

# Navigacija kroz nove programske jezike: izazovi, barijere i strategije savladavanja

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

Tamara Baranin 1029/2025, Luka Đekić 1021/2025,  
Petar Magenhajm 1116/2025, Nemanja Ršumović 1016/2025,  
mi251029@alas.matf.bg.ac.rs, mi251021@alas.matf.bg.ac.rs,  
mi251116@alas.matf.bg.ac.rs, mi251016@alas.matf.bg.ac.rs

25. decembar 2025.

## Sažetak

Ovaj rad analizira izazove sa kojima se suočavaju kako početnici, tako i iskusni programeri u procesu učenja novih programske jezike i paradigmi. Istraživanje je sprovedeno putem anketnog upitnika na uzorku od 95 ispitanika, pretežno studenata sa višegodišnjim iskustvom u programiranju, stečenim kroz akademsko obrazovanje, profesionalni rad i samostalno učenje. Fokus rada je na ispitivanju povezanosti između godina iskustva, poznavanja većeg broja programske jezike i paradigmi, kao i subjektivnih poteškoća pri savladavanju nove programske paradigme. Raznolikost ispitanika u pogledu iskustva, obrazovanja i pristupa učenju omogućava sagledavanje istraživanog problema iz više perspektiva, čime se dodatno povećava validnost donetih zaključaka.

## Sadržaj

<b>1 Uvod</b>	<b>2</b>
<b>2 Pregled osnovnih podataka o ispitanicima</b>	<b>2</b>
<b>3 Izazovi u učenju programske jezike</b>	<b>4</b>
3.1 Uticaj programske paradigme na proces učenja . . . . .	4
3.2 Uloga stila učenja i izvora znanja . . . . .	4
3.3 Završna razmatranja . . . . .	5
<b>4 Analiza ankete</b>	<b>5</b>
4.1 Strategije učenja i njihova efikasnost . . . . .	5
4.2 Barijere i izazovi u učenju novih programske jezike . . . . .	6
4.3 DODATI NAZIV . . . . .	7
<b>5 Informacije prikupljene pitanjima otvorenog karaktera</b>	<b>7</b>
5.1 Razumevanje paradigmi i algoritama je važnije od samog jezika . . . . .	8
5.2 Intenzivno pisanje koda kao osnovni mehanizam učenja . . . . .	8
5.3 Učenje kroz poznate zadatke u novom jeziku . . . . .	8
5.4 Ne postoji univerzalna strategija učenja . . . . .	8
5.5 Strpljenje, vreme i kontinuitet kao ključni faktori . . . . .	8
<b>6 Zaključak</b>	<b>8</b>
<b>Literatura</b>	<b>10</b>

## 1 Uvod

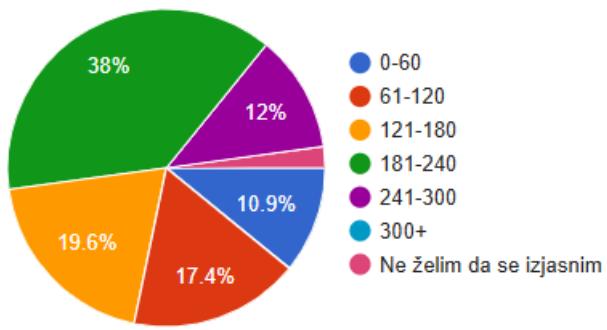
Učenje programskih jezika predstavlja složen i višeslojan proces koji se može značajno razlikovati od osobe do osobe, u zavisnosti od prethodnog znanja, iskustva i individualnih pristupa učenju. Brzina razumevanja, sposobnost dugoročnog pamćenja, kao i stepen sigurnosti u primeni novog programskog jezika zavise od niza faktora, među kojima se posebno izdvajaju višegodišnje iskustvo u programiranju, poznavanje različitih programskih paradigmi, kao i odabранe strategije učenja. Iako prethodno iskustvo i konceptualno razumevanje paradigmi u velikoj meri olakšavaju savladavanje novih jezika, značajan uticaj ima i način na koji se učenje odvija, bilo kroz kurseve, proučavanje zvanične dokumentacije ili rad na konkretnim projektima [4].

Ipak, izazovi sa kojima se programeri susreću prilikom učenja novih jezika nisu univerzalni, već su u velikoj meri individualni i zavise od više faktora. Upravo iz tog razloga, cilj ovog rada jeste da se, na osnovu većeg broja ispitanika, identifikuju i analiziraju obrasci i korelacije između uočenih prepreka i olakšavajućih faktora u procesu usvajanja različitih programskih jezika i paradigmi. Posebna pažnja posvećena je ispitanicima sa Matematičkog fakulteta, čiji obrazovni program karakteriše snažna teorijska osnova, što potencijalno olakšava apstraktno razumevanje novih koncepta i njihovo povezivanje sa postojećim znanjem. Istovremeno, uključivanje ispitanika sa drugih fakulteta omogućava poređenje različitih obrazovnih pristupa i njihovog uticaja na proces učenja programiranja.

Analiza rezultata ankete ukazuje na to da se kao jedan od ključnih izazova izdvaja učenje programskog jezika koji pripada novoj paradigmi, kao i kvalitet i dostupnost izvora za učenje. Ovi, ali i drugi identifikovani faktori, biće detaljnije ispitani u nastavku rada, sa ciljem da se pruži jasniji uvid u složenost procesa učenja programskih jezika i ponude smernice za efikasnije pristupe učenju u akademskom i samostalnom okruženju.

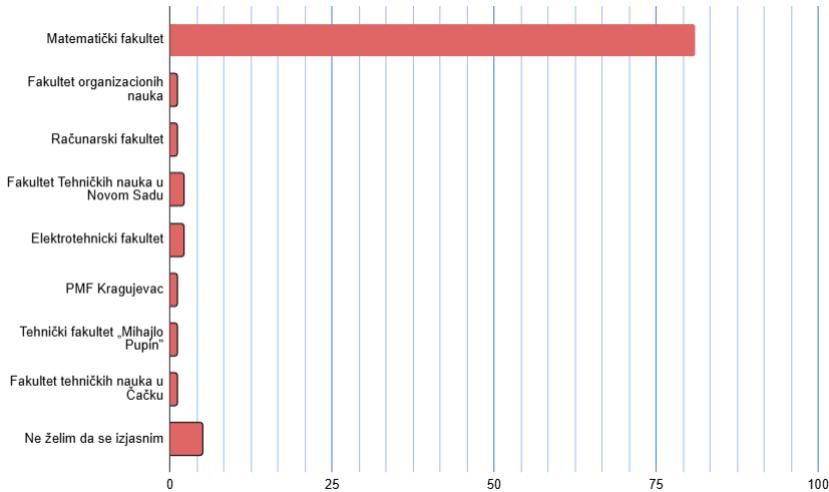
## 2 Pregled osnovnih podataka o ispitanicima

Većinu ispitanika čine studenti, dok je znatno manji broj onih koji su završili osnovne, master ili doktorske studije. Time se uzorak primarno oslanja na populaciju koja je aktivno uključena u proces učenja i akademskog usavršavanja, što je u skladu sa temom istraživanja. Među studentima postoji raznolikost u pogledu broja položenih ESPB bodova, što ukazuje na prisustvo ispitanika iz različitih faza studiranja, od početnih godina do završnih faza studija. Grafik odnosa broja ESBP poena među ispitanicima se može videti na slici 1.



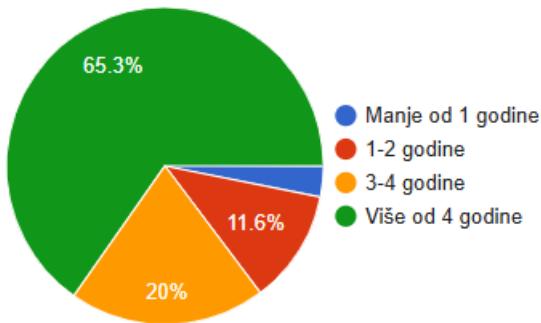
Slika 1: Položeni ESPB ispitanika koji su studenti

Analiza obrazovne strukture ispitanika pokazuje da najveći deo uzorka čine studenti Matematičkog fakulteta (slika 2), dok su studenti drugih tehničkih i srodnih fakulteta zastupljeni u manjem broju. Takva raspodela ukazuje da je istraživanje u najvećoj meri obuhvatilo populaciju koja je tokom studija kontinuirano u kontaktu sa programiranjem i srodnim računarskim disciplinama. Ovo doprinosi tome da prikupljeni podaci odražavaju iskustva ispitanika koji se aktivno bave učenjem programskih jezika, što je u skladu sa ciljevima istraživanja i temom seminarског rada.



Slika 2: Raspodele ispitanika po fakultetima

Kada je u pitanju iskustvo u programiranju, rezultati pokazuju da većina ispitanika ima višegodišnje iskustvo, pri čemu najveći deo navodi da se programiranjem bavi duže od četiri godine. Manji broj ispitanika ima između jedne i četiri godine iskustva, dok je najmanje onih koji se programiranjem bave kraće od jedne godine. Ovakva struktura uzorka omogućava analizu procesa učenja novih programskih jezika iz perspektive ispitanika koji već poseduju značajno prethodno znanje i iskustvo. Grafik odnosa godina iskustva među ispitanicima se može uočiti na slici 3.



Slika 3: Godine iskustva

### 3 Izazovi u učenju programskih jezika

Učenje programskih jezika predstavlja zahtevan proces koji podrazumeva razvoj sposobnosti analize problema, apstraktnog razmišljanja i razumevanja toka izvršavanja programa. Mnogi učenici, naročito početnici, pristupaju programiranju bez prethodnog planiranja i analize zadatka, fokusirajući se na pojedinačne linije koda umesto na celokupnu strukturu rešenja. Takav pristup često dovodi do nerazumevanja osnovnih programskih koncepta i otežava identifikaciju i ispravljanje grešaka, zbog čega se sintaksni problemi često javljaju kao posledica dubljih konceptualnih poteškoća[3].

#### 3.1 Uticaj programskih paradigmi na proces učenja

Poseban izazov u učenju novih programskih jezika predstavlja savladavanje jezika koji pripadaju drugačijoj programskoj paradigmiji. Promena paradigmе zahteva prilagodavanje postojećih i usvajanje novih načina razmišljanja o strukturi programa i rešavanju problema. Ovaj proces ne zavisi isključivo od tehničkog znanja ili iskustva, već od sposobnosti da se apstraktni koncepti povežu sa prethodnim znanjem. Zbog toga se problemi u učenju novih paradigmi javljaju i kod iskusnijih programera, a ne samo kod početnika.

#### 3.2 Uloga stila učenja i izvora znanja

Na uspešnost učenja programskih jezika značajno utiče i stil učenja, kao i kvalitet korišćenih izvora. Pasivni oblici učenja, poput oslanjanja isključivo na teorijska predavanja ili gotove primere, često nisu dovoljni za razvoj dubljeg razumevanja[1]. Suprotno tome, praktičan rad, rešavanje konkretnih problema i eksperimentisanje sa kodom omogućavaju postepeno građenje sigurnosti i povezivanje teorijskih znanja sa njihovom primenom. Kvalitetna dokumentacija, jasni primeri i dostupnost praktičnih zadataka dodatno olakšavaju proces usvajanja novih programskih koncepta.

### 3.3 Završna razmatranja

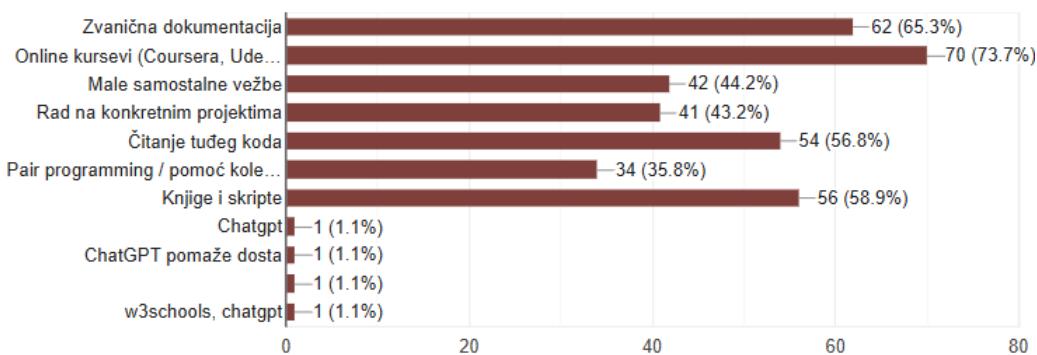
Iako se poteškoće u učenju programskih jezika mogu ispoljavati na različite načine, zajedničko im je to što ne proističu isključivo iz same složenosti jezika, već iz kombinacije kognitivnih zahteva, prethodnog znanja i načina učenja. Razumevanje ovih faktora omogućava sagledavanje učenja programiranja kao postepenog procesa u kome se tehničke veštine razvijaju uporedno sa sposobnošću apstraktног razmišljanja i rešavanja problema. Upravo iz tog razloga, identifikovanje obrazaca u izazovima i strategijama učenja, na osnovu većeg broja ispitanika, predstavlja važan korak ka unapređenju nastavnih pristupa i efikasnijem usvajanju novih programskih jezika i paradigmi [2].

## 4 Analiza ankete

U nastavku će biti prikazana analiza sprovedene ankete, identifikovani glavni problemi, kao i preporuke za rešavanje tih problema. Anketu je popunilo 95 ispitanika. Dobijeni podaci omogućavaju detaljniji uvid u navike ispitanika prilikom učenja novih programskih jezika, kao i u njihove subjektivne procene efikasnosti različitih pristupa. Analiza rezultata pruža osnov za razumevanje obrazaca ponašanja i uočavanje faktora koji mogu značajno uticati na uspešnost procesa učenja.

### 4.1 Strategije učenja i njihova efikasnost

Rezultati ankete ukazuju da ispitanici preferiraju aktivne strategije učenja, odnosno pristupe koji podrazumevaju neposrednu primenu znanja kroz kodiranje i rešavanje problema. Kada posmatramo koje strategije ispitanici najčešće koriste, dominiraju online resursi i samostalno istraživanje: online kursevi (70), zvanična dokumentacija (62), knjige i skripte (56) i čitanje tuđeg koda (54), dok su male samostalne vežbe (42) i rad na konkretnim projektima (41) takođe vrlo zastupljeni. Sve strategije koje su ispitanici naveli prikazane su na slici 4.



Slika 4: Strategije koje se koriste prilikom učenja novog jezika

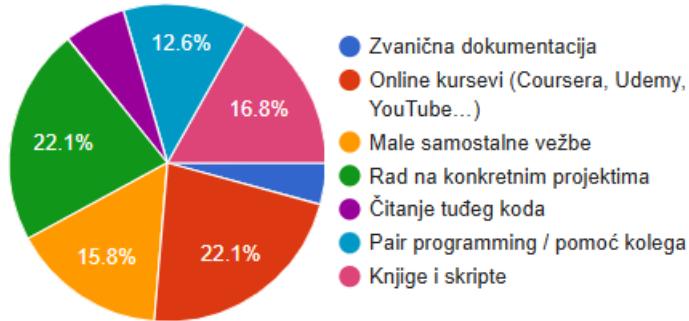
Međutim, kada se ispitanici opredeljuju za najefikasniju strategiju, slika postaje još jasnija: najviše glasova dele online kursevi (21) i rad na konkretnim projektima (21). To sugerira da pasivno ili informativno učenje

(npr. samo čitanje ili gledanje) najčešće dobija smisao tek kada se poveže sa praksom i konkretnim zadatkom.

Otvoreni odgovori dodatno potvrđuju ovaj obrazac i naglašavaju vrednost "učenja kroz rad" i iterativnog napredovanja kroz greske:

- „*Izrada konkretnog projekta*“
- „*Mali projekti, interaktivno učenje*“
- „*Rad na konkretnim primerima i projektima, uz stalno eksperimentisanje i greške iz kojih se uči.*“
- „*Bacanje u vatru i rad na konkretnom projektu više pomaže nego suvoporno učenje sintakse.*“

Ovi navodi su u skladu sa nalazom da ispitanici prepoznaju značaj iskustvenog učenja, gde se znanje proverava kroz implementaciju, što vremenom povećava sigurnost i sposobnost samostalnog rešavanja problema.



Slika 5: Najefikasnije strategije koje se koriste prilikom učenja novog jezika

Iako se u odgovorima jasno vidi sklonost ka učenju kroz praktičan rad i izradu projekata, prilikom procene najefikasnije strategije ne dolazi do jasne dominacije nijednog pristupa. Umesto toga, ispitanici su se opredeljivali za različite metode učenja, pri čemu su online kursevi, samostalne vežbe, rad na projektima i korišćenje literature zastupljeni u sličnom obimu. Ovakav nalaz potvrđuje da efikasnost strategije nije apsolutna kategorija, već zavisi od faze učenja, tipa programskog jezika i individualnog načina usvajanja znanja, zbog čega se različite strategije mogu smatrati podjednako korisnim u različitim okolnostima što je prikazano na slici 5.

## 4.2 Barijere i izazovi u učenju novih programskih jezika

Da bi se barijere sagledale preciznije, analizirani su različiti aspekti prilagođavanja novom jeziku (sintaksa, paradigma, tipovi, alati/okruženje, ekosistem, dokumentacija). Kada se za svakog ispitanika identificiše aspekt koji je dobio najveću ocenu težine, dobija se vrlo informativna slika dominantnih prepreka.

Najčešće isticana „najveća barijera“ je:

- Promena programske paradigmе
- Alati i razvojno okruženje

Dok su dokumentacija, sintaksa, ekosistem i tipovi navedeni kao najmanja prepreka.

Ovaj nalaz je važan jer sugeriše da poteškoće nisu samo „jezičke” (sintaksa), već da često nastaju u delu koji zahteva promenu načina razmišljanja (paradigma) ili praktično snalaženje u novom okruženju. To objašnjava zašto prelazak na novi jezik ponekad deluje „teže nego što se očekuje”, čak i kada je sintaksa relativno jednostavna.

Tabela 1: Faktori koji predstavljaju prepreku

	1	2	3	4	5
Razumevanje sintakse	34(35.8%)	38(40%)	17(19.9%)	5(5.3%)	1(1.1%)
Promena programske paradigmе	14(14.7%)	22(23.2%)	31(32.6%)	26(27.4%)	2(2.1%)
Razumevanje sistema tipova	31(32.6%)	26(27.4%)	26(27.4%)	11(11.6%)	1(1.1%)
Alati i razvojno okruženje	15(15.8%)	25(26.3%)	29(30.5%)	18(18.9%)	8(8.4%)
Biblioteke i ekosistem	8(8.4%)	31(32.6%)	44(46.3%)	8(8.4%)	4(4.2%)
Kvalitet ili dostupnost dokumentacije	20(21.1%)	23(24.2%)	27(28.4%)	21(22.1%)	4(4.2%)

Na osnovu prikazanih rezultata u tabeli 1 dat je pregled faktora koji mogu predstavljati prepreku u učenju novih programskih jezika, pri čemu su ispitanici ocenili stepen težine svakog od navedenih aspekata na skali od 1 do 5 (1 predstavlja „nije predstavljalo problem”, a 5 „veoma je predstavljalo problem”).

### 4.3 DODATI NAZIV

DODATI TEKST

## 5 Informacije prikupljene pitanjima otvorenog karaktera

Kako bi se dobio dublji uvid u individualna iskustva, stavove i strategije ispitanika koje nije moguće u potpunosti obuhvatiti zatvorenim pitanjima, u okviru ankete prikupljene su i informacije putem pitanja otvorenog karaktera, čija je analiza predstavljena u nastavku.

## **5.1 Razumevanje paradigmi i algoritama je važnije od samog jezika**

Jedan deo ispitanika smatra da konkretan programski jezik nije presudan, već da je ključno razumevanje algoritama i paradigmi, jer se na toj osnovi relativno lako prelazi na druge jezike. Ovakav stav ukazuje na viši nivo apstrakcije u razmišljanju o programiranju, gde se jezici posmatraju kao alati, a ne kao centralni predmet učenja. Ovo je tipično za iskusnije programere, kod kojih se znanje ne vezuje za sintaksu, već za koncepte.

## **5.2 Intenzivno pisanje koda kao osnovni mehanizam učenja**

Veliki broj ispitanika naglašava da je kontinuirano pisanje koda, često ponavljano u različitim formulacijama, najefikasniji način učenja novog jezika. Ovo sugerire da ispitanici doživljavaju učenje programiranja kao proces koji se dominantno odvija kroz praksu i ponavljanje, pri čemu se greške posmatraju kao sastavni deo učenja, a ne kao neuspeh.

## **5.3 Učenje kroz poznate zadatke u novom jeziku**

Pojedini ispitanici savetuju da se novi jezik uči kroz rešavanje problema koji su već ranije rađeni u drugim jezicima. Ovakav pristup smanjuje opterećenje pri učenju jer je problem već poznat, pa se pažnja može usmjeriti na razlike u sintaksi i specifičnosti njegove paradigmе.

Neki ispitanici upozoravaju da ne treba mehanički preslikavati rešenja iz drugih jezika, već učiti kako se problemi rešavaju na način koji je prirodan za dati jezik. Ovaj komentar ukazuje na svest o tome da svaki jezik ima sopstvenu filozofiju i namenu, te da površno poznavanje sintakse nije dovoljno za efikasno korišćenje jezika u praksi.

## **5.4 Ne postoji univerzalna strategija učenja**

Pojedini odgovori eksplisitno ističu da ne postoji jedinstvena strategija koja odgovara svima i da svako mora pronaći pristup koji mu lično najviše odgovara. Ovaj stav potvrđuje da je efikasnost uslovljena individualnim faktorima i prethodnim iskustvom.

## **5.5 Strpljenje, vreme i kontinuitet kao ključni faktori**

U više odgovora naglašava se potreba da se učenju pristupi strpljivo, bez preskakanja osnova, uz redovan i dugoročan rad. Ovi komentari ukazuju na shvatanje učenja programiranja kao sporog, ali stabilnog procesa, u kome se dobre navike stečene na početku dugoročno isplate, naročito pri učenju složenijih jezika i paradigm.

# **6 Zaključak**

Učenje novih programskih jezika predstavlja kontinuiran i višeslojan proces koji se ne završava usvajanjem sintakse, već podrazumeva razvoj načina razmišljanja, prilagođavanje različitim paradigmama i postepeno sticanje sigurnosti u praktičnoj primeni znanja. U savremenom okruženju,

u kome se tehnologije brzo menjaju, sposobnost da se efikasno savlada novi programski jezik postaje jednako važna kao i poznavanje samog jezika.

Rezultati ankete jasno ukazuju da ovaj proces predstavlja složen kognitivni i praktični izazov, ali da se uz odgovaraajuće strategije i motivaciju može značajno olakšati. Istraživanje potvrđuje da iskustvo igra ključnu ulogu, ali i da početnici mogu ostvariti napredak ukoliko primene aktivne metode učenja i imaju jasne ciljeve. Najveće poteškoće u učenju novih programskih jezika često se ne vezuju za tehničke detalje, već za promenu ustaljenih obrazaca razmišljanja i prilagođavanje drugačijim konceptima i alatima.

Takođe, izbor strategije učenja ima značajan uticaj na uspešnost usvajanja novih znanja. Aktivni pristupi, koji uključuju praktičan rad, eksperimentisanje i primenu naučenog na realne ili poznate zadatke, omogućavaju dublje razumevanje i dugoročnije pamćenje gradiva. Istovremeno, važno je naglasiti da ne postoji jedinstvena strategija koja odgovara svima, već da se efikasno učenje oslanja na prilagođavanje individualnim potrebama i iskustvu.

Na kraju, može se zaključiti da uspešno učenje programskih jezika zahteva strpljenje, kontinuitet i spremnost na suočavanje sa greškama kao sastavnim delom procesa. Razumevanje da su koncepti i paradigme važniji od same sintakse ne doprinosi samo efikasnijem učenju pojedinačnih jezika, već i dugoročnom razvoju veština neophodnih u savremenom softverskom okruženju.

## Literatura

- [1] Scott Freeman, Sarah L. Eddy, Miles McDonough, Michelle K. Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23):8410–8415, 2014. Available at: <https://www.pnas.org/doi/pdf/10.1073/pnas.1319030111>.
- [2] Rozita Kadar, Naemah Abdul Wahab, Jamal Othman, Maisurah Shamsuddin, and Siti Balqis Mahlan. A study of difficulties in teaching and learning programming: A systematic literature review. *International Journal of Academic Research in Progressive Education and Development*, 10(3):591–605, 2021.
- [3] Essi Lahtinen, Kirsti Ala-Mutka, and Hannu-Matti Järvinen. A study of the difficulties of novice programmers. *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, pages 14–18, 2005. Available at: <https://dl.acm.org/doi/10.1145/1067445.1067453>.
- [4] Anthony Robins, Janet Rountree, and Nathan Rountree. Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2):137–172, 2003. Available at: [https://www.researchgate.net/publication/220372478\\_Learning\\_and\\_Teaching\\_Programming\\_A\\_Review\\_and\\_Discussion](https://www.researchgate.net/publication/220372478_Learning_and_Teaching_Programming_A_Review_and_Discussion).