



Oblikovanje programske potpore

Međuispit

25. studeni 2014.



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

GRUPA A

1. (1 bod) Kao dioniku u ocjeni kakvoće programskog produkta kupcu je najvažnije:

- A. Lakoća oblikovanja.
- B. Lakoća održavanja.
- C. Lakoća ponovne uporabe dijelova.
- D. Nijedno od navedenog.
- E. Sve navedeno.

2. (1 bod) Odredite istinitu tvrdnju za svojstva modela procesa programskog inženjerstva:

- A. Iteracije su primjenjive samo na unificirani procesni model.
- B. Najmanje novog kôda generira se u modelu zasnovanom na komponentama.
- C. RUP (engl. Rational Unified Process) je najstariji model.
- D. Implementacija promjena je najlakša u vodopadnom modelu.
- E. Evolucijski model generira najbolju strukturu.

3. (1 bod) Koje se tri aktivnosti ponavljaju tijekom razvoja u evolucijskom modelu?

- A. Specifikacija, oblikovanje, ispitivanje.
- B. Specifikacija, razvoj, validacija.
- C. Analiza, oblikovanje, implementacija.
- D. Analiza, implementacija, ispitivanje.

4. (1 bod) Dobar obrazac uporabe u inženjerstvu zahtjeva prvenstveno specificira:

- A. Zahtjeve okoline.
- B. Mogućnosti razvojnog tima.
- C. Detaljne opise algoritma zahtjeva.
- D. Najvažnije funkcionalne zahtjeve.
- E. Sve navedeno.

5. (1 bod) Za sekvencijske dijagrame vrijede sljedeće tvrdnje (moguće više točnih odgovora):

- A. Prikazuju stanja objekata.
- B. Dobri su za grafički prikaz složenih algoritama.
- C. Dobri su za praćenje stanja sustava.

D. Prikazuju interakciju u sustavu.

E. Dobri su za prikaz komunikacije među objektima.

6. (1 bod) Koje od navedenih tvrdnji vrijede za odnos agregacije u UML dijagramu (moguće više točnih odgovora):

- A. Agregat predstavlja "je dio od" IsPartOf odnos.
- B. Kad se uništi agregat, uništavaju se i svi njegovi dijelovi.
- C. Agregacija je slabija od kompozicije.
- D. Agregat predstavlja „je" IS-A odnos.

7. (1 bod) Za princip oblikovanja koji preporuča zadržavanje razine apstrakcije (engl. *keep the level of abstraction as high as possible*) vrijedi:

- A. Traži poznavanje svih detalja podsustava.
- B. Povećava složenost prikaza problema.
- C. Nije pogodan za ciljanu implementaciju u objektnoj okolini.
- D. Osigurava odgodu razmatranja detalja.
- E. Nijedno od navedenog.

8. (1 bod) Koji koncept objektnog usmjerenja je primijenjen pri implementaciji metoda plus i minus u primjeru:

```
public class A {  
    private int a;  
    public A() { a = 0; }  
    public void plus(){ a++; }  
    public void minus(){ a--; }  
}  
  
public class B extends A {  
    public void plus () { a = a + 2; }  
    public void minus () { a = a - 2; }  
}
```

A. Višeobličje/polimorfizam (engl. *polymorphism*).

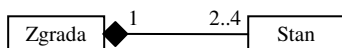
B. Nasljeđivanje (engl. *inheritance*).

C. Statičko povezivanje (engl. *static binding*).

D. Apstrakcija (engl. *abstraction*).

E. Izdvajanje sučelja (engl. *interface extraction*).

9. Što definira veza prikazana na UML dijagramu:



- A. U sustavu se u bilo kojem trenutku nalazi najviše dva objekta instancirana iz razreda Zgrada.
- B. U sustavu se u bilo kojem trenutku može nalaziti od dva do četiri objekata nastalih iz razreda Stan bez objekta razreda Zgrada.
- C. Ako se obriše jedan objekt razreda Stan, mora se obrisati i pripadajući objekt razreda Zgrada.
- D. Svaki objekt razreda Stan nasljeđuje attribute razreda Zgrada.
- E. Svaki objekt razreda Zgrada je povezan s dva do četiri objekta razreda Stan (nakon inicijalizacije).
10. (1 bod) Ako razred nasljeđuje apstraktni razred, onda mora implementirati sve njegove apstraktne metode da bi postao konkretan.
- A. Točno.
- B. Netočno.
- C. Ako ima više metoda, onda mora implementirati barem jednu.
11. (1 bod) Ključni zahtjevi sustava za upravljanje inačicama datoteka (npr. SVN) su (moguće više točnih odgovora):
- A. Lako upravljanje projektom (engl. *project management*).
- B. Jednostavnost pohrane/vraćanja sadržaja.
- C. Dobro grananje i spajanje razvojnih inačica.
- D. Podrška za modeliranje UML-om.
- E. Sustav automatski razrješava konflikte.

Napomena: u zadatku 11 priznavan je i dodatni odgovor E, ako je zacrtnjen skupa s B i C, budući da ponuđeni odgovor pod E nije imao dovoljno preciznu semantiku.

12. (1 bod) Navedite generičke aktivnosti procesa programskog inženjerstva:

- A. Specifikacija
- B. Oblikovanje i implementacija
- C. Validacija i verifikacija
- D. Evolucija

Napomena: Kao točne opcije, priznavalo se i: "razvoj i oblikovanje", "ispitivanje (ili testiranje)", "održavanje".

13. (1 bod) Prilikom izlučivanja i analize zahtjeva, koriste se UML dijagrami obrazaca uporabe, uz koje se u projektnoj dokumentaciji obično navodi i detaljan opis scenarija. Navedite najmanje 5 elemenata koje se pišu u okviru detaljnog opisa scenarija.

- A. _____
- B. _____
- C. _____
- D. _____
- E. _____

Moguća rješenja:

- opis početne situacije (ili preduvjeti).
- cilj obrasca uporabe (ili cilj scenarija)
- opis stanja gdje scenarij završava (ili rezultat).
- opis normalnog/standardnog tijeka događaja.
- opis što se eventualno može dogoditi krivo.
- informacija o paralelnim aktivnostima.
- glavni sudionik (ili glavni aktor)
- popis sudionika (ili aktori, dionici)

Napomena: rješenje "zahtjev" nije priznavano jer je preopćenito

14. (4 boda) Dijagram obrazaca uporabe

Dijagramom obrazaca uporabe modelirajte sustav za podizanje gotovine na bankomatu. Sustav koriste klijenti banke koji žele podići gotovinu. Klijent na bankomatu obavlja operacije: ubacivanja bankovne kartice u bankomat, pregledavanja stanja računa, podizanja gotovine s računa i izvlačenja bankovne kartice iz bankomata.

Ubacivanje bankovne kartice u automat podrazumijeva upisivanje PIN-a za bankovnu karticu. PIN se provjerava u bazi podataka o PIN-ovima.

Pregled stanja računa može biti pregled stanja na ekranu ili pregled stanja ispisom na papiru.

Podizanje gotovine s računa uključuje unos iznosa koji se podiže. Bankomat provjerava iznos s bankom i isplaćuje gotovinu klijentu. Pregled stanja računa i iznos koji se podiže provjeravaju se u bazi podataka klijentskih računa.

U slučaju da je, prilikom provjere iznosa koji se podiže, stanje na računu manje od iznosa koji se želi podići a korisnik time ne prekoračuje dozvoljeni minus na računu, klijentu se ispisuje upozorenje da će ući u minus. U slučaju da je stanje na računu takvo da se zadani iznos uopće ne može podići, klijentu se ispisuje greška da se novac ne može podići, a inače provjera iznosa prolazi bez neke posebne poruke.

Bankomat isplaćuje gotovinu klijentu ako je provjera prošla u redu ili ako je ispisano upozorenje. Kartica se u bankomatu zadržava tijekom pregledavanja stanja računa i podizanja gotovine.

15. (4 boda) Sekvencijski dijagrami

Sekvencijskim dijagramima obradite tipičnu uporabu bankomata prilikom podizanja gotovine iz prethodnog zadatka.

Nakon što ubaci karticu u bankomat, klijent ima pravo na tri pokušaja upisa PIN-a (string tip podatka), što bankomat pamti internim postupkom inkrementiranja. PIN se provjerava u bazi podataka s PIN-ovima i klijentu se vraća odgovor o statusu PIN-a.

Ako je unesen ispravan PIN, klijent upisuje traženi iznos gotovine kao cjelobrojnu vrijednost, inače bankomat izbacuje karticu. Upisani iznos se provjerava u bazi podataka o klijentskim računima. Nakon provjere, bankomat klijentu po potrebi ispisuje odgovarajuću poruku (upozorenje o ulasku u minus ili greška).

U slučaju da je isplata moguća, bankomat isplaćuje gotovinu i izbacuje karticu, inače se čeka da klijent unese dozvoljeni iznos gotovine.

Napomena: Sekvencijske dijagrame prikazati s razinom detalja potrebnom za implementaciju.

16. (4 boda) Dijagram razreda

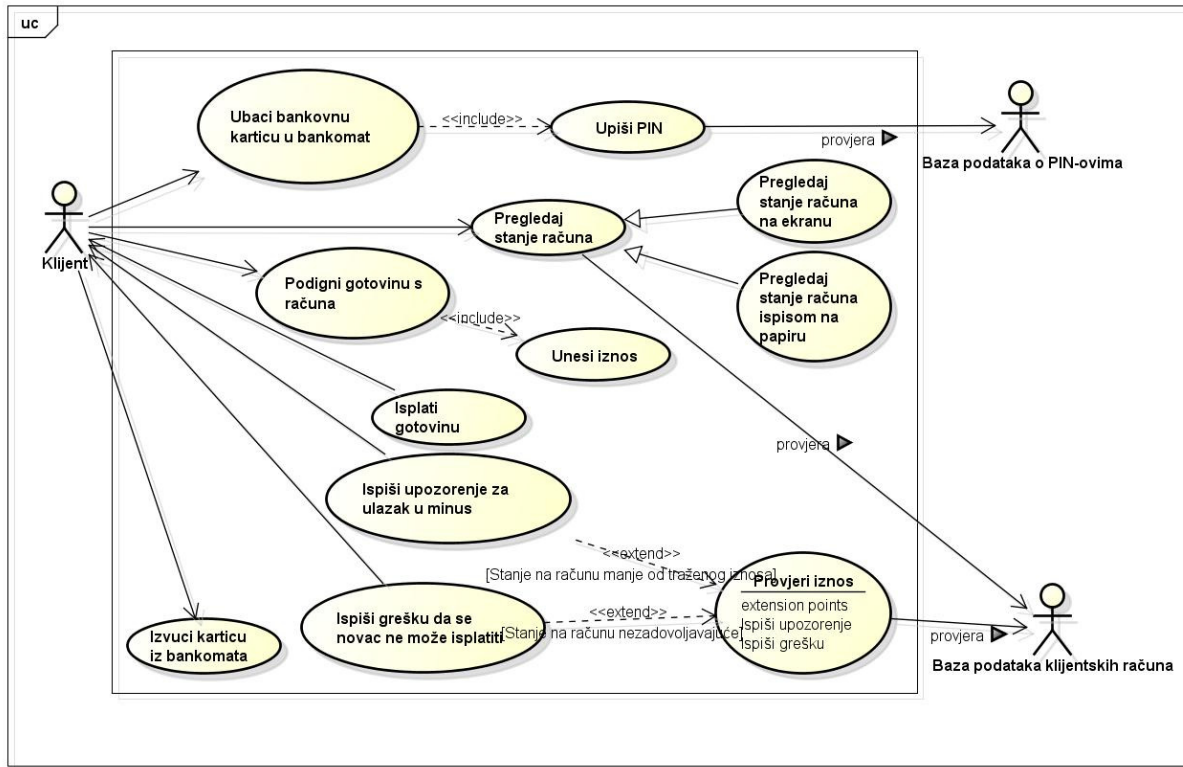
Dijagramom razreda modelirajte sportski klub. Sportski klub ima različite sekcije (nogomet, rukomet, stolni tenis...). Statutom kluba određeno je da klub neće imati više od 10 sekcija. Svaka sekcija ima atribut naziv (string, *public*). U sekciji može biti jedna ili više grupa, a svaka grupa ima jednog trenera. Trener može trenirati više grupa. Trener također može ujedno biti i voditelj najviše jedne sekcije. Svaka sekcija ima svog voditelja.

Klub ima dvije vrste članova: Redovni i Premium. Redovni članovi mogu biti članovi samo jedne sekcije, a Premium mogu biti članovi u proizvoljnom broju sekcija. Svaki član kluba mora biti uključen u najmanje jednu sekciju. Mjesečni iznos članstva računa metoda „mjesečniIznosClanstva()“, koja vraća rezultat tipa float. Članarina za Redovne članove je fiksna, a Premium članovi dobivaju popust ovisno o duljini trajanja članstva i osvojenim nagradnim bodovima koji se dodjeljuju samo Premium članovima.

Svaka grupa ima jedan ili više treninga tijekom tjedna. Ukoliko se grupa raspusti, neće biti ni treninga. Sportski klub ima nekoliko dvorana i za svaki trening se definira u kojoj se dvorani održava.

Napomena: Atributi i metode koji su eksplicitno zadani u tekstu zadatka moraju biti uključeni u dijagram razreda na odgovarajućim mjestima.

Rješenja:
14.

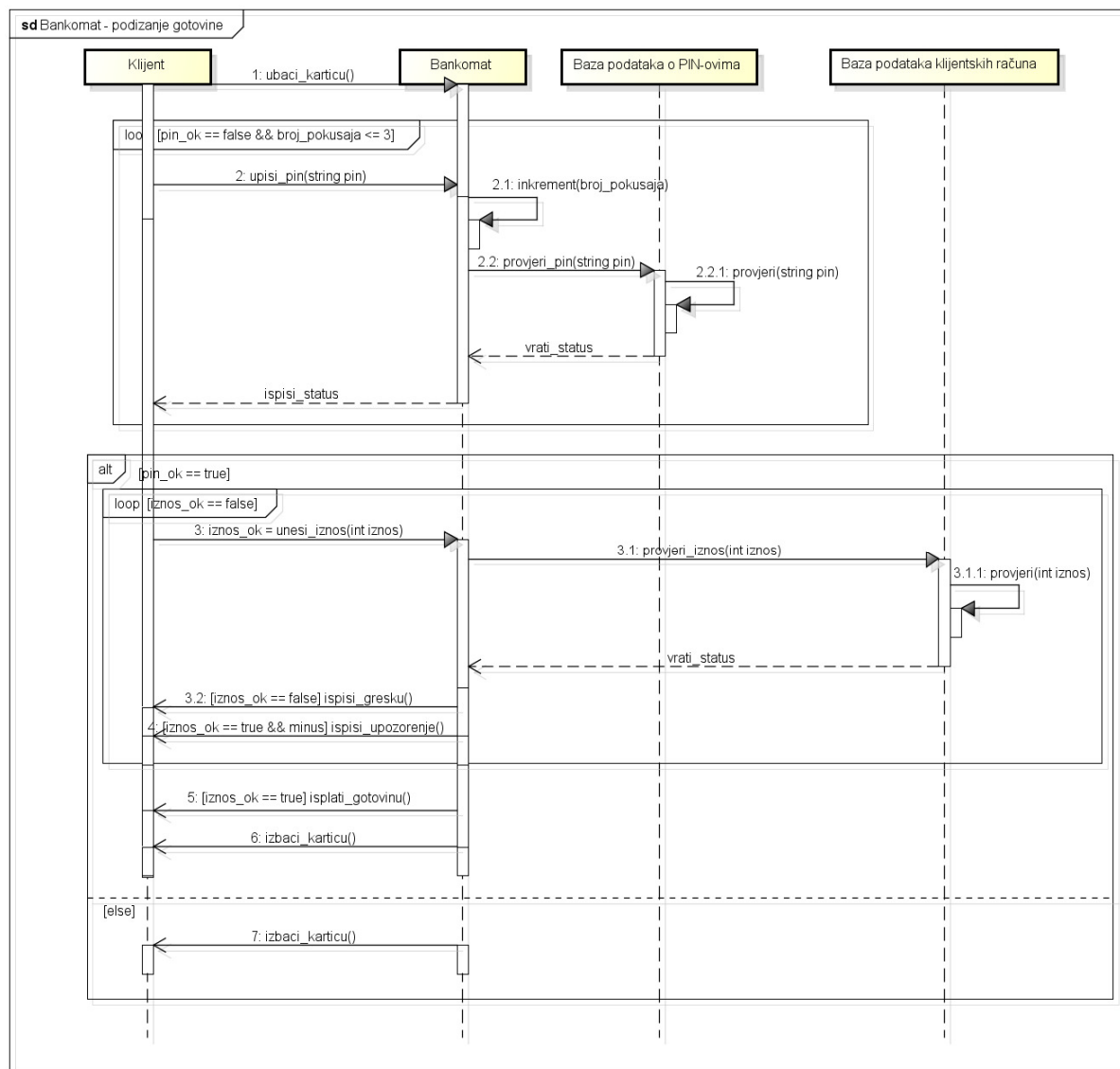


powered by Astah

Napomene:

1. Bankomat se ne modelira kao aktor. Bankomat je čitav sustav, koji se obično prikazuje kao imenovani ili neimenovani pravokutnik koji uokviruje unutarnje obrasce uporabe.
2. U tekstu zadatka postoje dvije baze, jedna o PIN-ovima, i druga, klijentskih računa, koje je trebalo kao aktore eksplicitno povezati s odgovarajućim obrascima uporabe (baze nisu povezane sa svim obrascima).
3. Priznavana su i rješenja gdje je obrazac "Unesi iznos" povezan relacijom <<include>> s "Provjeri iznos", kao i gdje je obrazac "Isplati gotovinu" povezan relacijama <<include>> ili <<extend>> (uz odgovarajući dodatni uvjet) s obrascem "Provjeri iznos".
4. Naziv "provjera" na asocijaciji s bazama mogao se modelirati i kao dodatni obrazac uporabe.

15.

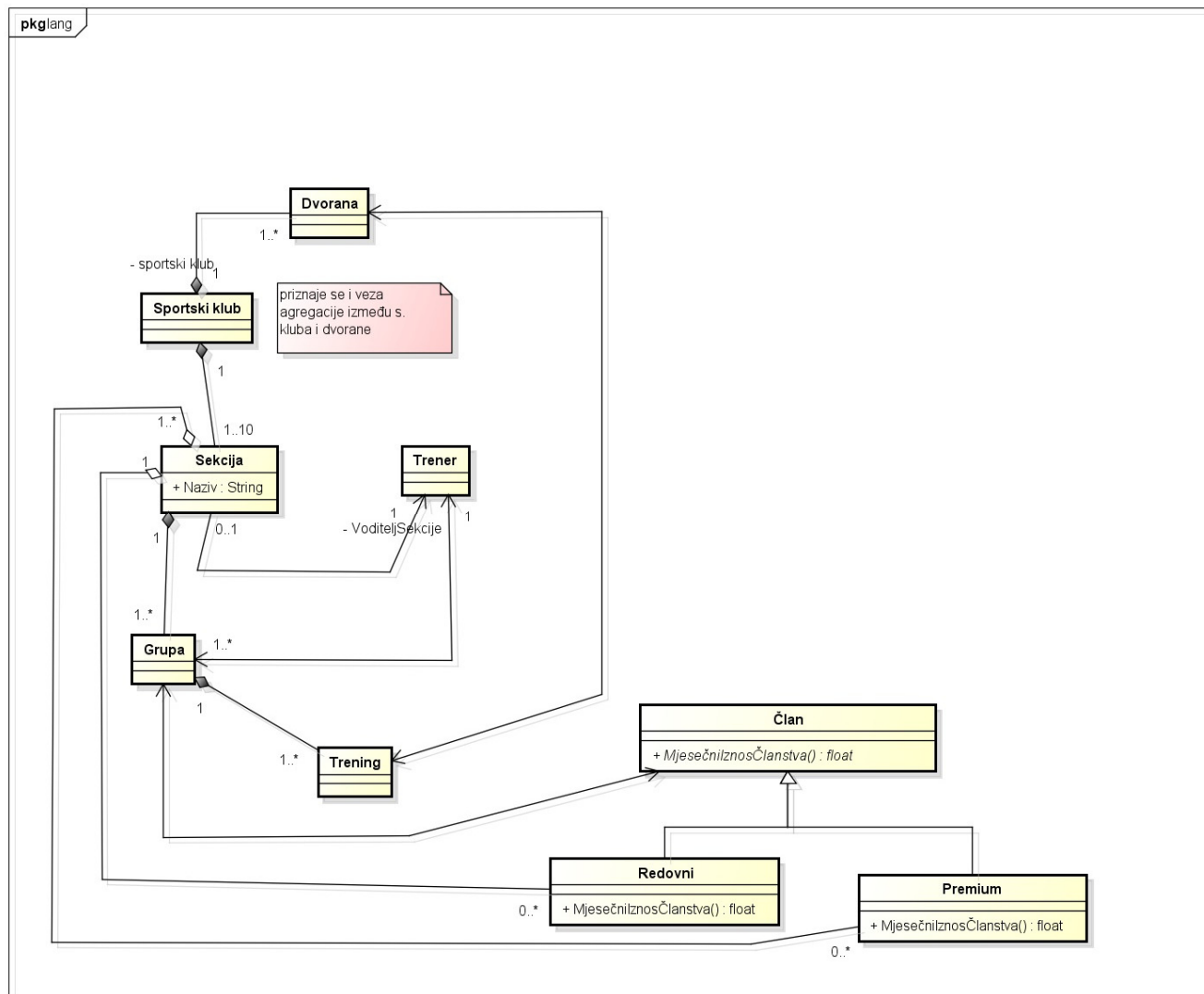


powered by Astah

Napomene:

1. Postoje dvije baze, jedna o PIN-ovima, i druga, klijentskih računa.
2. Potrebno je prikazati internu proceduru za inkrement pokušaja upisa PIN-a (kao poruku aktora Bankomat samom sebi).
3. Potrebno je navesti tipove podataka u argumentima poruka.
4. Pokušaji upis PIN-a se navode u petlji s pripadajućim uvjetom.
5. Pokušaji unosa iznosa se navode u petlji s pripadajućim uvjetom.

16.



powered by Astah