



Oblikovanje programske potpore Završni ispit

2. veljače 2015.



JMBAG

Ime i prezime

Vlastoručni potpis

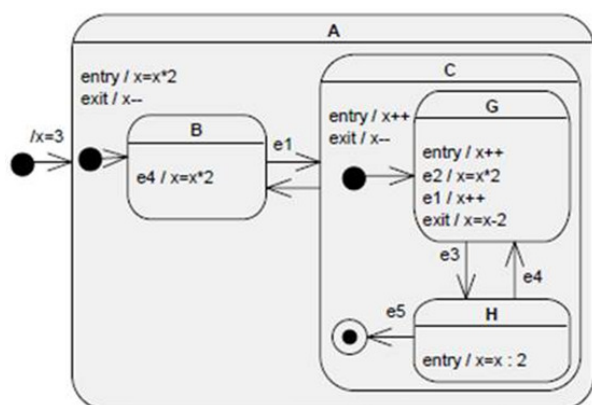
Minimum za prolaz je 12 bodova, maksimum 36 bodova

GRUPA B

1. (1 bod) Koje su osnovne grupe troškova koje ubrajamo u ukupnu cijenu programske potpore?
2. (1 bod) Navedite generičke aktivnosti prisutne u različitim modelima inženjerstva zahtjeva.
3. (1 bod) Nabrojite barem tri metode koje se koriste pri izlučivanju zahtjeva u oblikovanju programske potpore.
4. (1 bod) Opišite svojstva konzistentnih zahtjeva u procesu inženjerstva zahtjeva.
5. (1 bod) Odgovore na koja pitanja treba definirati cjeloviti proces razvoja programske potpore?

6. (1 bod) Navedite barem četiri prednosti definiranja arhitekture programske potpore.
7. (1 bod) Koji je osnovni cilj primjene principa oblikovanja: "Povećaj uporabu postojećeg"?
8. (1 bod) Navedite faze životnog ciklusa unificiranog procesa (engl. *Unified Process*).
9. (1 bod) Kako nazivamo grupu metoda/ metodu razvoja programske potpore kojima je zajednički iterativni razvoj uz male inkremente i naglaskom na ljude i međusobne odnose?
- 10.(1 bod) Što je metoda u objektno orijentiranim sustavima i kako se prikazuje poziv metode na UML-sekvencijskom dijagramu?

- 11.(1 bod) Za zadani dijagram napišite redoslijed događaja koji se izvode do dolaska u završno stanje uz pretpostavku da je trenutno stanje B.



- 12.(1 bod) Navedite UML-dijagrame interakcije koji su crtani i komentirani u dokumentaciji projekta iz kolegija Oblikovanje programske potpore.

- 13.(1 bod) Navedite osnovne korake pri oblikovanju ispitnih slučajeva.

- 14.(1 bod) Tijekom procesa strukturnog ispitivanja (engl. *white box*) modula programske potpore, generiran je graf tijeka programa (engl. *control flow graph*) koji sadrži 10 čvorova i 11 lukova bez dodatnih povezanih komponenti. Odredite najmanji broj potrebnih ispitnih slučajeva za sve temeljne putove programa.

- 15.(1 bod) Primjenom principa ekvivalencije particija, oblikujte ispitne slučajeve za ispitivanje jedne cjelobrojne varijable s ograničenjem vrijednosti $[170 - 9999>$. U rješenju odaberite pogodne konkretne vrijednosti ispitivanja i sve minimalne elemente ispitnog slučaja.

16.(2 boda) Navedite definiciju logičke ekvivalencije propozicijskih formula, te definiciju logičke posljedice.

17.(2 boda) Definirajte potrebne predikate i konstante te preslikajte rečenicu u dobro definiranu formulu predikatne logike prvoga reda:

"Ako na put krećeš radnim danom prije 6:30, idi Maksimirskom, a ako krećeš radnim danom nakon 7:00, idi Branimirovom."

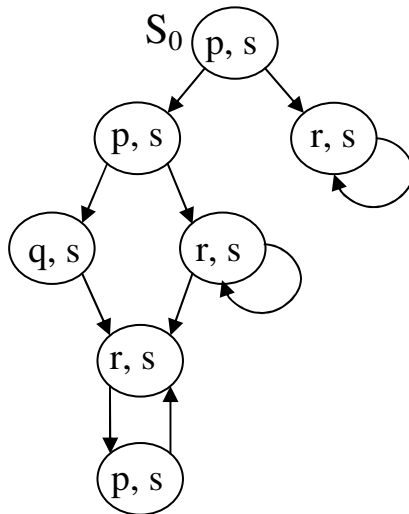
18.(2 boda) Preslikajte izjavu iz prirodnog jezika u CTL formulu koja će ispitati ispravnost tvrdnji te definirajte potrebne propozicijske simbole:

a) "Uvijek vrijedi da dretvu nije moguće terminirati u koraku koji slijedi nakon kreiranja dretve."

b) "Dretva nikada ne može doći u stanje od kojeg dalje zauvijek vrijedi da je u stanju čekanja".

19.(3 boda) Za zadani model implementacije M Kripke strukturom prema slici potrebno je:

- Dopuniti model i odrediti S (skup stanja), R (relaciju prijelaza), L (funkciju označavanja).
- Provjerite istinitost tvrdnje $M, S_0 \models AG \ EF \ r$ uz obrazloženje.
- Odrediti sva stanja koja zadovoljavaju formulu $E \ (p \ U \ E(r \ U \ q))$.



20.(1 bod) Opišite ulaze i značenje rezultata formalne verifikacija programske potpore metodom provjere modela.

Problemski dio- UML Dijagrami

Opći opis: Parkirni automat je namijenjen prodaji parkirnih karata na parkiralištu. Korisnik se služi parkirnim automatom tako što u njega jedan ili više puta ubacuje gotovinu, koja po tipu može biti novčanica ili kovanica. Na prednjoj strani uređaja nalazi se sučelje automata koje se sastoji od:

- o LCD zaslona na kojem je iskazano točno vrijeme, datum, parking zona u kojoj se nalazi uređaj i poruke prema korisniku,
- o dva otvora za ubacivanje novčanica i kovanica,
- o otvora za uzimanje ostatka novca,
- o tipke za unos trajanja parkiranja,
- o tipke za izdavanje karte,
- o tipke za odustajanje,
- o pisača za ispis karata.

Automat komunicira GSM modulom s bazom podataka Centrale, kojoj svakih sat vremena dojavljuje promet. Ako automat otkrije bilo kakvu pogrešku u svojem radu, onda to on također dojavljuje bazi podataka Centrale. Automat se napaja preko akumulatora koji mu daje autonomiju rada od 20 do 30 dana bez dopunjavanja preko solarnih ćelija ili iz električne mreže.

NAPOMENA: Prilikom izrade pojedinih dijagrama uz opći opis parkirnog automata i procesa uzmite u obzir i eksplicitno navedene zahtjeve uz pojedine zadatke.

21. (3 boda) Dijagram obrazaca uporabe

Odgovarajućim dijagramom obrazaca uporabe (engl. *use case diagram*) modelirajte procese plaćanja parkinga i dojave bazi podataka Centrale.

- o Korisnik, nakon ubacivanja gotovine, može unijeti broj sati za koji će platiti parking kao i pokrenuti izdavanje parkirne karte.
- o Parkirni automat obavlja naplatu parkinga po pokretanju izdavanja parkirne karte. Pri obavljanju naplate, u slučaju da korisnik nije unio željeno trajanje parkinga parkirni automat vraća korisniku ostatak novca u odnosu na zaokruženi satni iznos i to u obliku kovanica.
- o U slučaju da naplata nije moguća, jer nije uneseno dovoljno sredstava da pokriju satni iznos, korisniku automat ispisuje poruku „Neuspjela naplata“. U svakom drugom slučaju ispisuje se parkirna karta.

22.(4 boda) Dijagram stanja

Odgovarajućim UML-dijagramom stanja (engl. *statechart*) modelirajte unutarnja stanja parkirnog automata.

- o Automat je nakon uključivanja u „Stanju čekanja“ (na LCD-u ispisuje vrijeme, datum i zonu) sve dok korisnik ne započne s ubacivanjem novca čime prelazi u stanje „Prihvat novca“.
- o U stanju „Prihvat novca“ ostaje sve dok korisnik ne napravi: (1) odabir sati ili (2) izdavanje parkirne karte. Ukoliko korisnik napravi odabir sati, automat prelazi u novo stanje „Unos sati“ u kojem ostaje sve dok se ne odabere izdavanje parkirne karte.
- o Odabirom izdavanja parkirne karte izračunava se cijena i ako ima dovoljno sredstava naplaćuje se karta. U suprotnom se na LCD-u ispisuje poruka „Neuspjela naplata“ i ponovno omogućuje korisniku ubačaj dodatnog novca.
- o Prilikom ubacivanja novca ili odabira sati, korisnik u svakom trenutku može odustati i parkirni automat vraća odgovarajući iznos novca te se vraća u „Stanje čekanja“.
- o Pretpostavite da u bilo kojem trenutku može doći do pogreške (događaj „detektiran problem“) pri čemu automat prelazi u stanje „Kvar“ i dojavljuje poruku o pogrešci putem GSM-a bazi podataka Centrale. Pritom se u stanju „Kvara“ cijelo vrijeme ispisuje poruka „Automat nije u funkciji“. Nakon izvršenog popravka automat se vraća u „Stanje čekanja“.

23.(4 boda) Dijagram komponenata

Odgovarajućim UML-dijagramom komponenata (engl. *component diagram*) modelirajte komponente parkirnog automata i sučelja između njih.

- o Glavna komponenta parkirnog automata je izvršna datoteka „PParkirni“.
- o Napajanje parkirnog automata ima programske komponente „Akumulator“, „SolarneČelije“ i „ElektričnaMreža“ koje posjeduju sučelje „PokreniNapajanje“. Akumulator ima dodatna sučelja „DohvatiAutonomiju“ i „PokreniNadopunjavanje“. Inteligentni algoritmi autonomnog upravljanja podsustavom napajanja parkirnog automata implementirani su u izvršnoj datoteci „PNapajanje“ koje je putem sučelja „UpravljaNapajanjem“ povezano s izvršnom datotekom „PParkirni“.
- o Naplata parkirnog automata ima izvršnu datoteku „PNaplata“ sa sučeljem „UpravljaNaplatom“, te komponentu „ČitačNovčanica“ sa sučeljima „DohvatiPotvrduNaplate“ i „DohvatiIznosZaNaplata“. Cjenik parkinga u zonama zapisan je u XML datoteci „CjenikParkinga“ o kojoj ovisi te se upotrebljava u algoritmima komponente „PNaplata“. Izvršna datoteka „PParkirni“ komunicira s izvršnom komponentom „PNaplata“ koja samostalno obavlja sve ostale funkcije koje nije potrebno modelirati npr. procese poput brojanja gotovine, povrat kovanica i sl.
- o Sklopovlje koje koriste sve gore navedene komponente nije potrebno modelirati.
- o Označite sučelja komponenti gdje je to potrebno. Ispravno definirajte smjerove veza između komponenti u dijagramu. Grupirajte odgovarajuće komponente u zasebne pakete.