Preferencije programera prema programskim jezicima

Seminarski rad u okviru kursa
Metodologija stručnog i naučnog rada
Matematički fakultet

Mlađenovic Anđela, Marić Đorđe,
Cvijić Lazar, Stojanović Mateja
andjelaa.mladjenovic11@gmail.com, maricdjole0@gmail.com,
lcvijicc@gmail.com, stojanovic.mateja@gmail.com

26. decembar 2025.

Sažetak

U radu se ispituju faktori koji utiču na preferencije programskih jezika kod studenata i mladih programera u kontekstu promenljive popularnosti tehnologija i zahteva industrije. Istraživanje je sprovedeno putem anonimne ankete koja obuhvata demografske podatke, tehničke i eksterne karakteristike programskih jezika, kao i promene preferencija tokom obrazovanja i sticanja iskustva. Analiza odgovora 53 ispitanika ukazuje da izbor programskog jezika u velikoj meri zavisi od oblasti primene, dostupnosti ekosistema i nivoa praktičnog iskustva.

Sadržaj

1 Uvod

Različiti indeksi popularnosti, poput poznatog **TIOBE indeksa**, kontinuirano nam pokazuju da se distribucija korišćenja programskih jezika stalno menja. Broj jezika u upotrebi i dinamična priroda tržišta jasno ukazuju na to da svaki programer mora biti spreman na konstantno učenje i prilagođavanje novim tehnologijama[8].

Kao studenti i mladi programeri sa određenim predznanjem, pre nego što uđemo u dublju analizu preferencija prema specifičnim jezicima, naša je dužnost da se osvrnemo na značaj izučavanja **koncepata programskih jezika**. Različiti programski koncepti i paradigme direktno povećavaju kapacitet za izražavanje naših ideja. Opšte je prihvaćeno uverenje da dubina ljudskog razmišljanja zavisi od izražajne moći jezika kojim osoba komunicira; slično tome, programer je ograničen strukturama jezika koji koristi. Dodatno, proučavanje opštih koncepata programiranja pomaže programeru da bolje koristi jezike koje već poznaje, ali i da znatno brže usvaja nove.

Dizajn programskog jezika nije slučajan, jer je pod direktnim uticajem arhitekture mašine na kojoj će se izvršavati, kao i metodologija dizajna softvera koje dominiraju u određenom periodu. Ovi uticaji oblikuju različite **domene programiranja** – od naučnih proračuna i poslovnih aplikacija, do veb razvoja i ugrađenih (eng. *embedded*) sistema. Svaki od ovih domena zahteva specifičan balans između performansi, sigurnosti i brzine razvoja. Upravo ovi kompromisi (eng. *trade-offs*) čine suštinu dizajna i direktno utiču na to zašto određene grupe programera preferiraju jedne jezike nad drugima[7].

Kroz ovaj rad, pokušaćemo da povežemo ove teorijske postavke sa empirijskim podacima prikupljenim putem istraživanja među mladim programerima. Cilj je ispitati u kojoj meri svest o ovim konceptima, prethodno iskustvo i praktične potrebe industrije oblikuju njihove svakodnevne preferencije.

2 Metodologija istraživanja i izrada ankete

Istraživanja o preferencijama programskih jezika među studentima i praktikantima u literaturi još uvek je ograničena. Jedan od osnovnih izazova u ovoj oblasti je sama metodologija merenja popularnosti. S obzirom na to da je nemoguće direktno posmatrati svakog programera, industrija se oslanja na posredne indikatore (eng. *proxies*) kao što su Google pretrage, pitanja na Stack Exchange-u i aktivnost na GitHub-u. Savremeni razvoj veštačke inteligencije unosi dodatnu kompleksnost u ove metrike. Primećen je drastičan pad javno vidljivih signala interesovanja; na primer, broj pitanja na Stack Exchange-u u 2025. godini pao je na svega 22% u odnosu na 2024. godinu, jer programeri odgovore traže u privatnim razgovorima sa LLM modelima poput Claude-a ili ChatGPT-a[6].

U cilju identifikacije ključnih faktora koji oblikuju preferencije programera prema određenim jezicima, kao i analiziranje promene tih stavova tokom obrazovanja i karijere, sproveli smo istraživanje putem anonimne ankete.

Anketa se sastoji od 26 pitanja grupisanih u tri celine:

1. **Profil ispitanika (Demografski podaci):** Ova sekcija prikuplja podatke o trenutnom statusu ispitanika (student, praktikant ili zaposlen), fakultetu, godini studija i dužini profesionalnog bavljenja programiranjem.
2. **Evaluacija tehničkih i eksternih karakteristika:** U ovom delu ispitanici su ocenjivali (na skali od 1 do 5) uticaj osam ključnih faktora na njihov izbor jezika: čitljivost koda, brzina razvoja, performanse, stabilnost, dostupnost biblioteka, zajednica, popularnost na tržištu i open-source podrška.
3. **Iskustvo, evolucija i budući trendovi:** Poslednja sekcija fokusirana je na to kako se preferencije menjaju tokom vremena, da li izbor zavisi od tipa projekta i koje jezike ispitanici vide kao relevantne za svoju budućnost.

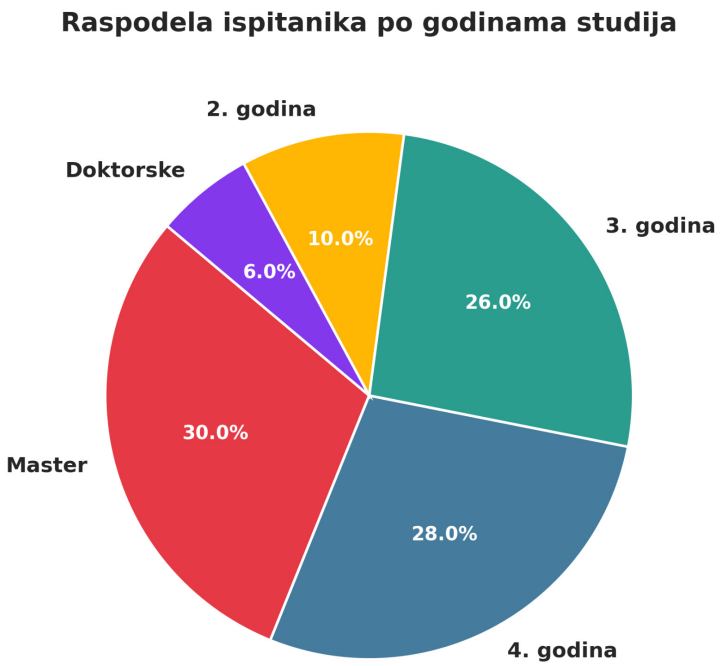
Svi podaci su prikupljeni preko Google Forms platforme. Pre statističke obrade dobijenih podataka, proverena je validnost podataka i eliminisani su neadekvatni i nepotpuni unosi. Nakon validacije, formiran je finalni uzorak od **53 ispitanika**, čiji su odgovori u potpunosti zadovoljavali kriterijume istraživanja.

3 Rezultati

3.1 Demografski profil ispitanika i nivo iskustva

Struktura uzorka pokazuje jasnu dominaciju studenata Matematičkog fakulteta u Beogradu, koji čine 90,6% ukupnog broja ispitanika. Iako uzorak nije ravnomerno raspoređen po fakultetima, ovakva struktura je očekivana s obzirom na kontekst sprovođenja ankete i relevantna je za istraživanje preferencija programskih jezika kod studenata tehničkih i prirodno-matematičkih fakulteta.

Godina studija i profesionalno iskustvo ispitanika predstavljaju ključne demografske parametre. Oni omogućavaju sagledavanje razlika u stavovima i iskustvima sa programskim jezicima tokom različitih faza obrazovnog i profesionalnog ciklusa. Raspodela ispitanika po ovim parametrima ukazuje na izbalansiran uzorak. Prisustvo studenata nižih i viših godina, kao i onih sa različitim stepenom iskustva u industriji, omogućava analizu promene preferencija tokom akademskog i profesionalnog napredovanja.



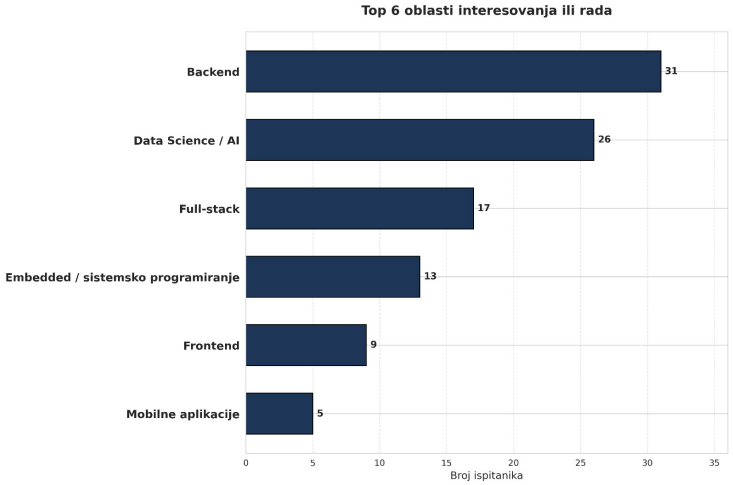
Slika 1: Raspodela ispitanika po godini studija

Slika 2: Raspodela ispitanika po godinama profesionalnog iskustva

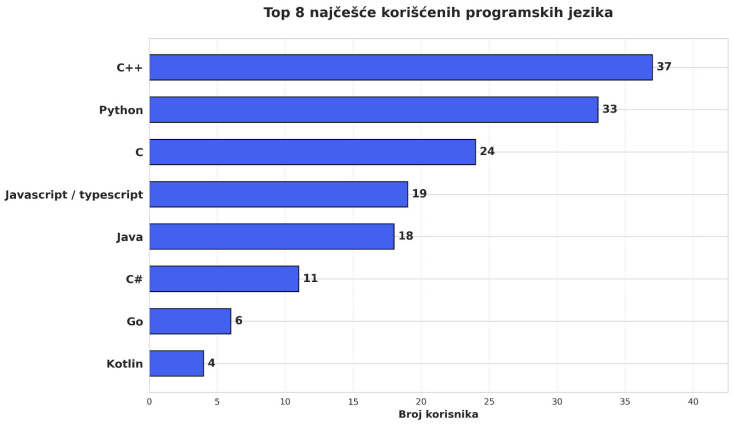
3.2 Oblasti interesovanja i omiljeni jezici ispitanika

Kada je reč o primarnim oblastima interesovanja, ispitanici su imali mogućnost višestrukog izbora, što pruža uvid u multidisciplinarnost uzorka i širinu njihovog tehničkog usmerenja. Dominantna oblast rada i interesovanja je **back-end** razvoj (31 ispitanik), zatim **nauka o podacima** (eng. *data science*) i veštačka inteligencija (26), a značajno su zastupljeni i full-stack razvoj (17) i sistemsko i embedded programiranje (13).

Ovakva raspodela interesovanja omogućava dublju analizu korelacije između izbora programskog jezika i specifičnih zahteva različitih oblasti programiranja.



Slika 3: Raspodela ispitanika po oblastima interesovanja



Slika 4: Najčešće korišćeni jezici

Prilikom selekcije jezika koje najčešće koriste, ispitanici su jasno izdvojili **C++** (37) i **Python** (33) kao dominantne jezike u svom radu.

Nakon identifikacije jezika koje koriste, od ispitanika je traženo da rangiraju svoja tri omiljena jezika. Rezultati potvrđuju dominaciju vodećeg dvojca, ali otkrivaju zanimljiv uvid u prvi izbor studenata. Programski jezik **C++** ne samo da je najkorišćeniji, već je i apsolutni favorit kada je u pitanju prvi izbor ispitanika (**22 glasa na prvom mestu**). Sledeći na listi nalazi se **Python**, koji zauzima drugo mesto sa **10 glasova** na prvom mestu, pri čemu je važno naglasiti da ga je preko 50% ispitanika uvrstilo u svoja tri omiljena programska jezika. Može se uočiti jasna korelacija između jezika koje najčešće koriste (a time i poznaju), sa onima koje najviše preferiraju.

U cilju razumevanja odnosa između primarne oblasti rada i preferencija prema konkretnim jezicima, sprovedena je analiza na uzorku ispitanika različitih profila. Ključno je bilo utvrditi da li specifična domenska ograničenja (npr. potreba za kontrolom na najnižem nivou u embedded sistemima naspram brže iteracije u frontend razvoju) direktno oblikuju listu omiljenih jezika programera, nezavisno od onih koje su primorani da koriste na radnom mestu.

Naredna tabela (Tabela 1) prikazuje procentualnu zastupljenost jezika koje su ispitanici rangirali u svoja tri omiljena izbora, kategorisane prema njihovim primarnim oblastima rada.

Oblast rada	**C++**	**Python**	**C**	**Java**	**JS/TS**	**C#**	**Go**	**Kotlin**
Backend	55%	41%	31%	24%	10%	10%	17%	3%
Full-stack	36%	43%	14%	36%	50%	14%	21%	0%
Data Science / AI	54%	65%	31%	19%	8%	12%	0%	4%
Embedded / sistemsko	69%	31%	77%	8%	8%	8%	15%	0%
Mobilne aplikacije	40%	40%	20%	60%	20%	0%	0%	40%
Razvoj igara	100%	0%	33%	0%	33%	66%	0%	0%
Frontend	0%	0%	0%	100%	100%	0%	0%	0%

Tabela 1: Zastupljenost programskih jezika po oblasti rada

Rezultati ukazuju na postojanje visoke korelacije između domena rada i tehnološkog afiniteta. C++ je popularan nezavisno od domena, verovatno zbog toga što su skoro svi ispitanici pohađali Matematički fakultet. Jezici poput Kotlin-a i C# ostaju strogo vezani za domene u kojima su idiomatski (mobilne aplikacije i razvoj igara). Posebno je interesantan podatak da full-stack programeri pokazuju najveći diverzitet u odgovorima, što sugeriše potrebu za širim tehnološkim spektrom u poređenju sa specijalizovanim ulogama.

3.3 Uticaj tehničkih aspekata na preference programskih jezika

Najviše je ocenjen faktor **Zajednica, dokumentacija i dostupnost materijala (4.49)** , što ukazuje na to da je usamljenost u rešavanju problema najveći rizik za produktivnost. Čak i tehnički bolji jezik može biti odbačen ukoliko ne nudi bogatu bazu znanja i brze odgovore na platformama poput Stack Overflow-a.[1]

Faktor	Prosečna ocena
Zajednica, dokumentacija i dostupnost materijala za učenje	4.49
Čitljivost koda	4.36
Stabilnost	4.34
Performanse	4.09
Dostupnost biblioteka, alata i framework-a	3.96
Open-source podrška	3.70
Brzina razvoja	3.66
Popularnost na tržištu rada	3.45

Tabela 2: Prosečne ocene faktora koji utiču na izbor programskog jezika

Visoko mesto **čitljivosti (4.36)** potvrđuje teorijsku promenu u dizajnu novijih jezika, gde se fokus pomerio sa mašinske orijentacije na ljudsku orijentaciju. Kako održavanje softvera čini najveći deo troškova njegovog životnog ciklusa, čitljivost je postala ključno merilo kvaliteta[7]. Naši ispitanici to prepoznaju, vrednujući razumljivost koda skoro podjednako kao i njegovu stabilnost.

Ispitanici daju prednost **performansama (4.09)** u odnosu na **brzinu razvoja (3.66)**. Ovakav sistem vrednosti objašnjava zašto je **C++** pobedio **Python** kao najomiljeniji jezik kod ispitanika[3].

Na samom kraju liste nalazi se **popularnost na tržištu rada (3.45)**. Ovo sugeriše da su ispitanici u ovoj analizi više vođeni kvalitetom i tehničkom izvrsnošću jezika nego trenutnim trendovima u zapošljavanju.

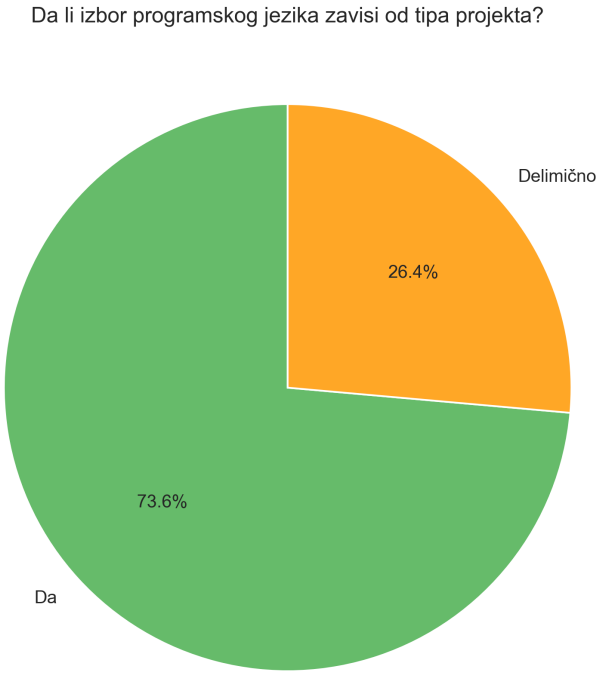
Istraživanje jasno pokazuje da su programeri spremni da prihvate određene mane u svom omiljenom jeziku zarad benefita koje konkretni jezik nudi. Neke od čestih zamerki koje se navode su:

- **Kompleksnost i težina savladavanja**
  Česta zamerka je komplikovana sintaksa, veliki broj koncepata i strma kriva učenja, posebno kod jezika sa niskim nivoom apstrakcije.
- **Upravljanje memorijom i bezbednost**
  Ručno upravljanje memorijom i manjak bezbednih apstrakcija često povećavaju rizik od grešaka i otežavaju razvoj.
- **Performanse ili neefikasnost**
  Dinamički tipizirani ili visoko apstraktni jezici se ponekad percipiraju kao sporiji ili zahtevniji po pitanju resursa.
- **Previše ili zastarelih koncepata**
  Navodi se prisustvo zastarelih mehanizama, prevelikog boilerplate koda ili obimnih biblioteka koje mogu otežati snalaženje.
- **Alati i ekosistemska ograničenja**
  Nedostatak određenih razvojnih alata, slabije integrisani build sistemi ili neujednačen razvoj tržišta rada takođe se ističu kao problemi.

3.4 Navike i razvoj mišljenja

Na pitanje u kojoj meri prethodno iskustvo utiče na izbor jezika, najučestalija ocena je 4 (izabralo 28 ispitanika, odnosno 52,8%), dok je srednja vrednost visokih 3,79. Međutim, oslanjanje na poznate alate nije prepreka za prilagođavanje, jer čak 73,4% ispitanika ističe da njihov **izbor primarno zavisi od tipa projekta**. Ovo nam govori da su programeri

često spremni zanemariti svoje preference kako bi zadovoljili potrebe konkretnog projekta.



Slika 5: Izbor jezika zavisno od projekta

Interesantno je razmotriti kako se preferencije programera menjaju tokom studija i rada. Većina ispitanika (62%) prijavljuje da su im se vremenom preference promenile.
Kao glavni razlozi se navode:

- **Sticanje praktičnog iskustva**
  Kroz rad na projektima i u realnom okruženju dolazi do jasnijeg uvida u prednosti i nedostatke programskih jezika, posebno u pogledu održavanja, skaliranja i timskog rada. Vremenom se razvija svest da ne postoji univerzalni jezik za sve probleme, već da izbor zavisi od vrste projekta i njegove namene.
- **Upoznavanje novih jezika, alata i tehnologija**
  Tokom studija i rada dolazi do stalnog susreta sa novim jezicima, bibliotekama i alatima, što proširuje perspektivu i utiče na promenu preferencija. Nastavni planovi, istraživački rad i zahtevi industrije često usmeravaju korišćenje određenih jezika, čime se formiraju nove preferencije.
- **Promena prioriteta u razvoju softvera**
  Fokus se može pomeriti sa jednostavnosti i brzine pisanja koda ka sigurnosti, robusnosti i jakom tipiziranju, ili obrnuto - ka većoj produktivnosti i brzom prototipisanju.

- **Razvoj interesovanja i specijalizacije**
  Promena oblasti interesovanja direktno utiče na izbor programskih jezika.
- **Bolje razumevanje apstrakcije i produktivnosti**
  Jezici sa višim nivoom apstrakcije vremenom se više cene zbog čitljivosti, komfora i efikasnosti u razvoju kompleksnih sistema.

Poslednji segment istraživanja pruža uvid u to kako ispitanici planiraju svoj dalji tehnički razvoj. Analiza odgovora na pitanje o jezicima koje bi želeli bolje da nauče ukazuje na to da i mlađi i stariji studenti teže detaljnijem usavršavanju jezika koje već poznaju i koriste. **Python** ostaje u vrhu prioriteta zbog svoje široke primene, **C++** se bira primarno zbog performansi, kontrole nad resursima i primene u zahtevnim oblastima poput gaming industrije i kvantitativnih analiza. Zanimljivo je pojavljivanje jezika kao što su **Go** i **Rust**, koji trenutno nisu dominantno zastupljeni u kursevima koje ispitanici pohađaju, ali su prepoznati kao tehnološki značajni. Među iskusnijim ispitanicima (zaposleni i diplomirani studenti) jasno se nagoveštava trend u kojem se **Rust vidi kao naslednik koji bi mogao da zameni C++**.

3.5 Diskusija

Rezultati pokazuju da ispitanici jasno prepoznaju prednosti i mane svojih omiljenih jezika, birajući ih spram konkretnih oblasti primene. C++ se ističe kao apsolutni favorit zbog performansi. Python dominira u sferi veštačke inteligencije i mašinskog učenja zahvaljujući visokoj produktivnosti, dok mlađi ispitanici i dalje prepoznaju C kao ključan alat za razvoj operativnih sistema i embedded rešenja. Ispitanici orijentisani ka vebu biraju JavaScript, dok se kod mobilnih aplikacija prirodno izdvaja Kotlin.

Pitanje kvaliteta i održavanja koda takođe je ključno. Istraživanje [2] koje je sprovedeno na GitHub projektima pokazalo je da korišćenje više jezika u jednom projektu može smanjiti kvalitet koda, što je u skladu sa našim nalazom da ispitanici visoko vrednuju čitljivost (4.36) i stabilnost (4.34).

Poseban uvid u istraživanje pružaju odgovori iskusnijih ispitanika (zaposlenih i studenata viših godina), koji prepoznaju jezike Go i Rust kao korisne za svoj rad i programersko iskustvo.
Go je nastao u kompaniji Google kao direktan odgovor na frustraciju inženjera procesima razvoja unutar kompanije, prvenstveno zbog spore kompilacije i glomaznih binarnih fajlova kod velikih serverskih sistema pisanih u C++-u. Dizajniran na „čistom listu papira", Go je kreiran sa jasnim ciljevima: softver mora da se gradi brzo, da efikasno koristi moderan hardver i da prirodno funkcioniše u mrežnom okruženju. Rezultat je jezik koji zadržava performanse i sigurnost C-a ili Jave, ali nudi „osećaj" programiranja u dinamičkim jezicima poput Pythona, čineći ga izuzetno prijatnim za korišćenje uz maksimalnu pragmatičnost. [4]

Sa druge strane, Rust se u odgovorima iskusnijih ispitanika izdvaja kao tehnološki najnapredniji naslednik C-olikih jezika. Iako je relativno mlad, omogućava jedinstven više paradigmatski pristup – on uspešno kombinuje elemente imperativnog programiranja i objektno-orijentisane karakteristike C++-a sa funkcionalnim karakteristikama jezika kao što su Haskell i

OCaml. Ključni razlog zašto ga profesionalci biraju je njegova sposobnost da obezbedi stroge garancije sigurnosti memorije i bezbedan konkurentni rad. Uvođenjem koncepata koji sprečavaju konflikte pri istovremenom pristupu podacima (eng. *data races*), Rust omogućava razvoj besprekorno stabilnog sistemskog koda, što ga u očima ispitanika postavlja kao primarnu alternativu za kritične infrastrukturne projekte. [5]

4 Zaključak

Ovo istraživanje je imalo za cilj da ispita faktore koji oblikuju preferencije programskih jezika kod studenata i mladih programera, kao i da analizira evoluciju tih stavova tokom obrazovanja i sticanja profesionalnog iskustva. Kroz analizu odgovora 53 ispitanika, potvrđene su ključne teorijske postavke iz literature, ali su i otkriveni novi uvidi specifični za kontekst mladih programera u Srbiji.

Rezultati jasno ukazuju da **izbor programskog jezika nije proizvoljne prirode**, već proističe iz kompleksne interakcije tehničkih zahteva, ekosistemske podrške i praktičnih potreba. Dominacija jezika C++ i Python u preferencijama ispitanika nije slučajna – ona odražava fundamentalni kompromis između performansi i produktivnosti koji se provlači kroz istoriju dizajna programskih jezika. Istovremeno, visoko vrednovanje faktora kao što su dostupnost dokumentacije (4.49) i čitljivost koda (4.36) potvrđuje da je programiranje pre svega **društvena i kolaborativna aktivnost**, gde kvalitet zajednice može biti presudniji od tehničke superiornosti samog jezika.

Značajan nalaz je da čak 73,4% ispitanika prilagođava svoj izbor jezika tipu projekta, što ukazuje na pragmatičan pristup i svest o domenskim ograničenjima. Ova fleksibilnost, zajedno sa činjenicom da 62% ispitanika prijavljuje promenu preferencija tokom vremena, sugeriše da mladi programeri usvajaju višeparadigmatski način razmišljanja – pristup koji Sebesta[7] identifikuje kao ključan za savremeni softverski razvoj. Promena stavova je najčešće rezultat sticanja praktičnog iskustva, što potvrđuje hipotezu da teorijsko poznavanje jezika nije dovoljno – tek rad na realnim projektima omogućava dublje razumevanje prednosti i kompromisa različitih rešenja.

Posebno je interesantno pojavljivanje jezika kao što su Go i Rust u planovima budućeg usavršavanja, uprkos njihovoj trenutno niskoj zastupljenosti u nastavnim planovima. Ova tendencija otkriva svest ispitanika o dinamičnoj prirodi industrije i potrebi za stalnim učenjem. Rust se posebno ističe kao tehnološki naslednik C++ jezika među iskusnijim ispitanicima, što sugeriše potencijalni pomak u sistemskom programiranju ka jezicima sa jačim garancijama sigurnosti memorije.

4.1 Ograničenja istraživanja

Kao i svako empirijsko istraživanje, i ovo ima određena ograničenja. Najpre, uzorak od 53 ispitanika, od kojih je 90,6% sa Matematičkog fakulteta u Beogradu, ograničava uopštavanje rezultata na širu grupu programera. Dominantna zastupljenost studenata jednog fakulteta može uticati na izraženu preferenciju prema C++-u, s obzirom na strukturu nastavnog plana. Takođe, kratkoročna priroda istraživanja ne omogućava dugoročno praćenje promene stavova, već se oslanja na retrospektivne izjave ispitanika.

4.2 Pravci budućih istraživanja

Buduća istraživanja bi mogla biti usmerena ka nekoliko pravaca. Prvo, praćenje iste grupe ispitanika tokom nekoliko godina omogućilo bi preciznije sagledavanje evolucije preferencija. Drugo, uporedna analiza stavova studenata različitih fakulteta mogla bi razotkriti uticaj nastavnih programa na formiranje preferencija. Konačno, s obzirom na dramatičan rast upotrebe AI alata u programiranju, buduća istraživanja bi trebalo da ispitaju kako veštačka inteligencija menja odnos programera prema izboru jezika - da li asistenti poput GitHub Copilot-a ili ChatGPT-a smanjuju značaj faktora kao što su sintaksa ili dokumentacija.

U zaključku, ovo istraživanje potvrđuje da su preferencije programskih jezika dinamična i kontekstualno uslovljena pojava. One nisu samo odraz ličnih sklonosti, već rezultat interakcije između obrazovanja, praktičnog iskustva, zahteva industrije i tehnoloških trendova. Razumevanje ovih faktora ne samo da pomaže obrazovnim institucijama da prilagode svoje nastavne planove, već i samim programerima da svesnije upravljaju svojom karijerom u brzo promenljivom tehnološkom okruženju.

Literatura

[1] Stack overflow. https://stackoverflow.com. Online; accessed 26-Dec-2025.

[2] Programming language usage by students, practitioners and hobbyists. *Issues in Information Systems*, 2020. There is limited literature on the usage of programming languages used by students, practitioners or hobbyists.

[3] Analysis of programming language usage trends. *European Journal of Theoretical and Applied Sciences*, 2021.

[4] I. Balbaert. *The Way to Go: Introduction to the Go Programming Language*. iUniverse, 2012.

[5] IBM Developer. Why you should learn the rust programming language. https://developer.ibm.com/articles/os-developers-know-rust/, 2024.

[6] IEEE Spectrum. Top programming languages 2025. https://spectrum.ieee.org/top-programming-languages-2025, 2025.

[7] Robert W. Sebesta. *Concepts of Programming Languages*. Addison-Wesley, 10 edition, 2012.

[8] TIOBE Software. Tiobe index. https://www.tiobe.com/tiobe-index/, 2025. Accessed: 2025.