UML obrasce uporabe vrijedi:

Odaberite jedan ili više odgovora:

- a. Opisuju sve detalje aktivnosti potrebnih za davanje odgovora aktoru.
- b. Izgradnja završava opisom svih slučajeva potrebnih aktoru.**
- c. Pišu se s gledišta aktora.**
- d. Izrada započinje zahtjevom od sustava prema aktoru.

21. Stvarima ponašanja (engl. behavioral things) odgovara kao UML opis:

- a. oznake elemenata modela
- b. paketi
- c. razredi, sučelja, slučajevi korištenja, komponentni čvorovi, aktori i sl.
- d. interakcije i automati stanja**

22. Vrsta poveznice u dijagramu obrazaca uporabe s prikazom:



naziva se:

- a. uključivanje
- b. pridruživanje/asocijacija
- c. generalizacija**
- d. proširenje

23. Vrsta poveznice u dijagramu obrazaca uporabe s prikazom:



naziva se:

- a. generalizacija
- b. proširenje
- c. uključivanje
- d. asocijacija**

24. U sekvencijskom dijagramu UML sustava padajuća linija poruke predstavlja:

- a. Asinkronu poruku.
- b. Sinkronu poruku.
- c. Poruku koja ima vremensko trajanje.**
- d. Poruku koja očekuje povratnu informaciju.
- e. Poruku koja ne očekuje povratnu informaciju.

25. UML sekvencijski dijagrami NISU pogodni za:

- a. pojašnjenje pojedinog obrasca uporabe (engl. use case)
- b. prikazivanje komunikacije aktora
- c. prikaz složene proceduralne logike**
- d. prikaz vremenskog uređenja među događajima
- e. prikazivanje objekata koji međusobno komuniciraju

26. Za pojam krize razvoja programske potpore vrijedi:

- termin smisljen 1968. Na nato konferenciji
- nije moguće prevazići poboljšanjem performansi pridruženog sklopolja
- manifestira se kašnjnjem isporuke programske potpore i nezadovljavanjem korisničkih zahtjeva
- uzročnici su povezani sa soženošću skopovalja i razvojnih procesa

27. Ako program u svom radu ne zadovoljava korisničke zahtjeve tu manifestaciju nazivamo:

O: Nedostatak

28. Uporabom se programska potpora kvari jer:

O: Pogreške se uočavaju tek kod intezivne i dugotrajne uporabe

29. Proces inženjerstva zahtjeva ne uključuje:

O: Validaciju prohrama potpore

30. U načine dokumentiranja zahtjeva sustava koji se koriste u praksi ne spada:

O: UML dijagram razreda

31. Što vrijedi za spiralni model razvoja programske potpore

O: Barem jedan dio specifikacija mora biti definiran prije njihovog kodiranja

32. Prednosti uporabe vodopadnog modela nad "ad hoc" modelom su:

- definirani prijelazi između strogo definiranih faza
- dobrodefinirana slijedna organizacija procesa
- nema iteracije

33. Proces razvoja programske potpore osigurava:

- potrebne informacije
- pravovremnost potrebnih informacija
- upotrebljiv/efikasan oblik informacija
- definira tko radi što i kada

34. Za funkcionalnu dekompoziciju problema u UML obrascima uporabe upotrebljavaju se poveznice vrste:

O: Uključivanja

35. Za UML obrasce uporabe NE vrijedi:

O: Specificiraju način implementacije sustava

36. Stvarima grupiranja (en. Grouping things) odgovara kao UML opis:

O: Paketi

37. Vrsta poveznice u dijagramu obrazaca uporabe s namjenom prikaza da osnovni obrazac sadrži ponašanje definirano u drugom obrascu naziva se:

O: Generalizacija

38. U procesu inženjerstva zahtjeva faza analize zahtjeva rezultira:

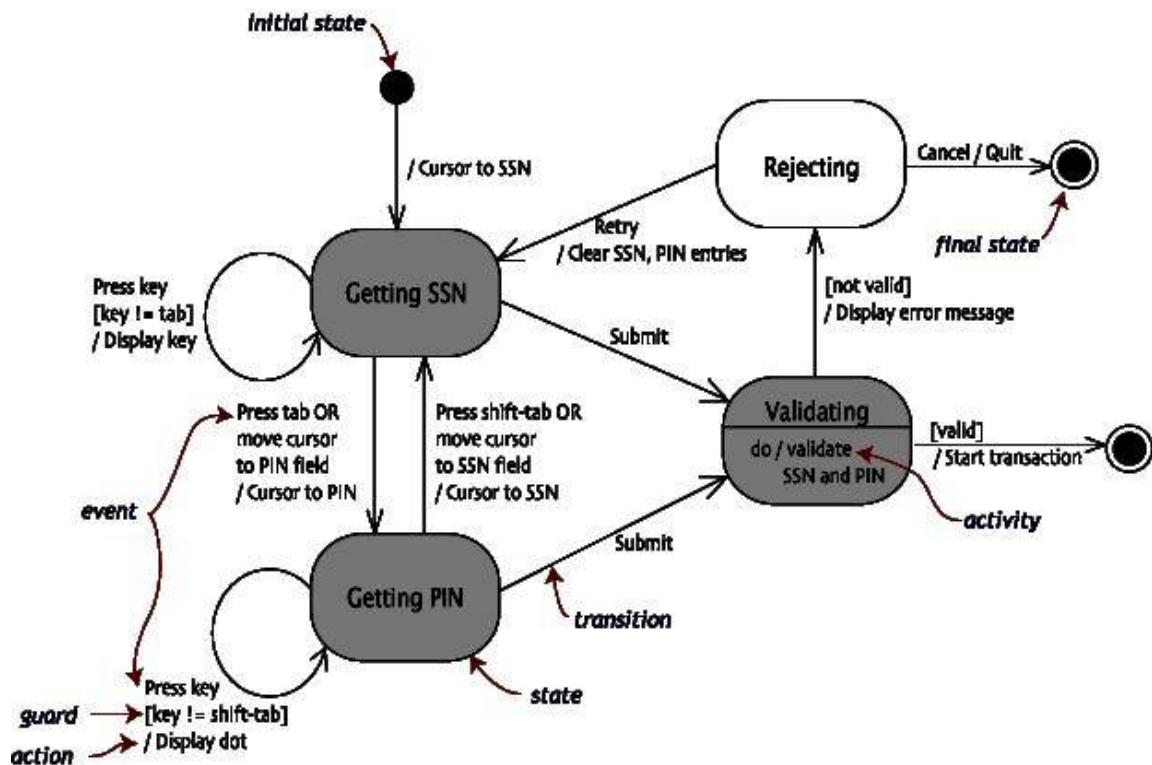
O: Funkcionalni i ponašajni model

OPP - Samoprovjera znanja 3

Na pitanja odgovorio Telefunken.

1. UML dijagrami koji se primjenjuju za opis modela toka upravljanja (engl. *control flow*) ili toka podataka su:
 - a. Sekvencijski dijagrami
 - b. Dijagrami stanja
 - c. Komunikacijski dijagrami
 - d. **Dijagrami aktivnosti**
2. UML dijagrami koji se temelje na prikazu fizičkih modularnih i zamjenjivih jedinki kojima se može pristupati samo kroz sučelja su:
 - a. **Dijagrami komponenti**
 - b. Dijagrami razreda
 - c. Dijagrami paketa
 - d. Dijagrami objekata
3. Za objekt u UML dijagramu objekata vrijede tvrdnje:
 - a. **Simbol objekta je pravokutnik s dva pretinca.**
 - b. Objekti imaju definiciju metoda.
 - c. Simbol objekta je pravokutnik s tri pretinca.
 - d. **Objekti imaju vrijednosti atributa.**
 - e. Objekti nemaju definiciju atributa.
4. Moguće razine vidljivosti unutar UML dijagrama paketa su:
 - a. **javno (engl. *public*)**
 - b. paket (engl. *package*)
 - c. **privatno (engl. *private*)**
 - d. **zaštićeno (engl. *protected*)**
5. Topologiju sustava te odnose njegovih sklopoških i programske dijelova prikazuju UML dijagrami:
 - a. komponenti
 - b. komunikacije
 - c. paketa
 - d. objekata
 - e. razmještaja

6. Prikazani UML dijagram je:



- a. dijagram razreda
- b. dijagram obrazaca uporabe
- c. sekvencijski dijagram
- d. dijagram stanja**
- e. dijagram objekata
- f. dijagram aktivnosti

7. U skupinu UML dijagrama međudjelovanja (engl. interaction diagrams) spadaju:

- a. komunikacijski dijagram**
- b. sekvencijski dijagram**
- c. dijagram obrazaca uporabe
- d. dijagram stanja
- e. dijagram aktivnosti

8. Klijent-poslužitelj arhitektura ima nekoliko dobrih strana. Koje od slijedećih NISU dobre strane ove arhitekture (koje izjave nisu točne)?
- a. konzistencija podataka
 - b. raspodjela posla na više računala
 - c. jednostavno upravljanje sigurnošću podataka
 - d. različiti poslužitelji
 - e. lakoća programiranja takvih sustava (ne samo za profesionalce)
 - f. mogućnost odvojenog oblikovanja klijenta i poslužitelja
 - g. različiti klijenti
 - h. jednostavnost sinkronizacije

9. Za razred ConnectionToClient vrijedi tvrdnja da je za komunikaciju sa svim klijentima potrebna jedna jedina instanca tog razreda.

Točno

Netočno

10. U SOA (*Software as a Service*) arhitekturi aplikacija se organizira kao kolekcija usluga koje međusobno komuniciraju uporabom dobro definiranih javnih sučelja.

11. Za kombinacijsko ispitivanje vrijede tvrdnje:

- a. na broj ispitnih slučajeva ne utječe broj grananja u programskom kodu
- b. broj ispitnih slučajeva ne ovisi o strukturi podataka
- c. Pogodno za ispitivanje međutjecaja varijabli
- d. Otkriva nedokumentirane odzive

12. Uporabom se programska potpora kvari jer:

- a. postaje nekompatibilno s okolinom
- b. sklopolje na kojem se izvodi podložno je starenju
- c. pogreške se uočavaju tek kod intezivne i dugotrajne uporabe
- d. rezervni dijelovi zastarijevaju
- e. krajnji korisnici se mijenju
- f. je izložena teškim radnim uvjetima

13. Programskom potporom nazivamo:

- a. razvojnu okolinu
- b. skup povezanih programa
- c. **program, pripadajuću dokumetaciju, korisničke upute**
- d. program i sklopošku dokumentaciju
- e. skup naredbi i procedura

14. Pri ispitivanju programske potpore upotrebljavamo termin ispitni slučaj. Odredi ispravne tvrdnje.

- a. bilježimo status/uspješnost provođenja ispitivanja
- b. prije provođenja ispitivanja moraju biti definirani ulazni podaci i očekivani izlaz
- c. nakon provođenja ispitni slučajevi se ne ponavljaju
- d. na očekivane rezultate ispitnih slučajeva utječe opis stanja prije izvođenja
- e. ispitne slučajeve jedinstveno obilježavamo

15. Za metriku pokrivanja stupnja gotovosti ispitivanja upotrebljavamo:

Odaberite jedan ili više odgovora:

- a. **Pokrivenost putova**
- b. Vrijeme izvođenja
- c. **Pokrivenost grana**
- d. Broj ispitnih slučajeva
- e. **Postotak programskih elemenata koji su izvedeni**
- f. **Pokrivenost linija koda**

16. Koji tipovi UML dijagrama prikazuju i jasno razlikuju aktivnosti (engl. *activity*) i akcije (engl. *action*) sudionika?

Odaberite jedan ili više odgovora:

- a. Komunikacijski dijagram (engl. communication diagram)
- b. Sekvencijski dijagram (engl. sequence diagram)
- c. Dijagram obrazaca uporabe (engl. use-case diagram)
- d. **Dijagram stanja (engl. statechart diagram)**
- e. **Dijagram aktivnosti (engl. activity diagram)**

17. Artefakti dijagrama komponenata mogu biti:

- Odaberite jedan ili više odgovora:
- a. sklopovske komponente
 - b. izvorni kodovi**
 - c. poslužitelji
 - d. izvršne datoteke
 - e. statičke i dinamičke biblioteke

18. Prednosti uporabe arhitektura s n razina su:

- Odaberite jedan ili više odgovora:
- a. Lako je odrediti optimalno preslikavanje odgovornosti na razine.
 - b. Olakšana potpora zahtjevima povećanja i poboljšanja sustava.
 - c. Postizanje poboljšanja performansi izričitim podržavanjem definicije n -razinske arhitekture pri oblikovanju.
 - d. Ograničeni utjecaj promjena jedne razine na cijeli sustav.**
 - e. Pojednostavljeni oblikovanje temeljem viših razina apstrakcije.
 - f. Dobra potpora ponovnom korištenju (engl. *reuse*), prenosivost i sl.

19. Koje metode korisnik OCSF radne klupe mora obavezno implementirati za klijentsku stranu?

- Odaberite jedan ili više odgovora:
- a. closeConnection()
 - b. handleMessageFromServer()**
 - c. sendToServer()
 - d. isConnected()
 - e. openConnection()

AbstractClient
«control»
openConnection()
sendToServer()
closeConnection()
«hook»
connectionEstablished()
connectionClosed()
connectionException()
«slot»
handleMessageFromServer()
«accessors»
isConnected()
getPort()
setPort()
getHost()
setHost()
getInetAddress()

20. Za tankog klijenta (*thin client*) vrijedi tvrdnja da se izvršni kod preuzima preko mreže.

Odaberite jedan odgovor:

Točno

Netočno

21. UML dijagrami koji se temelje na pojedincima nekog dijagrama razreda se nazivaju:

Odaberite jedan odgovor:

- a. Dijagrami komponenti (engl. component diagram)
- b. Dijagrami objekata (engl. object diagram)**
- c. Dijagrami paketa (engl. package diagram)
- d. Dijagrami razreda (engl. class diagram)

22. Koje su osnovne komponente svakog komunikacijskog protokola?

Odaberite jedan odgovor:

- a. Dva jezika i pravila konverzacije**
- b. Jeden jezik i pravila konverzacije
- c. Pravila konverzacije

23. Od navedenih koji razred(i) pripada(ju) poslužiteljskoj strani?

Odaberite jedan ili više odgovora:

- a. AbstractServer**
- b. AbstractClient
- c. ConnectionToClient

24. UML dijagram koji prikazuje uređen redoslijed razmjene poruka između sudionika bez izrazitog prikaza preciznijih vremenskih odnosa je:

Odaberite jedan odgovor:

- a. Komunikacijski dijagram (engl. communication diagram)
- b. Objektni dijagram (engl. object diagram)
- c. Dijagram aktivnosti (engl. activity diagram)
- d. Sekvencijski dijagram (engl. sequence diagram)**

25. Kada želimo napraviti liniju proizvoda koja će se imati zajedničku osnovnu funkcionalnost, a razlikovati se u pojedinim mogućnostima koristiti ćemo mehanizam koji (u OO arhitekturi) nazivamo:

- Odaberite jedan odgovor:
- a. Statični UML dijagrami
 - b. Scenarij
 - c. Radni okvir
 - d. Predefinirana arhitektura sustava

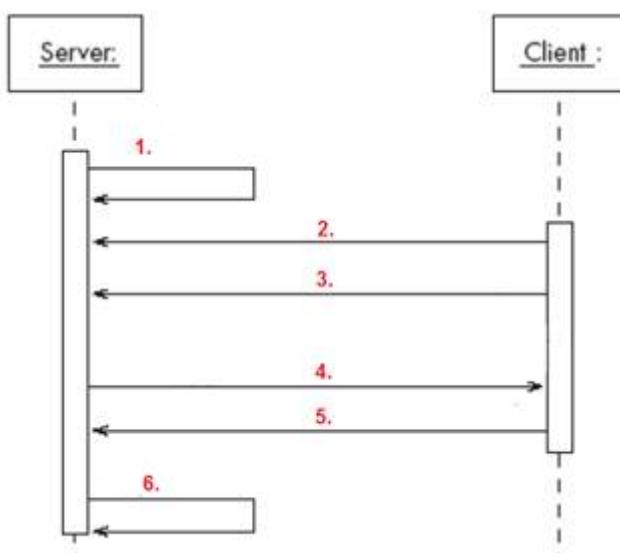
26. Koja od navedenih NISU pravila uporabe OCSF-a (*moguće više točnih odgovora*)?

Odaberite jedan ili više odgovora:

- a. Redefinirati metode u apstraktnim razredima.
- b. Kreirati podrazrede.
- c. Modificirati apstraktne razrede tako da se dobije puna funkcionalnost.
- d. Konkretizirati metode u podrazredima.

27. Na kojoj se web tehnologiji baziraju standardni komunikacijski protokoli u SOA-i? XML

28. Na donjoj slici prikazan je jednostavan sekvenčni dijagram koji prikazuje komunikaciju u arhitekturi klijent - poslužitelj. Povežite redni broj aktivnosti prikazan na strelicama s nekim od ponuđenih opisa aktivnosti.



- Aktivnost 1. Početak rada i slusanje
- Aktivnost 2. Spajanje
- Aktivnost 3. Slanje poruke (zahtjeva)
- Aktivnost 4. Slanje odgovora
- Aktivnost 5. Odspajanje
- Aktivnost 6. Prestanak slusanja

UVOD

1. Što je nedostatak tradicijskog postupka oblikovanja sustava?

- a) Zahtjevi nisu formalizirani
- b) Granični slučajevi se sve teže identificiraju
- c) Dokumentacija se zbog neodređenosti teško interpretira
- d) Točan odgovor pod a i c

2. Ispravnost se provjerava simulacijama i testiranjem koje:

- a) Mogu samo dokazati postojanje pogreške ali nikako njeno nepostojanje
- b) Daju absolutnu sigurnost u ispravnost sustava
- c) mogu se primjeniti samo na automate s konačnim brojem stanja
- d) ništa od navedenog

3. Nedostaci tradicijskog ispitivanja (testiranja) sustava su:

- a) Cijena testiranja čini najveći dio cijene produkta
- b) Pokrivenost skupa testiranih stanja se smanjuje
- c) Ljudski um nije u mogućnosti predvidjeti sve mogude interakcije
- d) Sve točno

4. U principu savladavanja problema složenosti spada:

- a) Trajna interakcija s okolinom
- b) Razvoj i validacija
- c) Uvođenje kordinacije u tim
- d) Uvođenje modela

5. Validacija je ?

- a) jesmo li ispravno napravili sustav?
- b) zadovoljava li sustav zahtjeve na ispravan način?
- c) jesmo li napravili ispravan sustav?
- d) Kako sustav radi?

6. Temelji formalne verifikacije (FV) sustava su:

- a) Specifikacija i implementacija
- b) Specifikacija i formalizacija
- c) Validacija i implementacija
- d) Apstrakcija i infrastruktura

7. Formalna verifikacija je postupak provjere da _____, odgovara formalnoj specifikaciji sa matematičkom izvjesnošću.

- a) Validirani model izvedenog sustava
- b) Poboljšani model izvedenog sustava
- c) Formalni model izvedenog sustava
- d) Analiza zahtjeva

8. Specifikacije nam govori:

- a) Da li sustav zadovoljava funkcionalne zahtjeve
- b) Kako sustav radi
- c) Da li je sustav ispravan
- d) Što sustav treba raditi

9. Da bi najlakše specificirali „što želimo“ koristimo:

- a) Formalan opis
- b) Grafički opis
- c) Gotove komponente
- d) Algoritamsku verifikaciju

10. Postupci zasnovani na dokazivanju teorema:

- a) Primjenjuju se na sustave s konačnim brojem stanja
- b) Primjenjuju se kod izračuna svih dosezljivih stanja
- c) Primjenjuju se na sustave koji imaju više od 10500 stanja
- d) Primjenjuju se na sustave s beskonačnim brojem stanja

11. Točna tvrdnja vezana uz formalnu verifikaciju je :

- a) FV daje absolutnu sigurnost u ispravnost sustava (programa /sklopolja)
- b) FV ne osigurava ispravnu relaciju između stvarne implementacije i specifikacije
- c) FV ne garantira ispravnu relaciju između formalnih (matematičkih) objekata I i S
- d) FV grantira izvan svake sumnje samo ispravnu relaciju između stvarne implementacije i specifikacije

12. Koja tvrdnja nije točna ?

- a) Uvesti inženjerski propisane postupke (procedure) u proces oblikovanja programske potpore
- b) Svaku fazu procesa dokumentirati (na standardan i formalan način)
- c) Programske potpore oblikovati u manjim cijelinama (komponentama)
- d) Ipak su sve točne

UVOD U PROGRAMSKO INŽENJERSTVO

13. Za oblikovanju programske potpore moramo koristiti:

- a) Društveni pristup
- b) Inženjerski pristup
- c) Slobodan pristup
- d) Brzi pristup

14. Karakteristika programske potpore je:

- a) Lagano je unijeti izmjene
- b) Teško ju je reproducirati
- c) Ne kvari se
- d) Nijedno od navedenog

15. Programska potpora:

- a) Bavi se teorijama, metodama i alatima za profesionalno oblikovanje računalnih programa
- b) To je računalni program i pridružena dokumentacija
- c) Ma to je skup aktivnosti čiji cilj je razvoj ili evolucija programskega produkta
- d) To je zapravo produkcija programskega produkta

16. Cijena _____ dominira u cijeni cjelokupnog sustava.

- a) Producije
- b) Ispitivanja sustava
- c) Programske potpore

- d) Nijedno od navedenog
17. Cijena održavanja programske potpore je _____ od cijene razvoja, oblikovanja i ispitivanja programskog produkta.
- a) Zanemariva
 - b) manja
 - c) jednaka ili veda
 - d) Nijedno od navedenog
18. Disciplina koja se bavi metodama i alatima za profesionalno oblikovanje i produkciju programske potpore uzimajući u obzir cjenovnu efikasnost je:
- a) Inženjerstvo sustava
 - b) Računarska znanost
 - c) Programsко inženjerstvo
 - d) Metoda Programskog inženjerstva
19. Inženjerstvo sustava se bavi:
- a) Teorijom i temeljima
 - b) Svim aspektima sustava zasnovanim na računalima
 - c) Praktičnim problemima razvoja
 - d) Razvojem programske infrastrukture
20. Programski produkti mogu se kreirati :
- a) Oblikovanjem novih programa
 - b) Konfiguracijom postojećih generičkih programa
 - c) Ponovnom uporabom postojećih programa
 - d) Sve točno
21. Kod programske potpore (programskog produkta):
- a) Vrijede zakoni fizike
 - b) Ne vrijede zakoni fizike
 - c) Vrijedi Benfordov zakon
 - d) Nijedno od navedenog
22. Za proces programskog inženjerstva bitan je :
- a) Timski rad
 - b) Razvoj i validacija
 - c) Opis modela
 - d) Trošak oblikovanja
23. Vodopadni, interaktivni, komponentni... su:
- a) Generičke aktivnosti
 - b) Generički modeli
 - c) Metode
 - d) Pravila
24. Generičke aktivnosti u svim procesima programskog inženjerstva su:
- a) Specifikacija , razvoj i oblikovanje , evolucija
 - b) Specifikacija, validacija i verifikacija , oblikovanje i implementacija

- c) Tijek podataka , razvoj i oblikovanje, evolucija
- d) Specifikacija , razvoj i oblikovanje, validacija i verifikacija , evolucija**

25. Rukovanje promjenama sukladno novim zahtjevima karakterizira:

- a) Validaciju
- b) Specifikaciju
- c) Evoluciju**
- d) CASE

26. Što je model procesa programskog inženjerstva?

- a) To je pojednostavljeni predstavljanje procesa iz određene perspektive**
- b) Sustav koji podupiru određenu metodu razvoja programske potpore
- c) To je struktturni pristup razvoju i oblikovanju programske potpore
- d) To su ograničenja primijenjena na modele sustava

27. Programsко inženjerstvo je proces rješavanja problema:

- a) Programera i rukovoditelja razvoja
- b) Kupca i programera
- c) Kupca i korisnika**
- d) Programera i korisnika

28. Koja generička aktivnost provjerava da li sustav čini samo ono što se od njega zahtjeva?

- a) Specifikacija
- b) Validacija i verifikacija**
- c) Evolucija
- d) Nijedna od navedenih

29. Razdioba troška ovisi o:

- a) modelu procesa programskog inženjerstva**
- b) procesu programskog inženjerstva
- c) okolini
- d) zahtjevima

30. Što je CASE (Computer-Aided Software Engineering) ?

- a) Programski produkti namijenjeni automatiziranoj podršci aktivnostima u procesu programskog inženjerstva**
- b) To je struktturni pristup razvoju i oblikovanju programske potpore, što uključuje modele sustava, notaciju, pravila, preporuke i naputke
- c) Pa to su sustavi za koje je karakteristično da ne podupiru određenu metodu razvoja programskog produkta
- d) Nijedno od navedenog

31. Viši CASE :

- a) Podupiru rane aktivnosti**
- b) Podupiru kasnije aktivnosti
- c) Isto što i niži CASE
- d) Ne postoje niži i viši CASE

32. Dobar programski produkt osim što mora osigurati traženu funkcionalnost i performanse mora paziti na:
- a) Pouzdanost, isplativost, vrijeme isporuke
 - b) Samo na pouzdanost
 - c) Pouzdanost, održavanje, pravila
 - d) Pouzdanost, održavanje, prihvatljivost**

33. Veliku vedinu projekata čine:

- a) Sasvim novi projekti
- b) Evolucijski i oni u svezi s održavanjem starijih sustava**
- c) Ma svi su podjednako zastupljeni
- d) Nijedno od ponuđenog

PROCESI PROGRAMSKOG INŽENJERSTVA

34. U kojem generičkom modelu imamo odvojene i specifične faze specifikacije i razvoja?

- a) Komponentno usmjerenom
- b) Evolucijskom
- c) Vodopadnom**
- d) U RUP-u

35. U evolucijskom modelu dva uobičajena postupka su:

- a) Istraživački razvoj i oblikovanje & metoda odbacivanja prototipa**
- b) Istraživački razvoj i oblikovanje & inkrementalna isporuka sustava
- c) Metoda odbacivanja prototipa & Spriralni razvoj i oblikovanje
- d) Spriralni razvoj i oblikovanje & inkrementalna isporuka sustava

36. Pojedina faza se mora dovršiti prije pokretanja nove faze je:

- a) Temeljna značajka vodopadnog modela**
- b) Temeljna značajka komponentno usmјerenog modela
- c) Temeljna značajka evolucijskog modela
- d) Nijedno od ponuđenog

37. U kojem pristupu započinjemo s grubo definiranim zahtjevima, a cilj je razumijevanje zahtjeva sustava?

- a) Istraživački razvoj i oblikovanje
- b) Metoda odbacivanja prototipa**
- c) Ekstremno programiranje
- d) Inkrementalni pristup

38. Koji su problemi kod evolucijskog modela razvoja i oblikovanja?

- a) Proces razvoja i oblikovanja je jasno vidljiv
- b) Vrlo je teško izvesti implementaciju promjena koje zahtjeva kupac
- c) Sustavi su često vrlo loše strukturirani**
- d) Kod evolucijskog modela razvoja i oblikovanja nema problema

39. Vodopadni model se uglavnom koristi ?

- a) Za male i srednje interaktivne sustave
- b) Za dijelove velikih sustava
- c) Samo za male interaktivne sustave

d) Za velike inženjerske projekte gdje se sustav razvija na nekoliko odvojenih mjestu

40. Što ne spada u stupnjeve procesa modela zasnovanog na komponentama?

- a) Analiza komponenata
- b) Modifikacija zahtjeva
- c) Razvoj i oblikovanje sustava i programske potpore
- d) Oblikovanje sustava s višestrukom uporabom komponenata

41. Iteracije procesa sastavni su dio:

- a) Velikih projekata
- b) Malih projekata
- c) Ne primjenjuju se u projektima
- d) Nijedno od navedenog

42. Kod inkrementalne isporuke sustava zahtjevi višega prioriteta:

- a) Isporučuju se u ranim inkrementima
- b) Isporučuju se u kasnijim inkrementima
- c) Zapravo sustav uopće nije podjeljen na inkrente i isporučuje se korisniku u cjelini
- d) Svejedno je u kojoj fazi ih isporučujemo

43. Ekstremno programiranje je?

- a) Poseban oblik inkrementalnog pristupa
- b) Poseban oblik spiralnog pristupa
- c) Poseban oblik specifikacija programskog produkta
- d) Jeden od načina otklanjanja pogrešaka

44. Spiralni razvoj i oblikovanje je?

- a) Jeden od dva međuvisna pristupa iteracijama
- b) Generički model u kojem su specifikacija, razvoj i validacija isprepleteni
- c) Sastavni dio inkrementalne isporuke sustava
- d) Nijedno od navedenog

45. Koji su sve sektori u spiralnom modelu iteracija?

- a) Postavljanje ciljeva, Procjena i smanjivanje rizika, Validacija i verifikacija, Analiza
- b) Postavljanje ciljeva, Procjena i smanjivanje rizika, Razvoj i validacija, Planiranje
- c) Specifikacija, Razvoj i oblikovanje, Procjena i smanjivanje rizika, Planiranje
- d) Postavljanje ciljeva, Procjena rizika, Oblikovanje i implementacija, Planiranje

46. Temeljni dio specifikacije čini proces inženjerstva zahtjeva koji uključuje:

- a) Studiju izvedivosti, Izlučivanje i analizu zahtjeva
- b) Specifikaciju zahtjeva, Validaciju zahtjeva
- c) Pod a i b
- d) Analizu zahtjeva i definicije, Studiju izvedivosti

47. Što je oblikovanje i implementacija programskog produkta?

- a) To je proces preslikavanja specifikacije u stvarni, realni sustav
- b) Postupak provjere da sustav odgovara specifikaciji i zadovoljava zahtjevima kupca
- c) To je proces određivanja potrebnih usluga i ograničenja u radu i razvoju sustava
- d) Ne postoji takva generička aktivnost u procesu programskog inženjerstva

48. Rezultat oblikovanja komponenata, kao jedne od podaktivnosti u oblikovanju je:

- a) Arhitektura sustava
- b) Specifikacija komponenata**
- c) Specifikacija programskog produkta
- d) Specifikacija sučelja

49. Pogreške se otklanjaju u okviru:

- a) Implementacije – programiranja**
- b) Validacije i verifikacije
- c) Procesa inženjerstva zahtjeva
- d) Jedne od faza ispitivanja sustava

50. Što treba pokazati validacija i verifikacija?

- a) Da su aktivnosti oblikovanja i implementacije povezane i mogu biti isprepletene
- b) Da sustav odgovara specifikaciji i zadovoljava zahtjevima kupca i korisnika**
- c) Koje su sve mogudnosti u oblikovanju sustava
- d) Koliki je rizik neuspjeha projekta

51. Validacija i verifikacija uz provjeru (matematički zasnovanu) i reviziju procesa uključuje i:

- a) Dokumentiranje odabrane arhitekture
- b) Programiranje i otklanjanje pogrešaka
- c) Ispitivanje (testiranje) sustava**
- d) Nijedno od ponuđenog

52. Koliko imamo stupnjeva u ispitivanju sustava ?

- a) 2
- b) 4
- c) 5
- d) 3**

53. Jedan od stupnjeva u ispitivanju sustava je?

- a) Dinamičko ispitivanje
- b) Statičko ispitivanje
- c) Ispitivanje značajki na temelju kojih kupac prihvada i preuzima sustav**
- d) Nijedno od ponuđenih

54. Koja je tvrdnja točna, a vezana uz evoluciju programskog produkta?

- a) Programski produkt se ne može mijenjati
- b) Zahtjevi na sustav se često mijenjaju ali programski produkt uvijek ostaje isti
- c) Kako se mijenjaju zahtjevi na sustav tako se mora mijenjati i programski produkt**
- d) Ako se zahtjevi promjene onda se krede s projektiranjem novog programskog produkta

55. Rational Unified Process (RUP) je:

- a) Model procesa programskog inženjerstva**
- b) Proces programskog inženjerstva
- c) Aktivnost programskog inženjerstva
- d) jezika za modeliranje

56. RUP je izведен na temelju:

- a) UML-a i pridruženih aktivnosti
- b) C++ i pridruženih aktivnosti
- c) Java
- d) Nijedno od ponuđenih

57. Kroz koliko je perspektiva opisan RUP?

- a) 3
- b) 4
- c) 2
- d) 6

58. Praktična perspektiva?

- a) Pokazuje slijed faza kroz vrijeme
- b) Pokazuje aktivnosti procesa
- c) Sugerira aktivnosti kroz iskustvo i dobru praksu
- d) Opisuje funkcionalne i ne-funkcionalne zahtjeve

59. Za specificiranje objektno usmjerene arhitekture programske potpore posebice je prikladan?

- a) UML
- b) Model
- c) Dijagram
- d) Proces

60. Koliko postoji standarnih dijagrama u RUP-u?

- a) 11
- b) 5
- c) 9
- d) 10

61. Dijagrami su definirani

- a) UML standardom
- b) Pomodu obrazaca uporabe
- c) Aktivnostima koje dokumentiraju
- d) Modelima koje dokumentiraju

62. Svakoj/svakom _____ u procesu programskog inženjerstva pridružen(a) je _____ za opis, koji/koje su dokumentirani _____.

- a) Aktivnost, model, dijagram
- b) Model, aktivnost, dijagram
- c) Dijagram, aktivnost, model
- d) Model, dijagram, aktivnost

63. Sve aktivnosti semantički povezuju?

- a) Obrasci uporabe
- b) Modeli
- c) Dijagrami
- d) Arhitektura

64. RUP definira i međusobno povezuje:

- a) Dinamičke faze i statičke modele procesa
- b) Statičke faze i dinamičke aktivnosti procesa
- c) Dinamičke faze i statičke modele procesa
- d) Dinamičke faze i statičke aktivnosti procesa**

65. Što je CASE tehnologija?

- a) To je proces izrade specifikacije programskog produkta
- b) Opisuje proces SE kao ciklus različitih aktivnosti unutar generičkih modela procesa
- c) Definira arhitekturu sustava kroz modele i dijagrame koristedi UML standard
- d) Podupire automatizaciju aktivnosti tijekom procesa programskog inženjerstva**

66. Zašto poboljšanja do kojih je dovela CASE tehnologija nisu sukladna očekivanjima?

- a) Jer CASE tehnologija podupire samo jedan dio procesa programskog inženjerstva
- b) Jer klasifikacija alata prema aktivnostima nije učinkovita
- c) Programsко inženjerstvo zahtjeva kreativnost, koju često nije mogude automatizirati**
- d) Ništa od navedenog

67. Koja tvrdnja nije točna?

- a) Programsko inženjerstvo je timska aktivnost
- b) Prema funkcionalnoj perspektivi alati se klasificiraju prema specifičnoj funkciji
- c) CASE tehnologija dobro podupire timski rad**
- d) Radne klupe podupiru pojedine faze procesa

68. Unutar CASE integracijske perspektive razlikujemo:

- a) Modele, aktivnosti i radne klupe
- b) Aktivnosti, radne klupe i razvojne okoline
- c) Alate , radne klupe i razvojne okoline**
- d) Aktivnosti, alate i razvijene okoline

69. S obzirom na integracijsku perspektivu ANALIZA I OBLIKOVANJE, PROGRAMIRANJE , ISPITIVANJE spadaju u:

- a) Alate
- b) Radne klupe**
- c) Razvojne okoline
- d) Ništa od navedenog

70. Koji je ispravan redoslijed faza u RUP-u?

- a) Početak, Konstrukcija, Elaboracija, Tranzicija
- b) Početak, Tranzicija, Elaboracija, Konstrukcija
- c) Početak, Elaboracija, Konstrukcija, Tranzicija**
- d) Elaboracija, Početak, Konstrukcija, Tranzicija

71. U koju fazu spada oblikovanje sustava, programiranje i ispitivanje?

- a) Početak
- b) Konstrukcija**
- c) Elaboracija
- d) Tranzicija

72. U koju fazu spada oblikovanje razumijevanja domene primjene i određivanje temeljne arhitekture sustava?
- a) Početak
 - b) Konstrukcija
 - c) Elaboracija
 - d) Tranzicija

INŽENJERSTVO ZAHTJEVA U OBLIKOVANJU PROGRAMSKE POTPORE

73. Što je inžinerstvo zahtjeva?
- a) To je postupak pronalaženja, analiziranja, dokumentiranja i provjere zahtijevanih usluga sustava
 - b) To je postupak provjere zahtijevanih usluga, te ograničenja u uporabi
 - c) To je postupak pronalaženja, analiziranja, dokumentiranja i provjere zahtijevanih usluga sustava, te ograničenja u uporabi
 - d) To je definiranje standardnih formulara i obrazaca u kojima se izražavaju zahtjevi

74. S obzirom na razinu detalja razlikujemo?

- a) Specifikacija niske razine, detaljna specifikacija , specifikacija programske potpore
- b) Specifikacija niske razine, detaljna specifikacija i specifikacija programske produkta
- c) Specifikacija visoke razine, detaljna specifikacija i specifikacija programske potpore
- d) Funkcionalni zahtjevi, nefunkcionalni zahtjevi, zahtjevi domene primjene

75. S obzirom na sadržaj zahtjeve dijelimo na?

- a) Kompletni zahtjevi, konzistentni zahtjevi
- b) Zahtjevi programskog produkta, organizacijski zahtjevi, vanjski zahtjevi
- c) Funkcionalni zahtjevi, nefunkcionalni zahtjevi, zahtjevi domene primjene
- d) Ništa od ponđenog

76. Ograničenja u uslugama i funkcijama značajka su:

- a) Funkcionalnih zahtjeva
- b) Zahtjeva domene primjene
- c) Detaljne specifikacije
- d) Nefunkcionalnih zahtjeva

77. Koji zahtjevi definiraju usluge koje sustav osigurava?

- a) Funkcionalni zahtjevi
- b) Zahtjevi domene primjene
- c) Kompletni zahtjevi
- d) Nefunkcionalni zahtjevi

78. Koji zahtjevi sadrže opise svih zahtijevanih mogudnosti?

- a) Funkcionalni zahtjevi
- b) Konzistentni zahtjevi
- c) Kompletni zahtjevi
- d) Nefunkcionalni zahtjevi

79. Koji zahtjevi ne smiju sadržavati konflikte ili kontradikcije u opisima zahtijevanih mogudnosti?

- a) Konzistentni zahtjevi
- b) Funkcionalni zahtjevi

- c) Nefunkcionalni zahtjevi
- d) Kompletni zahtjevi

80. Na koliko se podvrsta klasificiraju nefunkcionalni zahtjevi?

- a) 2
- b) 5
- c) 7
- d) 3**

81. Koji zahtjevi specificiraju da se isporučeni produkt mora ponašati na određeni način?

- a) Vanjski zahtjevi
- b) Organizacijski zahtjevi
- c) Zahtjevi programskog produkta**
- d) Zahtjevi domene primjene

82. Koji su problemi kod zahtjeva domene primjene?

- a) Razumljivost i implicitnost**
- b) Razumljivost i apstrakcija
- c) Procedure i implicitnost
- d) Razumnjivost i eksplisitnost

83. Zahtjevi sustava s obzirom na razinu detalja spadaju u:

- a) Specifikaciju visoke razine
- b) Detaljniju specifikaciju**
- c) Specifikaciju programske potpore
- d) Ništa od ponuđenog

84. Odredi točnu tvrdnju:

- a) Oblikovanje određuje što sustav mora raditi, a zahtjevi određuju kako de se to ostvariti
- b) Oblikovanje određuje kako sustav mora raditi, a zahtjevi određuju kada de se to ostvariti
- c) Zahtjevi određuju što sustav mora raditi, a oblikovanje određuje kako de se to ostvariti**
- d) Zahtjevi određuju što sustav mora raditi, a oblikovanje određuje gdje de se to ostvariti

85. Koji zahtjevi se mogu definirati ili ilustrirati jednim od modela sustava?

- a) Korisnički zahtjevi
- b) Zahtjevi sustava**
- c) Zahtjevi programske potpore
- d) Kompletni zahtjevi

86. Zahtjevi sustava izražavaju se:

- a) Strukturiranim prirodnim jezikom, posebnim jezicima za opis oblikovanja, grafičkom notacijom
- b) Prirodnim jezikom, grafičkim dijagramima
- c) Sve pod a) i matematičkom notacijom**
- d) Sve pod b) i matematičkom notacijom

87. UML spada u:

- a) grafičku notaciju**
- b) matematičku notaciju

- c) poseban jezik za opis oblikovanja
- d) ništa od ponuđenog

88. Koliko tipa sučelja postoji (zahtjevi sustava)?

- a) 5
- b) Vrlo mnogo
- c) Ne zna se
- d) 3**

89. Jednoznačno opisuje sučelje ali traži specijalističko znanje?

- a) Specifikacija sučelja
- b) Formalna notacija**
- c) Matematička notacija
- d) Dokument zahtjeva

90. Usklađen skup izjava o svim zahtjevima na sustav je ?

- a) IEEE standard
- b) Dokument zahtjeva programskog produkta**
- c) Korisnički i zahtjevi sustava
- d) Detaljna specifikacija

91. Tko definira način pisanja dokumenata zahtjeva?

- a) Korisnik
- b) Rukovoditelj za pisanje ugovora
- c) IEEE standard**
- d) Nema definiranog načina

1. Koja je razlika između modernog oblikovanja programske potpore u odnosu na tradicijski način?

- a) uvođenje analize zahtjeva
- b) specifikacija sustava
- c) evolucija sustava
- d) uvođenje i analiza modela sustava**
- e) sve gore navedeno

2. Prof. Dijkstra tvrdi da provjera sustava simuliranjem i testiranjem:

- a) identificira rubne slučajeve
- b) dokazuje moguće postojanje greške**
- c) dokazuje moguće nepostojanje greške
- d) smanjuje vrijeme izlaska proizvoda na tržište
- e) sve gore navedeno

3. Validacija programske potpore je:

- a) provjera da sustav zadovoljava zahtjeve na ispravan način
- b) provjera da formalni model zadovoljava specifikaciju
- c) provjera da model zadovoljava specifikaciju s matematičkom izvjesnošću
- d) provjera da program zadovoljava funkcione zahtjeve**

4. U modelu programskog inženjerstva temeljenog na komponentama postoji aktivnost „analiza komponenti“. Odaberi istinitu tvrdnju.

- a) Ulazni podaci za tu aktivnost su isključivo zahtjevi za programsku potporu.
 - b) Izlazni podaci te aktivnosti se proslijeđuju u sljedeću aktivnost „oblikovanje sustava s ponovnim korištenjem“.
 - c) Ulazni podaci te aktivnosti su isključivo specifikacija iz knjižnice komponenti.
 - d) Izlazni podaci te aktivnosti se proslijeđuju u aktivnost „razvoj i integracija“.
- e) Izlazni podaci te aktivnosti se proslijeđuju u aktivnost „modifikacija zahtjeva“.**

5. U okviru generičke aktivnosti „procesa programskog inženjerstva“ koja se naziva „oblikovanje i implementacija“, dio „oblikovanje programskog produkta“, postoji niz podaktivnosti i produkata tih podaktivnosti. Što su ulazne podaktivnosti za podaktivnost „oblikovanje algoritama“?

- a) Specifikacija programske potpore

b) Specifikacija strukture podataka

- c) Specifikacija arhitekture
- d) Specifikacija komponenti

6. RUP se razlikuje od generičkih procesa programskog inženjerstva jer:

- a) definira aktivnosti u procesu programskog inženjerstva
- b) definira faze životnog ciklusa programske potpore

c) uvodi iteraciju

- d) definira miljokaze
- e) povezuje temeljne aktivnosti s modelima, a modele s dijagramima

7. U kojoj fazi RUP-a je analiza zahtjeva najintenzivnija?

elaboraciji

8. Pogledi interakcije rade se iz perspektive dionika sustava:

- a) koji ne koriste sustav izravno, ali utječu na zahtjeve

b) koje izravno koriste sustav

- c) koji definiraju karakteristike domene i ograničenja i primjene
- d) a, b, c
- e) a, b

9. Za sekvenčni dijagram ne vrijedi ili e:

- a) petlje se crtaju samo nad područjem aktivnosti
- b) svi naknadno stvoreni objekti na istom dijagramu moraju biti uništeni
- c) sinkroni pozivi (poruke) ne moraju imati povratnu strelicu
- d) asinkroni pozivi mogu imati povratnu strelicu

e) sve prethodno je istina

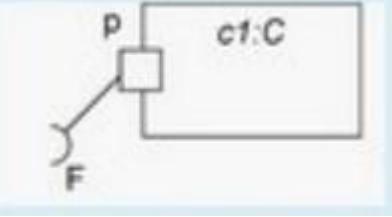
10. Što je etnografija?

imate definiciju na kraju 4. predavanja

11. Zajedničko obrascima uporabe i sekvenčnim dijagramima može biti:

- a) vremenska linija
- b) <<include>> obrasci
- c) <<extend>> obrasci
- d) neke poruke

e) akteri

PITANJE	ODOGOVOR
Za tehnike simulacije i ispitivanja programa u procesu dokazivanja ispravnosti vrijedi:	Može se dokazati postojanje, ali ne i nepostojanje pogreške
Koje su generičke aktivnosti u svakom procesu inženjerstva zahtjeva?	Specifikacija Razvoj i oblikovanje Validacija i verifikacija Evolucija
U iteracije spiralnog modela procesa inženjerstva zahtjeva ne pripadaju:	Izrada prototipa Studija izvedivosti
U načine dokumentiranja zahtjeva sustava koji se koriste u praksi NE spada:	UML dijagram razreda
Slika 	Objekt c1 je tipa komponenta, port p zahtjeva svojstva definirana sučeljem F
bla grupiranja (engl. grouping things) odgovara kao UML opis:	Paketi
U sekveničkom dijagramu UML kosa ili sl. padajuća linija poruke predstavlja:	Poruku koja ima vremensko trajanje
Kod oblikovanja kojih zahtjeva su bla UML obrasci uporabe:	funkcionalnih zahtjeva
Odnos između A i B (<< extend >>) 	Obrazac A proširuje osnovni način uporabe obrasca B
Što je potrebno uključiti u okument oblikovanja arhitekture programske potpore?	Prioritete koji su vodili proces oblikovanja Referenciju prema dokumentu zahtjeva Opis s najviše razine promatranja
Pristup zasnovan na razvoju oblikovanju i isporuci vrlo malih inkremenata funkcionalnosti svojstven je kojoj metodologiji razvoja programske potpore?	Agilnoj
Koji je model procesa programskog inženjerstva najpogodniji za male i srednje interaktivne sustave ?	Evolucijski
Koje od navedenih tvrdnji vrijede za spiralni model razvoja programske podrške ?	Sektori u spirali slijede vodopadni model Smanjuje se rizik razvoja Brza izrada prototipa
U drugu fazu unificiranog procesa (Faza elaboracije) ne spada:	Oblikovanje produkta Definiranje dosega projekta
Koja od navedenih je temeljna značajka vodopadnog modela?	Pojedina faza se mora dovršiti prije pokretanja nove faze
Što znači „oblikuj po ugovoru“	Sve metode imaju ugovor s pozivateljem

Koja je temeljna značajka svojstvena modernom načinu oblikovanja PP-e (PP = programska potpora nadalje)	Uvođenje modela sustava
Princip oblikovanja kojemu jecilj osigurati sakrivanje ili odgodu razmatranja detalja te time i smanjenje složenosti naziva se	Zadrži razinu apstrakcije
Razred može implementirati više sučelja:	Točno
Kod preopterećivanja (overloading) metode imaju _____ naziv , a _____ broj i tip parametara	istog , različitog
Metode razreda deklariraju se unutar razreda ključno riječi _____ i pozivaju se preko _____	static , imena razreda
Moguće je kreirati instance apstraktog razreda ali je tada potreban modifikator abstract:	Netočno
Kako u objektno usmjerenoj terminologiji nazivamo predložak (specifikaciju) koja definira vrstu objekta ?	Razred (Class)
Konstruktore upotrebljavamo za:	Inicijalizaciju novih objekata
_____ varijable određuje koje objekte ta varijabla može referencirati	tip ? PLS točan odg na FER2 postajte ako ovo nije
U procesu oblikovanja PP arhitektura programa rezultat je _____ - level , a implementacija _____ - level dizajna	visoki, niski PLS točan odgovor na FER2 postajte ako ovo nije
Koje su dvije osobine zajedničke svim objektima u OO?	Stanje i ponašanje
Koje od navedenih tvrdnji vrijede za odnos agregacije u UML dijagramu:	agregat predstavlja IS-PART-OF odnos, agregacija je slabija od kompozicije
Osnovu opisa funkcionalnih zahtjeva u dokumentaciji projekta čine:	UML dijagrami obrazaca uporabe UML sekvensijski dijagrami
Naredba Subversion sustava kojom se promjene lokalnog repozitorija prenose na središnji repozitorij: 90 % sam siguran da bi ovo moglo doći – novo pitanje	checkin
Koje su temeljne značajke dobrog programskog produkta ?	Mora se moći idržavati Mora biti prihvatljiv korisniku Mora biti pouzdan
Zahtjevi sustava mogu se izraziti:	Sve od navedenog (Strukturirnim..., Specijalnim..., Matematičkom..., Grafičkom...)
U iteracije spiralnog modela procesa inženjerstva zahtjeva ne pripadaju:	Studija izvedivosti Izrada protipa
Etnografija je:	zahtjevi izvedeni temeljem istraživanja kako ljudi stvarno rade
Osnovni elementi UML-a su:	stvari, relacije, objekti
Strukturne stvari u jeziku UML služe za opis:	razredi, sučelja, slučajevi korištenja, komponentni čvorovi, aktori i sl. Lakše vam je ovo zapamtiti kao najduži pdgovor haha

Za opis upravljačkog toka UML sekveničkim dijagramom vrijede tvrdnje:	Moguće je opisati uvjetno grananje i petlje
Kod oblikovanja kojih zahtjeva se koriste UML obrasci uporabe?	Funkcionalnih zahtjeva
Koji od navedenih tipa poveznica NE postoji u dijagramu obrazaca uporabe ?	Kompozicija (composition)
Princip višestrukog korištenja u oblikovanju PP koristi se u :	SVE ponuđeno je točno ponovnom korištenju fragmenata rješenja, konzistencije, dijelova arhitekture, arhitekture, cjelokupne arhitekture sustava
Koje se tri aktivnosti ponavljaju tijekom razvoja u evolucijskom modelu?	Specifikacija, razvoj, validacija
Što nije sektor u spiralnom modelu iteracija PI-a?	Pridjeljivanje zahtjeva inkrementima
Koja je temeljna značajka vodopadnog modela?	Pojedina faza se mora dovršiti prije pokretanja nove faze
Obrasci uporabe specificiraju _____, a arhitektura _____ izvedbu	PLS ODGOVOR NA FER2 postajte, ne znam ovo
Princip oblikovanja koji se temelji na obradi svih slučajeva u kojima se neka komponenta može neprikladno upotrijebiti naziva se:	Konzervativno oblikovanje
Princip oblikovanja povećanja ponovne uporabivnosti u pristupu oblikovanju je komplementaran principu	Povećaj uporabu postojećeg
Koji od navedenih pojmova NE predstavlja neki od osnovnih principa objektne paradigne	Nasljeđivanje (možda je ona napredan haha)
Svojstvo da različiti razredi imaju funkcije istog imena nazivamo: sad je tu slika koda koji je nebitan ionako	Polimorfizam
Što se događa tijekom izvođenja OO aplikacije?	Dio modela objekata se nalazi u memoriji i dinamički se stvaraju/brišu
Proces kojim jedan objekt može steći svojstva nekog drugog objekta nazivamo	Nasljeđivanje
Odnos kompozicije i agregacije u UML dijagramu razreda:	Kompozicija je jača od agregacije
UML dijagrami koji nisu nužni u inačici 1.0 su	Dijagram komponenti Dijagram Aktivnosti Dijagram razmještaja
Neki od alata za upravljanje inačicama datoteka unutar projekta su:	GIT, TFS(Team Foundation Server)
Kako se zove test u OOP kod provjere pravilnog nasljeđivanja	IS-A

MOJA PITANJA (vrlo jerojatna pogotovo iz SVN-a)	ODGOVORI
Što je Apache Subversion	Sustav za kontrolu inačica PP pod Apache licencom
Kako se zove test u OOP kod provjere pravilnog nasljeđivanja	„IS-A“
Koja operacija mijenja lokalnu radnu kopiju kako bi bila identična stanju središnjeg repozitorija	Update
Koja operacija promjene nastale u lokalnoj radnoj kopiji izvozi u središnji repozitorij	Commit
Prilikom stvaranja lokalne radne kopije (prvo dohvaćanje sadržaja središnjeg repozitorija) koristi se operacija:	Checkout
Koja operacija rješava sukob odbacivanjem vlastitih promjena	Revert
Operacija kojem se odabire koja će se inačica zadržati kod sukoba	Resolve
Apache SVN podržava isključivo:	komandno-linijski način rada
Tortoise SVN ima integrirano:	grafičko sučelje
Model u kojem ne postoji jedinstveni središnji repozitorij s kojim se klijenti sinkroniziraju, već svaka radna kopija repozitorija predstavlja repozitorij za sebe	Raspodijeljeni model
GIT podrazumijeva rad s dva repozitorija:	lokalnim i udaljenim

1. Za pojam krize razvoja programske potpore vrijedi:

Odaberite jedan ili više odgovora:

- proizlazi iz problema operacijskih sustava
- djelomično se rješava poboljšanjem performansi pridruženog sklopolja
- **nije moguće prevladati poboljšanjem performansi pridruženog sklopolja**
- **termin smislen 1968. godine na NATO konferenciji**
- proizlazi iz činjenice da samo trećina studentskih projekata programske potpore daje djelomičan rezultat
- **uzročnici su povezani sa složenošću sklopolja i razvojnih procesa**
- **manifestira se kašnjenjem isporuke programske potpore i nezadovoljavanjem korisničkih zahtjeva**
- stvarna kriza programske potpore ne postoji svi mogu raditi programe

2.U inženjerstvu zahtjeva programske potpore za zahtjeve vrijede tvrdnje:

Odaberite jedan ili više odgovora:

- **to su ograničenja programske potpore**
- **mogu se izražavati prirodnim jezikom**
- **to su opisi usluga sustava**
- izražavaju se isključivo posebnim jezicima za oblikovanje sustava, dijagramima i matematičkom notacijom
- to su utvrđena ograničenja implementirane programske potpore
- **to su opisi ograničenja sklopolja**

3.Odredite istinite tvrdnje za svojstva modela procesa programskog inženjerstva:

Odaberite jedan ili više odgovora:

- RUP (engl. Rational Unified Process) je najstariji model.
- Iteracije su primjenjive samo na RUP model.
- Implementacija promjena je najlakša u vodopadnom modelu.
- **Najmanje novog kôda generira se u modelu zasnovanom na komponentama.**
- Evolucijski model generira najbolju strukturu.

4.Ako svi zahtjevi obuhvaćeni u dokumentu inženjerstva zahtjeva imaju samo jednu moguću interpretaciju tada možemo tvrditi da je dokument specifikacije zahtjeva:

- **jednoznačan**

5.U koje skupine UML dijagrama spadaju dijagrami obrazaca uporabe i sekvencijski dijagrami?

- **i jedni i drugi su ponašajni**

6. Stvarima grupiranja (engl. grouping things) odgovara kao UML opis:

- **paketi**

7.Za sekvencijske dijagrame vrijede slijedeće tvrdnje (moguće više točnih odgovora):

Odaberite jedan ili više odgovora:

- prikazuju objekte kao životne crte
- prikazuju interakciju u sustavu
- dobri su za prikaz komunikacije među objektima
- dobri su za praćenje stanja sustava
- dobri su za grafički prikaz složenih algoritama

8.Koji je najbolji opis svrhe UML dijagrama obrazaca uporabe?

- **to je pogled na sustav izvana**

9.Vrsta poveznice u dijagramu obrazaca uporabe s prikazom (extend) naziva se:

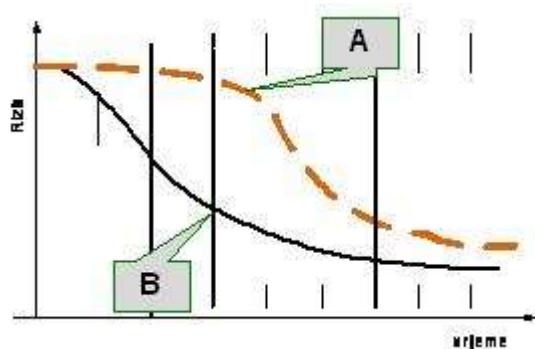
- **proširenje**

10. Što je potrebno uključiti u dokument oblikovanja arhitekture programske potpore?

Odaberite jedan ili više odgovora:

- **Opis s najviše razine promatranja**
- **Referenciju prema dokumentu zahtjeva**
- **Prioritete koji su vodili proces oblikovanja**
- Alternativna rješenja
- Dobro komentirani kod

11. Na slici je prikazana procjena rizika razvoja programske potpore tijekom vremena temeljena na vodopadnom i inkrementalnom modelu.



Odaberite točnu tvrdnju:

- **Krivulja B prikazuje inkrementalni model.**

12. Loša struktura sustava je najčešće nedostatak kojeg modela procesa programskog inženjerstva?

- **evolucijskog**

13.Što nije sektor u spiralnom modelu iteracija programskog inženjerstva?

- **Pridjeljivanje zahtjeva inkrementima**

14.U drugu fazu unificiranog procesa (engl. Unified Process) (Faza elaboracije) ne spada:

Odaberite jedan ili više odgovora:

- Izrada plana projekta.
- **Definiranje dosega projekta.**
- Specifikacija značajki.
- Definiranje temelja arhitekture sustava.
- **Oblikovanje produkta.**

15.Prednosti uporabe vodopadnog modela nad "ad hoc" modelom su:

Odaberite jedan ili više odgovora:

- nestrukturirani proces
- brze iteracije
- složena organizacija s jednostavnom podrškom
- **definirani prijelazi između strogo definiranih faza**

dobro definirana slijedna organizacija procesa

16.Koja temeljna značajka je svojstvena modernom načinu oblikovanja programske potpore?

- **uvodenje modela sustava**

17.Ako za neki sustav vrijedi da promjene na jedom mjestu zahtijevaju i promjene drugdje može se reći za imma veliku:

- **međuovisnost**

18.Princip oblikovanja povećanja ponovne uporabivosti u pristupu oblikovanju je komplementaran principu:

- **Povećaj uporabu postojećeg**

20.Liskov princip zamjene kaže da ako se u varijablu tipa **superrazreda** stavi objekt tipa **podrazreda**

program se mora korektno izvoditi

21. Svojstvo da različiti razredi imaju funkcije istog imena nazivamo:

Npr.

```
class A {  
    int a;  
public:  
    A() { a = 0; };  
    void X(){ a++; };  
    void Y(){ a--; };  
};  
class B: public A {  
public:
```

```
void X(){ a = a + 2; };
void Y(){ a = a - 2; };
};
```

Odaberite jedan odgovor:

- **polimorfizam**

21. Metode razreda deklariraju se unutar razreda ključnom riječi **static** i pozivaju se preko **imena razreda**.

22. Svaki razred posjeduje karakteristična svojstva. Obilježja razreda predstavljena su **atributima**, a ponašanje **metodama**.

23. Kada se jedan razred temelji na drugom razredu, on ujedno nasljeđuje:

- **Podatke i metode**

24. Objekti se dinamički stvaraju u dijelu radne memorije:

- **gomila (engl. heap)**

25. Koji od ponuđenih odgovora prikazuje ispravan redoslijed povijesnog razvoja modularizacije?

Odaberite jedan odgovor:

- oblikovanje zasnovano na komponentama-> glavni program i potprogrami -> objektno usmjerena arhitektura
- **glavni program i potprogrami -> objektno usmjerena arhitektura -> oblikovanje zasnovano na komponentama**
- glavni program i potprogrami -> oblikovanje zasnovano na komponentama-> objektno usmjerena arhitektura

26. Koji od ponuđenih odgovora predstavlja ispravan način instanciranja i pristupa varijabli razreda tipa MojRazred?

Odaberite jedan odgovor:

- i a i b je ispravno;
- **tmp = MojRazred.svojstvo1; //tmp definiran ranije u kodu**
- MojRazred mojaVarijabla = new MojRazred();
tmp = mojaVarijabla.svojstvo1; //tmp definiran ranije u kodu

27. Ako dva objekta imaju identične podatke (attribute) zapravo se radi o jednom te istom objektu koji je dvaput referenciran.

- **Svaki objekt je jedinstven, čak i ako ima sve attribute potpune jednakih vrijednosti kao i neki drugi objekt u sustavu.**

Točan odgovor je 'Ne'.

28.Kompozicija u UML dijagramu razreda znači da kad se uništi kompozit uništavaju se i svi njegovi dijelovi.

- **Točan odgovor je 'Da'.**

29.Zahtjevi kvalitete, zahtjevi platforme ostvarenja i zahtjevi procesa čine:

- **nefunkcionalne zahtjeve**

30.Naredba Subversion sustava kojom se promjene lokalnog repozitorija prenose u središnji repozitorij je:

- **checkin**
-

1.Koje su temeljne značajke dobrog programskog produkta?

Odaberite jedan ili više odgovora:

- **mora biti prihvatljiv korisniku**
- mora biti vizualno lijep
- **mora se moći održavati**
- **mora biti pouzdan**
- mora imati dobru reklamu

2.Opseg rezultirajućih dokumenata u inženjerstvu zahtjeva ovisi o:

Odaberite jedan odgovor:

- trajanju projekta
- upotrebljenim metodama
- ugovorenoj cijeni
- **opsegu problema**
- zahtjevima programske potpore

3.Dva uobičajena modela procesa inženjerstva zahtjeva su

Odgovor: klasični i spiralni

4. U načine dokumentiranja zahtjeva sustava koji se koriste u praksi NE spada:

Odaberite jedan odgovor:

- Izrada scenarija
- **UML dijagram razreda**
- Intervjuiranje
- UML dijagram obrazaca uporabe
- UML sekvensijski dijagram

5.Osnovni elementi UML-a su:

- dijagrami i tekstualni opisi
- aktori, relacije i objekti
- **stvari, relacije i dijagrami**
- ništa od navedenog

6.Stvarima ponašanja (engl. behavioral things) odgovara kao UML opis:

Odaberite jedan odgovor:

- razredi, sučelja, slučajevi korištenja, komponentni čvorovi, aktori i sl.
- **interakcije i automati stanja**
- oznake elemenata modela
- paketi

7.Za sekvencijske dijagrame vrijede slijedeće tvrdnje (moguće više točnih odgovora):

Odaberite jedan ili više odgovora:

- dobri su za praćenje stanja sustava
- **prikazuju interakciju u sustavu**
- dobri su za grafički prikaz složenih algoritama
- **prikazuju objekte kao životne crte**
- **dobri su za prikaz komunikacije među objektima**

8.Za funkcionalnu dekompoziciju problema u UML obrascima uporabe upotrebljavaju se poveznice vrste:

Odaberite jedan odgovor:

- **uključivanja**
- generalizacije
- proširenja
- pridruživanja/asocijacije

9. Vrsta poveznice u dijagramu obrazaca uporabe s prikazom: (strelica s praznim trokutom) naziva se:

Odaberite jedan odgovor:

- **generalizacija**
- proširenje
- pridruživanje/asocijacija
- uključivanje

10.U atributе programskog inženjerstva ne spada:

Odaberite jedan odgovor:

- održavanje
- proširivost
- pouzdanost
- **Vještina programerskog tima**
- efikasnost

12. Koje se tri aktivnosti ponavljaju tijekom razvoja u evolucijskom modelu?

Odaberite jedan odgovor:

- Specifikacija, oblikovanje, ispitivanje
- Analiza, oblikovanje, implementacija
- **Specifikacija, razvoj, validacija**
- Analiza, implementacija, ispitivanje

13. Analiza rizika u spiralnom modelu programskog inžinjerstva provodi se:

Odaberite jedan odgovor:

- u prvoj petlji
- prije početka modeliranja
- **u svakoj petlji**
- nije potrebna
- samo na završnom modelu

14. U drugu fazu unificiranog procesa (engl. Unified Process) (Faza elaboracije) ne spada:

Odaberite jedan ili više odgovora:

- **Oblikovanje produkta.**
- **Definiranje dosega projekta.**
- Definiranje temelja arhitekture sustava.
- Specifikacija značajki.
- Izrada plana projekta.

15. Koji model procesa programskog inženjerstva je najpogodniji za velike inženjerske projekte gdje se sustav razvija na nekoliko odvojenih mesta?

Odaberite jedan odgovor:

- **vodopadni**
- RUP proces
- evolucijski
- komponentno usmjereni

16. Princip oblikovanja kojemu je cilj osigurati sakrivanje ili odgodu razmatranja detalja te time i smanjenje složenosti naziva se:

Odaberite jedan odgovor:

- Povećaj ponovnu uporabivost
- Oblikuj za fleksibilnost
- Oblikuj po ugovoru
- Planiraj zastaru
- **Zadrži razinu apstrakcije**

17.Princip oblikovanja koji se temelji na obradi svih slučajeva u kojima se neka komponenta može neprikladno upotrijebiti naziva se:

Odaberite jedan odgovor:

- Oblikovanje za ispitivanje
- Oblikovanje po ugovoru
- **Konzervativno oblikovanje**
- Oblikovanje za fleksibilnost
- Podijeli pa vladaj

18.Koja temeljna značajka je svojstvena modernom načinu oblikovanja programske potpore?

Odaberite jedan odgovor:

- specifikacija sustava
- uvođenje analize zahtjeva
- **uvođenje modela sustava**
- evaluacija sustava
- sve navedeno

20.Poliformizam u objektno usmjerenim sustavima povezujemo s:

Odaberite jedan odgovor:

- dinamičkom alokacijom (engl. dynamic allocation)
- **dinamičkim povezivanjem (engl. Dynamic binding)**
- postojanjem više metoda istog naziva (engl. overloading)
- statičkom alokacijom (engl. static allocation)

21.Odgovor objekti su rezultat instanciranja Odgovor razreda

22.Metode razreda deklariraju se unutar razreda ključnom riječi Odgovor static i pozivaju se preko Odgovor naziva razreda

23.Mogu li se instancirati objekti iz apstraktnih razreda (engl. abstract classes)?

Odaberite jedan odgovor:

- Ne
- da

- Dinamičko povezivanje (engl. dynamic linking)

24.Kakve su preporuke za uporabu globalnih varijabli u OO aplikacijama?

Odaberite jedan odgovor:

- ne postoje
- ponekad, najbolje ne
- **ne savjetuje se nikad**
- ovisno o zahtjevima performansi

25.Koje od navedenih tvrdnji vrijede za oznake vidljivosti:

Odaberite jedan ili više odgovora:

- **public – vidljivo svima**
- protected – dostupno u paketu
- **protected – dostupno u hijerarhiji razreda**
- private – dostupno u hijerarhiji razreda
- **private – dostupno samo unutar razreda**

27.Koje su dvije osobine zajedničke svim objektima u OO?

Odaberite jedan odgovor:

- Metode i poruke
- **stanje i ponašanje**
- Metode i sučelja

28.Odnos kompozicije i agregacije u UML dijagramu razreda je slijedeći:

Odaberite jedan odgovor:

- **Kompozit je jači od agregata zato što kad se uništi kompozit uništavaju se i svi sastavni dijelovi, dok to kod aggregata nije slučaj.**

29.Osnovu opisa funkcionalnih zahtjeva u dokumentaciji projekta čine:

Odaberite jedan ili više odgovora:

- **UML dijagrami obrazaca uporabe**
- UML dijagrami stanja
- UML dijagrami razreda
- **UML sekvencijski dijagrami**

30.Za sustav Subversion vrijede tvrdnje:

Odaberite jedan ili više odgovora:

- Središnje mjesto pohrane je uvijek jedan od lokalnih repozitorija (prema dogovoru).
- **Izbrisane datoteke je moguće povratiti.**

- Sporovi se rješavaju isključio ručnim spajanjem promjena u datotekama.
- Inačice datoteka s različitim brojevima revizije nužno su različite.
- **Broj revizije globalno određuje stanje repozitorija u određenom trenutku.**

Oblikovanje programske potpore - 1. labos – 2007/08

Primjeri pitanja

by [Fusbaler](#)

1. Za skinuti projekt iz Subversion repozitorija 'svn://host/projekt' potrebno je navesti:

svn co svn://host/projekt --username xxxxxxxxx

- lozinka je potrebna ukoliko je projekt zaštićen;

- projekt će se skinuti u trenutni direktorij u kojem smo pozicionirani kad pozovemo naredbu

2. Naredba 'svn update' napraviti će sljedeće:

- ovom naredbom ćemo dobiti zadnju inačicu projekta. Zamjeniti će se svi file-ovi koji su bili mijenjani. Naredba se poziva kada smo pozicionirani u folderu koji je "working copy" našeg projekta (npr.: /home/x/G54). Također će se ispisati i broj revision-a.

Ako se ne nalazimo u working copy direktoriju na konzoli će se ispisati: Skipped '.'

3. Naredba 'svn rm datoteka' maknuti će datoteku :

- ova naredba će prvo maknuti datoteku iz working copy direktorija na našem računalu, a onda će prilikom sljedećeg commita-nja maknuti je i iz repozitorija tj. sa servera kojemu se pristupa.

4. Što treba napraviti da bi se 'datoteka.ext' dodala u repozitorij (radnu kopiju)?

- treba stvoriti file "datoteka.txt" i onda ga kopirati u radni direktorij pa onda naredba : svn add datoteka.txt da se datoteka doda i još :

svn ci -m "Poruka" --username xxxxxxxx , kako bi se promjene napravile u repozitoriju.

Datoteku smo mogli stvoriti na disku pa onda prenijeti u folder ili smo mogli iz konzole pomoću nekog editora.

5. Da bi se vidjele promjene u datotekama između radne kopije repozitorija i zadnje inačice tih datoteka u repozitoriju treba napisati:

svn diff

- ovom naredbom ćemo dobiti ispis promjena koje su napravljene u datoteci od zadnje inačice. Naredba ne skida iz repozitorija najnoviju inačicu nego ispisuje promjene od zadnje verzije koju je korisnik skinuo (tj. imaju na disku).

Promjene se očituju kao plusevi (+) i minusi (-) u redcima datoteke.

6. Autori dijelova datoteke 'test.txt' mogu se vidjeti naredbom:

Postoje samo dvije naredbe koje prikazuju autora, onda koja nama treba u ovom slučaju je:

svn blame test.txt

7. Ispis sadržaja repozitorija u kojem se vide revizije u kojima su se datoteke mijenjale te tko ih je

mijenjao može se dobiti sa:

svn log

8. Naredba 'svn import objekt' (objekt=datoteka ili direktorij) napraviti će slijedeće:

Hmmm.... Ovo ce napraviti sranje u najmanju ruku. Jer se import definira kao:

svn import [PATH] URL

Znaci PATH mozemo maknuti ali URL ne mozemo. Ako maknemo PATH uzme sve iz trenutnog direktorija, a posto je stavljen samo objekt ispalo bi da je objekt URL sto je nemoguce... Ne znam samo kako bi se import ponasao ako smo pozicionirani u repozitorij. Nek netko izvrti.

by [bojzi](#)

Evo isprobao sam tako da sam prvo napravio "touch hello3.c" da napravim dummy file, a nakon toga "svn import hello3.c".

Output:

"svn: Invalid URL 'hello3.c'"

Valjda bi odgovor onda trebao biti da će se dogoditi greška. :\

9. Razlika između 'svn import datoteka' i 'svn add datoteka' je:

Uzmimo zdravo za gotovo da import radi onako ako se nalazimo u repozitoriju. Basically import mozemo opisati kao add + commit. Znaci import ce dodati datoteku u repozitorij i automatski promjene javiti na server dok add samo promjene napravi lokalno i ceka commit da se to potvrди.

10. 5x Iz ispisa nakon ažuriranja radne kopije repozitorija s promjenama u repozitoriju na poslužitelju, uz datoteku se pojavila oznaka 'G|C|U|D|A' (samo jedno slovo). Značenje toga jest:

G ime filea - Merged - Datoteka je sadržavala promjene u radnoj kopiji, ali je Subversion uspješno spojio promjene sa onima iz repozitorija

C ime filea - Conflicted - Nastao je konflikt u datoteci kojeg korisnik sam treba razrješiti

U ime filea - Updated - Datoteka je uspješno ažurirana

D ime filea - Deleted - Datoteka ili direktorij je bio izbrisano iz radne kopije

A ime filea - Added - Datoteka ili direktorij je bio dodan radnoj kopiji

11. Prilikom ažuriranja radne kopije repozitorija s promjenama u repozitoriju ('svn up'), na poslužitelju za datoteku 'main.c' pojavio se konflikt. Ukoliko je korisnik prije ažuriranja koristio inačicu 10 (koju je mijenjao), a trenutna inačica u repozitoriju jest 13, koje sve datoteke postoje u radnom repozitoriju korisnika koje započinju sa 'main.c'?

Stvorit ce se 3 datoteke:

main.c.mine - moja verzija datoteke koju sam, tako genijalno, uspio sprckati

main.c.10 - zadnja verzija datoteke koju ja imam, 10ta revizija

main.c.13 - najnovija verzija koju je nasao na repozitoriju.

Sa svim tim bi ja trebao rjesiti konflikt tako da copy/pasteam okolo po datotekama. Vise tu:

<http://svnbook.red-bean.com/en/1.4/s....cycle.resolve>

12. Ažuriranje repozitorija s datotekama iz radne kopije uz komentar "komentar" obavlja se naredbom:

svn ci -m "Ovo je komentar" --username xxxxxxxxxxx

- Ovo svn ci je commit (radi i naredba svn commit), "-m" označava da će se ostaviti komentar, " --username xxxx" korisničko ime ako je potrebno

13. Nakon ažuriranja na zadnju inačicu iz repozitorija, datoteka "glavni.c" je mijenjana u radnom repozitoriju. Međutim, naknadno je odlučeno da te nove promjene treba odbaciti i vratiti se na sadržaj datoteke iz repozitorija. To će se napraviti pomoću naredbe:

svn revert

by Data :

Ako zelimo samo promjeniti glavni.c onda pisemo:

svn revert glavni.c

Jer sa cistim revertom bi vratili sve na prijasnu reviziju a nikad ne znas sto je sve promjenjeno u prosloj reviziji, možda sve

14. Nakon promjena datoteke u lokalnom repozitoriju, napravljen je "diff" te su se u njegovu ispisu pojavile neke linije koje započinju sa minus (-) i neke koje započinju sa plus (+). Koje je značenje tih oznaka (u odnosu na zadnju inačicu datoteke iz repozitorija)?

+ znaci da je taj redak koda dodan u novoj inaciji

- znaci da je taj redak koda postojao u staroj inaciji ali je izbrisano u novoj

15. Upisom naredbe 'svn status' pojavio se ispis kod kojeg se redci sastoje od jednog znaka te imena datoteke. Za datoteke kojima prethodi znak '?' vrijedi:

naredba svn status služi za provjeru promjena u radnoj verziji.

Ukoliko se ispred naziva datoteke nalazi ? to znači da ta datoteka nije pod kontrolom Subversiona

16. Subversion sustav karakterizira:

Recimo mogucnost vracanja stare verzije, kontrola nad dosadasnjim verzijama, pametno resolvanje problema, jednostavne naredbe, besplatnost... Ima toga. Ne znam drugi logican odgovor.

17. Da bi se naredba 'svn up' izvela kako smo željeli (ažurirala radna kopija repozitorija):

Da bi se up izveo kako spada prvo moraju biti rjeseni svi konflikti jer sve dok je na nekoj datoteci G zastavica, ta datoteka se nece promjeniti. Naravno, mora se biti u direktoriju gdje nam je repozitorij, i ... ne pada mi nist drugo na pamet.

by **bojzi**

Ovo nije loš odgovor, ali treba izmjenu...

Naime, G zastavica kod naredbe "svn update" kaže kako je lokalna verzija datoteke bila različita od one verzije koja se nalazi na svn-u, te isto tako govori kako je pomoću te naredbe datoteka "sređena", tj. promjene koje su se trebale dogoditi kako bi datoteka bila up-to-date, su se dogodile i da nema konflikta.

Problem se javlja ako se javi zastavica C koja govori da je došlo do konflikta koji se mora ručno riješiti, a više o tome pogledajte u koleginom seminaru koji je dosta lako štivo i prođe se za 30-45 minuta.

18. Dva programera koji rade na istom projektu, mijenjaju istu datoteku. Prepostavimo da je promjena koju su napravili samo u jednoj liniji. Prvi je staru liniju "int funckija()" zamjenio sa "void funkcija()", a drugi, pošto je krenuo od iste datoteke tu istu liniju ("int funckija()") zamjenio s "double funkcija()". Ako prvi programer sada ažurira glavni repozitorij s 'svn ci -m "int->void' što će se dogoditi kada drugi programer sada pokuša napraviti slično, tj. 'svn ci -m "int->double"'?

Je, je... Subversion je pametan sustav ali ipak ne toliko pametan. Moze razrjesiti probleme koji bi bili u drugim redovima datoteke (mozda) ali ovako pametan ipak nije jer ni sami nismo sigurni kaj bi htjeli u programu. double ili void. Uglavnom izbacit ce conflict, i da, datoteka ce biti označena da je pod konfliktom sve dok se sa resolved ne javi da je problem rjesen. ;)

19. Dva programera koji rade na istom projektu mijenjaju dvije različite datoteke (svaki svoju, prvi "prva.c", drugi "druga.c"). Kada prvi od njih napravi 'svn ci -m "prvi mijenjao"' što će se dogoditi kada drugi nakon toga napravi 'svn ci -m "drugi mijenjao"'?

E, pa sad je dovoljno pametan. Izgleda kao konflikt, ali posto nisu prckali medjusobno po tudjim datotekama subversion ce sam razrjesiti konflikt i javiti -G zastavicu kod commiranja... ;)

by **bojzi**

Ovo je također krivo.

Naime, radi se o tome da dvoje ljudi radi na dvije RAZLIČITE datoteke.

To konkretno znači da će obojica uspjeti normalno commitati bez da im svn kaže išta.

Kako bih to potvrdio dignuo sam si u ovom njihovom CENTOSu svoj svn i napravio dva korisnička računa - bojzi i bojzi2. Nakon toga sam na istom svnu mijenjao dvije datoteke. bojzi je mijenjao hello.c, a bojzi2 je mijenjao hello2.c.

Bez korištenja "svn update", bojzi je prvi komitao svoje promjene, a zatim bojzi2 i sve je prošlo potpuno u redu. nakon "svn update" na obje strane obojica su imali sve kako treba.

20. Ako je naredba "svn up" označila konflikt za datoteku "test.c" (oznaka 'C' uz ime te datoteke) što će napraviti naredba "svn resolved main.c" koja neposredno slijedi iza "svn up"?

- naredba svn resolved će izbrisati file main.c

Ova naredba služi kako bi se Subversionu dojavilo da je konflikt riješen. Konflikt se može riješiti na više načina, a ako koristimo naredbu svn revert nije potrebno kasnije koristiti svn resolved.

by **bojzi**

Uh, momci, ja se ovdje ne bih složio...

Ako pogledate onaj kolegin seminar vidjet ćete da s naredbom "svn resolved datoteka.ext" treba jako paziti pošto će se u svakom slučaju dojaviti da je konflikt riješen, čak i ako se to nije dogodilo.

U tom smislu, ako iza "svn up", koji će nam dojaviti da postoji konflikt, odmah izvedemo "svn resolved main.c", i ovo citiram, "Subversion će pustiti da se objavi verzija koja sadrži konfliktne markere."

by **Dux**

Molim kolege da ovo sve provjere, pa čemo onda sastaviti konačnu i 100% točnu verziju !!!

1. Za skinuti projekt iz Subversion repozitorija 'svn://host/projekt' potrebno je navesti:

```
svn co svn://host/projekt --username xxxxxxxxx
```

- username je potrebno navesti ukoliko je projekt zaštićen (nakon toga će sustav tražiti password), ako projekt nije zaštićen slobodno se izostavi
- projekt će se skinuti u trenutni direktorij u kojem smo pozicionirani kad pozovemo naredbu

2. Naredba 'svn update' napraviti će sljedeće:

- ovom naredbom ćemo dobiti zadnju inačicu projekta. Zamjeniti će se svi file-ovi koji su bili mijenjani. Naredba se poziva kada smo pozicionirani u folderu koji je "working copy" našeg projekta (npr.: /home/x/G54). Također će se ispisati i broj revision-a.
- Ako se ne nalazimo u working copy direktoriju na konzoli će se ispisati: Skipped '.'

3. Naredba 'svn rm datoteka' maknuti će datoteku iz:

- ova naredba će prvo maknuti datoteku iz working copy direktorija na našem računalu, a onda će prilikom sljedećeg commita-nja maknuti je i iz repozitorija tj. sa servera kojemu se pristupa.

4. Što treba napraviti da bi se 'datoteka.ext' dodala u repozitorij (radnu kopiju)?

- treba stvoriti file "datoteka.txt" i onda ga kopirati u radni direktorij pa onda naredba : svn add datoteka.txt da se datoteka doda i još :
svc ci -m "Poruka" --username xxxxxxxx , kako bi se promjene napravile u repozitoriju.
Datoteku smo mogli stvoriti na disku pa onda prenijeti u folder ili smo mogli iz konzole pomoću nekog editora.

5. Da bi se vidjele promjene u datotekama između radne kopije repozitorija i zadnje inačice tih datoteka u repozitoriju treba napisati:

```
svn diff
```

- ovom naredbom ćemo dobiti ispis promjena koje su napravljene u datoteci od zadnje inačice. Naredba ne skida iz repozitorija najnoviju inačicu nego ispisuje promjene od zadnje verzije koju je korisnik skinuo (tj. ima

ju na disku).

Promjene se očituju kao plusevi (+) i minusi (-) u redcima datoteke.

10. 5x Iz ispisa nakon ažuriranja radne kopije repozitorija s promjenama u repozitoriju na poslužitelju, uz datoteku se pojavila oznaka 'G|C|U|D|A' (samo jedno slovo). Značenje toga jest :

G *ime filea* - Merged - Datoteka je sadržavala promjene u radnoj kopiji, ali je Subversion uspješno spojio promjene sa onima iz repozitorija

C *ime filea* - Conflicted - Nastao je konflikt u datoteci kojeg korisnik sam treba razrješiti

U *ime filea* - Updated - Datoteka je uspješno ažurirana

D *ime filea* - Deleted - Datoteka ili direktorij je bio izbrisana iz radne kopije

A *ime filea* - Added - Datoteka ili direktorij je bio dodan radnoj kopiji

12. Ažuriranje repozitorija s datotekama iz radne kopije uz komentar "komentar" obavlja se naredbom:

```
svn ci -m "Ovo je komentar" --username xxxxxxxxxxx
```

- Ovo svn ci je commit (radi i naredba svn commit), "-m" označava da će se ostaviti komentar, " --username xxxx" korisničko ime ako je potrebno

13. Nakon ažuriranja na zadnju inačicu iz repozitorija, datoteka "glavni.c" je mijenjana u radnom repozitoriju. Međutim, naknadno je odlučeno da te nove promjene treba odbaciti i vratiti se na sadržaj datoteke iz repozitorija. To će se napraviti pomoću naredbe:

```
svn revert
```

by Data :

13:

Ako zelimo samo promjeniti glavni.c onda pisemo:

```
svn revert glavni.c
```

Jer sa cistim revertom bi vratili sve na prijasnu reviziju a nikad ne znas sto je sve promjenjeno u prosloj reviziji, mozda sve :yahoo:

15. Upisom naredbe 'svn status' pojavio se ispis kod kojeg se redci sastoje od jednog znaka te imena datoteke. Za datoteke kojima prethodi znak '?' vrijedi:

- naredba svn status služi za provjeru promjena u radnoj verziji.

Ukoliko se ispred naziva datoteke nalazi ? to znači da ta datoteka nije pod kontrolom Subversiona

17. Da bi se naredba 'svn up' izvela kako smo željeli (ažurirala radna kopija repozitorija):

- pozvati naredbu svn update kada smo pozicionirani u working copy direktoriju.
? ovo mi je malo čudno ... prelagano ... vidi pitanje br. 2 ?

by **Data** :

17. Da bi se up izveo kako spada prvo moraju biti rjeseni svi konflikti jer sve dok je na nekoj datoteci G zastavica, ta datoteka se nece promjeniti. Naravno, mora se biti u direktoriju gdje nam je repozitorij, i ... ne pada mi nist drugo na pamet.

20. Ako je naredba "svn up" označila konflikt za datoteku "test.c" (oznaka 'C' uz ime te datoteke) što će napraviti naredba "svn resolved main.c" koja neposredno slijedi iza "svn up"?

- naredba svn resolved će izbrisati file main.c

Ova naredba služi kako bi se Subversionu dojavilo da je konflikt riješen. Konflikt se može riješiti na više načina, a ako koristimo naredbu svn revert nije potrebno kasnije koristiti svn resolved.

by **Data**

6. Autori dijelova datoteke 'test.txt' mogu se vidjeti naredbom:

Postoje samo dvije naredbe koje prikazuju autora, onda koja nama treba u ovom slučaju je:
svn blame test.txt

7. Ispis sadržaja repozitorija u kojem se vide revizije u kojima su se datoteke mijenjale te tko ih je mijenjao može se dobiti sa:

svn log

8. Naredba 'svn import objekt' (objekt=datoteka ili direktorij) napraviti će slijedeće:

Hmmm.... Ovo ce napraviti sranje u najmanju ruku. Jer se import definira kao:
svn import [PATH] URL

Znaci PATH mozemo maknuti ali URL ne mozemo. Ako maknemo PATH uzme sve iz trenutnog direktorija, a posto je stavljeno samo *objekt* ispalo bi da je objekt URL sto je nemoguce... Ne znam samo kako bi se import ponasao ako smo pozicionirani u repozitorij. Nek netko izvrsti. Ovdje ima vise o import naredbi:
<http://svnbook.red-bean.com/nightly/....c.import.html>

9. Razlika između 'svn import datoteka' i 'svn add datoteka' je:

Uzmimo zdravo za gotovo da import radi onako ako se nalazimo u repozitoriju. Basically import mozemo opisati kao add + commit. Znaci import ce dodati datoteku u repozitorij i automatski promjene javiti na server dok add samo promjene napravi lokalno i ceka commit da se to potvrди.

11. Prilikom ažuriranja radne kopije repozitorija s promjenama u repozitoriju ('svn up'), na poslužitelju za datoteku 'main.c' pojavio se konflikt. Ukoliko je korisnik prije ažuriranja koristio inačicu 10 (koju je mijenjao), a trenutna inačica u repozitoriju jest 13, koje sve datoteke postoje u radnom repozitoriju korisnika koje započinju sa 'main.c'?

Stvorit ce se 3 datoteke:

main.c.mine - moja verzija datoteke koju sam, tako genijalno, uspio sprckati

main.c.10 - zadnja verzija datoteke koju ja imam, 10ta revizija

main.c.13 - najnovija verzija koju je nasao na repozitoriju.

Sa svim tim bi ja trebao rjesiti konflikt tako da copy/pasteam okolo po datotekama. Vise tu:

<http://svnbook.red-bean.com/en/1.4/s....cycle.resolve>

14. Nakon promjena datoteke u lokalnom repozitoriju, napravljen je "diff" te su se u njegovu ispisu pojavile neke linije koje započinju sa minus (-) i neke koje započinju sa plus (+). Koje je značenje tih oznaka (u odnosu na zadnju inačicu datoteke iz repozitorija)?

- + znaci da je taj redak koda dodan u novoj inacici
- znaci da je taj redak koda postojao u staroj inacici ali je izbrisana u novoj

16. Subversion sustav karakterizira:

Cute face, izgled slatkice, lijep smijesak, slatko ponasanje, nasmijava, bas je cute... OK KNOCK IT OFF!!! Covjek moze nabrojati 10 stvari... nadjite neke i nabrajajte. Recimo mogucnost vracanja stare verzije, kontrola nad dosadasnjim verzijama, pametno resolvanje problema, jednostavne naredbe, besplatnost... Ima toga. Ne znam drugi logican odgovor.

18. Dva programera koji rade na istom projektu, mijenjaju istu datoteku. Prepostavimo da je promjena koju su napravili samo u jednoj liniji. Prvi je staru liniju "int funkcija()" zamjenio sa "void funkcija()", a drugi, pošto je krenuo od iste datoteke tu istu liniju ("int funkcija()") zamjenio s "double funkcija()". Ako prvi programer sada ažurira glavni repozitorij s 'svn ci -m "int->void"' što će se dogoditi kada drugi programer sada pokuša napraviti slično, tj. 'svn ci -m "int->double"'?

Je, je... Subversion je pametan sustav ali ipak ne toliko pametan. Moze razrjesiti probleme koji bi bili u drugim redovima datoteke (mozda) ali ovako pametan ipak nije jer ni sami nismo sigurni kaj bi htjeli u programu. double ili void. Uglavnom izbacit ce conflict, i da, datoteka ce biti označena da je pod konfliktom sve dok se sa resolved ne javi da je problem rjesen. ;)

19. Dva programera koji rade na istom projektu mijenjaju dvije različite datoteke (svaki svoju, prvi "prva.c", drugi "druga.c"). Kada prvi od njih napravi 'svn ci -m "prvi mijenjao"' što će se dogoditi kada drugi nakon toga napravi 'svn ci -m "drugi mijenjao"'?

E, pa sad je dovoljno pametan. Izgleda kao konflikt, ali posto nisu prckali medjusobno po tudjim datotekama subversion ce sam razrjesiti konflikt i javiti -G zastavicu kod commiranja... ;)

Ajmo jos neke od ovih objasniti. Tocnije **#17**. Da bi se up izveo kako spada prvo moraju biti rjeseni svi konflikti jer sve dok je na nekoj datoteci G zastavica, ta datoteka se nece promjeniti. Naravno, mora se biti u direktoriju gdje nam je repozitorij, i ... ne pada mi nist drugo na pamet.

#13:

Ako zelimo samo promjeniti glavni.c onda pisemo:

svn revert glavni.c

Jer sa cistim revertom bi vratili sve na prijasnu reviziju a nikad ne znaš sto je sve promjenjeno u prosloj reviziji, možda sve

Oblikovanje programske potporeYou have gone full screen. [Exit full screen \(F11\)](#)

Navigacija u testu Pitanje 1 Završeno Marked out of 1,00 <input type="button" value="Flag question"/> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 Završi test ...	<p>Pitanje 1 Završeno Marked out of 1,00 <input type="button" value="Flag question"/></p> <p>Za sustave za formalnu verifikaciju vrijede tvrdnje:</p> <p>Odaberite jedan ili više odgovora:</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> a. mora postojati izgrađen model sustava <input checked="" type="checkbox"/> b. mora postojati definirana formalna specifikacija sustava <input type="checkbox"/> c. garantira izvan svake sumnje ispravnu relaciju modela i specifikacije <input type="checkbox"/> d. postupak se provodi na izgrađenom sustavu <input type="checkbox"/> e. osigurava ispravnu relaciju implementacije i specifikacije <p><input type="button" value="Check"/></p> <p>Točan odgovor je: mora postojati izgrađen model sustava, mora postojati definirana formalna specifikacija sustava, garantira izvan svake sumnje ispravnu relaciju modela i specifikacije.</p> <hr/> <p>Pitanje 2 Završeno Marked out of 1,00 <input type="button" value="Flag question"/></p> <p>Za tehnike simulacije i ispitivanja (engl. testing) programa u procesu dokazivanja ispravnost programa vrijede tvrdnje:</p> <p>Odaberite jedan odgovor:</p> <ul style="list-style-type: none"> <input type="radio"/> a. Ne može dokazati postojanje, ali može dokazati nepostojanje pogreške. <input checked="" type="radio"/> b. Može dokazati postojanje, ali ne može dokazati nepostojanje pogreške. <input type="radio"/> c. Ne može dokazati ni postojanje ni nepostojanje pogreške. <input type="radio"/> d. Može dokazati postojanje i nepostojanje pogreške. <p><input type="button" value="Check"/></p> <p>Točan odgovor je: Može dokazati postojanje, ali ne može dokazati nepostojanje pogreške..</p> <hr/> <p>Pitanje 3 Završeno Marked out of 1,00 <input type="button" value="Flag question"/></p> <p>Za pojam krize razvoja programske potpore vrijedi</p> <p>Odaberite jedan ili više odgovora:</p> <ul style="list-style-type: none"> <input type="checkbox"/> a. prolazi iz problema operacijskih sustava <input type="checkbox"/> b. nije moguće prevazići poboljšanjem performansi pridruženog sklopovja <input type="checkbox"/> c. termin smislen 1968. godine na NATO konferenciji <input type="checkbox"/> d. stvarna kriza programske potpore ne postoji svi mogu raditi programe <input type="checkbox"/> e. prolazi iz činjenice da samo trećina studentskih projekata programske potpore daje djelomičan rezultat <input type="checkbox"/> f. djelomično se riješava poboljšanjem performansi pridruženog sklopovja <input checked="" type="checkbox"/> g. manifestira se kašnjenjem isporuke programske potpore i nezadovljavanjem korisničkih zahtjeva <input checked="" type="checkbox"/> h. uzročnici su povezani sa složenošću sklopovja i razvojnih procesa <p><input type="button" value="Check"/></p> <p>Točan odgovor je: termin smislen 1968. godine na NATO konferenciji, nije moguće prevazići poboljšanjem performansi pridruženog sklopovja, manifestira se kašnjenjem isporuke programske potpore i nezadovljavanjem korisničkih zahtjeva, uzročnici su povezani sa složenošću sklopovja i razvojnih procesa.</p>
--	--

[Nastavak](#)

Navigacija u testu

1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
19 20 21 22 23 24
25

Završi test ...

Pitanje 4
Završeno
Marked out of 1,00
 Flag question

Ako program u svom radu ne zadovoljava korisničke zahtjeve tu manifestaciju nazivamo:

- Odaberite jedan odgovor:
- a. kvar
 - b. pogreška
 - c. zatajenje
 - d. nedostatak
 - e. sve navedeno

[Check](#)

Točan odgovor je: nedostatak.

Pitanje 5
Završeno
Marked out of 1,00
 Flag question

Uporabom se programska potpora kvari jer:

- Odaberite jedan odgovor:
- a. rezervni dijelovi zastarijevaju
 - b. je izložena teškim radnim uvjetima
 - c. sklopovje na kojem se izvodi podložno je stareno
 - d. pogreške se uočavaju tek kod intezivne i dugotrajne uporabe
 - e. postaje nekompatibilno s okolinom
 - f. krajnji korisnici se mijenju

[Check](#)

Točan odgovor je: pogreške se uočavaju tek kod intezivne i dugotrajne uporabe.

[Nastavi](#)

Navigacija u testu
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 Završi test ...

Pitanje 6

Završeno

Marked out of 1,00

Flag question

Proces inženjerstva zahtjeva ne uključuje:

- Odaberite jedan odgovor:
- a. Studiju izvedivosti
 - b. Specifikaciju zahtjeva
 - c. Rukovanje promjenama u zahtjevima
 - d. Izlučivanje zahtjeva i analizu
 - e. Validaciju programske potpore

[Check](#)

Točan odgovor je: Validaciju programske potpore.

Pitanje 7

Završeno

Marked out of 1,00

Flag question

Etnografija je:

Odaberite jedan odgovor:

- a.
 - b. metoda brze izrade prototipa
 - c. metoda strukturiranja formalnih poslovnih procesa u svrhu izlučivanja zahtjeva
 - d. pregled utjecaja svih dionika na zahtjeve sustava
 - e.
- Kvalitativno promatranje i opis ponašanja ljudi pri obavljanju poslova

[Check](#)

Točan odgovor je: Zahtjevi izvedeni temeljem istraživanja kako ljudi stvarno rade.

Pitanje 8

Završeno

Marked out of 1,00

Flag question

U načine dokumentiranja zahtjeva sustava koji se koriste u praksi NE spada:

- Odaberite jedan odgovor:
- a. Intervjuiranje
 - b. UML dijagram razreda
 - c. Izrada scenarija
 - d. UML sekvensijski dijagram
 - e. UML dijagram obrazaca uporabe

[Check](#)

Točan odgovor je: UML dijagram razreda.

Pitanje 9

Završeno

Marked out of 1,00

Flag question

Ako svi zahtjevi obuhvaćeni u dokumentu inženjerstva zahtjeva imaju samo jednu moguću interpretaciju tada možemo tvrditi da je dokument specifikacije zahtjeva:

- Odaberite jedan odgovor:
- a. sve navedeno
 - b. jednoznačan
 - c. konzistentan
 - d. ispravan
 - e. moguće verificirati

[Check](#)

Točan odgovor je: jednoznačan.

[Nastavi](#)

Navigacija u testu

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25					

Završi test ...

Pitanje 10
Završeno
Marked out of 1,00
 Flag question

Opseg rezultirajućih dokumenata inžinjertvu zahtjeva ovisi:

Odaberite jedan odgovor:

a. trajanju projekta
 b. zahtjevima programske podrške
 c. uporebljenim metodama
 d. opsegu problema
 e. ugovorenoj cijeni

You have gone full screen. [Exit full screen \(F11\)](#)

Točan odgovor je: opsegu problema.

Pitanje 11
Završeno
Marked out of 1,00
 Flag question

Tipične metode u izlučivanju i analizi zahtjeva su:

Odaberite jedan ili više odgovora:

a. Specificiranje dinamičkih interakcija u sustavu
 b. specificiranje zahtjeva obrascima uporabe
 c. Intervjuiranje
 d. obavezno analizira doprinose sustava postavljenim ciljevima
 e. Izrada scenarija

Točan odgovor je: Intervjuiranje , Izrada scenarija, specificiranje zahtjeva obrascima uporabe, Specificiranje dinamičkih interakcija u sustavu.

Pitanje 12
Završeno
Marked out of 1,00
 Flag question

Primarna uloga dokument specifikacije zahtjeva obuhvaća:

Odaberite jedan ili više odgovora:

a. definiranje sustava razumljivo korisniku
 b. osigurava razumljiv opis zahtjeva
 c. osigurava tehničku specifikaciju za razvojni tim
 d. osigurava komunikaciju s klijentima i korisnicima
 e. služi za komunikaciju s razvojnim timom

Točan odgovor je: definiranje sustava razumljivo korisniku, osigurava razumljiv opis zahtjeva, osigurava komunikaciju s klijentima i korisnicima .

Pitanje 13
Završeno
Marked out of 1,00
 Flag question

U inžinjertvu zahtjeva programske potpore za zahteve vrijede tvrdnje:

Odaberite jedan ili više odgovora:

a. to su opisi usluga sustava
 b. to su utvrđena ograničenja implementirane programske podrške
 c. to su ograničenja programske podrške
 d. izražavaju se isključivo posebnim jezicima za oblikovanje sustava, dijagramima i matematičkom notacijom
 e. to su opisi ograničenja sklopovija
 f. mogu se izražavati prirodnim jezikom

Točan odgovor je: to su opisi usluga sustava, to su opisi ograničenja sklopovija, to su ograničenja programske podrške, mogu se izražavati prirodnim jezikom.

Nastavi

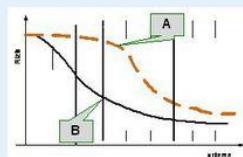
Oblikovanje programske potpore

You have gone full screen. [Exit full screen \(F11\)](#)

Navigacija u testu					
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25					
Završi test ...					

Pitanje 14
Završeno
Marked out of 1,00
[Flag question](#)

Na slici je prikazana procjena rizika razvoja programske potpore tijekom vremena temeljena na vodopadnom i inkrementalnom modelu.



Odaberite točnu tvrdnju:

Odaberite jedan odgovor:

- a. Krivulja B prikazuje inkrementalni model.
- b. Krivulja A prikazuje inkrementalni model.

[Check](#)

Točan odgovor je: Krivulja B prikazuje inkrementalni model..

Pitanje 15
Završeno
Marked out of 1,00
[Flag question](#)

Što vrijedi za spiralni model razvoja programske potpore?

Odaberite jedan odgovor:

- a. većina ispitnih slučajeva mora biti završena prije početka implementacije
- b. barem jedan dio implementacije mora biti završen prije pisanja ispitnih slučajeva
- c. većina specifikacija je definirana prije kodiranja
- d. nije potrebna
- e. barem jedan dio specifikacija mora biti definiran prije njihovog kodiranja

[Check](#)

Točan odgovor je: barem jedan dio specifikacija mora biti definiran prije njihovog kodiranja.

Pitanje 16
Završeno
Marked out of 1,00
[Flag question](#)

Prednosti uporabe vodopadnog modela nad "ad hoc" modelom su:

Odaberite jedan ili više odgovora:

- a. složena organizacija s jednostavnom podrškom
- b. nema iteracija
- c. brze iteracije
- d. definirani prijelazi između strogo definiranih faza
- e. nestrukturirani proces
- f. dobro definirana slijedna organizacija procesa

[Check](#)

Točan odgovor je: definirani prijelazi između strogo definiranih faza, dobro definirana slijedna organizacija procesa, nema iteracija.

[Nastavi](#)

Oblikovanje programske potpore

You have gone full screen. [Exit full screen \(F11\)](#)

Navigacija u testu	
Pitanje 17	Završeno
Marked out of 1,00	
<input type="checkbox"/> Flag question	
1 2 3 4 5 6	
7 8 9 10 11 12	
13 14 15 16 17 18	
19 20 21 22 23 24	
25	

Završi test ...

Pitanje 17

Završeno

Marked out of 1,00

Flag question

Proces razvoja programske potpore osigurava:

- Odaberite jedan ili više odgovora:
- a. definira tko radi što i kada
 - b. upotrebljiv/efikasan oblik informacija
 - c. pravovremenost potrebnih informacija
 - d. predstavlja jedinstveni predložak razvoja
 - e. potrebne informacije

[Check](#)

Točan odgovor je: potrebne informacije, pravovremenost potrebnih informacija, upotrebljiv/efikasan oblik informacija, definira tko radi što i kada.

Pitanje 18

Završeno

Marked out of 1,00

Flag question

Unificirani model procesa (engl. *Unified Process Model*) razlikuje se od generičkih procesa programskog inženjerstva jer:

- Odaberite jedan ili više odgovora:
- a. Definira kљučne točke (engl. *milestones*) u procesu programskog inženjerstva.
 - b. Definira aktivnosti u procesu programskog inženjerstva.
 - c. Uvodi iteracije u proces programskog inženjerstva.
 - d. Definira faze životnog ciklusa programske potpore.
 - e. Povezuje temeljne aktivnosti s modelima, a modele s dijagramima.

[Check](#)

Točan odgovor je: Povezuje temeljne aktivnosti s modelima, a modele s dijagramima., Definira kљučne točke (engl. *milestones*) u procesu programskog inženjerstva., Definira faze životnog ciklusa programske potpore..

[Nastavi](#)

Oblikovanje programske potpore

Navigacija u testu					
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25					

Završi test ...

Pitanje 19

Završeno

Marked out of 1,00

Flag question

Za funkcionalnu dekompoziciju problema u UML obrascima uporabe upotrebljavaju se poveznice vrste:

Odaberite jedan odgovor:

- a. uključivanja
- b. generalizacije
- c. proširenja
- d. pridruživanja/asocijacije

Check

Točan odgovor je: uključivanja.

Pitanje 20

Završeno

Marked out of 1,00

Flag question

Za UML obrasce uporabe NE vrijedi:

- a. jednostavni su za razumijevanje
- b. služe za izlučivanje zahtjeva prema pogledu interakcije
- c. modeliraju funkcionalne zahtjeve sustava
- d. specificiraju način implementacije sustava
- e. temeljeni su na idejni scenarija

Check

Točan odgovor je: specificiraju način implementacije sustava.

Pitanje 21

Završeno

Marked out of 1,00

Flag question

Stvarima grupiranja (engl. grouping things) odgovara kao UML opis:

- a. oznake elemenata modela
- b. interakcije i automati stanja
- c. razredi, sučelja, slučajevi korištenja, komponentni čvorovi, aktori i sl.
- d. paketi

Check

Točan odgovor je: paketi.

[Nastavi](#)

Oblikovanje programske potpore

Navigacija u testu					
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25					
Završi test ...					

Pitanje 22

Završeno
Marked out of 1,00
 Flag question

Vrsta poveznice u dijagramu obrazaca uporabe s prikazom:



naziva se:

Odaberite jedan odgovor:

- a. generalizacija
- b. pridruživanje/asocijacija
- c. uključivanje
- d. proširenje

Točan odgovor je: generalizacija.

Pitanje 23

Završeno
Marked out of 1,00
 Flag question

Vrsta poveznice u dijagramu obrazaca uporabe s namjenom prikaza da osnovni obrazac sadrži ponašanje definirano u drugom obrascu naziva se:

- Odaberite jedan odgovor:
- a. generalizacija
 - b. proširenje
 - c. uključivanje
 - d. pridruživanje/asocijacija

Točan odgovor je: uključivanje.

Oblikovanje programske potpore

You have gone full screen. [Exit full screen \(F11\)](#)

Navigacija u testu

Pitanje 24
Završeno
Marked out of 1,00
Flag question

UML sekvensijski dijagrami NISU pogodni za:

Odaberite jedan ili više odgovora:

a. prikazivanje objekata koji međusobno komuniciraju
 b. pojašnjenje pojedinog obrazca uporabe (engl. use case)
 c. prikaz vremenskog uređenja među događajima
 d. prikazivanje komunikacije aktora
 e. prikaz složene proceduralne logike

[Check](#)

Točan odgovor je: prikaz složene proceduralne logike.

Pitanje 25
Završeno
Marked out of 1,00
Flag question

U sekvensijskom dijagramu UML sustava padajuća linija poruke predstavlja:

Odaberite jedan odgovor:

a. Asinkronu poruku.
 b. Sinkronu poruku.
 c. Poruku koja ne očekuje povratnu informaciju.
 d. Poruku koja očekuje povratnu informaciju.
 e. Poruku koja ima vremensko trajanje.

[Check](#)

Točan odgovor je: Poruku koja ima vremensko trajanje..

[Nastavi](#)

1. Najznačajnija odlika oblikovanja programske potpore principom "od vrha prema dolje" (engl. *top down*) je:

- A) Definiraju se moduli koji su pogodni za kompoziciju na višoj razini.
- B) Ostvaruje se dobra struktura cijelog sustava.
- C) U najvećoj mjeri ostvaruje se princip ponovne uporabe (engl. *reuse*).

2. Uporaba globalnih varijabli u programskom kodu u najvećoj mjeri:

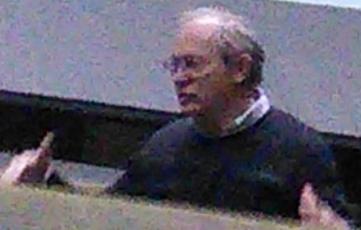
- A) Povećava koheziju.
- B) Povećava fleksibilnost
- C) Povećava međuovisnost.
- D) Povećava ponovnu uporabivost.

3. U objektno usmjerenom oblikovanju programske potpore
apstrakcija se POVEĆAVA s:

- A) Izbjegavanjem generalizacije i nasljeđivanja.
- B) Smanjenjem broja javnih procedura (metoda).
- C) Smanjenjem broja privatnih varijabli a povećanjem javnih.

4. Princip oblikovanja procedura po ugovoru (engl. *by contract*) gdje sve metode imaju ugovor s pozivateljem najviše doprinosi:

- A) Fleksibilnosti sustava.
- B) Jednostavnom ispitivanju.
- C) Jednostavnoj prenosivosti.
- D) Planiranju zastare
- E) Konzervativnom (obrambenom) oblikovanju.



5. Iz navedenih parova pojma izdvoji moguće dobre generalizacije u objektno usmjerenom oblikovanju.

- A) Student - Ivalvic_2
- B) StedniRacun - TekuciRacun
- C) Osoba - Klijent
- D) Kontinent - Drzava

6. U objektno usmjerenoj programskoj potpori tip varijable može biti neki razred.

- A) Jedan objekt može biti referenciran samo od jedne varijable koja je tipa razreda toga objekta.
- B) Jedan objekt (isti) može biti referenciran od više varijabli koje su tipa toga objekta. Sve te varijable mogu u jednom trenutku sadržavati memorijsku adresu objekta.
- C) Jedan objekt (isti) može biti referenciran od više varijabli koje su tipa toga objekta ali uvijek samo jedna od njih može u jednom trenutku sadržavati memorijsku adresu objekta.

7. Aktivnosti u nekom objektno usmjerenom sustavu možemo specificirati na više razina apstrakcije. Koji je ispravan slijed razina po principu "s najviše razine apstrakcije prema najniže?"

- A) odgovornost, metoda, operacija
- B) operacija, metoda, odgovornost,
- C) odgovornost, operacija, metoda
- D) metoda, operacija, odgovornost

8. U UML dijagramu razreda neka postoje hijerarhija generalizacije (engl. "is a") i hijerarhija cjelina-dio (engl. "part-of").

Što je ispravno ?

- A) Prijenos atributa i metoda nasljeđivanjem u hijerarhiji generalizacije mora se posebno isprogramirati (kodirati) a također u hijerarhiji cjelina-dio učinak operacija iz cjeline na njene dijelove.
- B) Ne mora se isprogramirati nasljeđivanje u generalizaciji ni učinak operacija iz cjeline na njene dijelove.
- C) Ne mora se isprogramirati nasljeđivanje u generalizaciji ali se mora učinak operacija iz cjeline na njene dijelove.
- D) Mora se isprogramirati nasljeđivanje u generalizaciji ali se ne mora učinak operacija iz cjeline na njene dijelove.



9. U razredu tipa "sučelje" (engl. *interface*):

- A) Mogu se implementirati (kodirati) metode koje su polimorfno uporabive.
- B) Ne mogu se implementirati (kodirati) metode već ih **sve** mora implementirati neki drugi ali **jedan** razred.
- C) Ne mogu se implementirati (kodirati) metode već ih mogu implementirati **razni** razredi.



10 Neka je za neku programsку potporu zadan dijagram razreda:



Koliko se objekata iz razreda "*Tvrta*" može **najviše** generirati u toj programskoj potpori ?

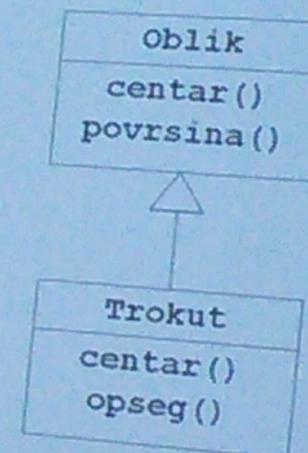
- A) 1
- B) 5
- C) 1 do 5
- D) više od 5

11. Neka u objektno usmjerenom oblikovanju programske potpore postoe dva konkretna razreda u hijerarhiji kao na slici.

Neka u jednom trenutku izvođenja varijabla **var** tipa **Oblik** referencira objekt stvoren iz razreda **Trokut**.

Pozivom metode **centar()** preko te varijable radi se o:

- A) Statičkom povezivanju
- B) Dinamičkom povezivanju

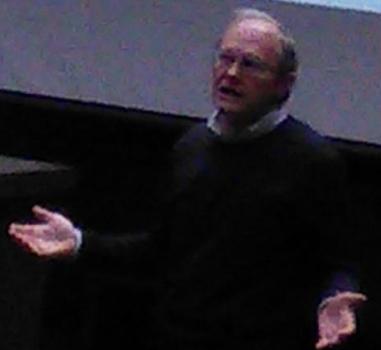


12. Zadano je oblikovanje sustava:

- Sustav osigurava temeljne usluge za rukovanje bankovnim računima u banci koja se zove OOBank. Ta banka ima više poslovnica, koje imaju svoje adrese i oznaku poslovnice. Pojedini klijent otvara račun u poslovniči banke.

Koji skup imenica čini dobre kandidate za razrede
(ispravan odgovor mora sadržavati sve dobre kandidate):

- A) sustav, banka, oznaka_poslovnice
- B) poslovница, račun, OOBank
- C) klijent, poslovница, račun
- D) oznaka_poslovnice, poslovica, banka



Pitanje 1

Fakultet:

Broj bodova: 1,00

od 1,00

Pit. Oznaci pitanje

Za tehnike simulacije i ispitivanja (engl. testing) programa u procesu dokazivanja ispravnosti vrijede tvrdnje:

- Odaberite jedan odgovor:
- a. Ne može dokazati ni postojanje ni nepostojanje pogreške.
 - b. Može dokazati postojanje i nepostojanje pogreške.
 - c. Ne može dokazati postojanje, ali može dokazati nepostojanje pogreške.
 - d. Može dokazati postojanje, ali ne može dokazati nepostojanje pogreške. ✓

Točan odgovor je: Može dokazati postojanje, ali ne može dokazati nepostojanje pogreške.

Pitanje 2

Fakultet:

Broj bodova: 1,00

od 1,00

Pit. Oznaci pitanje

Koje su generičke aktivnosti u svakom procesu inženjerstva zahtjeva?

Odaberite jedan ili više odgovora:

- a. evolucija ✓
- b. razvoj i oblikovanje ✓
- c. specifikacija ✓
- d. validacija i verifikacija ✓

Točan odgovor je: specifikacija, razvoj i oblikovanje, validacija i verifikacija, evolucija

Pitanje 3

Nekoliko:

Broj bodova: 0,00

od 1,00

Pit. Oznaci pitanje

U iteracije spiralnog modela procesa inženjerstva zahtjeva ne pripadaju:

Odaberite jedan ili više odgovora:

- a. Izlučivanje zahtjeva ✗
- b. Specifikacija zahtjeva ✗
- c. Validacija zahtjeva ✗
- d. Studija izvedivosti
- e. Izrada prototipa

Točan odgovor je: Izrada prototipa, Studija izvedivosti

Pitanje 4

Nekoliko:

Broj bodova: 0,00

od 1,00

Pit. Oznaci pitanje

U načine dokumentiranja zahtjeva sustava koji se koriste u praksi NE spada:

Odaberite jedan odgovor:

- a. intervjuiranje ✗
- b. UML sekvenčni dijagram
- c. UML dijagram razreda
- d. Izrada scenarija
- e. UML dijagram obrazaca uporabe

Točan odgovor je: UML dijagram razreda

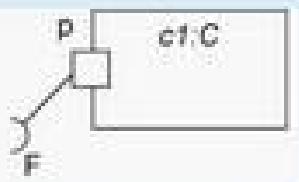
Pitanje 5

Nastavak:

Široj kategorije:
0,00 od 1,00

u: Čitavac
pitanja

Očekivana izložba rješenja:



Očekivana je lična rješenja:

- a. puniti je zadržava uvođenja i uklanjanja učilištem?
- b. objekt c1 je liga komponente
- c. prikazano je informacija struktura nazivata C
- d. puniti je provođen na objekt F? **x**
- e. puniti je učinkuje uvođenja i uklanjanja učilištem?

Tučan rješenje je: objekt c1 je liga komponente, puni je učinkuje uvođenja i uklanjanja učilištem?

Pitanje 6

Nastavak:

Široj kategorije:
1,00 od 1,00

u: Čitavac
pitanja

Složenje grupiranje (eng. grouping) učinkova kao UML objekat:

Očekivana ječan rješenja:

- a. Inicijacija i automatski otvaranje
- b. nazivni, uobičajeni, oblikovanji kontekstni, komponentni čvorovi, aktori i sli.
- c. pokazi ↗
- d. razne slavarske metode

Tučan rješenje je: pokazi ↗

Pitanje 7

Nastavak:

Široj kategorije:
0,00 od 1,00

u: Čitavac
pitanja

U sekvenčnom dijagramu UML, koštana predajući funkcija puniti je prethodno

Očekivana ječan rješenja:

- a. Poruka koja oblikuje povratnu informaciju
- b. Zadovoljena poruka
- c. Poruka koja ne oblikuje povratnu informaciju **x**
- d. Poruka koja ima vremensku trajanje
- e. Asinhrona poruka

Tučan rješenje je: Poruka koja ima vremensku trajanje.

Pitanje 8

Nastavak:

Široj kategorije:
1,00 od 1,00

u: Čitavac
pitanja

Kontroliranje koji zadržava se kroz UML obvezni upoznati?

Očekivana ječan rješenja:

- a. funkcionirajući izvješće ↗
- b. zadržava procesna razlikovanja
- c. zadržava platformne razlikovanja
- d. zadržava kvalitete

Tučan rješenje je: funkcionirajući izvješće ↗

Pitanje 9

Nastavak:

Široj kategorije:
-0,25 od 1,00

u: Čitavac
pitanja

Rukov vodnik mijedi između objekata A i B na sljedeći način?



Očekivana ječan rješenja:

- a. Objekat A pružajući razne vrste upoznati objektu B
- b. Objekat B u svih svrha povezivanje objekta A
- c. Objekat B pružajući razne vrste upoznati objektu A **x**
- d. Objekat A svrha u svih povezivanje objektu B

Tučan rješenje je: Objekat A pružajući razne vrste upoznati objektu B

Pitanje 10

Djelomično točno

Broj bodova: 0,33

od 1,00

Oznaci pitanje

Što je potrebno uključiti u dokument oblikovanja arhitekture programske potpore?

Odaberite jedan ili više odgovora:

- a. Dobro komentirani kod
- b. Prioritete koji su vodili proces oblikovanja ✓
- c. Alternativna rješenja
- d. Opis s najviše razine promatranja
- e. Referenciju prema dokumentu zahtjeva

Točan odgovor je: Prioriteti koji su vodili proces oblikovanja, Referenciju prema dokumentu zahtjeva, Opis s najviše razine promatranja

Pitanje 11

Netočno

Broj bodova: 0,00

od 1,00

Oznaci pitanje

Pristup zasnovan na razvoju, oblikovanju i isporuci vrlo malih inkremenata funkcionalnosti svojstven je kojoj metodologiji razvoja programske potpore?

Odaberite jedan odgovor:

- a. RUP ✗
- b. agilnoj
- c. iterativnoj
- d. spiralnoj

Vaš odgovor nije točan.

Točan odgovor je: agilnoj

Pitanje 12

Netočno

Broj bodova: 0,00

od 1,00

Oznaci pitanje

Koji je model procesa programskog inženjerstva najpogodniji za male i srednje interaktivne sustave?

Odaberite jedan odgovor:

- a. evolucijski
- b. RUP proces
- c. komponentno usmjeren ✗
- d. vodopadni

Točan odgovor je: evolucijski

Pitanje 13

Netočno

Broj bodova: 0,00
od 1,00

▼ Oznaci pitanje

Koje od navedenih tvrdnji vrijede za spiralni model razvoja programske podrške?

Odaberite jedan ili više odgovora:

- a. Smanjuje se rizik razvoja
- b. Malo administrativno opterećenje
- c. Sektori u spirali slijede vodopadni model
- d. Brza izrada prototipa.
- e. Razumljiv tehničkom i netehničkom rukovodstvu ✗

Točan odgovor je: Sektori u spirali slijede vodopadni model , Smanjuje se rizik razvoja , Brza izrada prototipa.

Pitanje 14

Točno

Broj bodova: 1,00
od 1,00

▼ Oznaci pitanje

U drugu fazu unificiranog procesa (engl. *Unified Process*) (Faza elaboracije) ne spada:

Odaberite jedan ili više odgovora:

- a. Izrada plana projekta.
- b. Specifikacija značajki.
- c. Oblikovanje produkta. ✓
- d. Definiranje dosega projekta. ✓
- e. Definiranje temelja arhitekture sustava.

Točan odgovor je: Definiranje dosega projekta., Oblikovanje produkta.

Pitanje 15

Netočno

Broj bodova: -0,25
od 1,00

▼ Oznaci pitanje

Koja od navedenih je temeljna značajka vodopadnog modela:

Odaberite jedan odgovor:

- a. Funkcionalnosti se razvijaju postepeno, tako da se u svakom koraku implementira nova funkcionalnost ✗
- b. Korisnik zna što treba i nisu mu potrebna pojašnjenja
- c. Zahtjevi se rijetko mijenjaju
- d. Pojedina faza se mora dovršiti prije pokretanja nove faze

Točan odgovor je: Pojedina faza se mora dovršiti prije pokretanja nove faze

Pitajte 16

Nalazite:

Svoj bodova: -0,33
od 1,00

Opis: Odmot pitanje

Što znači princip „oblikuj po ugovoru“?

Odaberite jedan odgovor:

- a. izbjegavanje specifičnosti neke okoline
- b. izbjegavanje pretpostavki kako će neko koristiti program
- c. sve navedeno
- d. određivanje stavki iz ugovora s naručiteljem ✗
- e. sve metode imaju ugovor s pozivateljem

Točan odgovor je: sve metode imaju ugovor s pozivateljem

Pitajte 17

Nalazite:

Svoj bodova: 1,00
od 1,00

Opis: Odmot pitanje

Koja temeljna značajka je svojstvena modelom nadimnog oblikovanja programske potpore?

Odaberite jedan odgovor:

- a. evalvacija sustava
- b. specifikacija sustava
- c. uvođenje analize zahtjeva
- d. sve navedeno
- e. uvođenje modela sustava ✗

Točan odgovor je: uvođenje modela sustava

Pitajte 18

Nalazite:

Svoj bodova: 1,00
od 1,00

Opis: Odmot pitanje

Princip oblikovanja kojemu je cilj osigurati sakrivanje ili odgodu razmatranja detalja te time i smanjenje složenosti naziva se:

Odaberite jedan odgovor:

- a. Oblikuj za fleksibilnost
- b. Povećaj ponovnu uporabljivost
- c. Zadrži razinu apstrakcije ✗
- d. Planira; zastavi
- e. Oblikuj po ugovoru

Vaš odgovor je točan.

Točan odgovor je: Zadrži razinu apstrakcije

Pitanje 19

Takmičenje

Družište/član: 1.00

od: 1.00

V: Oznaci pitanje

Razred može implementirati više sučelja.

Odaberite jedan odgovor:

- Točno ✓
- Netočno

Točan odgovor je 'Da'.

Pitanje 20

Takmičenje

Družište/član: 0.00

od: 1.00

V: Oznaci pitanje

Kod preopterećivanja (overloading) metode imaju **jednak** ✗ naziv, a **različiti** ✗ broj i tip parametara.

Pitanje 21

Takmičenje

Družište/član: 0.00

od: 1.00

V: Oznaci pitanje

Metode razreda deklaraju se unutar razreda klijentom (npr.) **public** ✗ i pozivaju se preko **razreda** ✗.

Pitanje 22

Takmičenje

Družište/član: 0.00

od: 1.00

V: Oznaci pitanje

Moguce je kreirati instance saobraćajnog razreda ali je tada potreban modifikator **abstract**.

Odaberite jedan odgovor:

- Točno ✗
- Netočno

Točan odgovor je 'Ne'.

Pitanje 23

Takmičenje

Družište/član: 1.00

od: 1.00

V: Oznaci pitanje

Kako u objektno usmjerenoj terminologiji nazivamo predložak/specifikacija koja definira vrstu objekta?

Odaberite jedan odgovor:

- a. metoda (engl. method)
- b. Razred (engl. class) ✗
- c. Prototip (engl. prototype)
- d. Okvir (engl. frame)

Točan odgovor je: Razred (engl. class).

Pitanje 24

Netočno

Broj bodova: 1,00
od 1,00

Poznati pitanje

Konstruktore (engl. constructor) upotrebljavamo za:

- Odaberite jedan odgovor:
- a. Izgradnju sučelja objekata
 - b. stvaranje podrazreda
 - c. Dinamičko povezivanje objekata
 - d. oslobađanje memorije
 - e. Inicijalizaciju novih objekta ✓
 - f. ništa od navedenog

Točan odgovor je: Inicijalizaciju novih objekta

Pitanje 25

Netočno

Broj bodova: 0,00
od 1,00

Poznati pitanje

U procesu oblikovanja programske potpore, arhitektura programa rezultuje niski X - level dizajna, a implementacija visoki X - /eve dizajna.**Pitanje 26**

Netočno

Broj bodova: 0,00
od 1,00

Poznati pitanje

Dostupnost X varijable određuje koje objekte ta varijable može referencirati.

Točan odgovor je: Tip

Pitanje 27

Netočno

Broj bodova: 0,00
od 1,00

Poznati pitanje

Koje su dvije osobine zajedničke svim objektima u OOT?

Odaberite jedan odgovor:

- a. stanje i ponavljanje
- b. Metode i sučelja
- c. Metode i ponuke X

Točan odgovor je: stanje i ponavljanje

Pitanje 28

Netočno

Broj bodova: 0,00
od 1,00

Poznati pitanje

Koje od navedenih tvrdnji vrijede za odnos agregacije u UML dijagramu:

Odaberite jedan ili više odgovora:

- a. agregat predstavlja IS-A odnos X
- b. agregacija je slična kompoziciji
- c. agregat predstavlja IS-PART-OF odnos
- d. kad se uništiti agregat uništavaju se i svi njegovi djelovi

Točan odgovor je: agregat predstavlja IS-PART-OF odnos, agregacija je slična kompoziciji

Pitanje 29

Testni

Štaj brokava: 1.00
za 1.00

✓ Oznaci pitanje

Osnovnu opise funkcionalnih zahtjeva u dokumentaciji projekta čine:

Odaberite jedan ili više odgovora:

- a. UML dijagrami razreda
- b. UML dijagrami okreznica uporabe ✓
- c. UML sekvenčijski dijagrami ✓
- d. UML dijagrami stanja

Vaš odgovor je točan.

Točan odgovor je: UML sekvenčijski dijagrami, UML dijagrami okreznica uporabe

Pitanje 30

Testni

Štaj brokava: 1.00
za 1.00

✓ Oznaci pitanje

Naredba Subversion sustava kojom se promjene lokalnog rezervorija prenose u središnji rezervorij je:

Odaberite jedan odgovor:

- a. update
- b. checkout ✘
- c. add
- d. checkin

Vaš odgovor nije točan.

Točan odgovor je: checkin

Pitanje 1

Fakultet

Uraž budžeta: 1,00
od 1,00

P: Oznaci pitanje

Koje su temeljne značajke dobrog programskog produkta?

Odaberite jedan ili više odgovora:

- a. mora biti vizualno lijep
- b. mora imati dobru reklamu
- c. mora se moći održavati ✓
- d. mora biti privlačljiv korisniku ✓
- e. mora biti pouzdan ✓

Točan odgovor je: mora biti privlačljiv korisniku, mora biti pouzdan, mora se moći održavati

Pitanje 2

Fakultet

Uraž budžeta: -0,25
od 1,00

P: Oznaci pitanje

Zahjevi sustava mogu se izraziti:

Odaberite jedan odgovor:

- a. Strukturiranim prirodnim jezikom (formulari i obresci)
- b. Specijalnim jezikom za opis otkrivanja (SDL)
- c. Grafičkom notacijom (UML) X
- d. Matematičkom specifikacijom (FBM...)
- e. Sve od navedenog

Točan odgovor je: Sve od navedenog

Pitanje 3

Fakultet

Uraž budžeta: 1,00
od 1,00

P: Oznaci pitanje

U iteracije spiralnog modela procesa inženjerstva zahjeva ne pripadaju:

Odaberite jedan ili više odgovora:

- a. Izlučivanje zahjeva
- b. Studija izvedivosti ✓
- c. Validacija zahjeva
- d. Izrada prototipa ✓
- e. Specifikacija zahjeva

Točan odgovor je: izrada prototipa, Studija izvedivosti

Pitanje 4

Fakultet

Uraž budžeta: 1,00
od 1,00

P: Oznaci pitanje

Ethnografija je:

Odaberite jedan odgovor:

- a. pregled utjecaja svih dionika na zahjeve sustava
- b. metoda vrze izrade prototipa
- c. zahjevi izvedeni temeljem istraživanja kako ljudi stvarno rade ✓
- d. kvalitativno promatranje i opis ponavljanja ljudi pri obavljanju poslova
- e. metoda struktuiranja formalnih poslovnih procesa u svrhu izlucivanja zahjeva

Pitanje 5

Netočno

Broj bodova: -0,33
od 1,00

▼ Oznaci pitanje

Osnovni elementi UML-a su:

Odaberite jedan odgovor:

- a. dijagrami i tekstualni opisi
- b. aktori, relacije i objekti ✗
- c. stvari, relacije i dijagrami
- d. ništa od navedenog

Točan odgovor je: stvari, relacije i dijagrami

Pitanje 6

Točno

Broj bodova: 1,00
od 1,00

▼ Oznaci pitanje

Strukturne stvari (engl. *structural things*) u jeziku UML služe za opis:

Odaberite jedan odgovor:

- a. grupa, paketa
- b. interakcija, automata stanja
- c. opisnih elemenata
- d. razredi, sučelja, slučajevi korištenja, komponentni čvorovi, aktori i sl. ✓

Točan odgovor je: razredi, sučelja, slučajevi korištenja, komponentni čvorovi, aktori i sl.

Pitanje 7

Točno

Broj bodova: 1,00
od 1,00

▼ Oznaci pitanje

Za opis upravljačkog toka (engl. control flow) UML sekveničkim dijagramom vrijede tvrdnje:

Odaberite jedan odgovor:

- a. prema definiciji UML sekvenički dijagrami ne smiju opisivati upravljački tok u vremenu
- b. moguće je opisati uvjetna grananja, a petlje ne.
- c. moguće je opisati uvjetno grananje i petlje. ✓
- d. nije moguće opisati uvjetno grananje ni petlje.

Točan odgovor je: moguće je opisati uvjetno grananje i petlje.

Pitanje 8

Totno

Broj bodova: 1,00
od 1,00

Označi pitanje

Kod oblikovanja kojih zahtjeva se koriste UML obrasci uporabe?

Odaberite jedan ili više odgovora:

- a. zahtjeva kvalitete
- b. funkcionalnih zahtjeva ✓
- c. zahtjeva procesa oblikovanja
- d. zahtjeva platforme ostvarenja

Točan odgovor je: funkcionalnih zahtjeva

Pitanje 9

Netočno

Broj bodova: -0,33
od 1,00

Označi pitanje

Koji od navedenih tipa poveznica NE postoji u dijagramu obrazaca uporabe?

Odaberite jedan odgovor:

- a. proširenje (extension)
- b. sve navedene poveznice postoje ✗
- c. obuhvaćanje (inclusion)
- d. kompozicija (composition)
- e. asocijacija (association)

Točan odgovor je: kompozicija (composition)

Pitanje 10

Djelomično točno

Broj bodova: 0,40
od 1,00

▼ Oznaci pitanje

Princip višestrukog korištenja u oblikovanju programske potpore koristi se u:

Odaberite jedan ili više odgovora:

- a. ponovnom korištenju fragmenata rješenja (knjižnice) ✓
- b. ponovnom korištenju konzistencije (programski jezici)
- c. ponovnom korištenju dijelova arhitekture (arhitekturni obrasci - engl. architectural patterns)
- d. ponovnom korištenju arhitekture (radni okviri - engl. frameworks) ✓
- e. ponovnom korištenju cijelokupne arhitekture sustava

Točan odgovor je: ponovnom korištenju konzistencije (programski jezici), ponovnom korištenju fragmenata rješenja (knjižnice), ponovnom korištenju dijelova arhitekture (arhitekturni obrasci - engl. architectural patterns), ponovnom korištenju arhitekture (radni okviri - engl. frameworks), ponovnom korištenju cijelokupne arhitekture sustava

Pitanje 11

Netočno

Broj bodova: 0,00
od 1,00

▼ Oznaci pitanje

Pristup zasnovan na razvoju, oblikovanju i isporuci vrlo malih inkremenata funkcionalnosti svojstven je kojoj metodologiji razvoja programske potpore?

Odaberite jedan odgovor:

- a. iterativnoj
- b. spiralnoj ✗
- c. RUP
- d. agilnoj

Vaš odgovor nije točan.

Točan odgovor je: agilnoj

Pitanje 12

Točno

Broj bodova: 1,00
od 1,00

▼ Oznaci pitanje

Koje se tri aktivnosti ponavljaju tijekom razvoja u evolucijskom modelu?

Odaberite jedan odgovor:

- a. Specifikacija, razvoj, validacija ✓
- b. Analiza, implementacija, ispitivanje
- c. Specifikacija, oblikovanje, ispitivanje
- d. Analiza, oblikovanje, implementacija

Pitanje 13

Netočno

Broj bodova: 0,00
od 1,00

▼ Oznaci pitanje

Što nije sektor u spiralnom modelu iteracija programskog inženjerstva?

Odaberite jedan odgovor:

- a. Razvoj i validacija X
- b. Postavljanje ciljeva
- c. Pridjeljivanje zahtjeva inkrementima
- d. Planiranje
- e. Procjena i smanjivanje rizika

Točan odgovor je: Pridjeljivanje zahtjeva inkrementima

Pitanje 14

Netočno

Broj bodova: 0,00
od 1,00

▼ Oznaci pitanje

U drugu fazu unificiranog procesa (engl. *Unified Process*) (Faza elaboracije) ne spada:

Odaberite jedan ili više odgovora:

- a. Specifikacija značajki.
- b. Izrada plana projekta. X
- c. Definiranje temelja arhitekture sustava.
- d. Oblikovanje produkta.
- e. Definiranje dosega projekta.

Točan odgovor je: Definiranje dosega projekta., Oblikovanje produkta.

Pitanje 15

Netočno

Broj bodova: 0,25
od 1,00

▼ Oznaci pitanje

Koja od navedenih je temeljna značajka vodopadnog modela:

Odaberite jedan odgovor:

- a. Funkcionalnosti se razvijaju postepeno, tako da se u svakom koraku implementira nova funkcionalnost X
- b. Pojedina faza se mora dovršiti prije pokretanja nove faze
- c. Korisnik zna što treba i nisu mu potrebna pojašnjenja
- d. Zahtjevi se rijetko mijenjaju

Točan odgovor je: Pojedina faza se mora dovršiti prije pokretanja nove faze

Pitanje 16

Netočno

Broj bodova: 0,00
od 1,00

▼ Oznaci pitanje

Dopunite slijedeću tvrdnju: „Obrasci uporabe specificiraju **način uporabe**  , a arhitektura **izvedbu**  .

Pitanje 17

Točno

Broj bodova: 1,00
od 1,00

▼ Oznaci pitanje

Princip oblikovanja koji se temelji na obradi svih slučajeva u kojima se neka komponenta može neprikladno upotrijebiti naziva se:

Odaberite jedan odgovor:

- a. Oblikovanje po ugovoru
- b. Podijeli pa vladaj
- c. Oblikovanje za ispitivanje
- d. Konzervativno oblikovanje 
- e. Oblikovanje za fleksibilnost

Vaš odgovor je točan.

Točan odgovor je: Konzervativno oblikovanje

Pitanje 18

Netočno

Broj bodova: 0,00
od 1,00

▼ Oznaci pitanje

Princip oblikovanja povećanja ponovne uporabivosti u pristupu oblikovanju je komplementaran principu:

Odaberite jedan odgovor:

- a. Povećaj uporabu postojećeg
- b. Planiraj zastaru
- c. Oblikuj za prenosivost 
- d. Oblikuj za ispitivanje
- e. Oblikuj po ugovoru

Vaš odgovor nije točan.

Točan odgovor je: Povećaj uporabu postojećeg

Pitanje 19

Netočno

Broj bodova: 0,33
od 1,00

Oznaci pitanje

Koji od navedenih pojmlja NE predstavlja neki od osnovnih principa objektne paradigme?

Odaberite jedan odgovor:

- a. modularnost
- b. apstrakcija
- c. enkapsulacija
- d. Ipak se svi odnose na objektnu paradigmu. X
- e. nasljeđivanje

Točan odgovor je: nasljeđivanje

Pitanje 20

Točno

Broj bodova: 1,00
od 1,00

Oznaci pitanje

Svojstvo da različiti razredi imaju funkcije istog imena nazivamo:

Npr.

```
class A {
    int a;
public:
    A() { a = 0; }
    void X(){ a++; }
    void Y(){ a--; }
};

class B: public A {
public:
    void X(){ a = a + 2; }
    void Y(){ a = a - 2; }
};
```

Odaberite jedan odgovor:

- a. polimorfizam ✓
- b. dinamičko povezivanje
- c. statičko povezivanje
- d. nasljeđivanje
- e. overriding ("nadjačavanje", "neobaziranje")

Točan odgovor je: polimorfizam

Pitanje 21

Očekomjeno točno

Broj bodova: 0.50
od 1.00

Oznaci pitanje

Metode razreda deklariraju se unutar razreda ključnom riječi **public** i pozivaju se preko **imena razreda** .

Pitanje 22

Netočno

Broj bodova: 0.00
od 1.00

Oznaci pitanje

Moguće je kreirati instance apstraktog razreda ali je tada potreban modifikator **abstract**.

Odaberite jedan odgovor:

- Točno
- Netočno

Točan odgovor je 'Ne'.

Pitanje 23

Točno

Broj bodova: 1.00
od 1.00

Oznaci pitanje

Kako u objektno usmjerenoj terminologiji nazivamo predložak(specifikaciju) koja definira vrstu objekta?

Odaberite jedan odgovor:

- a. Prototip (engl. prototype)
- b. Okvir (engl. frame)
- c. Razred (engl. class)
- d. metoda (engl. method)

Točan odgovor je: Razred (engl. class)

Pitanje 24

Netočno

Broj bodova: 0,00
od 1,00

▼ Oznaci pitanje

Što se događa tijekom izvođenja OO aplikacije?

Odaberite jedan odgovor:

- a. skup zadnjih objekata se učitava u memoriju
- b. dio modela objekata se nalazi u memoriji i dinamički se stvaraju/ brišu
- c. svi objekti se stvaraju u memoriji prema odgovarajućem dijagramu klasa
- d. model objekata se instancira iz baze podataka prema potrebi nakon čega se objekti uništavaju ✗

Točan odgovor je: dio modela objekata se nalazi u memoriji i dinamički se stvaraju/ brišu

Pitanje 25

Netočno

Broj bodova: 0,00
od 1,00

▼ Oznaci pitanje

U procesu oblikovanja programske potpore, arhitektura programa rezultat je niski ✗ - level dizajna, a implementacijavisoki ✗ - level dizajna.**Pitanje 26**

Netočno

Broj bodova: 0,00
od 1,00

▼ Oznaci pitanje

Dostupnost ✗ variabile određuje koje objekte ta varijabla može referencirati.

Točan odgovor je: Tip

Pitanje 27Ocjenočno točno
Broj bodova: 0,50
od 1,00

▼ Oznaci pitanje

Proces kojim jedan objekt može stići svojstva nekog drugog objekta nazivamo:

Odaberite jedan ili više odgovora:

- a. nasljeđivanje ✓
- b.instanciranje
- c. polimorfizam
- d. sakrivanje informacija
- e. enkapsulacija
- f. indoktrinacija

Pitanje 28

Točno

Broj bodova: 1,00
od 1,00

Oznaci pitanje

Odnos kompozicije i agregacije u UML dijagramu razreda je slijedeći:

Odaberite jedan odgovor:

- a. Agregacija je jača od kompozicija.
- b. Kompozicija je jača od agregacije. ✓ Zato što kad se uništi kompozit uništavaju se i svi sastavni dijelovi, dok to kod aggregata nije slučaj.

Točan odgovor je: Kompozicija je jača od agregacije.

Pitanje 29

Točno

Broj bodova: 1,00
od 1,00

Označi pitanje

UML dijagrami koji nisu nužni u Inačici 1.0 dokumentacije projekta su:

Odaberite jedan ili više odgovora:

- a. Dijagram razreda
- b. Dijagram komponenti ✓
- c. Dijagram aktivnosti ✓
- d. Sekvencijski dijagram
- e. Dijagram razmještaja ✓

Vaš odgovor je točan.

Točan odgovor je: Dijagram razmještaja, Dijagram aktivnosti, Dijagram komponenti

Pitanje 30

Netočno

Broj bodova: 0,00
od 1,00

Označi pitanje

Neki od alata za upravljanje inačicama datoteka unutar projekta su:

Odaberite jedan ili više odgovora:

- a. Astah Community ✗
- b. ArgoUML
- c. GIT
- d. Team Foundation Server (TFS)

Vaš odgovor nije točan.

Točan odgovor je: GIT, Team Foundation Server (TFS)

Pitanje 1

Točno

Mark 1,00 out of
1,00

Flag question

Razredi se grupiraju u pakete prema:

Odaberite jedan odgovor:

- a. organizaciji tima
- b. dijagramu razreda u kojem se pojavljuju
- c. razredima koji su definirani zajedno na sekvencijskom dijagramu
- d. razredima koji su povezani s zajedničkom uslugom ✓

Točan odgovor je: razredima koji su povezani s zajedničkom uslugom.

Pitanje 2

Točno

Mark 1,00 out of
1,00

Flag question

Za apstraktni razred (engl. abstract class) vrijedi:

Odaberite jedan odgovor:

- a. ne može seinstancirati ✓
- b. može sadržavati metode ali ne podatke
- c. ne sadrži podrazrede
- d. ne može se proširiti

Točan odgovor je: ne može seinstancirati.

Pitanje 3

Netočno

Mark 0,00 out of
1,00

Flag question

Što se zapravo događa kada se poruka šalje na objekt?

Odaberite jedan odgovor:

- a. inicijaliziraju se podaci objekta ✗
- b. pozva se njegova metoda
- c. postavljaju se vrijednosti privatnih varijabli

Točan odgovor je: pozva se njegova metoda.

Pitanje 4

Netočno

Mark 0,00 out of
1,00

Flag question

Kada se jedan razred temelji na drugom razredu, on ujedno nasljeđuje:

Odaberite jedan odgovor:

- a. Podatke i metode
- b. metode i poruke ✗
- c. samo vidljive podatke
- d. sve podatke

Točan odgovor je: Podatke i metode .

Pitanje 5

Djelomično točno

Mark 0,50 out of
1,00

Flag question

Koje od navedenih značajki ne spadaju u objektno usmjerenje:

Odaberite jedan ili više odgovora:

- a. udaljeni pozivi operacija
- b. instanciranje
- c. sakrivanje informacija
- d. polimorfizam
- e. višezadačnost ✓
- f. provjera tipova

Točan odgovor je: višezadačnost, udaljeni pozivi operacija.

Pitanje 6

Djelomično točno

Mark 0,50 out of
1,00

Flag question

Proces kojim jedan objekt može stići svojstva nekog drugog objekta nazivamo:

Odaberite jedan ili više odgovora:

- a. polimorfizam
- b. sakrivanje informacija
- c. indoktrinacija
- d. instanciranje
- e. enkapsulacija
- f. nasljeđivanje ✓

Točan odgovor je: nasljeđivanje, enkapsulacija.

Pitanje 7

Točno

Mark 1,00 out of
1,00

Flag question

Kako u objektno usmjerenoj terminologiji nazivamo predložak(specifikaciju) koja definira vrstu objekta?

Odaberite jedan odgovor:

- a. Prototip (engl. prototype)
- b. Okvir (engl. frame)
- c. Razred (engl. class) ✓
- d. metoda (engl. method)

Točan odgovor je: Razred (engl. class).

Pitanje 8

Točno

Mark 1,00 out of
1,00

Flag question

Za Liskov princip zamjene (engl. *Liskov Substitution Principle*) vrijedi:

Odaberite jedan ili više odgovora:

- a. Ako je objekt proširenje drugog objekta tada ga je moguće uporebljavati na svim mjestima tog objekta ✓
- b. jedan objekt može sigurno zamijeniti drugi samo u istom sloju arhitekture
- c. odnosi se na interoperabilnost objekta u hijerarhiji nasljeđivanja ✓
- d. objekt koji proširuje drugi objekt može nadjačati (engl. override) njegove metode

Točan odgovor je: Ako je objekt proširenje drugog objekta tada ga je moguće uporebljavati na svim mjestima tog objekta, odnosi se na interoperabilnost objekta u hijerarhiji nasljeđivanja.

Pitanje 9

Točno

Mark 1,00 out of
1,00

Flag question

Metode uvijek moraju vratiti vrijednost pozivatelju (uspjeh)?

Odaberite jedan odgovor:

- a. netočno ✓
- b. točno

Točan odgovor je: netočno.

Pitanje 10

Točno

Mark 1,00 out of
1,00

Flag question

Odnos podrazreda prema svojem superrazredu (engl. superclass) nazivamo:

Odaberite jedan odgovor:

- a. obuhvaćanje (engl. encapsulation)
- b. Dinamičko povezivanje (engl. dynamic linking)
- c. agregacija (engl. aggregation)
- d. specijalizacija ✓
- e. ništa od navedenog
- f. apstrakcija (engl. abstraction)

Točan odgovor je: specijalizacija.

Pitanje 11

Točno

Mark 1,00 out of
1,00

Flag question

Svojstvo da različiti razredi imaju funkcije istog imena nazivamo:

Npr.

```
class A {  
    int a;  
    public:  
        A() { a = 0; };  
        void X(){ a++; };  
        void Y(){ a--; };  
};  
class B: public A {  
    public:  
        void X(){ a = a + 2; };  
        void Y(){ a = a - 2; };  
};
```

Odaberite jedan odgovor:

- a. overriding ("nadjačavanje", "neobaziranje")
- b. dinamičko povezivanje
- c. nasljedivanje
- d. polimorfizam ✓
- e. statičko povezivanje

Točan odgovor je: polimorfizam.

Pitanje 12

Netočno

Mark 0,00 out of
1,00

Flag question

Na glavnoj metodi programa smijemo koristiti nadjačavanje metoda (engl. method overriding)?

Odaberite jedan odgovor:

- a. netočno
- b. točno ✗
- c. samo u iznimnim slučajevima

Točan odgovor je: netočno.

Pitanje 13

Točno

Mark 1,00 out of
1,00

Flag question

Agregacija se razlikuje od kompozicije

Odaberite jedan odgovor:

- a. ništa od navedenog
- b. agregirani objekt nasljeđuje svojstva pojedinih objekata, a objekt nastao kompozicijom nasljeđuje svojstva uporabom drugih razreda
- c. članovi agregacije moraju biti instancirani u jednom koraku
- d. članovi kompozicije moraju biti instancirani u jednom koraku
- e. objekti nastali agregacijom imaju postojanje nezavisno od njihovih članova ✓

Točan odgovor je: objekti nastali agregacijom imaju postojanje nezavisno od njihovih članova .

Pitanje 14

Netočno

Mark 0,00 out of
1,00

Flag question

Odnos podrazred prema svojem superrazredu (engl. superclass) nazivamo:

Odaberite jedan odgovor:

- a. ništa od navedenog
- b. agregacija (engl. aggregation)
- c. specijalizacija
- d. apstrakcija (engl. abstraction)
- e. Dinamičko povezivanje (engl. dynamic linking)
- f. obuhvaćanje (engl. encapsulation) ✗

Točan odgovor je: specijalizacija.

Pitanje 15

Netočno

Mark 0,00 out of
1,00

Flag question

Konstruktore (engl. constructor) upotrebljavamo za:

Odaberite jedan odgovor:

- a. oslobađanje memorije
- b. izgradnju sučelja objekata ✗
- c. stvaranje podrazreda
- d. ništa od navedenog
- e. Dinamičko povezivanje objekata
- f. inicijalizaciju novih objekta

Točan odgovor je: inicijalizaciju novih objekta.

Pitanje 1

Točno

Broj bodova: 1,00
od 1,00

▼ Označi pitanje

Kao dioniku u ocjeni kakvoće programskog produkta kupcu je najvažnije:

Odaberite jedan odgovor:

- a. Nijedno od navedenog ✓
- b. Lakoća ponovne uporabe dijelova
- c. Lakoća održavanja
- d. Sve navedeno
- e. Lakoća oblikovanja

Točan odgovor je: Nijedno od navedenog

Pitanje 2

Djelomično točno

Broj bodova: 0,55
od 1,00

▼ Označi pitanje

U inženjerstvu zahtjeva programske potpore za zahtjeve vrijede tvrdnje:

Odaberite jedan ili više odgovora:

- a. to su opisi ograničenja sklopovlja
- b. to su utvrđena ograničenja implementirane programske potpore
- c. izražavaju se isključivo posebnim jezicima za oblikovanje sustava, dijagramima i matematičkom notacijom
- d. to su opisi usluga sustava ✓
- e. to su ograničenja programske potpore ✓
- f. mogu se izražavati prirodnim jezikom

Točan odgovor je: to su opisi usluga sustava, to su opisi ograničenja sklopovlja, to su ograničenja programske potpore, mogu se izražavati prirodnim jezikom

Pitanje 3

Netočno

Broj bodova: 0,00
od 1,00

▼ Označi pitanje

U iteracije spiralnog modela procesa inženjerstva zahtjeva **ne pripadaju**:

Odaberite jedan ili više odgovora:

- a. Specifikacija zahtjeva X
- b. Izlučivanje zahtjeva
- c. Validacija zahtjeva X
- d. Studija izvedivosti
- e. Izrada prototipa

Točan odgovor je: Izrada prototipa, Studija izvedivosti

Pitanje 4

Točno

Broj bodova: 1,00
od 1,00

▼ Označi pitanje

Ako svi zahtjevi obuhvaćeni u dokumentu inženjerstva zahtjeva imaju samo jednu moguću interpretaciju tada možemo tvrditi da je dokument specifikacije zahtjeva:

Odaberite jedan odgovor:

- a. moguće verificirati
- b. sve navedeno
- c. Ispravan
- d. jednoznačan ✓
- e. konzistentan

Točan odgovor je: jednoznačan

Pitanje 5

Točno

Broj bodova: 1,00
od 1,00

▼ Označi pitanje

U koje skupine UML dijagrama spadaju dijagrami obrazaca uporabe i sekvenčni dijagrami?

Odaberite jedan odgovor:

- a. Oba su dinamični tipovi dijagrama.
- b. Dijagram obrazaca uporabe je dinamični tip dijagrama, sekvenčni dijagram je statični tip dijagrama.
- c. Oba su statični tipovi dijagrama.
- d. Dijagram obrazaca uporabe je statični tip dijagrama, sekvenčni dijagram je dinamični tip dijagrama. ✓

Točan odgovor je: Dijagram obrazaca uporabe je statični tip dijagrama, sekvenčni dijagram je dinamični tip dijagrama.

Pitanje 6

Točno

Broj bodova: 1,00
od 1,00

▼ Označi pitanje

Stvarima označavanja odgovara kao UML opis:

Odaberite jedan odgovor:

- a. paketi
- b. razredi, sučelja, slučajevi korištenja, komponentni čvorovi, aktori i sl.
- c. oznake elemenata modela ✓
- d. interakcije i automati stanja

Točan odgovor je: oznake elemenata modela

Pitanje 7

Djelomično točno

Broj bodova: 0,50
od 1,00

▼ Označi pitanje

Za sinkrone poruke u sustavu vrijede slijedeće tvrdnje (moguće više točnih odgovora):

Odaberite jedan ili više odgovora:

- a. pošiljatelj čeka na odgovor ✓
- b. na sekvencijskom dijagramu odgovor se ne prikazuje (podrazumijeva se)
- c. pošiljatelj ne čeka na odgovor
- d. na sekvencijskom dijagramu odgovor se prikazuje isprekidanim crtom

Točan odgovor je: pošiljatelj čeka na odgovor, na sekvencijskom dijagramu odgovor se ne prikazuje (podrazumijeva se)

Pitanje 8

Netočno

Broj bodova: -0,33
od 1,00

▼ Označi pitanje

Granica sustava (*boundary*) u UML dijagramu obrazaca uporabe (*use case diagram*) predstavlja:

Odaberite jedan odgovor:

- a. granicu između aktora i sustava
- b. grafički element koji postoji radi bolje preglednosti dijagrama ✗
- c. granicu između različitih obrazaca uporabe
- d. ništa od navedenog

Točan odgovor je: granicu između aktora i sustava

Pitanje 9

Točno

Broj bodova: 1,00
od 1,00

▼ Označi pitanje

Vrsta poveznice u dijagramu obrazaca uporabe s prikazom:



Naziva se:

Odaberite jedan odgovor:

- a. generalizacija
- b. proširenje
- c. asocijacija ✓
- d. uključivanje

Točan odgovor je: asocijacija

Pitanje 10

Djelomično točno

Broj bodova: 0,33
od 1,00

▼ Označi pitanje

Što je potrebno uključiti u dokument oblikovanja arhitekture programske potpore?

Odaberite jedan ili više odgovora:

- a. Prioritete koji su vodili proces oblikovanja
- b. Dobro komentirani kod
- c. Referenciju prema dokumentu zahtjeva
- d. Opis s najviše razine promatranja ✓
- e. Alternativna rješenja

Točan odgovor je: Prioritete koji su vodili proces oblikovanja, Referenciju prema dokumentu zahtjeva, Opis s najviše razine promatranja

Pitanje 11

Točno

Broj bodova: 1,00
od 1,00

▼ Označi pitanje

Odredite istinite tvrdnje za svojstva modela procesa programskog inženjerstva:

Odaberite jedan ili više odgovora:

- a. Iteracije su primjenjive samo na RUP model.
- b. RUP (engl. Rational Unified Process) je najstariji model.
- c. Implementacija promjena je najlakša u vodopadnom modelu.
- d. Najmanje novog koda generira se u modelu zasnovanom na komponentama. ✓
- e. Evolucijski model generira najbolju strukturu.

Vaš odgovor je točan.

Točan odgovor je: Najmanje novog koda generira se u modelu zasnovanom na komponentama.

Pitanje 12

Točno

Broj bodova: 1,00
od 1,00

▼ Označi pitanje

Koji je model procesa programskog inženjerstva najpogodniji za male i srednje interaktivne sustave?

Odaberite jedan odgovor:

- a. vodopadni
- b. komponentno usmjeren
- c. RUP proces
- d. evolucijski ✓

Točan odgovor je: evolucijski

Pitanje 13

Netočno

Broj bodova: 0,00
od 1,00

▼ Označi pitanje

Što vrijedi za spiralni model razvoja programske potpore?

Odaberite jedan odgovor:

- a. barem jedan dio specifikacija mora biti definiran prije njihovog kodiranja
- b. većina ispitnih slučajeva mora biti završena prije početka implementacije ✗
- c. nije potrebna
- d. barem jedan dio implementacije mora biti završen prije pisanja ispitnih slučajeva
- e. većina specifikacija je definirana prije kodiranja

Točan odgovor je: barem jedan dio specifikacija mora biti definiran prije njihovog kodiranja

Pitanje 14

Točno

Broj bodova: 1,00
od 1,00

▼ Označi pitanje

Koje od navedenih su glavne značajke unificiranog procesa (*moguće više točnih odgovora*)?

Odaberite jedan ili više odgovora:

- a. Obrasci uporabe (engl. *use cases*) ✓
- b. Fokus na arhitekturi sustava ✓
- c. Iterativni i inkrementalni pristup ✓
- d. Uporaba matematičke specifikacije
- e. Programiranje u paru (engl. *pairwise programming*)

Točan odgovor je: Iterativni i inkrementalni pristup, Obrasci uporabe (engl. *use cases*), Fokus na arhitekturi sustava

Pitanje 18

Nije odgovoren

Broj bodova od 1,00

▼ Označi pitanje

Dopunite slijedeću tvrdnju: „Obrasci uporabe specifickiraju X, a arhitektura X.

funkcije

formu

Pitanje 19

Nije odgovoren

Broj bodova od 1,00

▼ Označi pitanje

Koje su od navedenih tvrdnji točne?

Odaberite jedan ili više odgovora:

- a. Polimorfizam se razrješava staticki.
- b. Overloading predstavlja inačicu polimorfizma.
- c. Overriding predstavlja inačicu polimorfizma.
- d. Polimorfizam označava postojanje više metoda istog naziva, tipa te broja i vrste parametara u podrazredima koji implementiraju istu apstraktnu operaciju superrazreda.
- e. Svi podrazredi implicitno nasljeđuju sve značajke definirane u superrazredu.

Točan odgovor je: Polimorfizam označava postojanje više metoda istog naziva, tipa te broja i vrste parametara u podrazredima koji implementiraju istu apstraktnu operaciju superrazreda., Overriding predstavlja inačicu polimorfizma., Svi podrazredi implicitno nasljeđuju sve značajke definirane u superrazredu.

Pitanje 20

Nije odgovoreno

Broj bodova od 1,00

▼ Označi pitanje

U primjeru:

```
class A {  
    int a;  
    public:  
        A() { a = 0; };  
        void X(){ a++; };  
        void Y(){ a--; };  
};  
class B: public A {  
    public:  
        void X(){ a = a + 2; };  
        void Y(){ a = a - 2; };  
};  
...  
//u glavnom programu  
class A *m, *n;  
m = new A();  
m->Y();  
n = new B();  
n->Y();
```

razred B nanovo definira funkcije X i Y.

Postupak nazivamo:

Odaberite jedan odgovor:

- a. dinamičko povezivanje
- b. polimorfizam
- c. statičko povezivanje
- d. nadjačavanje (engl. overriding)
- e. nasljeđivanje

Točan odgovor je: nadjačavanje (engl. overriding)

Pitanje 21

Nije odgovoreno

Broj bodova od 1,00

▼ Označi pitanje

Ako razred nasljeđuje apstraktni razred mora implementirati sve apstraktne metode da bi postao konkretn.

Odaberite jedan odgovor:

- Točno
- Netočno

Točan odgovor je 'Da'.

Pitanje 29

Djelomično točno

Broj bodova: 0,50
od 1,00

▼ Označi pitanje

Osnovu opisa funkcionalnih zahtjeva u dokumentaciji projekta čine:

Odaberite jedan ili više odgovora:

- a. UML sekvencijski dijagrami
- b. UML dijagrami stanja
- c. UML dijagrami razreda
- d. UML dijagrami obrazaca uporabe ✓

Vaš odgovor je djelomično točan.

Broj točnih odgovora: 1

Točan odgovor je: UML sekvencijski dijagrami, UML dijagrami obrazaca uporabe

Pitanje 30

Nije odgovoreno

Broj bodova od 1,00

▼ Označi pitanje

Za sustav *Subversion* vrijede tvrdnje:

Odaberite jedan ili više odgovora:

- a. Središnje mjesto pohrane je uvijek jedan od lokalnih rezitorija (prema dogovoru).
- b. Inačice datoteka s različitim brojevima revizije nužno su različite.
- c. Sporovi se rješavaju isključivo ručnim spajanjem promjena u datotekama.
- d. Izbrisane datoteke je moguće povratiti.
- e. Broj revizije globalno određuje stanje rezitorija u određenom trenutku.

Vaš odgovor nije točan.

Točan odgovor je: Izbrisane datoteke je moguće povratiti., Broj revizije globalno određuje stanje rezitorija u određenom trenutku.

Pitanje 15

Netočno

Broj bodova: 0,00
od 1,00

▼ Označi pitanje

Koji model procesa programskog inženjerstva je najpogodniji za velike inženjerske projekte gdje se sustav razvija na nekoliko odvojenih mesta?

Odaberite jedan odgovor:

- a. vodopadni
- b. RUP proces ✗
- c. komponentno usmjeren
- d. evolucijski

Točan odgovor je: vodopadni

Pitanje 16

Točno

Broj bodova: 1,00
od 1,00

▼ Označi pitanje

Koji od ponuđenih odgovora NE PREDSTAVLJA jedan od 12 principa oblikovanja?

Odaberite jedan odgovor:

- a. povećaj ponovnu iskoristivost
- b. smanji međuvisnost
- c. oblikuj za ispitivanje
- d. podijeli pa vladaj
- e. oblikuj za budućnost ✓

Točan odgovor je: oblikuj za budućnost

Pitanje 17

Točno

Broj bodova: 1,00
od 1,00

▼ Označi pitanje

Koji principi oblikovanja su direktno povezani s principom *povećanja ponovne uporabivosti (reusability)*?

Odaberite jedan ili više odgovora:

- a. viša razina apstrakcije ✓
- b. smanjenje međuvisnosti ✓
- c. povećanje kohezije ✓
- d. povećanje uporabe postojećeg

Točan odgovor je: povećanje kohezije, smanjenje međuvisnosti, viša razina apstrakcije

Pitanje 26

Nije odgovoreno

Broj bodova od 1,00

▼ Označi pitanje

Koji od ponuđenih odgovora predstavlja ispravan način instanciranja i pristupa **varijabli razreda** tipa MojRazred?

Odaberite jedan odgovor:

- a. `MojRazred mojaVarijabla = new MojRazred();`
`tmp = mojaVarijabla.svojstvol; //tmp definiran ranije u kodu`
- b. i a i b je ispravno;
- c. `tmp = MojRazred.svojstvol; //tmp definiran ranije u kodu`

Točan odgovor je: `tmp = MojRazred.svojstvol; //tmp definiran ranije u kodu`

Pitanje 27

Nije odgovoreno

Broj bodova od 1,00

▼ Označi pitanje

Ako dva objekta imaju identične podatke (attribute) zapravo se radi o jednom te istom objektu koji je dvaput referenciran.

Odaberite jedan odgovor:

- Točno
- Netočno

Točan odgovor je 'Ne'.

Pitanje 28

Nije odgovoreno

Broj bodova od 1,00

▼ Označi pitanje

Koje od navedenih tvrdnji vrijede za odnos **agregacije** u UML dijagramu:

Odaberite jedan ili više odgovora:

- a. agregat predstavlja IS-PART-OF odnos
- b. agregat predstavlja IS-A odnos
- c. agregacija je slabija od kompozicije
- d. kad se uništi agregat uništavaju se i svi njegovi dijelovi

Točan odgovor je: agregat predstavlja IS-PART-OF odnos, agregacija je slabija od kompozicije

Pitanje 22

Nije odgovoreno

Broj bodova od 1,00

▼ Označi pitanje

Koja se od ponuđenih tvrdnji NE odnosi na pojam **razred** u objektnoj paradigmi?

Odaberite jedan odgovor:

- a. Razred sadrži podatkovne i proceduralne apstrakcije.
- b. Razred može biti apstrakcija koncepta iz stvarnog svijeta.
- c. Razred predstavlja jedan definirani tip podataka.
- d. Program u radu može se sagledati kao skup razreda u međusobnoj kolaboraciji.
- e. Sve navedene tvrdnje su točne.

Točan odgovor je: Program u radu može se sagledati kao skup razreda u međusobnoj kolaboraciji.

Pitanje 23

Nije odgovoreno

Broj bodova od 1,00

▼ Označi pitanje

Odnos podrazred prema svojem superrazredu (engl. superclass) nazivamo:

Odaberite jedan odgovor:

- a. Dinamičko povezivanje (engl. dynamic linking)
- b. obuhvaćanje (engl. encapsulation)
- c. agregacija (engl. aggregation)
- d. specijalizacija
- e. ništa od navedenog
- f. apstrakcija (engl. abstraction)

Točan odgovor je: specijalizacija

Pitanje 24

Nije odgovoreno

Broj bodova od 1,00

▼ Označi pitanje

Konstruktore (engl. constructor) upotrebljavamo za:

Odaberite jedan odgovor:

- a. ništa od navedenog
- b. oslobođanje memorije
- c. inicijalizaciju novih objekata
- d. izgradnju sučelja objekata
- e. Dinamičko povezivanje objekata
- f. stvaranje podrazreda

Točan odgovor je: inicijalizaciju novih objekta

Pitanje 25

Nije odgovoreno

Broj bodova od 1,00

▼ Označi pitanje

U procesu oblikovanja programske potpore, **arhitektura programa** rezultat je X - level dizajna, a **implementacija**

high

 X - /level/ dizajna.