

Stilovi učenja u tehničkim disciplinama: šta najbolje pomaže programeru?

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

Boško Andrić, Kristijan Petronijević
mi251012@alas.matf.bg.ac.rs, mi251012@alas.matf.bg.ac.rs

25. decembar 2025.

Sažetak

U ovom radu analizirani su stilovi učenja i obrazovne navike studenata tehničkih usmerenja, sa posebnim fokusom na učenje programiranja. Kroz sprovedenu anketu ispitani su najčešće korišćeni resursi, stilovi učenja i percepcija njihove efikasnosti. Dobijeni rezultati pružaju uvid u to kako studenti pristupaju učenju i mogu poslužiti kao osnova za unapređenje nastavne prakse.

Sadržaj

1	Uvod	2
1.1	Stilovi učenja i njihov značaj	2
1.2	Cilj rada	2
2	Stilovi učenja u tehničkim disciplinama	3
2.1	Dominantni pristupi učenju	3
2.2	Učenje sintakse i razumevanje algoritama	4
2.3	Učenje u grupi	5
3	Korišćeni resursi u učenju programiranja	6
3.1	Resursi	6
3.2	Nekonvencionalni resursi	7
4	Način rada i stavovi o nastavi	8
4.1	Problemi pri učenju programiranja	10
5	Zaključak	10
	Literatura	11

1 Uvod

Različiti studenti uče na različite načine, a razumevanje koji stilovi učenja im najviše pomažu ključno je za poboljšanje procesa obrazovanja. U ovom radu fokusirali smo se na identifikaciju dominantnih stilova učenja i procenu njihove efikasnosti, posebno u kontekstu učenja programiranja i tehničkih koncepata. Cilj je bio da uporedimo sopstvena iskustva sa percepcijama i praksama drugih studenata i da kroz analizu ankete dodemo do uvida u to koje metode učenja najbolje funkcionišu u praksi.

1.1 Stilovi učenja i njihov značaj

Jedan od najpoznatijih pristupa proučavanju stilova učenja jeste VARK(eng. *Visual, Aural, Read/Write, Kinesthetic*) model, koji je razvio Neil D. Fleming. Ovaj model polazi od pretpostavke da se pojedinci razlikuju prema dominantnom načinu na koji primaju i obrađuju informacije, te razlikuje četiri osnovna stila učenja: vizuelni, auditivni, praktični/kinestetički i čitalački/pisani. Prema Flemingu, prepoznavanje sopstvenog stila učenja može doprineti efikasnijem usvajanju znanja, većoj motivaciji i boljem akademskom postignuću, posebno u oblastima koje zahtevaju razumevanje apstraktnih i tehnički složenih koncepata [4].

U kontekstu programiranja i srodnih tehničkih disciplina, stil učenja može imati značajan uticaj na način na koji studenti razumeju sintaksu, algoritamsko razmišljanje i rešavanje problema. Dok pojedini studenti preferiraju vizuelne prikaze i dijagrame, drugi lakše usvajaju znanje kroz praktičan rad, pisanje koda i rešavanje konkretnih zadataka.

Iako studenti znaju njihove, lične preferense u učenju, čak skoro 70% njih ne uspeva da koristi tehnikе učenja koji podržavaju te preferense [1]. Ovo nam govori da moramo biti opreznι pri razmatranju i izučavanju stilova učenja.

1.2 Cilj rada

Cilj ovog rada jeste da se ispita koji stilovi učenja dominiraju među studentima tehničkih usmerenja i kako studenti procenjuju efikasnost različitih metoda učenja, sa posebnim fokusom na učenje programiranja. U tu svrhu sprovedena je anketa, a prikupljeni podaci analizirani su kako bi omogućio uvid u dominantne stilove učenja, resurse koji studenti koriste, kao i u njihove navike i stavove u vezi sa procesom učenja.

Posebna pažnja posvećena je razumevanju načina na koji studenti kombinuju različite pristupe učenju, kao i identifikovanju metoda koje smatraju najefikasnijim u savladavanju kompleksnih programerskih i tehničkih koncepata. Dobijeni rezultati mogu doprineti boljem razumevanju potreba studenata i poslužiti kao osnova za unapređenje nastavnih metoda i organizacije nastave na tehničkim fakultetima.

Rad je organizovan na sledeći način: nakon uvodnog dela, u narednim poglavljima biće predstavljena metodologija istraživanja, struktura ankete, kao i analiza i diskusija dobijenih rezultata, dok se u završnom delu iznose zaključci i razmatraju mogući pravci za unapređenje nastavne prakse u tehničkim disciplinama.

2 Stilovi učenja u tehničkim disciplinama

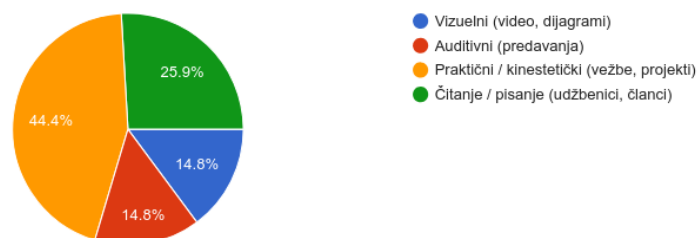
U tehničkim disciplinama, a naročito u oblasti programiranja, način na koji studenti usvajaju znanja ima ključnu ulogu u razumevanju gradiva i njegovoj uspešnoj primeni u praksi. Proces učenja u ovim oblastima često podrazumeva rad sa apstraktnim konceptima, formalnim pravilima i složenim problemima, zbog čega stilovi učenja predstavljaju važan faktor koji može značajno olakšati ili otežati savladavanje gradiva.

2.1 Dominantni pristupi učenju

Na osnovu rezultata ankete može se uočiti da studenti tehničkih disciplina najčešće primenjuju pristupe učenja koji uključuju aktivno učestvovanje u procesu rada. Direktan rad na zadacima, samostalno rešavanje problema i pisanje koda izdvajaju se kao centralni elementi efikasnog učenja. Ovakav pristup je u skladu sa prirodom programiranja, gde se teorijsko znanje vrlo brzo proverava i potvrđuje kroz praktičnu primenu.

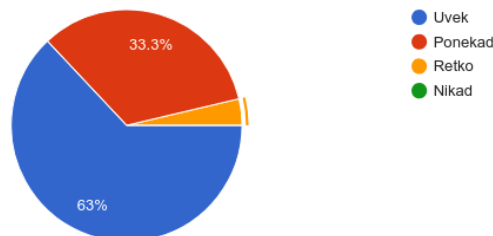
Pored praktičnog pristupa, značajan broj studenata koristi i učenje kroz čitanje i pisanje, što ukazuje na važnost razumevanja dokumentacije, primera koda i teorijskih objašnjenja. Vizuelni i auditivni pristupi takođe imaju svoju ulogu, naročito kao dopuna drugim stilovima učenja. Vizuelni elementi poput dijagrama, grafova i šema pomažu u organizaciji informacija, dok auditivni pristup može biti koristan kroz objašnjenja, diskusije i video-materijale.

Na slici 1 prikazana je raspodela dominantnih stilova učenja među ispitanicima.



Slika 1: Stilovi učenja

Rezultati ankete dodatno ukazuju na to da se učenje u tehničkim disciplinama retko oslanja na jedan isključivi stil. Studenti najčešće kombinuju više pristupa, povezujući teorijsko gradivo sa praktičnim primerima i vizuelnim objašnjenjima. Ovakav multimodalni pristup omogućava fleksibilnije usvajanje znanja i prilagođavanje različitim tipovima sadržaja. Na slici Na slici 2 možemo videti da preko 95% ispitanika bar povremeno kombinuju stilove pri učenju.



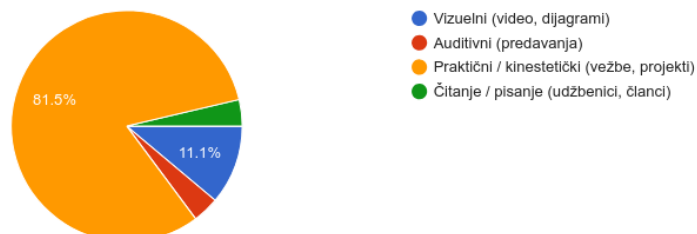
Slika 2: Kombinovanje stilova učenja

2.2 Učenje sintakse i razumevanje algoritama

Prava korist od korišćenja vizuelizacije možda ne leži u njihovom sadržaju, već u njihovoj upotrebi kao motivacioni faktor koji pokreće studente da se zalažu više[5].

Kod savladavanja sintakse novog programskog jezika, praktični pristup se pokazuje kao posebno značajan. Pisanje koda, eksperimentisanje sa primerima i ispravljanje grešaka omogućavaju studentima da kroz neposredno iskustvo razumeju pravila jezika i njihovu primenu. Ovakav način učenja doprinosi bržem usvajanju sintakse i većem samopouzdanju prilikom rada u novom programskom okruženju.

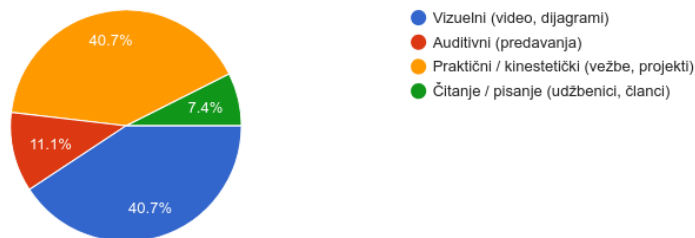
Na slici 3 prikazan je odnos stilova učenja koji se koriste prilikom učenja sintakse programskih jezika.



Slika 3: Stil učenja za razumevanje sintakse novog programskog jezika

Sa druge strane, kod razumevanja algoritama i programskih koncepata, uočava se veći značaj vizuelnog pristupa. Algoritmi često uključuju apstraktne procese i višekoračne tokove izvršavanja, zbog čega vizualizacija ima važnu ulogu u njihovom razumevanju. Grafički prikazi, dijagrami toka, animacije algoritama i vizualizacija struktura podataka omogućavaju studentima da jasnije sagledaju tok izvršavanja, međukorake i međusobne odnose između elemenata algoritma.

Kombinacija vizuelnog pristupa sa praktičnom implementacijom algoritama u kodu dodatno doprinosi dubljem razumevanju gradiva. Studenti na taj način ne samo da vide kako algoritam funkcioniše, već i kako se njegova logika prevodi u konkretno programsko rešenje, što olakšava kasniju



Slika 4: Stil učenja za razumevanje algoritama i programskih koncepata

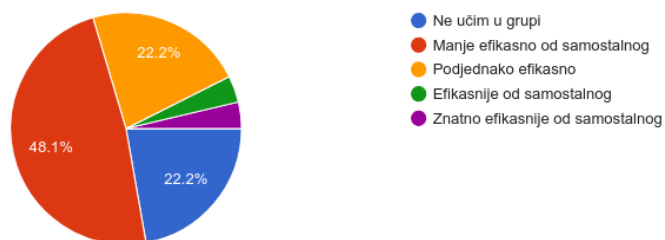
primenu i modifikaciju algoritama.

Na slici 4 prikazani su stilovi učenja koji se najčešće koriste pri razumevanju algoritama i programskih koncepata.

2.3 Učenje u grupi

Učenje programiranja u grupi predstavlja oblik kolaborativnog učenja u kojem studenti kroz međusobnu razmenu znanja, diskusiju i zajedničko rešavanje problema razvijaju svoje tehničke i analitičke veštine. Ovakav pristup omogućava razmenu različitih perspektiva, brže uočavanje grešaka i efikasnije razumevanje kompleksnih koncepata, posebno kroz rad u manjim timovima ili parovima.

Studenti u kolaborativnom okruženju učenja moraju da iznose svoje lične stavove, ali i da slušaju stavove ostalih kako bi zajedno došli do novih saznanja i smisla zajedno. Studenti u grupi poštuju i ističu sposobnosti i doprinose individualnih članova grupe [6].



Slika 5: Misljenje o efikasnosti učenja u grupi

Rezultati ankete ukazuju na to da učenje u grupi ima umerenu, ali ne i dominantnu ulogu u procesu savladavanja programerskih znanja. Većina ispitanika navodi da je učenje u grupi manje efikasno od samostalnog, što može ukazivati na to da studenti prepoznaju njegove prednosti, ali ga ne doživljavaju kao primarni način učenja. Razlog za ovakav rezultat može se pronaći u prirodi programiranja, koje često zahteva individualnu koncentraciju i samostalno rešavanje problema, dok se grupno učenje češće

primenjuje za razmenu ideja.

3 Korišćeni resursi u učenju programiranja

Učenje programiranja i srodnih tehničkih disciplina zahteva korišćenje različitih izvora i metoda kako bi se savladalo kako teorijsko znanje, tako i praktične veštine. U okviru naše ankete, ispitanici su odgovarali na pitanja vezana za resurse koje najčešće koriste u procesu učenja, kao i o efikasnosti tih resursa u savladavanju gradiva. Analiza ovih podataka omogućava uvid u to koje vrste materijala studenti smatraju najkorisnijim, koliko im praktične vežbe pomažu u učenju i u kojoj meri koriste savremene alate poput AI sistema ili tuđeg koda za podršku svom obrazovanju.

3.1 Resursi

Analiza odgovora na pitanje o najčešće korišćenim izvorima za učenje programiranja pokazala je da studenti pretežno koriste video tutorijale (81,5%), zvaničnu dokumentaciju (70,4%), udžbenike ili članke (63%) i praktične projekte (59,3%), dok su interaktivni kursevi i mentorski rad zastupljeni u znatno manjoj meri. Takođe, možemo primetiti i dobar broj poklapanja među ova četiri najzastupljenija resursa, odnosno veliki broj ispitanika koji su izabrali dokumentaciju, izabrali su i video tutorijale, čak njih (84,2%), slično se može uočiti i za ostale resurse.

Video tutorijali, dokumentacija i udžbenici/članci su najčešće izabrani resursi jer omogućavaju studentima da brzo i jasno pristupe potrebnim informacijama. Video materijali pružaju vizuelnu i praktičnu demonstraciju koncepta, dok dokumentacija i udžbenici omogućavaju detaljno razumevanje i proveru informacija. Ove kombinacije resursa omogućavaju balans između teorijskog učenja i praktične primene, što je naročito važno u savladavanju programiranja. Ovaj rezultat je jako dobra slika popularnih stilova učenja dobijenih anketom. Video tutorijali u potpunosti odgovaraju vizuelnom stilu učenja, dok u kombinaciji sa dokumentacijom i udžbenicima daju dobru podlogu za učenje praktičnim stilom.

Slični rezultati se mogu pronaći i u naučnom radu "Kako programeri pronalaze onlajn resurse za učenje" (eng. *How Programmers Find Online Learning Resources*) [2]. U ovom radu dobijeni rezultati ukazuju da su studenti najviše koristili nekonvencionalne izvore, poput raznih foruma, a potom zvaničnu dokumentaciju i video materijale, što se u dobroj meri poklapa sa našim rezultatima, imajući u vidu da se u naše ponuđene resurse uvrstavaju podgrupe netradicionalnih izvora.

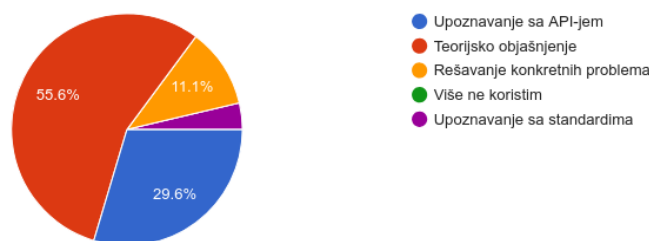
Ocena	1	2	3	4	5
Koliko pomažu praktične vežbe	0	1	3	4	19
Koliko pomažu video materijali	0	2	4	10	11

Tabela 1: Ocene efikasnosti praktičnih vežbi i video materijala u učenju programiranja, na skali od 1 do 5.

Rezultati vezani za procenu korisnosti praktičnih vežbi i video materijala se mogu videti u tabeli 1. Ispitanici su davali ocene od 1 do 5 u zavisnosti od toga koliko smatraju da im odgovarajući resursi pomažu.

Što se tiče korisnosti praktičnih vežbi pokazuju se izrazito pozitivan stav ispitanika. Najveći broj studenata ocenio je praktične vežbe najvišom ocenom, dok je zanemarljiv broj onih koji smatraju da one malo ili nimalo doprinose učenju. Ovakva raspodela ukazuje na to da studenti u velikoj meri prepoznaju praktičan rad kao ključni element u savladavanju programerskih znanja.

Sličan trend primećen je i kod procene korisnosti video materijala. Većina ispitanika ocenila je video tutorijale visokim ocenama, što potvrđuje da vizuelni i auditivni sadržaji predstavljaju značajnu podršku u učenju. Iako je broj ispitanika koji su dali najvišu ocena nešto manji u poređenju sa praktičnim vežbama, rezultati ukazuju da video materijali imaju važnu ulogu u objašnjavanju novih pojmova i dopunjuju druge oblike učenja, posebno u ranim fazama savladavanja gradiva.



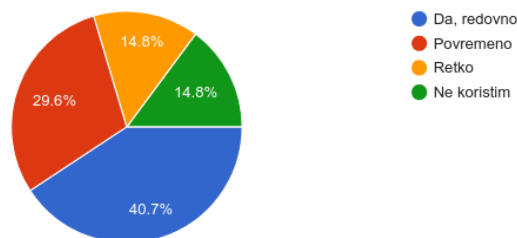
Slika 6: Upotreba dokumentacije

Što se tiče dokumentacije i za šta se ona koristi, na osnovu rezultata koji se nalaze na slici 6 možemo videti da čak 85.2% ispitanika najčešće koristi dokumentaciju za teorijska objašnjenja ili upoznavanje se API-jem (eng. *Application Programming Interface*). Ovakav rezultat ukazuje na to da ispitanici dokumentaciju pre svega doživljavaju kao pouzdan izvor formalnih i preciznih informacija, neophodnih za razumevanje strukture i načina rada određenih tehnologija. Dokumentacija se, u tom smislu, koristi kao referentni materijal kada je potrebno tačno i provereno objašnjenje.

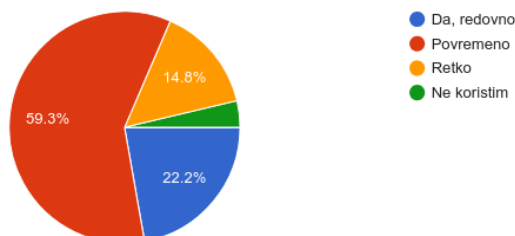
Istovremeno, manja zastupljenost korišćenja dokumentacije za rešavanje konkretnih problema može ukazivati na to da se u tim situacijama češće poseže za primerima iz prakse, video materijalima ili diskusionim forumima, gde se odgovori mogu pronaći brže i u kontekstu stvarnih problema.

3.2 Nekonvencionalni resursi

U savremenom obrazovnom okruženju, pored tradicionalnih izvora učenja, sve značajniju ulogu imaju i tzv. nekonvencionalni resursi. Ovi resursi obuhvataju alate i prakse koje nisu deo klasičnih nastavnih materijala, ali se sve češće koriste kao podrška u učenju, naročito u oblasti programiranja. Među njima se izdvajaju sistemi zasnovani na veštačkoj inteligenciji, kao i javno dostupni repozitorijumi koda, koji omogućavaju brži pristup informacijama, primerima i rešenjima iz realnih projekata.



Slika 7: Upotreba AI alata



Slika 8: Pregled tuđeg koda

Nesumnjivo je da ChatGPT i druga generativna veštačka inteligencija već pomeraju obrazovne granice i pokreću značajnu promenu paradigme u postojećoj obrazovnoj praksi[3]. Imajući to u vidu možemo videti na osnovu rezultata na slici 7 da 70,3% ispitanika bar povremeno koristi alata zasnovane na veštačkoj inteligenciji u učenju.

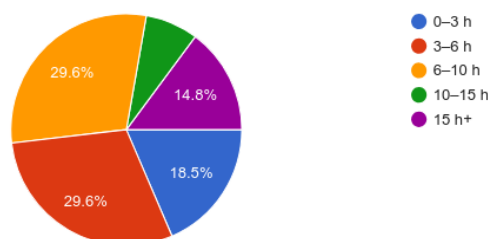
Slično možemo zaključiti i kod pregleda tuđeg koda, što se može videti na slici 8.

4 Način rada i stavovi o nastavi

Način na koji studenti organizuju svoje vreme za učenje, kao i njihov odnos prema fakultetskoj nastavi, ima značajan uticaj na kvalitet usvojenog znanja u tehničkim disciplinama. Programiranje zahteva kontinuitet, aktivno angažovanje i prilagođavanje metoda učenja sopstvenom nivou znanja i iskustva.

Analiza odgovora ispitanika pokazuje da većina studenata redovno izdvaja značajan deo nedeljnog vremena za aktivno učenje programiranja. U praksi, to najčešće podrazumeva samostalno pisanje koda, rešavanje zadataka, rad na projektima i eksperimentisanje sa novim tehnologijama. Manji deo ispitanika provodi relativno malo vremena u aktivnom učenju, dok se određeni broj studenata izdvaja po veoma visokom nivou angažovanosti, što ukazuje na snažnu motivaciju i orijentaciju ka praktičnom usavršavanju.

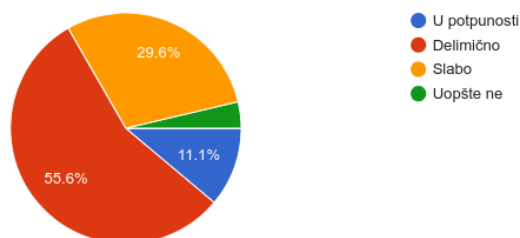
Radi ilustracije raspodele vremena koje studenti posvećuju aktivnom učenju programiranja, na slici 9 prikazan je odgovarajući kružni dijagram.



Slika 9: Vreme posvećeno učenju programiranja tokom nedelje

Kada je u pitanju odnos između fakultetske nastave i ličnih stilova učenja, može se uočiti da većina studenata smatra da nastava samo delimično odgovara njihovim potrebama. Ovo sugeriše da, iako formalna nastava pruža dobru teorijsku osnovu, studenti često osećaju potrebu da dodatno prilagode način učenja kroz samostalni rad, praktične zadatke i korišćenje dodatnih resursa. Manji broj studenata u potpunosti pronalazi svoj stil u okviru postojećeg nastavnog modela, dok deo ispitanika smatra da im nastava slabo ili uopšte ne odgovara, što može predstavljati izazov za njihovu motivaciju i napredovanje.

Na slici 10 prikazan je stav studenata o usklađenosti fakultetske nastave sa njihovim stilom učenja.



Slika 10: Usklađenosti fakultetske nastave sa stilom učenja

Važan aspekt procesa učenja u tehničkim disciplinama predstavlja i promena stilova učenja tokom vremena. Većina studenata primećuje da se njihov način učenja menja kako stiču više iskustva u programiranju. Ova promena se najčešće ogleda u većem oslanjanju na praktičan rad, svesnijem izboru izvora znanja i boljoj organizaciji vremena. Iskustvo omogućava studentima da lakše procene kompleksnost gradiva, izdvoje relevantne informacije i prilagode tempo učenja sopstvenim ciljevima.

Na osnovu kvalitativnih odgovora, može se zaključiti da studenti vremenom razvijaju zreliji i efikasniji pristup učenju. Razlozi za promenu stila uključuju potrebu za većom produktivnošću, uviđanje značaja kon-

tinuiranog pisanja koda, ali i bolje razumevanje teorijskih koncepata koji stoje u osnovi programiranja. Pojedini ispitanici navode i da su sa iskustvom počeli više da cene teorijski deo gradiva, jer im on pomaže da lakše razumeju i implementiraju složenije programske koncepte.

Ovi rezultati ukazuju na to da učenje u tehničkim disciplinama predstavlja dinamičan proces, u kome se stilovi učenja prirodno razvijaju i prilagođavaju u skladu sa rastom znanja, iskustva i zahteva koje programiranje postavlja pred studente.

4.1 Problemi pri učenju programiranja

Učenje programiranja može biti zastrašujuće – nije lako, ali je daleko od nemogućeg. Kao i svaka nova veština, kodiranje zahteva strpljenje, doslednu vežbu i pravi način razmišljanja. Početni izazovi, poput nepoznate sintakse i teže otklanjanje grešaka, mogu delovati zastrašujuće, ali uz odlučnost i pristup kvalitetnim resursima, i ti izazovi mogu postati savladivi [7].

Učenje programiranja predstavlja složen proces koji zahteva istovremeno razumevanje apstraktnih koncepata, logičkog razmišljanja i praktične primene stečenog znanja. Tokom tog procesa studenti se često susreću sa različitim poteškoćama, koje mogu usporiti napredak i negativno uticati na motivaciju. Ovi problemi mogu proisteci iz nedostatka dobrih primera i kvalitetnog video sadržaja iz kojih može da se učim, na neke utiču i psihološke stvari poput koncentracije i strpljenja.

Identifikovanje najčešćih izazova u učenju programiranja predstavlja važan korak ka unapređenju nastavnih metoda i prilagođavanju načina učenja potrebama studenata.

5 Zaključak

U ovom radu analizirani su stilovi učenja i obrazovne navike studenata tehničkih usmerenja, sa posebnim fokusom na učenje programiranja. Na osnovu rezultata sprovedene ankete može se zaključiti da studenti u velikoj meri primenjuju kombinaciju različitih pristupa učenju, pri čemu se kao najzastupljeniji izdvajaju praktičan rad, korišćenje dokumentacije i video materijala. Ovakav multimodalni pristup pokazuje se kao najefikasniji način savladavanja složenih tehničkih koncepata.

Rezultati takođe ukazuju na značaj savremenih alata i resursa, poput online tutorijala, repozitorijuma koda i sistema zasnovanih na veštačkoj inteligenciji, koji postaju neizostavan deo procesa učenja. Istovremeno, uočeno je da studenti prepoznaju vrednost tradicionalnih metoda, ali ih sve češće kombinuju sa novim oblicima samostalnog i kolaborativnog rada.

Na osnovu dobijenih nalaza može se zaključiti da efikasno učenje programiranja zahteva fleksibilan pristup koji uvažava različite stilove učenja, individualne potrebe studenata i prirodu samog gradiva. Ovi rezultati mogu poslužiti kao osnova za unapređenje nastavnih metoda i prilagođavanje obrazovnih programa savremenim zahtevima tehničkog obrazovanja.

Literatura

- [1] Scientific American. The problem with "learning styles", 2018.
- [2] Deeksha Arya, Jin L.C. Guo, and Martin Robillard. How programmers find online learning resources. *Empirical Software Engineering*, 28, 12 2022.
- [3] David Baidoo-anu and Leticia Owusu Ansah. Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of chatgpt in promoting teaching and learning. *Journal of AI*, 7(1):52–62, 2023.
- [4] Neil D. Fleming. *VARK Strategies: The definitive guide to VARK*. Neil D. Fleming, 2019.
- [5] Scott Grissom, Myles F. McNally, and Tom Naps. Algorithm visualization in cs education: comparing levels of student engagement. In *Proceedings of the 2003 ACM Symposium on Software Visualization*, SoftVis '03, page 87–94, New York, NY, USA, 2003. Association for Computing Machinery.
- [6] Ruey-Shiang Shaw. The relationships among group size, participation, and performance of programming language learning supported with online forums. *Computers Education*, 62:196–207, 2013.
- [7] Springboard. Is coding hard to learn? a guide to programming in 2025, 2023.