4 boda) Definirajte ulogu odgovornosti u objektno-usmjerenim sustavima? Razmotrite odnos odgovornosti i funkcionalnih zahtjeva. Na koji način se odgovornosti označavaju u dijagramu razreda?  
  
2b) Nužna obilježja koja mora posjedovati programski jezik za svrstavanje u kategoriju objektno usmjerenih jezika su:  
  
2b)Čime su opisani različiti stilovi arhitektura programske potpore? Navedite najmanje 3 primjera stila arhitekture.  
  
2b)Na koje dijelove progr. pot. je usmjereno ispitivanje programskog koda usmjereno na otkrivanje pogrešaka u algoritmima, podacima i sintaksi? Navedite i primjere što se ispituje. Što je referenca za provođenje tih ispitivanja?  
  
2b) Navedite vrstu arhitekture prog. pot. u koju se svrstava model-pogled-nadglednik (Model View Controller, MUC) i njezine osnovne značajke.  
  
2b) Navedite osnovne prednosti udomaćenih virtulanih strojeva ( hosted virtual machine). Jesu li udomaćeni virtualni strojevi efikasniji od hipervizorskih?  
  
4b) Control flow graph nacrtati  
  
2b) Skicirajte dijagram osnovnih koraka procesa otklanjana pogrešaka u fazi implementacije programiranja.  
  
1b) Na koji način se opisuje arh. prog. pot?  
  
1b) Navedite cilj analize zahtjeva u procesu izlučivanja zahtjeva..  
  
1b) Što je osnovni cilj oblikovanja prog. pot. zasnovanog na modelima?  
  
1b) .. osnovne značajke dobrog programskog produkta  
  
1b) RUP- 3 stvari u fazi razreda ( elaboration)

može ovo neko poredat točnim redosljedom, iz završno 08/09 (navodno dolazi na rokovima?)  
  
test komponenti  
test integracije  
funkcijski test  
test performansi  
test prihvatljjivosti  
test uporabe  
test instalacije

**Zimski 14/15:**

Teorija je bila skoro sva iz onog PDF-a na materijalima, bilo je OCSF- okruzenje, objektno orijentirana paradigma, šta će se ispisat, nadopisat kod za implementacijom sučelja.. uglavnom najosnovnije stvari.. dijagrami su isto bili dosta lagani po meni.

+

Ukupno 30(ili 29, ne sjećam se točno) zadataka, 10 na zaokruživanje(1 bod), 4 sa dijagramima(5 bodova). Bio je jedan zadatak za crtanje flowcharta nekog programa i još neki računi vezani uz njega koji se isto morao pisati na košuljicu/papir(3 boda), a ostali zadaci su bili onako kao na završnom, pitanje i odgovori(2 boda). Od dijagrama su došli use-case, komunikacijski, dij. stanja i dij. razreda.

**Zimski 11/12:**

|  |
| --- |
| *što se događa sa klijentima kad se pokrene metoda stoplistening na serveru?* |

posluzitelj prestane osluskivati za nove klijente, a sa onima sa kojima komunicira nastavlja ''razgovor''

.NET - na binarnoj razini? na razini izvornog koda*? (mislim da je na binarnoj razini?)*  
Navedi barem dvije razlike između programske komponente u komponentno temeljenoj arhitekturi i objekta u objektno usmjerenoj arhitekturi programske potpore.  
  
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 moze doci (iliti iz kojih postoji prijelaz) u stanje u kojem vrijedi p.)*

Pitanje je bilo " Upisati DA i NE ovisno o standardizaciji na binarnoj razini i standardizaciji na razini izvornog koda." za CORBA i .NET. Za CORBA je NE binarno, DA izvorno, ali za .NET obrnuto. (gradivo: arhitektura posluzitelj klijent)

"U ispitivanju programske potpore koristi se termin „testni slučaj“.Što je to?". **Jedan par: ulaz i očekivani izlaz (pri testiranju).**

**OCSF -** Općenito trebaš znati stvaranje veze, prekidanje veze, početak slušanja servera implementirati (par redaka), znati koje metode se gdje moraju implementirati, a koje su opcionalne te što se događa s klijentima nakon prestanka slušanja servera bla bla.  
  
Također treba znati koliko se dretvi za što stvara i mislim da je to to .;)  
Naravno da ne morate znati sve metode svih klasa niti onaj chat program ni tako slično. Samo svrhu tog paketa i neke važnije stvari. (imho)

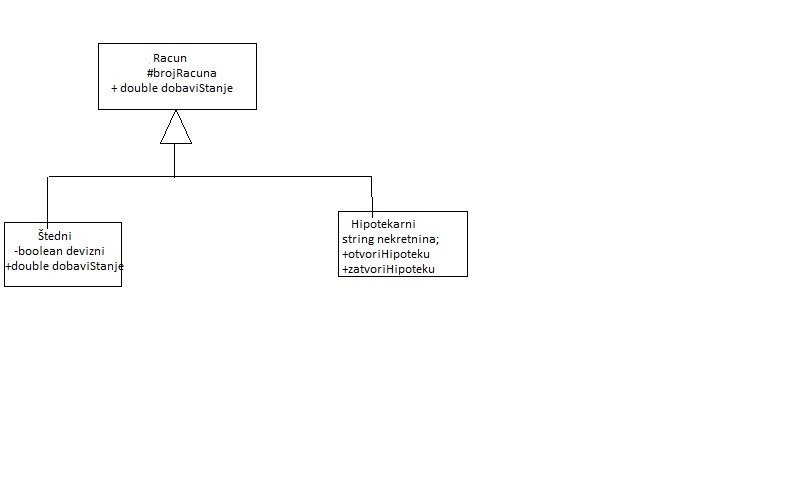
**Zadatak s testiranjem programske potpore -** Bio je zadan nekakav potprogram za sortiranje, trebalo je nacrtat one čvorove i lukove, izračunat CV(G), i za zadano polje, [2,7,5] izračunati postotak pokrivenosti testiranja stanja (ili tako nešto).

Da se nađe... teorijska pitanja su bila nešto detaljnija, nekoliko pitanja od prošlih godina je bilo spojeno 2 u 1, trazio se neki manji dijagram stanja za nacrtat (dok se garažna vrata otvaraju/zatvaraju svijetli žuto svijetlo, kod otključavanja/zaključavanja zvuk), objasnit općenito dijagram stanja, dijagram za testiranje nekakav, koji testovi su neophodni, use case dijagrami najbolje opisuju koji tip zahtjeva, pitanje sa "povećaj koheziju" i koje su prednosti, onaj dijagram sa geom. tijelima, kakvo je povezivanje korišteno za Rectangle i getBoundingRect() metodu, što je dinamičko povezivanje i što kako utječe na performanse, 4 osnovna tipa veza u use case, značajke dobre dokumentacije, kako izgleda nešto prije faze implementacije i za kog je namijenjeno, osim tražene funkcionalnosti i performansi 4 osnovne značajke dobrog softvera, dalje se ne sjećam.   
Use case dijagram je bio ok, sekvencijski lakši nego na 1. mi, class dijagram dosta opširan.

Jesenski 14/15

- pitanja sa 13/14 (bold i u 14/15)

**1. Što je programsko inženjerstvo? Koja je razlika između programskog inženjerstva i računalne znanosti?**  
2. Što su CASE sustavi?  
**3. Koje su vrste projekata (nabroji bar 4 tipa) u programskom inženjerstvu? Što znači pojam legacy system?**  
4. Nabroji generičke aktivnosti inženjerstva zahtjeva  
5. Cilj validacije zahtjeva i zašto se provodi  
6. Koja je temeljna karakteristika modela PI zasnovanog na komponentama koja ga razlikuje od ostalih modela?  
7. Ukratko i precizno objasni ekstremno programiranje, koje su prednosti, koje nedostaci tog pristupa oblikovanja programske potpore  
**8. 3 vrste verifikacije PP. U koju spada testiranje(ispitivanje)**9. Cilj, čemu služi i na koji se konkretan način koristi konceptualna razina arh. sustava  
10. Povećanje kohezije je jedan od temeljnih principa razvoja PP i kriterija. Objasni funkcijsku kojeziju!  
**11. Kod OO arhitekture, objasni metodu razreda  
12. Objasni nasljeđivanje i njegova svojstva u OO programiranju. (slika dole)**



i bilo je pitanje nešto tipa hoće li se izvesti statičko ili dinamičko povezivanje za dobaviStanje u razredu Štedni  
**13. Taj nisam zapisao, ali bila je slika klijentskih dretvi i serverskih te njihova komunikacija i pitanja o tome**14. Razlika tradicionalnog pristupa ispitivanja programa temeljenog na pronalaženju pogreške i modernog pristupa ispitivanju, posebno u svijetu moderne tehnologije razvoja PP  
15. Potpuno ispitivanje? kako se provodi?  
16. Od čega se sastoji ispitni slučaj  
**17. Navedi koje bi vrijednosti trebalo isprobati kako bi se provelo uspješni funkcijsko ispitivanje metode koja na ulazu treba primiti cijeli broj u rasponu [-100, 100]  
18. Preslikaj rečenicu u predikatnu logiku prvog reda tako da najprije definiraš sve potrebne predikate i konstante** (\*ne identično na 14/15)  
"Ako Ana voli Milovana, tada postoji netko tko voli Anu a ne voli Milovana"  
19. Preslikaj u vremensku logiku CTL:  
a) Sustav će, gledajući iz bilo kojeg stanja uvijek konačno doći u stanje "odobreno"  
b) Iz početnog stanja, postoji put u kojem se konačno ulazi u stanje gdje vrijedi "p"; od tog stanja dalje to "p" vrijedi uvijek i u idućem stanju  
c) Uvijek vrijedi da ako vrijedi "p" tada će na svim putevima vrijediti "q" dok ne počne vrijediti "r"  
20. Objasni što je to, kako funkcionira implicitno pozivanje kod arh zasnovane na događajima

**+ 14/15:** Uz to, bilo je još uloga i svrha arhitekture u procesu prog. inž., nešto sa big bang što u životu nisam čuo, arhitektura protoka podataka (njena 3 obilježja ili što već), spiralni model objasniti detaljnije, što su iteracije u procesu prog. inž. i koji modeli to koriste, od CTL logike je bilo samo ono da su zadane svakakve formule i označi koje su neispravne.  
Zadatak sa OCSF-om je bilo broj dretvi na poslužiteljskoj strani ako su spojena 2 klijenta (4), broj dretvi na klijentskoj strani (2) i napisati onu famoznu metodu koja mora biti implementirana: handleMessageFromServer().  
  
Od dijagrama su bili use-case, sekvencijski, razreda i objekta.

Bilo je nešto manje zadataka s dijagramima (samo 16 bodava - 3 zadatka - obrasci uporabe, dijagram razreda i komponenti), ali pitanja su očekivana (generički procesi i to), bilo je zadataka s formalnom verifikacijom (definicija logičke posljedice i modela), kripke + CTL logika, predikatna logika (napiši varijable/konstante i napiši "rečenicu" u predikatnoj logici).  
  
Od iznenađenja bio je zadatak s dinamičkim i statičkim povezivanjem (binding), napisati primjer koda za svaki (mislim da to nismo službeno radili na OPP-u), + 2 boda na dijagram stanja koji je bio relativno složen pa sam ga preskočio.  
  
Da, i zadatak s obrascima uporabe je IDENTIČAN onome iz UML priručnika. Dijagram razreda se mogao dosta dobro napisati, bilo je sitnica tipa enumeration (ako sam ja to dobro napravio?), napisati stereotipe na <<interface>> i <<abstract>> i puno klasa koje ovise jedne o drugima (svaki djelatnik ovisi o direktoru, svaki radnik ovisi o voditelju odjela, odjel ovisi o voditelju odjela pa je ispalo dosta neuredno).  
  
Zadnji je bio dijagram komponenti, pisalo je doslovno SVE (samo je trebalo znati kako se crtaju komponente i paketi), ako ste riješili ijedan primjer nije trebao biti problem potpuno točno nacrtati (ja nisam riješio NIJEDAN pa mislim da mi je dobro rješenje). :)

++  naučit stara pitanja (napamet sve podjele(nabrajanja), u ispitu je bilo 8 zadataka ak se ne varam koji kazu navedi, nabroji itd),

++ dijagrame i logiku (ona 2 predavanja, zadatke, dobro je ak ste imali umjetnu inteligenciju jer se tamo jos podrobnije radi logika). Bilo bi suludo ic kroz sva ona predavanja i mislit si sta je najbitnije, to ako se ima vremena, onda malo citas pa se nadas da ti nesto ostane u glavi. I da, nabavit rucni sat za svaki slucaj da vidis koliko se vremena jebes na nekom zadatku, jer ode vrijeme, nema se puno vremena za kemijanje i ono najgore, nema se vremena za provjeravanje jesi li tocno rjesio zadatak.

Ljetni 13/14:

Pitanja rok ljetni  
Bilo nas je 5,sva su bila opisna,evo nekih kojih se sjecam

Genericke aktivnosti u procesu programskog inzenjerstva? specifikacija, razvoj i oblikovanje,validacija i verifikacija,evolucija  
Koji model procesa programskog se primjenjuje kod malih i srednjih interaktivnih sustava? evolucijski  
Koji model ima najmanje novog koda? komponentni  
RUP koje aktivnosti ima? Zahtjevi,analiza,oblikovanje,implementacija,test  
Dva pristupa iteracijama u procesu programskog inzenjerstva? Inkrementalni i spiralni   
Kod onih 12. Principa oblikovanja i kriterija za odabir arhitekture, zasto je dobro povecat koheziju?(barem dvije prednosti): dakle veca je kohezija ako se grupira medusobno povezane elemente, sve ostalo je izvan, prednost je ta sta olaksava razumjevanje i promjene u sustavu  
Bilo je nesto s izražavanjem zahtjeva, nisam sigurna jesu li bile metode u izlučivanju(ako da to su:Intervjuiranje,scenarij,obrasci uporabe,dijagrami interakcija(sekv.dij)) ili je bilo kako izrazavamo korisnicke zahtjeve(strukturiran prirodni jezik,poebni jezici za opis oblikovanja, uml,matematica specifikacija)  
Za ovo nisam bas sigurna jer nisam znala odgovorit.Različiti dionici imaju različit pogleda na zahtjeve. Koji su moguci pogledi i koje korisnici sustava koriste? Pogledi interakcije,indirektni pogledi,pogledi domene primjene. Korisnici koriste poglede interakcije  
Zahtjevi s obzirom na sadrzaj: funkcionalni,nefunkcionalni,domene primjene  
Dvije formule, ro i omega, koja je razlika u ekvivalenciji koja u logickoj poseljdici  
Pretvorit 3 recenice u predikatnu logiku(Ante ima najvise jednu sestru, Ante ima tocno jednu sestru, Ante ima barem dvije seste)  
Bilo je ono sa racunanjem kod ispitivanja a da se produ svi putovi tako nekeko:zadane grane,čvorovi i P se valjda uvrsti jedan jer je sve povezano pa je formula: lukovi-cvorovi+2P  
Bilo je zadano ispitivanja, poredat ih po vremenu: ispitivanje komponenti,funkcijsko ispitivanje,ispitivanje performansi,ispitivanje prihvatljivosti,ispitivanje instalacije, ne znam jel ovo dobro poredano i bila su jos dva zadana cini mi se  
Bilo je neko stablo i nekoliko izraza i vidit dal vrijede,ono s CTL logikom, tipa AF q i tako to  
Bilo je zadan dijagram klasa i napisat za koje se izvodi dinamicko za koje staticko povezivanje, razredi su bili Shape2D, Rectangle i tako nesto  
Trebalo je nacrtat dijagram razreda knjiga,sadrzaj knjige,indeks pojmova,poglavlje u knjizi  
Nacrtan neki dijagram razreda, napisat sta je sta i sta dijagram predstavlja  
OCSF, bio je dijagram razreda, predavanje o OCSF-u str 62., tri potpitanja, koje sucelje implementira neki razred, koju metodu zove metoda HandlemessageFromServer i jos neko s main  
Tri zadatka s crtanjem dijagrama, jedan usecase:administrator i korisnik pa zadano sta koji radi, drugi sekvencijski:klijent A i B salju poruke Posluzitelju, i treci:nacrtat dijagram razreda za alarmni sustav, auto vrata na auto,sirena tako neke stvari su bile  
Ako se jos cega sjetim napisem

OCSF iz AR1(mislim da se tako zove preznetacija) i pogledaj staticko i dinamicko povezivanje iz prezentacije OO1 i nauci crtati control flow graph... Moja preporuka