

Podrška objektno orijentisanom programiranju
u jezicima C++, Objective C, Java, C#, Ada i
Ruby

Katarina Popović, Dušan Pantelić, Dejan Bokić, Nikola Stojević

30. april 2019

Sadržaj

1	Uputstva	2
2	Recenzent — ocena: 4	3
2.1	O čemu rad govori?	3
2.2	Krupne primedbe i sugestije	3
2.3	Sitne primedbe	3
2.4	Provera sadržajnosti i forme seminarskog rada	5
2.5	Ocenite sebe	5
3	Dodatne izmene	6

Glava 1

Uputstva

Prilikom predavanja odgovora na recenziju, obrišite ovo poglavlje.

Neophodno je odgovoriti na sve zamerke koje su navedene u okviru recenzija. Svaki odgovor pišete u okviru okruženja \odgovor, [kako bi vaši odgovori bili lakše uočljivi](#).

1. Odgovor treba da sadrži na koji način ste izmenili rad da bi adresirali problem koji je recenzent naveo. Na primer, to može biti neka dodata rečenica ili dodat pasus. Ukoliko je u pitanju kraći tekst onda ga možete navesti direktno u ovom dokumentu, ukoliko je u pitanju duži tekst, onda navedete samo na kojoj strani i gde tačno se taj novi tekst nalazi. Ukoliko je izmenjeno ime nekog poglavlja, navedite na koji način je izmenjeno, i slično, u zavisnosti od izmena koje ste napravili.
2. Ukoliko ništa niste izmenili povodom neke zamerke, detaljno obrazložite zašto zahtev recenzenta nije uvažen.
3. Ukoliko ste napravili i neke izmene koje recenzenti nisu tražili, njih navedite u poslednjem poglavlju tj u poglavlju Dodatne izmene.

Za svakog recenzenta dodajte ocenu od 1 do 5 koja označava koliko vam je recenzija bila korisna, odnosno koliko vam je pomogla da unapredite rad. Ocena 1 označava da vam recenzija nije bila korisna, ocena 5 označava da vam je recenzija bila veoma korisna.

NAPOMENA: Recenzije će biti ocenjene nezavisno od vaših ocena. Na osnovu recenzije ja znam da li je ona korisna ili ne, pa na taj način vama idu negativni poeni ukoliko kažete da je korisno nešto što nije korisno. Vašim kolegama šteti da kažete da im je recenzija korisna jer će misliti da su je dobro uradili, iako to zapravo nisu. Isto važi i na drugu stranu, tj nemojte reći da nije korisno ono što jeste korisno. Prema tome, trudite se da budete objektivni.

Glava 2

Recenzent — ocena: 4

2.1 O čemu rad govori?

U ovom radu su objašnjeni osnovni koncepti objektno orijentisanih jezika i njihova implementacija u jezicima C++, Objective C, Java, C#, Ada i Ruby. Za svaki jezik posebno su objašnjeni enkapsulacija, nasleđivanje, polimorfizam i apstrakcija. Navedene su i neke osobine koje dati jezik odvajaju od ostalih.

2.2 Krupne primedbe i sugestije

U poglavlju 2, koje se bavi programskim jezikom C++, i poglavlju 5, koje je vezano za C#, nedostaju reference.

Trebalo bi razmisliti da li je u delovima 2.3, 3.3, 4.3, umesto termina premošćavanje i koncept važnosti metoda, bolje koristiti termin preklapanje (eng. overriding). Takođe, mislim da je u poglavlju 5.1 bolje koristiti termine metoda i atribut umesto funkcija i promenljiva jer je reč o pravima pristupa unutar klase.

U delovima 3.3, 4.3 koje govore o polimorfizmu izmenjen je termin važnosti metoda u termin preklapanja, jer se čini intuitivnijim i adekvatnijim. U poglavlju 5.1 zamenjeni termini predloženim terminima.

2.3 Sitne primedbe

U Uvodu, kod objašnjenja nasleđivanja prva rečenica nije najjasnija, bilo bi lepo da se preformuliše.

Deo 2.2 - "+dok - nedostaje zatvoren navodnik, umesto slova d stoji slovo d.

Poglavlje 3 - Svaka klasa je izvedena iz superklase NSObect, čiji konstruktor init je podrazumevani konstruktor, može predefinisati. - ova rečenica je nejasna, potrebno ju je preformulisati. Nije poštovano pravilo za razmake posle tačke, zareza i zagrada.

Preformulisana je problematična rečenica, podeljena je u dve zapravo. Popravljeni razmaci. "Svaka klasa nasleđuje baznu klasu NSObject, koja sadrži konstruktor init. Ovaj konstruktor se može predefinisati."

Deo 3.3 - Koncept važnosti metoda(eng. *overriding*) postoji, i prikazan je u primeru(3). Gde klase zaposleni i vozač imaju isti metod `display` i poziv [`empldisplay`] će izvršiti metod klase vozač, zato što promenljiva zaposleni `empl` referiše na objekat tipa vozač. - ovo je potrebno popraviti (predlog, spojiti rečenice u jednu ili izbaciti reč 'Gde').

Ranije je već navedena zamerka oko termina važnosti metoda i to je ispravljeno. Nije lepo bilo objašnjeno, popravljene su rečenice u poglavlju 3.3. Izbačena je reč 'Gde', ostale su dve rečenice. "Koncept preklapanja metoda (eng. *overriding*) postoji, i prikazan je u primeru (3). Klase zaposleni i vozač imaju isti metod `display` i poziv [`empl display`] će izvršiti metod klase vozač, zato što promenljiva zaposleni `empl` referiše na objekat tipa vozač. "

Deo 3.4 - Prvu rečenicu bi bilo lepo podeliti u dve. U poslednjoj rečenici dodati zareze iza reči `@required`, klasi i iza reči `@optional`.

Podeljena je prva rečenica. "Jezik Objective C nema definisan koncept apstraktnih klasa [4]. Sličan efekat je moguće postići programerskom snalažljivošću i ne instancirati klasu koja bi trebalo biti apstraktna." Ispravljene gramatičke greške sa zarezom. " Sekcija protokola se završava sa **@end** i može sadžati dve podoblasti **@required**, za metode koji se obavezno moraju implementirati u klasi i **@optional**, za metode čija je implementacija opcionala."

Poglavlje 4 - U prvoj rečenici višak je drugo 'su'. Uklonjen je višak. "Kako su klase u centru zbivanja krenućemo od njih."

Deo 4.4 - Smatram da je u duhu srpskog jezika bolje napisati 'prema članku' nego 'prema artiklu'. Usvojena primedba, promenjeno u 'prema članku'. "Izdvajanje skupa metoda sa kojima spoljašnji korisnik komunicira, prema članku [3], vršimo pomoću apstraktnih klasa ili interfejsa."

Poglavlje 5 - Veznik 'i' pisati malim slovom (greška u celom poglavlju). Rečenica - Veome je slična podrška OOP-u kao kod Java programskog jezika takođe su iste I klase I strukture. - nejasno, popraviti (predlog, zarez iza takođe).

Prepravljen veznik 'i' sa velikom na malo slovo u celom poglavlju. Navedena recenica je preformulisana.

Deo 5.1 - U prvoj rečenici je 'od' višak. Ošišana latinica na nekoliko mesta (izlozi, clan).

Uklonjen visak 'od'. Prepravljena latinica.

Deo 5.2 - Ošišana latinica (moze, sto).

Prepravljena latinica.

Deo 5.3 - U prvoj rečenici jedno 'da' je višak.

Uklonjen visak 'da'.

Deo 5.4 - Ošišana latinica (nasih).

Prepravljena latinica.

Poglavlje 6 - Nisu poštovana pravila o razmacima pre i posle zagrade (greška u celom poglavlju).

Poglavlje 7 - Nisu poštovana pravila o razmacima pre i posle zagrade (greška u celom poglavlju).

Deo 7.1 - Ošišana latinica (mogucnost). Potreban je zarez posle `public`.

Deo 7.2 - Ošišana latinica (nasledjuje).

Deo 7.3 - Ošišana latinica (sto, ce, razliciti)

Deo 7.4 - Ošišana latinica (pokrecu, pomocu, olaksava). Potreban je razmak posle `"NotImplementedError"`.

Poglavlje 8 - U poslednjoj rečenici izostavljeno slovo 'h' (programski(h) jezika).

U delovima 4.4 i 5.3 postoje pasusi koji sadrže samo jednu rečenicu. U poglavlju 4.4 izdvojena je rečenica u pasus je predstavlja opšti uvod. Možda nije efektivno, pa je promenjeno.

2.4 Provera sadržajnosti i forme seminarskog rada

1. Da li rad dobro odgovara na zadatu temu?
Da. Pokriveni su osnovni koncepti objektno orijentisanih jezika, objašnjeni su načini za njihovo ostvarivanje kao i razlike u odnosu na druge objektno orijentisane jezike.
2. Da li je nešto važno propušteno?
Nije.
3. Da li ima suštinskih grešaka i propusta?
Ima manjih grešaka i propusta koji su gore navedeni
4. Da li je naslov rada dobro izabran?
Naslov rada je isti kao naslov teme.
5. Da li sažetak sadrži prave podatke o radu?
Sadrži. Na osnovu sažetka je jasno čime se rad bavi.
6. Da li je rad lak-težak za čitanje?
Rad je lak za čitanje.
7. Da li je za razumevanje teksta potrebno predznanje i u kolikoj meri?
Nije potrebno veliko predznanje. Dovoljno je poznavati osnove oop-a (šta su klase, metodi, atributi).
8. Da li je u radu navedena odgovarajuća literatura?
Nedostaje literatura za C++ i C#.
[Dodata literatura za C#.](#)
9. Da li su u radu reference korektno navedene?
Da.
10. Da li je struktura rada adekvatna?
Jeste.
11. Da li rad sadrži sve elemente propisane uslovom seminarskog rada (slike, tabele, broj strana...)?
Sadrži odgovarajući broj slika, tabela i strana.
12. Da li su slike i tabele funkcionalne i adekvatne?
Slika i tabela su funkcionalne i adekvatne za deo teksta u kom se nalaze.

2.5 Ocenite sebe

Srednje upućen.

Poznajem navedene koncepte oop-a u meri u kojoj smo ih izučavali na ranijim kursevima, i imam iskustva sa programiranjem u pojedinim jezicima iz teksta, ali ne u svim.

Glava 3

Dodatne izmene