



# Programsko inženjerstvo

Završni ispit

2. veljače 2021.



GRUPA A

Izjavljujem da tijekom izrade ove zadaće neću od drugoga primiti niti drugome pružiti pomoć, te da se neću koristiti nedopuštenim sredstvima. Ove su radnje teška povreda Kodeksa ponašanja te mogu uzrokovati i trajno isključenje s Fakulteta. Također izjavljujem da mi zdravstveno stanje dozvoljava pisanje ove zadaće.

JMBAG

Ime i prezime

Vlastoručni potpis

**Završni ispit nosi ukupno 36 bodova, a minimum za prolaz završnog ispita je 12 bodova.**

- (1 bod) Navedite klasifikaciju zahtjeva programske potpore prema razini detalja.  
**Korisnički zahtjevi, zahtjevi sustava, specifikacija programske potpore.**
- (1 bod) Navedite UML dijagrame pomoću kojih ste u projektnoj dokumentaciji opisali funkcionalne zahtjeve.  
**Obrasci uporabe, sekvencijski dijagrami**
- (1 bod) U kakvoj su vezi arhitektura i modeli u Unificiranom procesu (engl. *Unified process*) razvoja programske potpore?  
**Modeli su prijenosnici za vizualizaciju, specifikaciju, konstruiranje (oblikovanje, implementacija) i dokumentiranje arhitekture. / UP promiče oblikovanje programske potpore zasnovano na modelima - arhitektura sadrži skup pogleda u modele.**
- (1 bod) Objasnite odnos pojmova: *odgovornost, operacija i metoda* (engl. *responsibility, operation, method*) u kontekstu razreda.  
**Svaki razred ima barem jednu odgovornost (nešto što treba izvršiti) koja se ostvaruje skupom operacija. Operacije se implementiraju metodama u programskom jeziku.**
- (1 bod) Koliko tipično traje jedan sprint u Scrumu?  
**2-4 tjedna**
- (1 bod) Navedite četiri CASE alata koje ste koristili na projektu.  
**Alati za komunikaciju: Teams, WhatsApp, Slack...  
Alati za verzioniranje: Git, Gitlab ...  
Alati za modeliranje: Astah, Visual paradigm ...  
Alati za oblikovanje sustava: Eclipse IDE, Visual Studio ...**
- (1 bod) Na koji način ste koristili princip oblikovanja *Povećaj uporabu postojećeg* (engl. *Increase reusability where possible*) u izradi projekta. Navedite primjer.  
**Npr. Korištenje gotovih radnih okvira tipa Spring Boot, Bootstrap, Flutter, korištenje knjižnica, korištenje vlastitih razvijenih razreda na više mjesta u kodu, predložaka i sl....**
- (2 boda) Navedite kategorizaciju oblikovnih obrazaca (engl. *design pattern*), kratko opišite svaku kategoriju i navedite barem jedan primjer obrasca.  
**Stvaralački (engl. *creational*) - usredotočeni na načine stvaranja razreda i objekata, npr. builder, factory, prototype, singleton  
Strukturni (engl. *structural*) - usredotočeni na odnose razreda i objekata, npr. adapter, bridge, composite, decorator, façade, flyweight, proxy**

Ponašajni (engl. behavioral) - usredotočeni na međudjelovanja razreda i objekata, npr. command, interpreter, iterator, mediator, memento, observer

Specifični (engl. specific) - specifični za neku primjenu npr. J2EE Patterns

9. (1 bod) Pojasnite razliku između horizontalnih i vertikalnih radnih okvira (engl. *framework*).

Horizontalni radni okvir osigurava opće usluge koje mogu koristiti primjenski programi (aplikacije). Ovaj tip radnog okvira moguće je upotrijebiti u izgradnji aplikacija šire domene primjene. (Okviri su općenitiji a time i složeniji). Vertikalni radni okvir sužava primjenjivost na usku domenu aplikacija i općenito zahtijeva manji broj prilagodbi od horizontalnog, ali još uvijek traži popunu nekih nedefiniranih mjesta kako bi se prilagodio specifičnoj primjeni.

10. (1 bod) Na koji način u modelu web uslužne arhitekture (engl. *web services*) tražitelji usluge (engl. *service requestor*) pronalaze željenu uslugu?

Service requestor pronalazi opis usluge (engl. *service description*) na nekom od komercijalnih ili open source UDDI poslužitelja (engl. *Service registry*) korištenjem protokola SOAP.

11. (1 bod) Navedite osnovnu zadaću posredničke razine (engl. *middleware*) između klijenta i poslužitelja.

Sveobuhvatna programska podrška koja omogućava uzajamno djelovanje aplikacija bez potrebe za poznavanjem i kodiranjem svih operacija nužnih za implementaciju usluge.

- skriva osobama koje oblikuju raspodijeljeni sustav detalje operacijskog sustava i druge specifičnosti implementacije,
- preuzima detalje komunikacijske mreže.
- omogućuje osobama koje oblikuju raspodijeljeni sustav da se usredotoče na primjenski (aplikacijski dio).

12. (1 bod) Kako se naziva koncept u kojem radni okvir (engl. *framework*) preuzima od korisničke aplikacije ulogu organizatora povezivanja između slojeva kod višeslojne arhitekture (engl. *multilayer architecture*)?

Inverzija upravljanja.

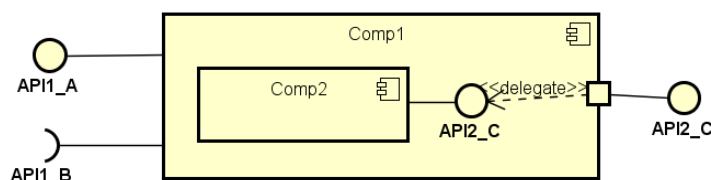
13. (1 bod) Od kojih se slojeva sastoji višeslojna organizacije aplikacije u Springu? U kojem sloju se pruža vanjsko sučelje web aplikacije najčešće kao RESTful web uslugu (REST API)?

Slojevi su: Controller, Service i Repository. U controller sloju se pruža vanjsko sučelje.

14. (1 bod) Kako je moguće ostvariti komunikaciju između ortogonalnih područja kod UML dijagrama stanja?

Uporabom dijeljenih varijabli.

15. (2 boda) Na UML dijagramu komponenti prikazite primjer komponente Comp1 s ponuđenim (implementiranim) sučeljem API1\_A i zahtijevanim (importiranim) sučeljem API1\_B. Također prikazite delegaciju ponuđenog sučelja API2\_C komponente Comp2 koja je sastavni dio komponente Comp1.



16. (1 bod) Opišite osnovna svojstva i ponašanje programskog defekta (često upotrebljavani sinonim engl. *error*, *fault*, *failure*, *bug*) radi čega program ne udovoljava zahtjevima ili specifikacijama, klasificiranog kao *mandelbug*.

Aktivacija i/ili širenje tog tipa defekta je složeno i teško ga je izolirati te sustavno ponoviti, npr. pogreška uslijed raspoređivanja.

17. (1 bod) Za koju tehniku razvoja programske potpore je jedno od svojstvenih pravila: „Ne pisati više programskog koda nego što je dovoljno za prolaz jediničnog ispitivanja (“*You may not write more production code than is sufficient to pass the currently failing test.*”)?

Razvoj upravljan ispitivanjem (*Test Driven Development*, *Test First Development*)

18. (2 boda) Primjenom tehnika ispitivanja standarda ISO/IEC 29119 dvovrijednosne granice (engl. *two value boundary testing*) i ekvivalentnih particija (engl. *equivalence partitioning*) odredite ispitne slučajeve (engl. *test case*) za mjesečni obračun cijene električne energije uz uvjete: potrošnja do 50 kWh obračunava se 0.5 kn/kWh, potrošnja za sljedeće kWh do 100kWh obračunava se po tarifi 1 kn/kWh. Ograničenje mjesečne potrošnje je 100kWh. Brojilo očitava cjelobrojne vrijednosti. Odredite ekvivalentne particije i granične vrijednosti. Tablično prikažite konkretne ispitne slučajeve.

E1:  $0 \leq p < 50$  G: 0,1; 49,50; 100, 101

E2:  $50 \leq p \leq 100$

E3:  $p \leq 0$

E4:  $p > 100$

TC	Ulaz	Očekivani izlaz	Rezultat
TC1	-10	Poruka greške	
TC2	0	0	
TC3	1	0.5	
TC4	34	17	
TC5	49	24,5	
TC6	50	50	
TC7	77	77	
TC8	100	100	
TC9	101	Poruka greške	
TC10	200	Poruka greške	

19. (1 bod) Za općeniti model sustava zadan Kripke strukturom  $M = (S, R, L)$ , interpretirajte formulu  $\mathbf{M, s_0} \models \neg \mathbf{AX (q \wedge r)}$ .

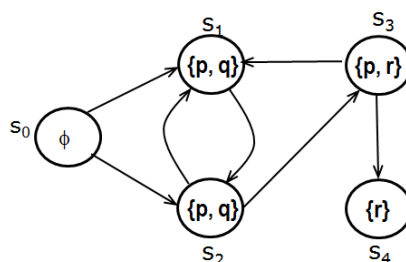
20. Postoji barem jedan put od početnog stanja  $s_0$  na kojem ne vrijedi u sljedećem stanju  $(q \wedge r)$  ili

Nije istina da na svim putevima od početnog stanja  $s_0$  u sljedećem stanju vrijedi  $(q \wedge r)$

21. (2 boda) Za zadanu Kripke strukturu odredite skup stanja koja zadovoljavaju formule. Nije potrebno obrazlagati postupak.

A.  $\phi = A(p \cup q)$ .

B.  $\phi = EF(EG p)$



A. Odgovor: {S1, S2}

B. Odgovor: {S0, S1, S2, S3}

## Problemski dio

Korisnici aplikacije na početku mjeseca dobivaju obavijesti o događanjima za taj mjesec. Putem aplikacije korisnik može kupiti jednu ili više godišnjih karata što omogućava neograničeni broj posjeta svim muzejima. Karta vrijedi 365 dana od uplate, a za provedbu plaćanja aplikacija se spaja na vanjski servis za plaćanje *ExpressTransakcija* koji kupcima omogućava sigurno plaćanje kreditnim karticama. U jednoj kupnji moguće je kupiti samo jednu kartu. Kupnjom godišnje karte korisnik dobiva status *MUZAFriend* i ostvaruje popust od 10% na kupnju nove godišnje karte. Nakon proteklih 365 dana od zadnje kupnje godišnje karte, korisnik prelazi u status *MUZAfan* i sljedećih 365 dana može ostvariti popust od 5% na kupnju godišnje karte. Ako niti u tom periodu ne kupi novu godišnju kartu, korisnik se vraća u status običnog člana.

21. (4 boda) UML dijagramom stanja modelirajte status članstva u društvu MUZA.

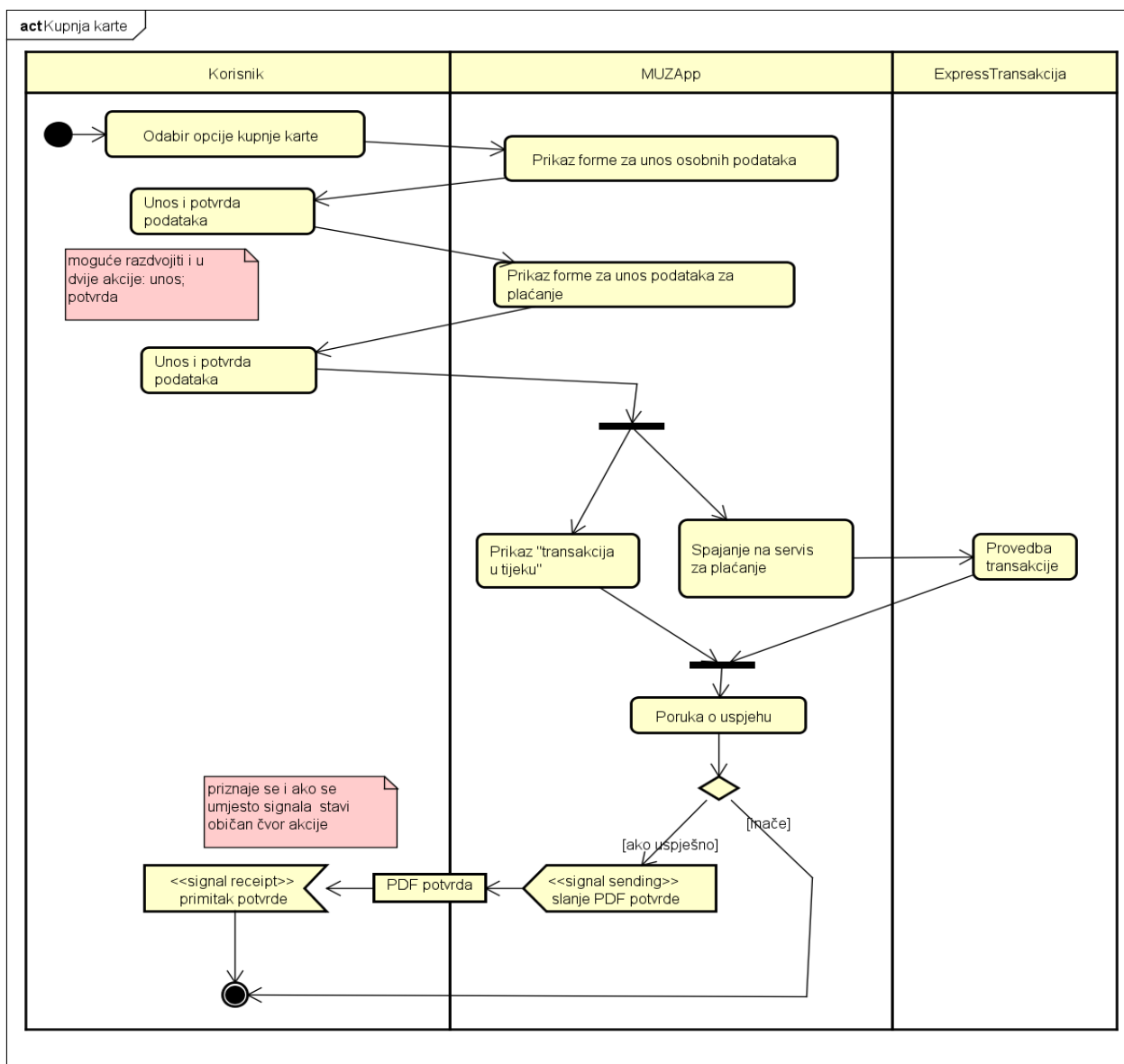


- 0.75 za navedena „vanjska“ stanja: Neregistrirani, Zahtijevano članstvo (ili neki sličan naziv) i Registrirani
  - 0.25 po točno navedenom stanju
- 0.5 za prijelaze između „vanjskih“ stanja: registracija, potvrda, istek 24h, brisanje računa
  - semantički ispravno, sintaksno neispravno (zgrade...): 0.25
- 0.75 za podstanja od stanja Registrirani: Redovni, MUZAfriend, MUZAfan
  - bodovanje: 0.25 po točno navedenom stanju
- 0.5 za prijelaze unutar stanja Registrirani, osim prijelaza vezanih uz odluku
  - semantički ispravno, sintaksno neispravno (zgrade...): 0.25
- 0.5 za prijelaz vezan uz odluku
  - čvor odluke ne mora nužno biti prikazan, ali onda moraju postojati 2 uvjetovana prijelaza
- 0.25 za akcije slanja e-pošte i brisanja računa
  - bodovanje: sve ili ništa

- priznaju se moguće manje razlike u rješenjima: npr. brisanje zahtjeva može biti i exit akcija od stanja Registrirani i sl.
- 0.5 za varijable stanja:
  - popust 0.25, br karata 0.25
- 0.25 za „do/slanje mjesečnih obavijesti“
- ako nema hijerarhije stanja, nego su sva stanja na istoj razini boduju se navedena stanja i prijelazi (kao da su u hijerarhiji) i onda se od toga oduzima -0.25 (takvo rješenje je moguće, ali je nezgrapno).
- stanja i prijelazi viška:
  - ako ne mijenjaju značenje dijagrama – priznati
  - ako nemaju smisla: oduzeto ukupno od -0.25 do -1 po procjeni ispravljača

22. (4 boda) UML dijagramom aktivnosti modelirajte proces kupnje jedne karte. Uz osnovni tekst zadatka uzmite i sljedeće informacije u obzir.

Proces kupnje karte započinje tako što korisnik odabere opciju kupnje karte i zatim mu se otvara forma za unos podataka o osobi na koju će glasiti karta. Nakon potvrde upisanih podataka korisnik unosi podatke o kreditnoj kartici s kojom će platiti kartu i potvrđuje kupnju. Mobilna aplikacija se spaja na servis za plaćanje i pokušava provesti transakciju, a za to vrijeme korisniku na zaslonu prikazuje da je transakcija u tijeku. Nakon završetka transakcije korisnika se obavještava o uspješnosti i ako je transakcija bila uspješna, šalje se potvrda o kupovini u PDF formatu na adresu e-pošte.

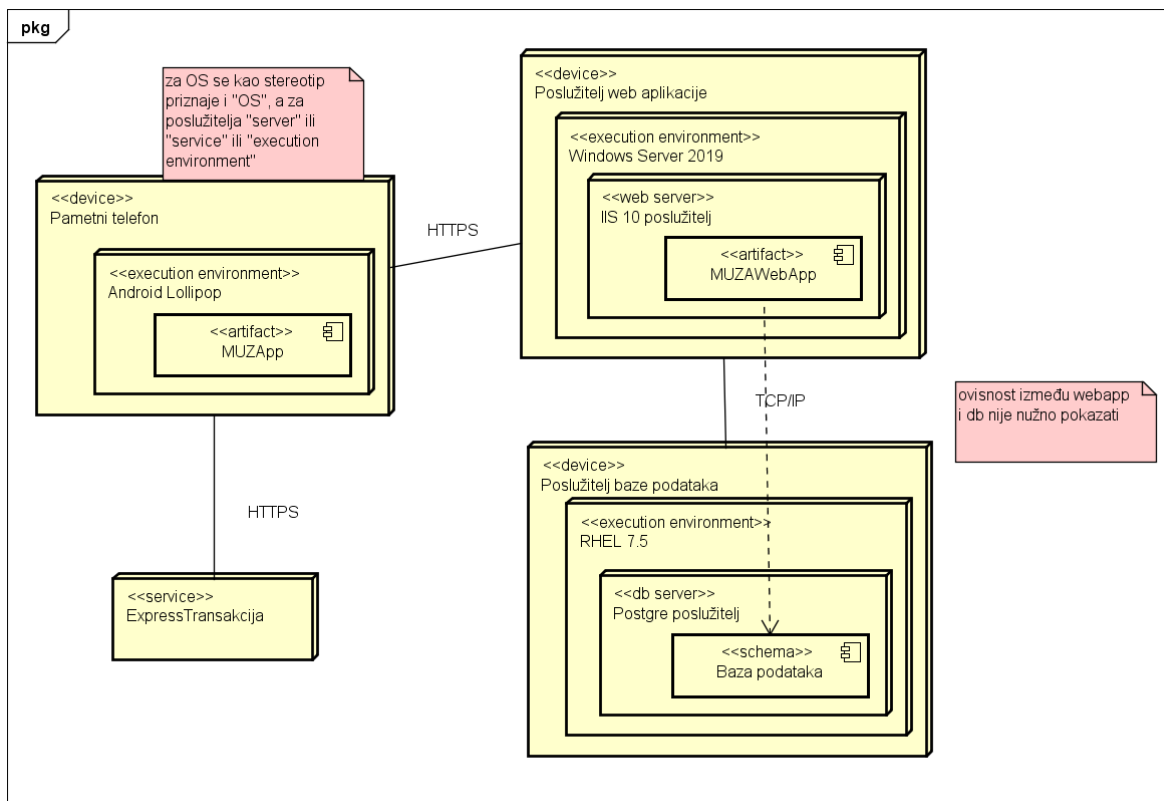


powered by Astah

### Bodovanje:

- 0.75 particije:
  - 0.25 po točno navedenoj particiji
- 0.5 početni i završni čvor:
  - 0.25 početni, 0.25 završni
- 1 ispravan slijed akcija za unos osobnih i podataka za plaćanje
  - ovaj slijed akcija moguće je modelirati s više ili manje čvorova akcije, ali mora ostati isto značenje (tijek aktivnosti) i mora biti sintaktički točno
  - ukoliko jedan dio ovog slijeda nije ispravno - 0 do 0.75 po procjeni ispravljача
- 0.75 fork/join + akcije između
  - bodovanje: 0.25 ako je prikazano kao jedan tok bez paralelizma i navedene su sve akcije
- 0.5 odluka
  - ako sintaktički nije točno: 0b
  - priznaje se ako se grana "inače" umjesto na kraj toka, vraća na neku smislenu prethodnu akciju (odabir opcije kupnje karte, potvrda kupnje..)
- 0.5 ispravno prikazano slanje potvrde
  - Priznaje se ako su umjesto signala korišteni čvorovi akcija

23. (4 boda) UML dijagramom razmještaja modelirajte cijeli sustav.



powered by Astah

### Bodovanje:

- 0.25 po ispravno prikazanoj komponenti/čvoru (ukupno  $12 \times 0.25 = 3$ )
  - ispravno prikazanom komponentom se smatra ona koja ima ispravan simbol (kvadar/pravokutnik) i naziv
  - u slučaju da su čvorovi prikazani dvodimenzionalno - 0.1 bod po čvoru
- 0.5 za sve ispravno definirane stereotipove
  - bodovanje sve ili ništa
  - priznaju se mogući različiti stereotipovi koji se smatraju jednako ispravnima tj. točno opisuju čvor/komponentu u kojoj je riječ:
    - za operacijski sustav (Android, RHEL, WIN..): `<<execution environment>>`, `<<OS>>`
    - za poslužitelj (web i db): `<<execution environment>>`, `<<service>>`, `<<server>>`, `<<web server>>`, `<<db server>>`...
    - za komponentu: `<<artifact>>`, `<<schema>>`, `<<executable>>`...
- 0.5 za sve ispravne veze između čvorova
  - bodovanje sve ili ništa