

Oblikovanje programske potpore – Završni ispit, 27. 1. 2014. (min 12 bodova, maks 36 bodova)

IME I PREZIME: _____ **JMBAG:** _____

1. (1 bod) Navedi generičke aktivnosti u svakom procesu programskog inženjerstva.

2. (1 bod) Koji model procesa programskog inženjerstva je najpogodniji za male i srednje interaktivne sustave ?

3. (1 bod) Navedi četiri generičke aktivnosti u svakom procesu **inženjerstva zahtjeva**.

4. (1 bod) Nabroji barem tri metode koje se koriste pri izlučivanju zahtjeva u oblikovanju programske potpore.

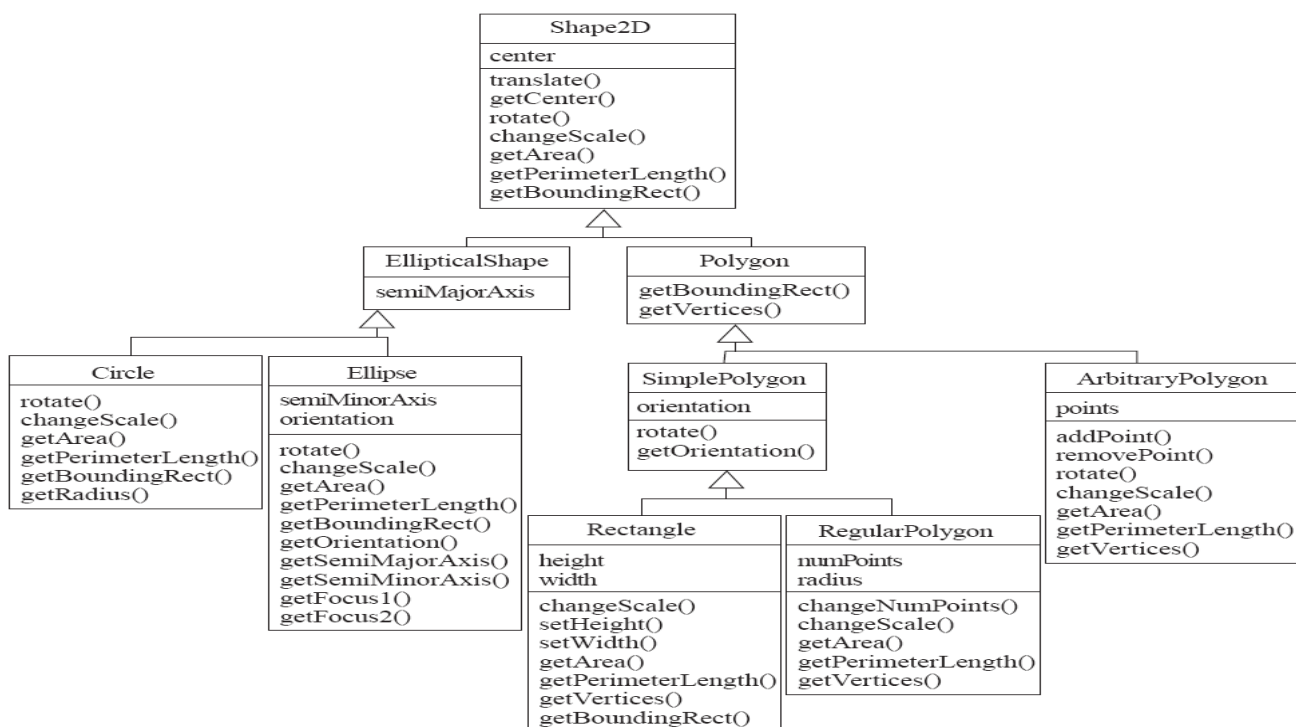
5. (1 bod) Na koje se sve općenite načine (navedi barem tri) mogu izraziti (opisati) zahtjevi sustava? (Pomoć: odgovor tipa "UML obrasci uporabe" nije ispravan jer je previše specifičan, traže se općeniti načini.)

6. (1 bod) Pogledi (engl. *viewpoints*) su načini strukturiranja zahtjeva tako da oslikavaju perspektivu i fokus različitih dionika. Navedi sva tri tipa pogleda.

7. (4 boda, po 1 bod za svaki ispravan odgovor) Zadana je hijerarhija razreda kao na slici. Neka je također zadana varijabla **var_a** određenog tipa. To znači da ta varijabla može sadržavati bilo koji objekt iz hijerarhije toga zadanog tipa. Pod strogim uvjetom da se ne stvaraju novi razredi ni metode, navedi za svaki zadani tip **var_a** iz kojih se sve razreda može izvesti zadana operacija (metoda).

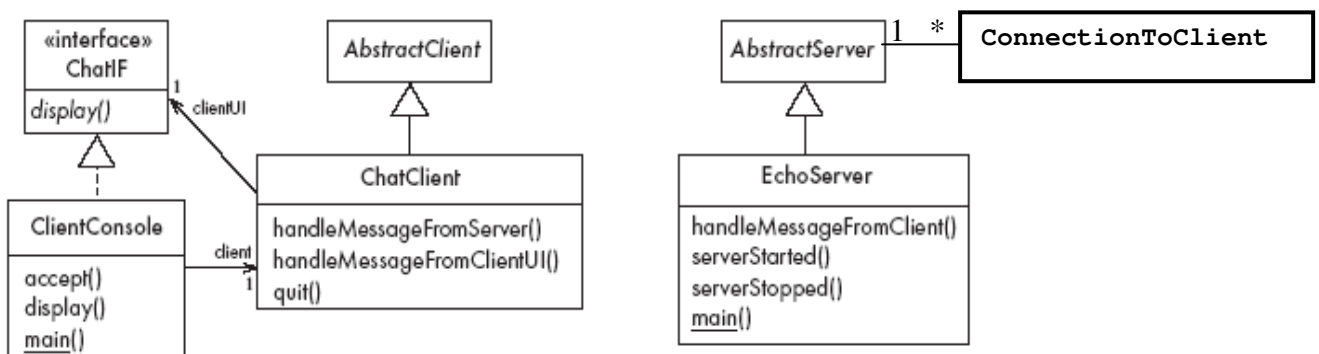
Priznaje se pojedini odgovor samo ako su navedeni SVI ispravni razredi. Također navedi za svaki pojedini slučaj da li se radi o dinamičkom povezivanju (engl. *dynamic binding*) ili statičkom povezivanju (engl. *static binding*).

- | | Tip varijable var_a : | Operacije s varijablom var_a : |
|----|------------------------------|---------------------------------------|
| a) | Circle | --> getArea() |
| b) | Circle | --> getCenter() |
| c) | Polygon | --> getBoundingRect() |
| d) | EllipticalShape | --> rotate() |

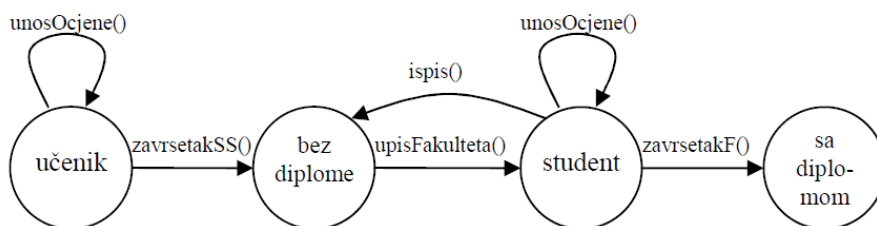


8. (2 boda) Skiciraj jednostavan UML-dijagram razreda (engl. *class diagram*) koji povezuje razrede: Knjiga, SadržajKnjige, IndeksUKnjizi (popis značajnih pojmova navedenih u knjizi), PoglavljeUKnjizi. Obrati pažnju na oznake brojnosti na vezama između razreda.

9. (4 boda – po 1 za svaki ispravan odgovor) Na slici je prikazana primjena radnog okvira objektnog klijent-poslužitelj raspodijeljenog sustava (OCSF) kroz tzv. "Jednostavan Chat".
- Nabroji posebno sve apstraktne i posebno sve konkretne razrede.
 - S kojom se operacijom/metodom na poslužiteljskoj strani šalju poruke (priznaje se i približan ali semantički ispravan naziv), te u kojem razredu je početno definirana?
 - U kojoj se metodi kreiraju objekti razreda **ClientConsole** i **ChatClient** na klijentskoj strani?
 - Koju operaciju/metodu zove metoda **handleMessageFromServer()** na klijentskoj strani?



10. (2 boda) Za razred **StatusOsobe** zadan je dijagram stanja prema slici. Prijelazi su uzrokovani pozivima metoda razreda.
- Na što treba obratiti pažnju pri oblikovanju testnog slučaja za ovu komponentu?
 - Navedi potrebni testni slučaj ili testne slučajeve.



11. (1 bod) Tijekom procesa strukturnog ispitivanja (engl. *white box*) modula programske potpore, generiran je graf tijeka programa (engl. *control flow graph*) koji sadrži 12 čvorova i 18 lukova bez dodatnih povezanih komponenti. Koliko je najmanje potrebno testova (navedi i formulu) da se ispitaju svi temeljni putovi (staze) kroz program?
12. (3 boda) Navedi dvije definicije logičke ekvivalencije propozicijskih formula (blažu i strožu), te definiciju logičke posljedice.

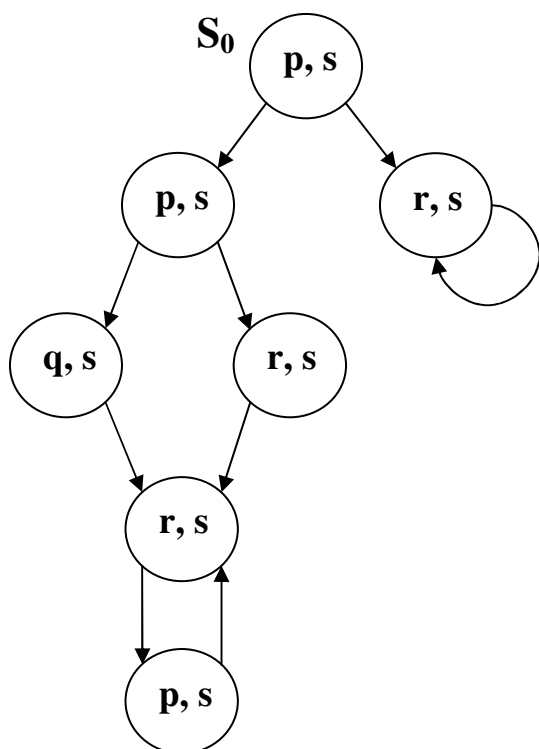
13. (2 boda) Definiraj potrebne predikate i konstante te preslikaj rečenicu u dobro definiranu formulu predikatne logike prvoga reda:
 "Svaki sin mog oca je moj brat."

14. (2 boda) Preslikaj izjave iz prirodnog jezika u odgovarajuću formulu u CTL logici:
 a) "Iz svih stanja gdje je p =istinito (dakle ima stanja u kojima p =neistinito), izvođenje programa će svakako doći do stanja gdje je q =istinito."

b) "Nije moguće doći u stanje od kojeg dalje zauvijek vrijedi "zaustavljen".

15. (2 boda) Zadana je Kripke struktura s propozicijskim simbolima p, q, r, s koji su istiniti u određenim stanjima. Vrijede li za stanje S_0 na vrhu (tj. jesu li istinite) sljedeće CTL formule:
 a) $AG\ EF\ r$
 b) $E\ (p\ U\ E(r\ U\ q))$

Odgovori moraju sadržavati objašnjenje zašto formule vrijede ili ne vrijede.



16. (5 bodova) UML-dijagramom razreda modeliraj sustav u banci.

Neka banka ima barem jednu filijalu, od kojih svaka ima svoj identifikacijski broj i adresu. Banka sadrži više organizacijskih jedinica koje mogu sadržavati vlastite organizacijske jedinice. Filijala je podvrsta organizacijske jedinice. Svaka organizacijska jedinica sadržava jednog vlastitog upravitelja i niz ostalih zaposlenika. Ako se organizacijska jedinica zatvori, upravitelj i zaposlenici prelaze u drugu jedinicu.

Svakom klijentu banke može biti pridružen jedan osobni bankar koji je tip zaposlenika, a osobni bankar može imati više klijenata. Svaki zaposlenik ima vlastiti jedinstveni broj unutar banke. Klijent i zaposlenik su osobe. Osoba ima svoj OIB, ime i prezime.

Klijent banke ima barem jedan račun otvoren u jednoj od filijala. Svaki račun je jedinstveno identificiran s brojem računa. Račun sadrži: stanje na računu, kreditni limit i limit prekoračenja.

Postoji više tipova računa: hipotekarni račun, štedni račun i tekući račun. Svaki račun može sadržavati ne više od dvije kreditne kartice. Ako se račun ukine, kreditne kartice se brišu. Hipotekarni račun sadrži opis nekretnine, njezinu vrijednost, te oznaku da li je nekretnina osigurana ili nije. Štedni račun može biti kunski ili devizni štedni račun. Tekući račun sadrži rok valjanosti dodatne kartice, i može imati ne više od pet dodatnih kreditnih kartica. Nadalje, svaki tip računa ima kamatnu stopu i mjesečni rashod. Štedni račun ima operaciju `Prebaci_stednju()` koju kunski i devizni štedni račun drugačije implementiraju. Hipotekarni račun ima vlastitu operaciju `Osiguraj_nekretninu()`.

Kreditna kartica i dodatna kreditna kartica realiziraju zajedničko sučelje *IKartica* s operacijama `Dohvati_raspoloziv_limit()` i `Dohvati_transakcije()`.

U izradi dijagrama razreda navedite nazive uloga gdje je to potrebno, te označite vidljivosti svih atributa i operacija pomoću simbola. Dijagram izradite koristeći minimalni potreban broj razreda i njihovih međusobnih odnosa poštujući zadana ograničenja višestrukosti.

17. (3 boda) UML-dijagramom komponentata prikaži sljedeću razvijenu programsku potporu.

Računalni sustav na poslužiteljskoj strani sastoji se od tri složene komponente pod nazivima: *Model*, *View* i *Controller* te od dodatne komponente: baze podataka. Komponenta *Model* sastoji se od četiri podkomponente: "Korisnik.java", "Poruka.java", "Parametri.java" i "Obrada.jar".

Komponenta *Model* eksportira dva sučelja: *IView* i *IController*. Također, komponenta *Model* importira sučelje *JDBC* od baze podataka. Baza podataka eksportira sučelje *JDBC*. Dodatno, unutar komponente *Model*, komponenta "Poruka.java" importira anonimno sučelje od komponente - knjižnice "Obrada.jar". Knjižnica "Obrada.jar" eksportira anonimno sučelje.

Komponenta *Controller* sastoji se od dvije podkomponente: "GlavnaServlet.java" i "GreskaServlet.java". Komponenta *Controller* importira sučelje *IController* od komponente *Model*.

Komponenta *View* sastoji se od dvije podkomponente: "glavna.jsp" i "greska.jsp". Komponenta *View* importira sučelje *IView* od komponente *Model*.

Pri izradi dijagrama, uz sve komponente navedite njihove odgovarajuće stereotipove (<<file>>, <<library>>, <<database>>, <<executable>>, <<package>>, ...), pri čemu tri složene komponente možete opisati kao pakete. Koristite standardnu oznaku za komponentu i sučelje komponente u UML-u.