

Clean CaDET

- Povratne informacije -



Iskustvo učenja



Iskustvo učenja

- Komunikaciona veza između nastavnika i studenata.
- Instrument za evaluaciju virtuelnog okruženja za učenje.
- Prikupljanje dve vrste povratnih informacija:
 - Iskustvo učenja
 - Unapređenje Tutora.
- Povratne informacije se prikupljaju na srpskom jeziku:
 - Kreiranje resursa za obradu prirodnog jezika
 - Nedostatak (javnih) skupova podataka za evaluiranje
 - Proširenje oblasti i na srpski jezik
 - Automatizacija evaluacija na srpskom jeziku.



Iskustvo učenja

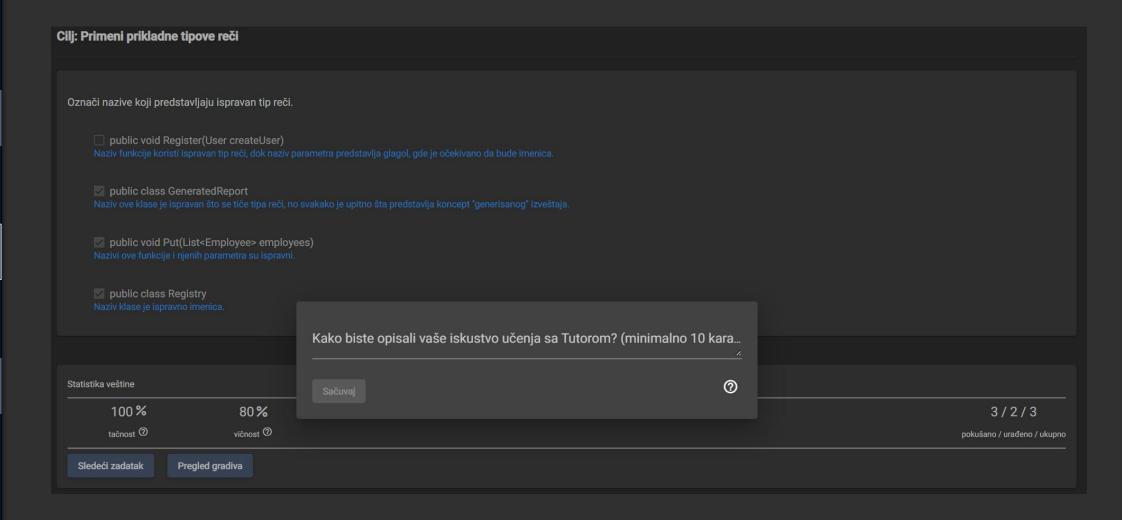
Unapređenje Tutora

- Osećanja i mišljenja o iskustvu učenja:
 - Razumevanje ponašanja tokom učenja
 - Afekti imaju uticaj na učenje
 - Razvoj afektivnog sistema učenja.

 Forma se automatski prikazuje nakon svakih 5 poslatih rešenja zadataka.



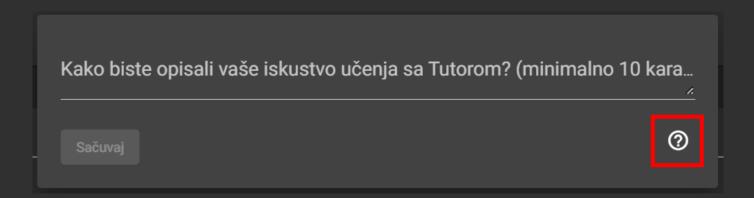
Iskustvo učenja





Iskustvo učenja

Unapređenje Tutora



Primeri:

- Lekcija je bila lagana za razumevanje i veoma korisna.
- Bilo mi je naporno.
- Nisam razumeo primere.
- Svideli su mi se izazovi.
- Nemam dovoljno znanja o ovome. Sve mi je preteško.



Iskustvo učenja

- Dragoceni izvori ideja i komentara koji dolaze direktno od korisnika za unapređenje:
 - Tutora kao softvera
 - Edukativnog sadržaja.
- Korisnici mogu da:
 - prijave potencijalne greške
 - daju predloge za:
 - nove funkcionalnosti/sadržaj
 - redizajn onih koje već postoje.
- Nastavnicima je:
 - Evaluacija njihovog rada
 - Osnov za unapređenje edukativnog sadržaja
 - Resurs za prikupljanje novih zahteva.



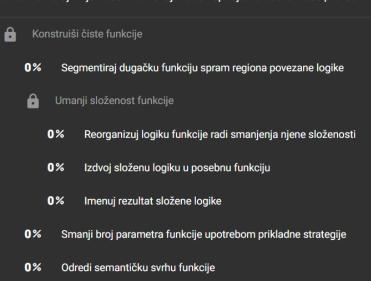
Iskustvo učenja

Unapređenje Tutora

 Formu studenti otvaraju sami klikom na dugme na stranici lekcije.

Čiste funkcije

Tradicionalna funkcija predstavlja *imenovani* blok koda koji obuhvata smislen zadatak. U objektno-orijentisanom programiranju funkcije su često *metode* koje definišu ponašanje objekta nad kojim se pozivaju. Principi koje poštujemo za formiranje čistih funkcija su jednako primenjivi na metode. Čista funkcija je *fokusirana na jedan zadatak*. Ovaj zadatak je jasno opisan kroz nazive u zaglavlju funkcije, što uključuje naziv funkcije i nazive njenih parametra. Čista funkcija ima jednostavno telo i sastoji se od koda koji zahteva malo mentalnog napora da se razume. Kao posledica, ovakve funkcije često sadrže mali broj linija koda. Kroz ovaj modul ispitujemo dobre i loše prakse za pisanje funkcija.





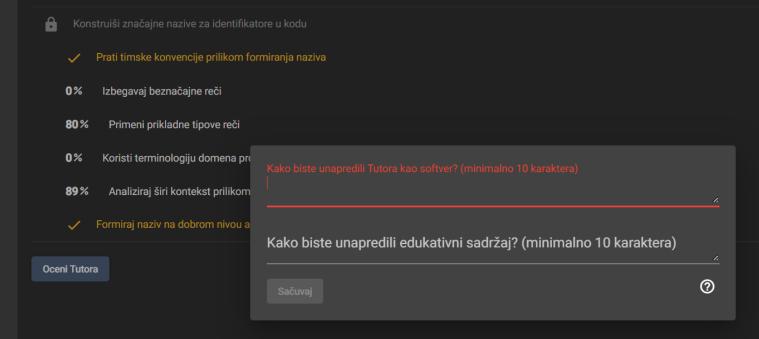


Iskustvo učenja

Unapređenje Tutora

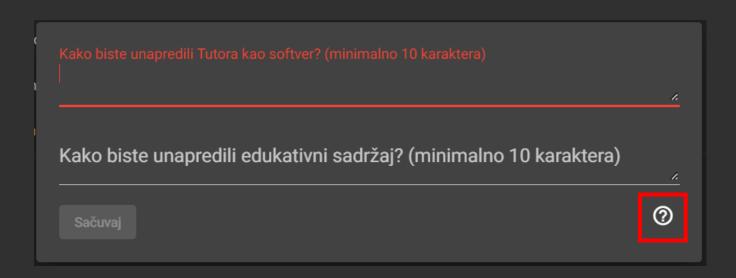
Značajni nazivi

Nazive pronalazimo u svim segmentima razvoja softvera - kod identifikatora promenljive, funkcije, klase, ali i biblioteke i aplikacije. Jasan naziv funkcije nas oslobađa od čitanja njenog tela, dok će misteriozni naziv zahtevati dodatan mentalni napor da razumemo svrhu koncepta koji opisuje. U najgorem slučaju, loš naziv će nas naterati da formiramo pogrešnu pretpostavku, što će nam drastično produžiti vreme razvoja. Kroz ovaj modul ispitujemo dobre i loše prakse za formiranja naziva identifikatora u kodu.





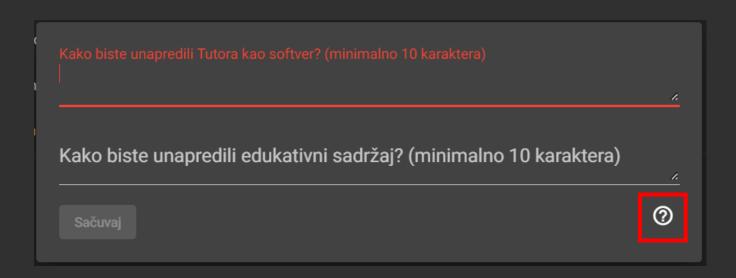
Iskustvo učenja



- Primeri za Tutor kao softver:
 - Noćni mod i dalje zaslepljuje :)
 - Znaci procenta su veći od brojeva u tekstovima.
 - Voleo bih da mogu da hajlajtujem delove teksta i da im dodeljujem komentare.
 - Imena klasa su čudno ispisana u tekstu zadatka. Trebalo bi formatirati tektst da bude isti font.



Iskustvo učenja



- Primeri za edukativni sadržaj:
 - Treba dodati još izazova.
 - Voleo bih više pitanja sa feedbackom.
 - Manje tekstualnih priča, više slika i primera. Slika vredi hiljadu reči :)
 - Izazovi su previše lagani. Treba ih otežati ili smanjiti njihov broj.



Hvala na pažnji!

Pitanja?

