

**1. Navedi generičke aktivnosti u svakome procesu programskog inženjerstva.**

Specifikacija, razvoj i oblikovanje, validacija i verifikacija, evolucija.

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

Evolucijski.

**3. Nabroji pet jezgrenih (temeljnih) aktivnosti (engl. Core Workflows) RUP modela procesa koje se pojavljuju s različitim intenzitetima u svim fazama.(četiri RUP faze su početak, elaboracija, konstrukcija i prijenos/tranzicija korisnicima).**

Model poslovnog procesa, zahtjevi, analiza i oblikovanje, implementacija, test.

*Potporne aktivnosti: menadžment konfiguracija i promjena, menadžment projekta, briga o okolišu*

**4. U inženjerstvu zahtjeva u oblikovanju programske potpore, navedi klasifikacije zahtjeva s obzirom na sadržaj.**

Funkcionalni zahtjevi, nefunkcionalni zahtjevi, zahtjevi domene pripreme.

**5. Nabroji četiri metode u izlučivanju zahtjeva u oblikovanju programske potpore.**

Intervjuiranje, scenarij, Izlučivanje i specficiranje zahtjeva obrascima uporabe, specficiranje dinamičkih interakcija u sustavu (UML sekvencijski dijagrami).

**6. Jedan od 12 principa oblikovanja i izbora arhitektura programske potpore traži povećanje kohezije komponenata(modula).Zašto se nastoji ostvariti taj princip (navedi barem dva osnovna razloga)?**

Olakšano razumijevanje, povećana ponovna uporabljivost modula, lakša zamjena.

**7. Zadani sekvencijski pretvoriti u komunikacijski.**

**8. Nešto o class dijagramu i polimorfizmu.**

**9. OCSF a) koje metode se obavezno moraju implementirati?**

HandleMessageFromServer(), HandleMessageFromClient()

**b) što se događa sa klijentima kad se pokrene metoda stoplistening na serveru?**

Poslužitelj prestane osluškivati za nove klijente, sa trenutnima i dalje komunicira.

**10. Neka je stvorena raspodijeljena arhitektura prema objektnom klijent-poslužiteljskom radnom okviru (OCSF). Ako poslužitelj ima spojena dva klijenta i provodi interakciju sa svojim administratorom, koliko je ukupno aktivnih dretvi na poslužiteljskoj strani?**

2 klijenta + 1 admin + 1 koja čeka nove konekcije

**11. Upisati DA i NE ovisno o standardizaciji na binarnoj razini i standardizaciji na razini izvornog koda.**

	Binarni kod	Izvorni kod
CORBA	NE	DA
.NET	DA	NE

**12. Navedite 3 entiteta koja čine temeljni model web uslužne arhitekture.(ne treba protokole)**

Registar usluga i posrednik, ponuditelj usluge, tražitelj usluge.

**13. Navedi barem dvije razlike između programske komponente u komponentno temeljenoj arhitekturi i objekta u objektno usmjerenoj arhitekturi programske potpore.**

Definicija objekta je tehnička; ne uključuje pojam nezavisnosti.

Kompozicija objekta nije namijenjena širokom krugu korisnika.

Ne postoji, niti će postojati, tržište objekata.

Objekti ne podržavaju paradigmu „plug-and-play“.

**14. Svako dijete je mlade od svoje majke.**

$\forall x (dijete(x) \Rightarrow \exists y (majka(y,x) \wedge mlađi(x,y)))$

**15. Tko god ima majku ima i oca.**

$\forall x (ima\_majku(x) \Rightarrow ima\_oca(x))$

**16. Ante ima barem 2 sestre. koristiti  $=(x,y)$  – x je jednak y**

$\exists x \exists y (sestra(x,Ante) \wedge sestra(y,Ante) \wedge \neg=(x,y))$

**17. pretvoriti u CNF.  $(P \wedge \neg Q) \vee (\neg R \vee P)$**

$(P \wedge \neg Q) \vee (\neg R \vee P) = [(P \wedge \neg Q) \vee \neg R] \vee P =$

$= [(\neg R \vee P) \wedge (\neg R \vee \neg Q)] \vee P =$

$= [P \vee (\neg R \vee P)] \wedge [P \vee (\neg R \vee \neg Q)] =$

$= (P \vee \neg R) \wedge (P \vee \neg R \vee \neg Q)$

kaj se, ako se ne varam, može skratiti na  $(P \vee \neg R)$

**18. Što znači ispravan (sound), a što kompletan (complete) formalni logički sustav koji sadrži skup logičkih izraza  $\{T\}$  i skup dozvoljenih pravila  $\{L\}$ .**

Sustav je **ispravan** ako  $\Gamma \models \omega_i$  kad god je  $\Gamma \vdash_L \omega_i$  tj. svaka pravilima definirana formula je ujedno i logička posljedica skupa  $\Gamma$ .

$\Gamma \vdash_L \omega_i$  **implicira**  $\Gamma \models \omega_i$ .

Sustav je **kompletan** ako je svaku logičku posljedicu skupa  $\Gamma$  moguće dokazati pravilima  $L$ .

$\Gamma \models \omega_i$  **implicira**  $\Gamma \vdash_L \omega_i$ .

**19. U ispitivanju programske potpore koristi se termin „testni slučaj“.Što je to?**

Jedan par: ulaz i očekivani izlaz.

**20. Koliko je najmanje potrebno testova za broj\_lukova=12, broj\_čvorova=8  
Nema dodatnih komponenti?**

$CV(G) = \text{Lukovi} - \text{Čvorovi} + 2 * P = 6$

**21. Zaokruži netočno:**

- a)  $A [p \cup EF \neg r]$
- b)  $AEFr$  – kvantifikatori ne mogu biti u niz
- c)  $FGr$  – operatori ne mogu biti u nizu
- d)  $AF [(r \cup g) \wedge (p \cup r)]$  – operatori ne mogu biti u nizu
- e)  $A [\neg p \cup A[q \cup r]]$
- f)  $A \neg G \neg p$  – operatori se ne mogu negirati

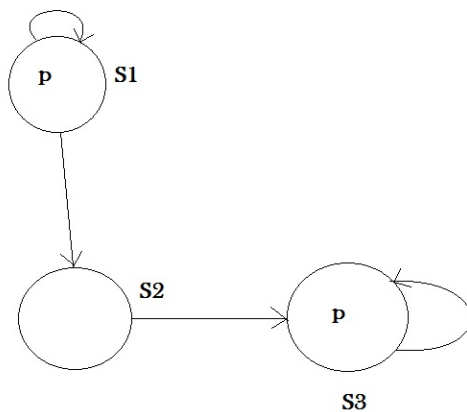
**22. Napišite CTL formulu za sljedeću izjavu: Iz svih stanja gdje je  $p$ =istinito (ima stanje u kojima je  $p$ =neistinito) izvođenje programa će svakako konačno završiti u stanju gdje je  $q$ =istinito**

$AG(p \Rightarrow AF q)$

**23. Neka je poznat skup stanja u kojima je istinit propozicijski simbol  $p$ . Objasniti riječima za koja stanja je istinita formula vremenske logike  $EXp$ .**

To su sva stanja iz kojih se može doći, tj. iz kojih postoji prijelaz u stanje u kojem  $p=T$ .

**24. Za (S1, S2, S3) odrediti da li je istinita CTL formula  $AF (AG p)$ .**



S1: ne, S2: da, S3: da