

Poteškoće u razumijevanju osnovnih programskih koncepata na prijediplomskoj razini studija

Sikavica, Antonela

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, Faculty of Science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:166:950483>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-17**

Repository / Repozitorij:

[Repository of Faculty of Science](#)



SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

**POTEŠKOĆE U RAZUMIJEVANJU
OSNOVNIH PROGRAMSKIH KONCEPATA
NA PRIJEDIPLOMSKOJ RAZINI STUDIJA**

Antonela Sikavica

Split, travanj 2024.

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

**POTEŠKOĆE U RAZUMIJEVANJU
OSNOVNIH PROGRAMSKIH KONCEPATA
NA PRIJEDIPLOMSKOJ RAZINI STUDIJA**

Antonela Sikavica

Mentorica: doc. dr. sc. Monika Mladenović

Split, travanj 2024.

Temeljna dokumentacijska kartica

Diplomski rad

Sveučilište u Splitu
Prirodoslovno-matematički fakultet
Odjel za informatiku
Ruđera Boškovića 33, 21000 Split, Hrvatska

POTEŠKOĆE U RAZUMIJEVANJU OSNOVNIH PROGRAMSKIH KONCEPATA

Antonela Sikavica

SAŽETAK

Ovaj rad bavi se poteškoćama u osnovnim konceptima programiranja. Mnogi se suočavaju sa izazovom miskoncepcija, što često proizlazi iz nedovoljnog razumijevanja koncepata. Miskoncepcije mogu biti vezane uz različite aspekte programiranja, što rezultira nepotpunim ili neispravnim kodom. U ovome radu korišten je test, kojeg su ispunjavali studenti koji su pohađali kolegij OOP, kao i studenti koji se bave drugim oblicima programiranja. Istraživanje je provedeno u ožujku 2023. godine u učionicama Prirodoslovno-matematičkog fakulteta u Splitu, u kojem je sudjelovalo 223 studenata već spomenutog fakulteta. Analizom testa su identificirane miskoncepcije putem varijabli, grananja, petlji, nizova, funkcija i metoda. Ovi rezultati pružaju konkretnije uvide koji mogu pomoći pri unapređenju nastave programiranja, što je ključno za bolje razumijevanje osnovnih koncepata programiranja i pripremu studenata za izazove u stvarnom svijetu programiranja.

Ključne riječi: miskoncepcije, test, studenti, ključni koncepti

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad sadrži: 56 stranica, 17 grafičkih prikaza, 23 tablice i 21 literaturnih navoda. Izvornik je na hrvatskom jeziku.

Mentor: **doc. dr. sc. Monika Mladenović**, docent Prirodoslovno - matematičkog fakulteta, Sveučilišta u Splitu

Ocjenjivači: **doc. dr. sc. Monika Mladenović**, docent Prirodoslovno - matematičkog fakulteta, Sveučilišta u Splitu

doc. dr. sc. Divna Krpan, docent Prirodoslovno – matematičkog fakulteta, Sveučilišta u Splitu

doc. dr. sc. Goran Zaharija, docent Prirodoslovno - matematičkog fakulteta, Sveučilišta u Splitu

Rad prihvaćen: **travanj 2024.**

Basic documentation card

Thesis

University of Split
Faculty of Science
Department of Computer Science
Ruđera Boškovića 33, 21000 Split, Croatia

DIFFICULTIES IN UNDERSTANDING BASIC PROGRAMMING CONCEPTS

Antonela Sikavica

ABSTRACT

This paper addresses the difficulties in understanding fundamental programming concepts. Many encounter the challenge of misconceptions, which often stem from insufficient understanding of various concepts. Misconceptions can be related to different aspects of programming, resulting in incomplete or faulty code. In this study, a test was used, filled out by students who attended the OOP course, as well as students engaged in other forms of programming. The research was conducted in March 2023 in classrooms at the Faculty of Science, University of Split, with the participation of 223 students from the aforementioned faculty. Various misconceptions were identified through the analysis of the test, including variables, branching, loops, arrays, functions, and methods. These results provide more specific insights that can aid in improving programming education, which is crucial for better understanding of basic programming concepts and preparing students for challenges in the real-world programming.

Key words: misconceptions, test, students, key concepts

Thesis deposited in library of Faculty of science, University of Split

Thesis consists of: 56 pages, 17 figures, 23 tables and 21 references Original language: Croatian

Mentor: **Monika Mladenović, Ph.D.** Assistant Professor of Faculty of Science, University of Split

Reviewers: **Monika Mladenović, Ph.D.** Assistant Professor of Faculty of Science, University of Split
Divna Krpan, Ph.D. Assistant Professor of Faculty of Science, University of Split
Goran Zaharija, Ph.D. Assistant Professor of Faculty of Science, University of Split

Thesis accepted: **April 2024.**

IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam diplomski rad s naslovom POTEŠKOĆE U RAZUMIJEVANJU OSNOVNIH PROGRAMSKIH KONCEPATA NA PRIJEDIPLOMSKOJ RAZINI STUDIJA izradila samostalno pod voditeljstvom doc. dr. sc. Monike Mladenović. U radu sam primijenila metodologiju znanstvenoistraživačkog rada i koristila literaturu koja je navedena na kraju diplomskog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući navela u završnom radu na uobičajen, standardan način citirala sam i povezala s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika.

Studentica

Antonela Sikavica

Sadržaj

Uvod.....	1
1. Pregled područja	2
1.1. Programiranje.....	2
1.2. Otkrivanje miskonceptija u vezi s konceptima programiranja	2
2. Metodologija istraživanja	7
2.1. Predmet i cilj rada.....	7
2.2. Instrumenti	7
2.3. Sudionici	11
3. Rezultati.....	13
3.1. Razlika studijskih grupa u uspjehu na testu iz programiranja	13
3.1.1. Grupe OOP i PP – usporedba uspješnosti studenata na testu iz programiranja u odnosu na studijsku grupu.....	13
3.1.2. Grupa P2 – usporedba uspješnosti studenata na testu iz programiranja u odnosu na studijsku grupu	14
3.2. Razlika između studenata i studentica u uspjehu na testu iz programiranja	15
3.2.1. Grupe OOP i PP – usporedba uspješnosti studenata na testu iz programiranja u odnosu na spol.....	15
3.2.2. Grupa P2 – usporedba uspješnosti studenata na testu iz programiranja u odnosu na spol.....	15
3.3. Razlika između studenata koji ponavljaju i onih koji ne ponavljaju kolegij	16
3.3.1. Grupe OOP i PP – usporedba uspješnosti studenata na testu iz programiranja u odnosu na to jesu li studenti ponavljači ili ne	16
3.3.2. Grupa P2 – usporedba uspješnosti studenata na testu iz programiranja u odnosu na to jesu li studenti ponavljači ili ne	16
3.4. Analiza otkrivenih miskonceptija	17
4. Ograničenja istraživanja	41

5. Zaključak	42
Literatura	44
Popis slika.....	46
Popis tablica.....	47
Prilozi	49

Uvod

U digitalnom dobu, gdje tehnologija oblikuje svaki aspekt našeg društva, od obrazovanja do poslovanja, vještina programiranja postala je neizostavna [13]. Programiranje više nije rezervirano samo za odabrane, nego se smatra temeljnom pismenošću neophodnom za svakoga tko želi biti kompetentan član suvremenog tehnološki orijentiranog društva. Što je uopće programiranje, da je toliko važno? Programiranjem se računala čine korisnima [13], a autor A. Perlis je rekao da bi svi trebali učiti programirati [15]. Međutim, putovanje kroz svijet programiranja često je obilježeno zabudama i miskoncepcijama koje mogu znatno otežati ili čak onemogućiti razumijevanje i primjenu osnovnih programskih koncepata [14][16][6][10][3].

Miskonceptije nisu samo prepreke u akademskom napretku; one su i barijere na putu razvoja kritičkog mišljenja i rješavanja problema, ključnih vještina u svakom profesionalnom okruženju. Često se pojavljuju tijekom učenja programiranja, posebno među početnicima, te mogu izazvati zablude u vezi s konceptima kao što su algoritamsko razmišljanje, struktura podataka, funkcionalnost petlji i uvjetnog izvođenja [11][17]. Može se reći da se miskonceptije javljaju kada se razviju netočni mentalni modeli, a mentalni modeli o konceptima programiranja se razvijaju tijekom procesa učenja. Različiti čimbenici mogu utjecati na razvoj ili sprječavanje miskonceptija, a to su dob, programski jezik i programski kontekst [20].

Ovaj rad istražuje identifikaciju i analizu miskonceptija koje imaju studenti Prirodoslovno-matematičkog fakulteta u Splitu o osnovnim programskim konceptima. Kroz dubinsku analizu i istraživanje, cilj nam je ne samo prepoznati miskonceptije, već i razumjeti njihove uzroke, kao i kako prethodno znanje utječe na njihovu pojavu. Na temelju prikupljenih podataka, predlažemo strategije i pristupe za njihovo ispravljanje, s ciljem poboljšanja pedagoških metoda i obrazovnih ishoda u području informatike.

Ovom analizom nastojimo osvijetliti put prema razvoju efikasnijih obrazovnih strategija koje ne samo da će se nositi s postojećim miskoncepcijama, već će i poticati dublje i trajnije razumijevanje programiranja. To će omogućiti studentima da postanu sposobni i samouvjereni programeri, spremni za izazove suvremenog tehnološkog okruženja.

1. Pregled područja

1.1. Programiranje

Kao što je rečeno, programiranje je vještina koja je postala neizostavna te se programiranjem računala čine korisnima [13]. Podučavanje osnovnih koncepata programiranja studentima je izazovno, ali ključno ne samo zbog ocjene iz kolegija nego i zbog razvoja računalnog razmišljanja. Računalno razmišljanje se opisuje kao vještina za rješavanje problema [9].

1.2. Otkrivanje miskonceptija u vezi s konceptima programiranja

Programiranje je teško za učenje, razumijevanje, svladavanje i poučavanje, a osobito je teško za ljude koji se tek upuštaju u svijet programiranja [1][2][5] – programiranje je zahtjevno [4] i na razini koja nije toliko zahtjevna [8].

Mnogi imaju probleme u ranim fazama učenja pa je stopa neuspjeha i odustajanja visoka [1]. Postoje koncepti koji su problematični od osamdesetih godina dvadesetog stoljeća do danas te se iz toga može vidjeti da je programiranje zahtjevno i za poučavanje [7].

Ovo istraživanje usmjereno je na dublje razumijevanje miskonceptija koje se javljaju kod studenata Prirodoslovno-matematičkog fakulteta u Splitu u vezi s osnovnim konceptima programiranja.

Nitko nije tabula rasa (prazna ploča) pa tako ni učenici. Učenici u škole dolaze s predznanjem o nečemu, a to može i ne mora biti usklađeno sa znanstvenim spoznajama. Predznanje učenika je predkonceptija, a ona nastaje osobnim iskustvom ili prethodnim učenjem. Predkonceptije koje nisu pogrešne omogućuju nadogradnju znanja, a prekonceptije koje su pogrešne dovode do razvoja miskonceptija (koncepti koji nisu usklađeni sa znanstvenim spoznajama). Miskonceptije nisu ništa novo te se može reći da su miskonceptije dio procesa koji se zove učenje. U svijetu je poznata poslovice „Na

greškama se uči!“ te se ta poslovice može primijeniti i na miskoncepcije jer iza učenikovih pogrešaka je prilika za ispraviti tu pogrešku i prilika za naučiti nešto novo.

U digitalnom dobu, gdje tehnološki napredak oblikuje svaki aspekt našeg društva, vještina programiranja postaje sve važnija. No, unatoč tomu, istraživanja su pokazala da studenti posjeduju miskoncepcije o programskim konceptima [14][19][12][20][21], što značajno otežava njihovo učenje i razvoj kao programera. Napravljena su mnoga istraživanja koja su doprinijela boljem razumijevanju poteškoća u uspješnom svladavanju vještine programiranja. Istraživanja su rezultirala činjenicama da ljudi (učenici) nemaju ista predznanja te da imaju pogrešna shvaćanja tijekom usvajanja koncepata. Od velike je važnosti da ljudi koji poučavaju programiranje znaju miskoncepcije jer njihovo pravovremeno prepoznavanje, ispravljanje te na kraju uklanjanje dovodi do manjeg utjecaja u procesu stjecanja znanja te poboljšanja učinkovitosti učenja, poučavanja i konceptualnog razumijevanja.

Autori su objavili članak naslova *Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level* [14] te u njemu postavljaju pitanje o miskoncepcijama na razini osnovne škole. Također, zanima ih može li se drugačijim pedagoškim pristupom, vizualnim programskim jezikom i pomicanjem programskog konteksta prema programiranju igrice „spriječiti“ miskoncepcije te su na ta pitanja pokušali odgovoriti u radu. Njihovo istraživanje se temeljilo na miskoncepcijama o petljama te je istraživanje provedeno u predmetnoj nastavi među 207 učenika osnovnih škola. Učenici (sudionici ovog istraživanja) su učili programirati u programskim jezicima: *Scratch*, *Logo* i *Python*. Rezultati istraživanja ukazala su na to da učenici s prosječnim i slabijim sposobnostima postižu bolje rezultate kada programiraju u *Scratch-u* nego kada programiraju u *Logo-u* ili *Python-u*. Također, analizom rezultata utvrđeno je da su pogrešna shvaćanja učenika o petljama minimalna kada koriste *Scratch* u usporedbi s tekstualnim programskim jezicima – *Logo* i *Python*. Što zadaci postaju složeniji (poput korištenja ugniježđenih petlji) razlike postaju očitije. Kada se usporede odgovori učenika iz *Loga* i *Pythona*, učenici su podjednako uspješni, osim kada zadatak uključuje ugniježdene petlje. Kada je riječ o ugniježđenim petljama, učenici koji koriste *Python* većinom ih nisu prepoznali, već su prepoznali dvije neovisne petlje. Ova poteškoća se odnosi na sintaksu *Pythona* gdje su sve naredbe unutar ugniježdene *for* petlje dvostruko uvučene, što je očito zbunjujuće za učenike. Rezultati ovog istraživanja potvrđuju opravdanost korištenja

vizualnog programskog jezika *Scratch* za poučavanje početnika u programiranju kroz programiranje igara (problemi sa sintaksom su eliminirani tako da se učenici mogu usredotočiti na algoritme).

Misconceptions about variables at the K-12 level [21] je istraživanju u kojem su autori proveli Pilot istraživanje među 127 učenika 5. i 6. razreda osnovne škole koje se odnosilo na miskoncepcije o varijablama. Učenici koji su sudjelovali u ovome istraživanju su učili programirati u tekstualnim programskim jezicima *Logo* i *Python* s grafičkim (*Logo*) i proceduralnim (*Python*) pristupima. Što se tiče pogrešnih shvaćanja (miskoncepcija), statistički značajne razlike između rezultata u *Logo-u* i *Python-u* pojavile su se pri ponovnoj dodjeli ($a = a + 1$). Ovo pogrešno shvaćanje vjerojatno proizlazi iz matematičkih pretpostavki gdje izraz izgleda kao nerješiva jednačba pa učenici misle da varijabla sadrži izraz $(a+1)$ kao *string*.

Isti autori su objavili dva članka. Naslov prvog članka je *Analysis of school students' misconceptions about basic programming concepts* [19], a naslov drugog članka je *Mediated transfer: impact on programming misconceptions* [20]. U prvom članku autorima je cilj bio utvrditi miskoncepcije u programiranju za početnike u programiranju na K-12 razini koristeći Python kao tekstualni programski jezik. Devedeset osam učenika petih razreda triju osnovnih škola je riješilo dva testa te su autori analizirali prikupljene podatke iz ta dva testa. Prvi test je bio nakon predavanja koje se odnosilo na programiranje, a drugi test je bio pet mjeseci kasnije. Autori su ovim istraživanjem otkrili sedam miskoncepcija, a to su:

- 1) Dodjeljivanje izraza umjesto izračunate vrijednosti,
- 2) Prilikom ponovnog dodjeljivanja vrijednosti varijable, dodjeljivanje vrijednosti zbroja varijabli,
- 3) Korištenje simboličkog naziva varijable umjesto njezine vrijednosti,
- 4) Korištene prve ili prethodno dodijeljene vrijednosti varijable,
- 5) Pogrešna predodžba tipa podataka,
- 6) Pogrešna predodžba sekvenciranja,
- 7) Zamjena vrijednosti varijabla.

U drugom članku autori su željeli otkriti mogu li se koristiti programski jezici bazirani na blokovima za posredovani prijenos kako bi se smanjile miskonceptije u tekstualnim programskim jezicima. Istraživanje je provedeno među četrdeset devet učenika šestih razreda osnovne škole tijekom nastave Informatike. Tijekom istraživanja, učenici su učili programiranje koristeći *Python*, a dva mjeseca kasnije koristili su MakeCode za micro:bit. Autori su ostvarili posredovani prijenos koristeći iste zadatke u oba okruženja. Tijekom predavanja o micro:bitu, nastavnik je pokazao ekvivalentna rješenja u *Pythonu* i analizirao ih, ali učenici nisu programirali u *Pythonu*. Analizirajući rezultate moglo se zaključiti da su rezultati bolji u *Pythonu* nakon korištenja BPL-a za posredovani prijenos programskih koncepata prema TBPL-u, tj. da BPL može smanjiti pojavu miskonceptija u TBPL-u (poboljšanje je najznačajnije bilo za zadatke petlji). U ovome radu uočeno je dvanaest miskonceptija, a to su:

- 1) Dodjela izraza umjesto izračunate vrijednosti,
- 2) Prilikom ponovnog dodjeljivanja vrijednosti varijabli, dodjeljivanje zbroja vrijednosti varijabli,
- 3) Korištenje prve dodijeljene vrijednosti varijabli,
- 4) Korištenje simboličkog imena varijable umjesto njezine vrijednosti,
- 5) Pogrešno shvaćanje tipa podataka,
- 6) Ispisivanje vrijednosti pogrešnog uvjeta kada nedostaje *else* blok,
- 7) Pogrešno tumačenje graničnih uvjeta,
- 8) Uključivanje vrijednosti <end> unutar petlje,
- 9) Početna vrijednost petlje je 1,
- 10) Zanemarena petlja *for*,
- 11) Prirodnojezično ime varijable utječe na vrijednost varijable unutar petlje,
- 12) Zanemarivanje zbrajanja unutar petlje.

Autori koji su napisali članak *Learning to use functions: The relationship between misconceptions and self-efficacy* [12] su imali dva cilja. Prvi cilj je bio istražiti miskonceptije o funkcijama, a drugi cilj je bio istražiti utjecaj miskonceptija na samoučinkovitost učenika. Ova studija je otkrila ukupno sedam miskonceptija, a to su:

- 1) Studenti misle da bi naredbe za povrat trebale biti korištene nakon poziva funkcije kako bi se vratio i ispisao izlaz funkcije,
- 2) Studenti smatraju da bi povratna vrijednost trebala imati isti naziv kao i funkcija,
- 3) Studenti smatraju da bi parametri trebali biti eksplicitno definirani kako bi se koristili unutar funkcije,
- 4) Studenti smatraju da bi parametri i argumenti trebali imati isto ime,
- 5) Studenti misle da bi argumenti trebali uključivati povratnu varijablu,
- 6) Studenti misle da se lokalnim varijablama može pristupiti iz jedne funkcije u drugu,
- 7) Studenti misle da se pozivom funkcije ispisuje rezultat.

U ovome istraživanju rezultati su pokazali da prethodno iskustvo u programiranju utječe na samopouzdanje, a samopouzdanje utječe na uspjeh studenata te da studenti koji nemaju pogrešne predodžbe podcjenjuju svoje rezultate, dok studenti s pogrešnim predodžbama precjenjuju svoje rezultate. To dovodi do zaključka da se studenti s pogrešnim predodžbama čvrsto drže krivih predodžbi i stoga ne mogu točno procijeniti svoje rezultate.

Nadalje, iako su prethodna istraživanja već dotakla ovo područje, postoje potrebe za daljnjim istraživanjem kako bi se detaljnije istražile miskoncepcije vezane uz osnovne koncepte programiranja, kao i proširila populacija ispitanika na širi skup studenata. Ovo istraživanje nastoji upravo ispuniti tu prazninu, identificirajući, analizirajući i predlažući strategije za rješavanje miskoncepcija kod studenata.

2. Metodologija istraživanja

Metodologija razvoja ovog istraživačkog rada osigurava sveobuhvatan pristup prikupljanju, analizi i interpretaciji podataka kako bi se postigli postavljeni ciljevi istraživanja. Kvantitativna metoda omogućuje detaljno istraživanje miskoncepcija, pružajući temelj za razvoj adekvatnih obrazovnih strategija i poboljšanje procesa učenja programiranja.

2.1. Predmet i cilj rada

Predmet ovog istraživačkog rada je proučavanje miskoncepcija među studentima Prirodoslovno-matematičkog fakulteta u Splitu (istraživanje je provedeno u ožujku 2023. godine). Glavni cilj istraživanja je identificirati miskoncepcije koje se pojavljuju kod studenata. Također, u ovome istraživanju su se nametnula četiri istraživačka pitanja (uključujući pitanje koje se odnosi na miskoncepcije) koja su se kasnije analizirala.

Istraživačka pitanja su:

1. Postoji li razlika u uspješnosti studenata na testu iz programiranja u odnosu na studijsku grupu?
2. Postoji li razlika u uspješnosti studenata na testu iz programiranja u odnosu na spol?
3. Postoji li razlika u uspješnosti studenata na testu iz programiranja u odnosu na to je li student ponavlja kolegij ili ne?
4. Postoje li miskoncepcije u razumijevanju osnovnih programskih koncepata među studentima?

2.2. Instrumenti

Za prikupljanje podataka korištena su dva testa koja su napravljena u *Microsoft Forms* obliku. Prvi test se sastojao od 16 zadataka pisanih u *JavaScript* programskom jeziku, a ispunjavali su ga studenti prijediplomskih studija PMF-a na početku drugog semestra u sklopu kolegija Objektno-orijentirano programiranje (OOP) i studenti diplomskog studija

PMF-a u sklopu kolegija Programske paradigme (PP). Drugi test se sastojao od 13 zadataka pisanih u *Python* programskom jeziku, a ispunjavali su ga studenti prve godine prijediplomskih studija PMF-a na početku drugog semestra u sklopu kolegija Programiranje 2 (P2). Studenti kolegija P2 se do tada se nisu susreli s kolegijem OOP jer kolegij P2 prethodi kolegiju OOP.

Testovi su sadržavali zadatke koji su bili usmjereni na utvrđivanje miskoncepcija.

Studentima kolegija P2 zadaci su prilagođeni tako da su se zadaci iz programskog jezika *JavaScript* napisali u programski jezik *Python* (ovi studenti se još nisu susreli s *JavaScript* programskim jezikom, dok s programskim jezikom *Python* jesu) te u odnosu na studente OOP-a i PP-a kod njih je bilo tri zadatka manje (tj. oni su imali trinaest zadataka) jer su tri zadatka (koja su vezana isključivo za OOP) izbačena. Tablica 1. prikazuje primjer istog zadatka u dva testa.

Tablica 1. *JavaScript vs. Python*

<pre> var a = 1; var b = 2; for (var i = 0; i<2; i++) { console.log(b); for (var j = 1; j<=2; j++) { console.log(a); } } </pre>	<pre> a = 1 b = 2 for i in range(2): print(b) for j in range(1, 3): print(a) </pre>
---	---

Odgovor na zadatke nije bio obavezan kako bi se smanjila vjerojatnost studentovog točnog odgovora kojim je došao pogađanjem (engl. *by guessing*) – u protivnom svaki student za svaki zadatak bi imao vjerojatnost od 20% da će ponuditi točan odgovor jer je jedan točan odgovor, a pet ponuđenih. Također, studentima je dano na raspolaganje petnaest minuta za rješavanja zadataka. Test se sastojao od zadataka koji su grupirani tako da se par zadataka odnosi na procjenu jedne moguće vrste miskoncepcije (npr. test za studente koji trenutno slušaju OOP i za studente PP-a je sadržavao: četiri zadatka kojima su se procjenjivale miskoncepcije vezane za varijable, četiri zadatka kojima su se procjenjivale miskoncepcije vezane za grananja, četiri zadatka kojima su se procjenjivale miskoncepcije vezane za petlje, dva zadatka kojima su se procjenjivale miskoncepcije vezane za nizove, jedan zadatak kojim su se procjenjivale miskoncepcije vezane za funkcije te jedan zadatak kojim su se procjenjivale miskoncepcije vezane za klase). U nekim zadacima ponuđen je odgovor *Other* koji student odabire ako smatra da ispravan odgovor nije ponuđen te mora na predviđeno mjesto upisati koji odgovor smatra ispravnim. Odgovor *Other* je u nekim

zadacima, ponajviše, ponuđen zbog mogućnosti otkrivanja novih miskoncepcija (npr. ako odgovor *Other* nije ispravan, a student ga odgovori te upiše odgovor za koji smatra da je ispravan, onda taj odgovor može ukazati na neku novu miskoncepciju vezanu za promatrani koncept).

Za obradu podataka korišten je programski paket SPSS (verzija 20). Podaci koji su korišteni su o: smjeru, semestru, spolu, ponovnom upisu kolegija, studentovim odgovorima na sva pitanja, bodovima za svako pitanje, dodatnim bodovima za pitanja gdje je točan odgovor *Other* te postotak koji označava koliko je svaki student ukupno točno riješio zadatke (nezavisne varijable su se odnosile na: spol, smjer i ponavljanje kolegija, a zavisne varijable su odgovori na zadatke te ukupni broj bodova). Varijable koje u svom imenu sadrže B služile su za broj bodova za svaki pojedini zadatak, a varijable koje u svom imenu sadrže DB služile su za dodatne bodove za zadatke u kojima je točan odgovor *Other*.

Pouzdanosti testova su se mjerile *Cronbach's Alphe* koeficijentom, a provjera distribucije podataka *Kolmogorov-Smirnov* testom. Vrijednost *Cronbach's Alphe* kreće se od 0 do 1 gdje veće vrijednosti ukazuju na veću unutarnju konzistentnost, i to:

- > 0.90 – vrlo visoko pouzdani
- 0.80-0.90 – visoko pouzdani
- 0.70-0.79 – pouzdani
- 0.60-0.69 – granično/minimalno pouzdani
- <0.60 – neprihvatljivo.

Tablica 2 prikazuje osnovne metrijske vrijednosti testova.

Tablica 2. Osnovne metrijske vrijednosti testova

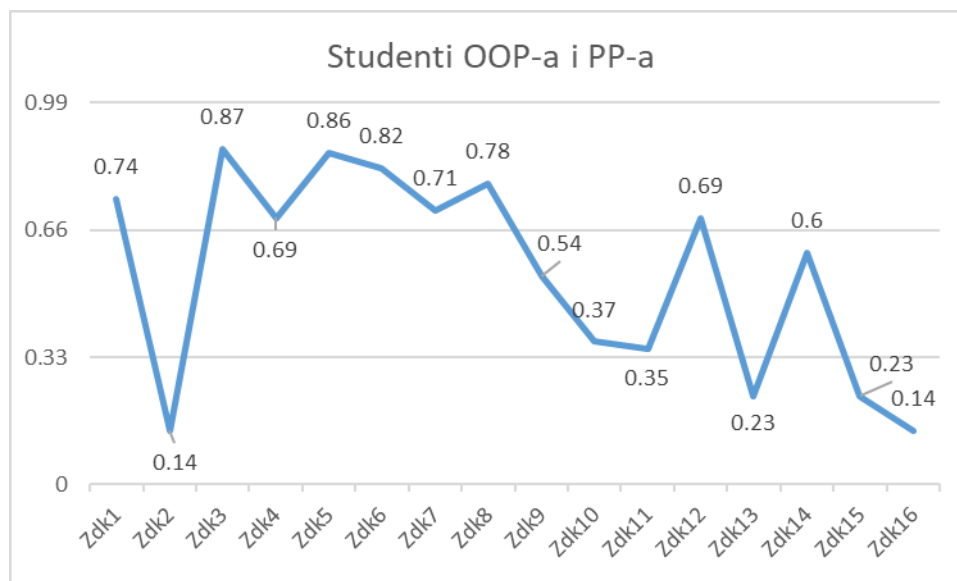
	<i>P2</i>	<i>OOP+PP</i>
N	144	78
Aritmetička sredina (%)	50.91	49.87
Mod	31.25	30
M (Medijan)	50	48.75
SD	22.40	20.88
Minimum (%)	6.25	5.00
Maksimum (%)	93.75	90.00
Maksimalan mogući %	100	100
Broj čestica	13	16

Cronbach α	0.72	0.772
Kolmogorov-Simirnov test	0.00	0.2

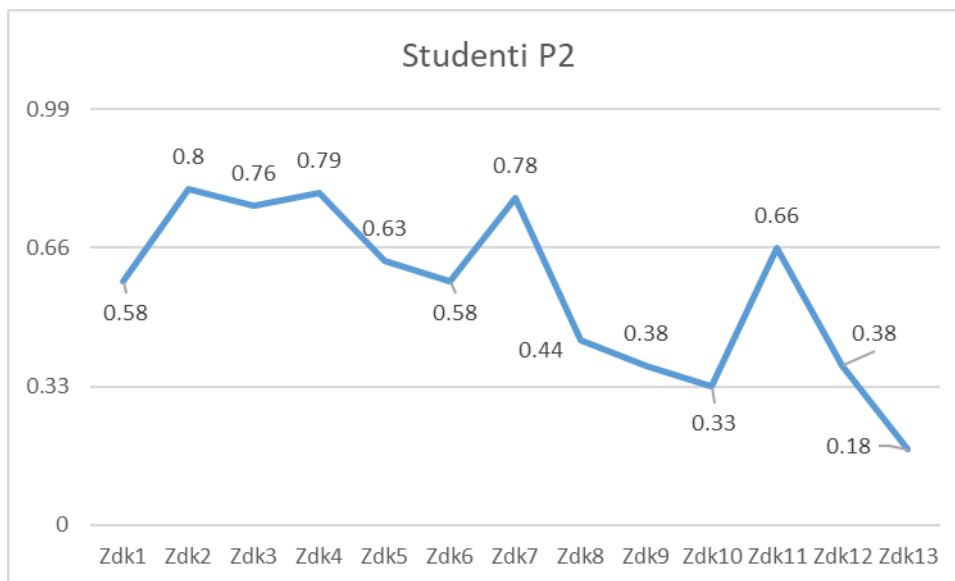
Za utvrđivanje težine zadataka koristio se Indeks težine čije vrijednosti određuju težinu pojedinog zadatka i to prema sljedećoj podjeli:

- > 0.66 – jednostavan zadatak
- $0.33-0.66$ – zadatak srednje težine
- <0.33 – težak zadatak

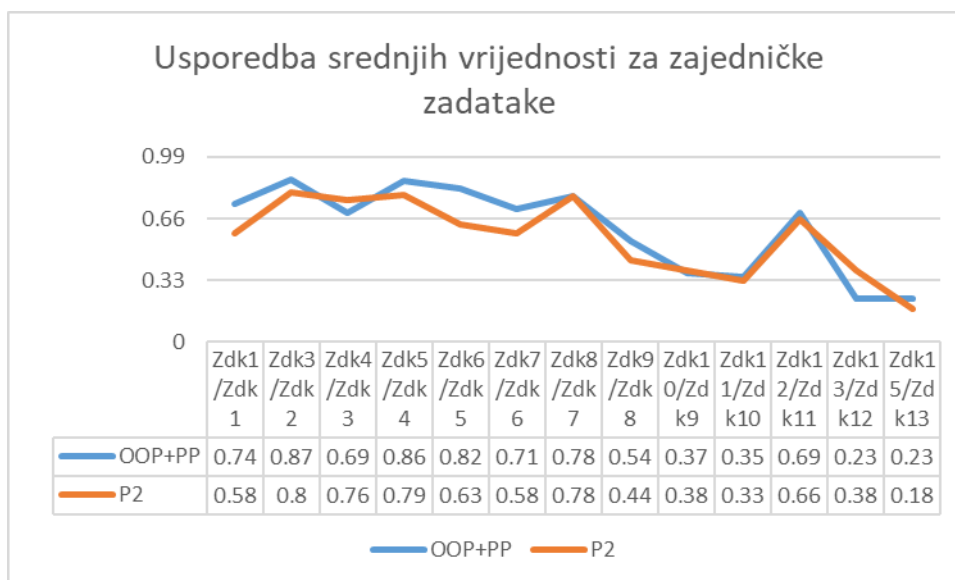
Na „Slika 1. Indeks težine – OOP i PP“ je prikazano kako su se kretale srednje vrijednosti ostvarenih bodova za sve zadatke koje je rješavala prva grupa, a na „Slika 2. Indeks težine – P2“ su prikazane srednje vrijednosti ostvarenih bodova za sve zadatke koje je rješavala druga grupa. Kao što je vidljivo na te dvije slike, većina zadataka pripada grupi zadataka srednje težine. Također, na slici pod rednim brojem tri možete vidjeti usporedbu indeksa težine za zadatke koji su dani svim studentima (takvih zadataka je bilo trinaest).



Slika 1. Indeks težine – OOP i PP



Slika 2. Indeks težine – P2



Slika 3. Usporedba indeksa težine za zajedničke zadatke

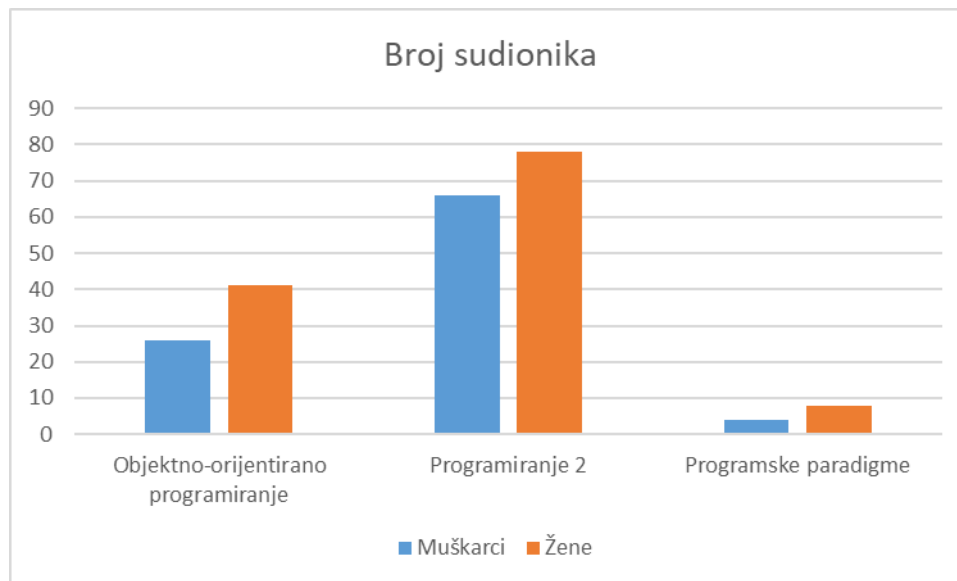
2.3. Sudionici

Sudionici testova bili su studenti Prirodoslovno-matematičkog fakulteta u Splitu (n=223).

Sudionici su pohađali jedan od tri kolegija (OOP, P2 ili PP). Kolegij P2 je pohađalo njih sto četrdeset četiri (144) - sedamdeset osam (78) žena i šezdeset šest (66) muškaraca. Četrdeset devet studenata kolegija P2 je bilo upisano na smjer Informatika, pedeset dvoje studenata je bilo upisano na smjer Informatika i tehnika, na smjer Matematika je bilo

upisano trideset pet studenata, a preostalih osam studenata je bilo upisano na smjer Matematika i informatika - smjer nastavnički.

Šezdeset sedam (67) sudionika je pohađalo kolegij OOP - četrdeset jedna (41) žena i dvadeset šest (26) muškaraca, a njih dvanaest (12) je pohađalo kolegij PP - osam (8) žena i četiri (4) muškarca. Studenti OOP-a i PP-a su bili upisani na smjerove: informatika, matematika i informatika, nastavnički smjer informatike, informatika i tehnika, matematika (računarska) i matematika (statistika i računarstvo).



Slika 4. Razdioba sudionika u odnosu na spol i kolegij

3. Rezultati

U ovom poglavlju bit će opisani rezultati te će se dati odgovori na postavljena istraživačka pitanja.

3.1. Razlika studijskih grupa u uspjehu na testu iz programiranja

Nakon što je utvrđeno postoji li normalna razdioba/distribucija te koji će se testovi koristiti došlo je vrijeme za analizu prvog istraživačkog pitanja koje se odnosi na razlike studijskih grupa u uspjehu na testu iz programiranja.

3.1.1. Grupe OOP i PP – usporedba uspješnosti studenata na testu iz programiranja u odnosu na studijsku grupu

Prvo što je bilo urađeno je reduciranje broja smjerova jer ima par smjerova koje pohađa mali broj studenata (npr. za Matematika i informatika; smjer: nastavnički $n=5$) pa su se takvi smjerovi spojili sa smjerovima koji imaju više studenata na način da im je srednja vrijednost što bliža jedna drugoj (smjer Matematika i informatika; smjer: nastavnički je spojen sa smjerom Informatika i smjerovi Matematika Racunarski i Matematika Matematicki su spojeni u jedan zajednički smjer pod nazivom Matematika ing). Također, jedan student ($n=1$) pohađa smjer Fizika pa je zbog toga taj cijeli podatak izuzet iz daljnje analize (znači u ovome trenutku $n=78$). Također, smjer Informatika (jednopedmetni); smjer: nastavnički je pripojen smjerovima Informatika i Matematika i informatika; smjer: nastavnički, a smjer Matematika Statistika i racunarstvo je pripojen smjeru Matematika ing.

One-way analysis of variance (ANOVA) test se koristio za utvrđivanje postoji li statistički značajnih razlika između smjerova. Rezultati ANOVA testa pokazali su da postoji statistički značajna razlika između smjerova ($F=10.295$, $p=0.000$).

Tablica 3. Deskriptivna statistika

Smjerovi	N	Srednja vrijednost	Minimum	Maksimum
Informatika+MI+Informatika nastav.	39	48.5256	5.00	87.50
Informatika i tehnika	17	35.8824	15.00	72.50
Matematika ing	22	63.0682	30.00	90.00

Nakon što smo se uvjerali da postoji statistički značajna razlika došlo je vrijeme za *Tukey post hoc* test pomoću kojega će se vidjeti gdje su značajne razlike.

Promatranjem rezultata dobivenih *Tukey post hoc* testom utvrđeno je da postoji statistički značajna razlika u postotku točnosti, prilikom rješavanja zadataka, između grupa u kojoj se nalaze studenti: informatike, matematike i informatike (MI) te studenti nastavničkog smjera informatike i grupe u kojoj su studenti inženjerske matematike ($p=0.013$), kao i između grupe gdje su studenti informatike i tehnike i grupe gdje su studenti inženjerske matematike ($p=0.000$). Statistički bitnih razlika nije bilo između preostalih grupa.

Nakon što su se p vrijednosti analizirale može se zaključiti da (za studente OOP-a i PP-a) postoje statistički značajne razlike između smjera inženjerske matematike i svih ostalih smjerova uzevši u obzir srednje vrijednosti koje označavaju koliko su (prosječno) studenti različitih smjerova točno riješili dane zadatke (Tablica 3) – studenti inženjerske matematike su najbolje riješili test (srednja vrijednost = 63.0682).

3.1.2. Grupa P2 – usporedba uspješnosti studenata na testu iz programiranja u odnosu na studijsku grupu

Prvo što je bilo provedeno je neparametrijski test *Kruskal-Wallisov H* temeljen na rangu. *Kruskal-Wallis H* test pokazao je da postoji statistički značajna razlika u postotcima između različitih smjerova studenata ($\chi^2(2) = 17.950$, $p = 0.000$) sa srednjim rezultatima za svaki smjer.

Nakon što je utvrđeno da postoje statistički značajne razlike, korišten je *Mann-Whitney U* test koji je neparametrijski test te se koristi za uspoređivanje srednjih vrijednosti dvije nezavisne skupine. Korištenjem *Mann-Whitney U* utvrđeno je da statistički značajne razlike postoje između smjerova informatika/informatika i tehnika ($U=855.00$, $p=0.04$) te između informatika i tehnika/matematika ($U=453.00$, $p=0.000$).

Prema rezultatima, gledajući srednje vrijednosti, zaključilo se da su studenti informatike (srednja vrijednost = 59.55) bolje riješili test od studenata informatike i tehnike (srednja vrijednost = 42.94). Također, studenti matematike (srednja vrijednost = 57.06) su bolje riješili test od studenata informatike i tehnike.

3.2. Razlika između studenata i studentica u uspjehu na testu iz programiranja

Nakon što je analizirano prvo istraživačko pitanje, prešlo se na analizu drugog istraživačkog pitanja.

3.2.1. Grupe OOP i PP – usporedba uspješnosti studenata na testu iz programiranja u odnosu na spol

Za analizu drugog istraživačkog pitanja, za studente OOP-a i PP-a, korišten je t-test.

Tablica *Group Statistics* sadržavala je deskriptivna statistiku za obje uspoređene grupe (M - muškarci i Z - žene), uključujući srednju vrijednost. Ukupno je bilo 29 studenata i 49 studentica. Nakon analize tablice *Independent Samples Test* uočeni su rezultati neovisnog t-testa. Iz dobivenih rezultata može se reći da ne postoje statistički značajne razlike između studenata i studentica OOP-a i PP-a gledajući točnost pri rješavanju testa ($p=0.55$).

Tablica 4. Deskriptivna statistika

Spol		N	Srednje vrijednosti
Postotak	M	29	51.7241
	Z	49	48.7755

3.2.2. Grupa P2 – usporedba uspješnosti studenata na testu iz programiranja u odnosu na spol

Da bi se analiziralo drugo istraživačko pitanje koje se odnosi na razlike studenata i studentica kolegija P2 u uspjehu na testu programiranja korišten je *Mann-Whitney* test. Analizom rezultata moglo se zaključiti da postoji statistički značajna razlika između studenata i studentica, tj. između spolova sudionika ($U=2029.50$, $p=0.029$). Nakon što su

se usporedile srednje vrijednosti za svaku grupu (M i Z), zaključilo se da su studenti bolje riješili test od studentica (Tablica 5.).

Tablica 5. Deskriptivna statistika

Spol		N	Srednje vrijednosti
Postotak	M	66	80.75
	Z	78	65.52

3.3. Razlika između studenata koji ponavljaju i onih koji ne ponavljaju kolegij

U ovom poglavlju obradit će se treće istraživačko pitanje.

3.3.1. Grupe OOP i PP – usporedba uspješnosti studenata na testu iz programiranja u odnosu na to jesu li studenti ponavljači ili ne

Ovo istraživačko pitanje za studente OOP-a i PP-a analiziralo se na isti način kao i drugo (korišten je t-test). Tablica *Group Statistics* prikazivala je deskriptivnu statistiku za dvije grupe koje su uspoređene (ponovni upis: Ne i Da). Među svim studentima (n=78) bilo je 60 studenata koji su prvi put upisali kolegije te 18 ponavljača. Kada se analizirala tablica *Independent Samples Test* moglo se vidjeti da srednje vrijednosti grupa (Ne i Da) nisu statistički značajno različite (p=0.061).

3.3.2. Grupa P2 – usporedba uspješnosti studenata na testu iz programiranja u odnosu na to jesu li studenti ponavljači ili ne

Kako bi utvrdili postoji li razlika u uspješnosti na testu između studenata koji ponavljaju i onih koji su prvi put upisali kolegij P2 korišten je *Mann-Whitney U* test.

Rezultat *Mann-Whitney U* testa pokazao je da postoji statistički značajna razlika (U=1700, p=0.023) između studenata koji ponavljaju kolegij (n=45) i studenata koji ne ponavljaju (n=99). Također, zaključilo se da su bolje riješili zadatke oni studenti koji ne ponavljaju

kolegij (srednja vrijednost = 77.83, a srednja vrijednost studenata koji ponavljaju kolegij iznosila je 60.78).

3.4. Analiza otkrivenih miskoncepcija

U ovome dijelu rada bit će opisane sve miskoncepcije koje su otkrivene u istraživanju te će za svaku miskoncepciju biti napisani prijedlozi za smanjivanje njenog pojavljivanja. Također, u ovome dijelu rada bit će prikazani kodovi zadataka (zadaci će biti prikazani u *JavaScript* programskom jeziku), ponuđeni odgovori, odgovori koji su miskoncepcije te postotci koji se odnose na udjel studenata koji su miskoncepcije smatrali točnim odgovorima.

Da bi se nešto proglasilo miskoncepcijom ne postoji striktno pravilo, a u ovome radu odgovor je proglašen miskoncepcijom ako je minimalno 15% studenata taj odgovor označilo kao odgovor koji smatraju ispravnim.

Zadaci i ponuđeni odgovori, koji su dani studentima, su pomno osmišljeni s ciljem otkrivanja miskoncepcija. Svaki zadatak je bio usmjeren na neki od osnovnih koncepata programiranja. Također, svaki ponuđeni odgovor je bio usmjeren na jednu od poznatih miskoncepcija (npr. kod zadataka koji su usmjereni na grananja jedan od ponuđenih odgovora je ciljao na miskoncepciju graničnih uvjeta).

Otkrivanje i analiza miskoncepcija izvršena je detaljno, provjeravajući svaki zadatak pojedinačno upotrebom *Crosstabs* i *Chi-square*.

Prvi zadatak (za sve studente)

U prvom zadatku ispitivao se pojam varijable. Tablica 6 prikazuje zadatak, ponuđene odgovore, odgovor koji se može smatrati miskoncepcijom i učestalost miskoncepcije za obje grupe.

Tablica 6. Prvi zadatak (za sve studente) - miskoncepcija

Kod zadatka	Ponuđeni odgovori	Miskoncepcija	Učestalost miskoncepcije	
			OOP i PP	P2

let najmanji = 3; let najveći = 1; let broj = 2; najmanji = najveći; najveći = najmanji - broj; console.log(najmanji, najveći);	a) -1 1 b) 1 -1 c) 1 1 d) 1 3 e) 3 1	c)	17.95%	21.53%
--	--	----	--------	--------

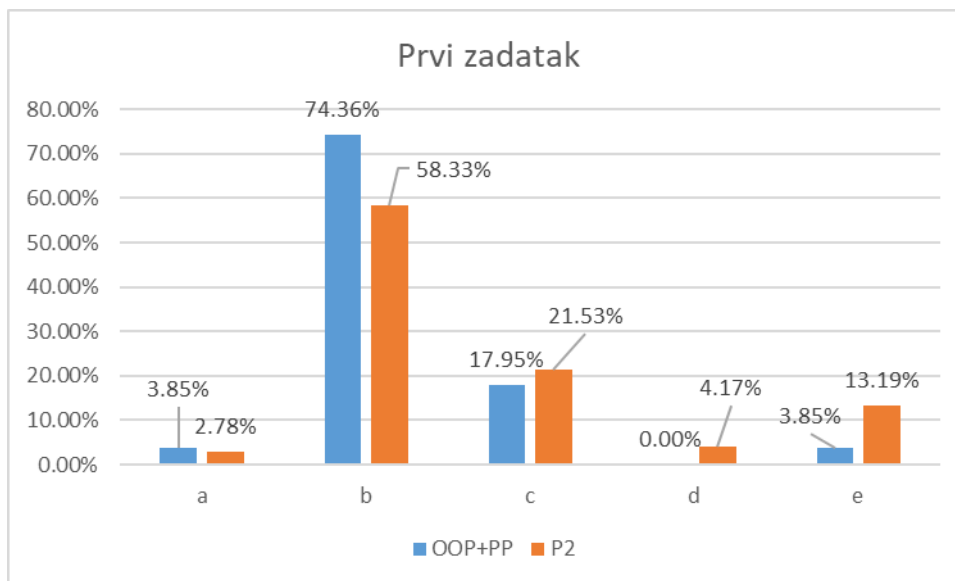
U prethodnoj tablici se može vidjeti da se odgovor c) 1 1 može smatrati miskoncepcijom. 17.95% studenata OOP-a i PP-a i 21.53% studenata P2 je odgovor 1 1 smatralo točnim.

Miskoncepcija koja bi se mogla povezati s odgovorom c) je ta da studenti misle da slijed kojim se izvode naredbe nije bitan. Ako je prvo napisana naredba kojom se izravno mijenja vrijednost varijable (najmanji = najveći;), a zatim naredba kojom se mijenja vrijednost varijable matematičkom operacijom (najveći = najmanji – broj;) onda se točno tim slijedom naredbe moraju izvršiti. Do odgovora c) studenti su, najvjerojatnije, došli tako da su zamijenili slijed izvršavanja dviju spomenutih naredbi te je za njih program u tom slučaju izgledao kao što je prikazano u tablici ispod.

Tablica 7. Mogući mentalni model studenata za navedenu miskoncepciju

let najmanji = 3; let najveći = 1; let broj = 2; najveći = najmanji - broj; najmanji = najveći; console.log(najmanji, najveći);
--

Jedan od savjeta za smanjenje učestalosti pojavljivanje ove miskoncepcije je da se studentima pokaže s par istih linija koda, ali s različitim slijedom izvršavanja da se neće uvijek dobiti isti rezultat. U ovom zadatku je vidljivo da promjena redoslijeda dviju naredbi ne daje isti rezultat. Na slici 5 je prikazana raspodjela odgovora kod studenata OOP-a i PP-a te raspodjela odgovora za studente P2 u prvom zadatku.



Slika 5. Raspodjela odgovora kod studenata – prvi zadatak

Na prethodnoj slici je vidljivo da je odgovor b) odabralo najviše studenata u obje grupe, a to je ujedno i točan odgovor. Idući najčešći odgovor koji su studenti davali u obje grupe je odgovor c), a taj odgovor se može smatrati kao miskoncepcija.

Drugi zadatak (za studente OOP-a i PP-a)

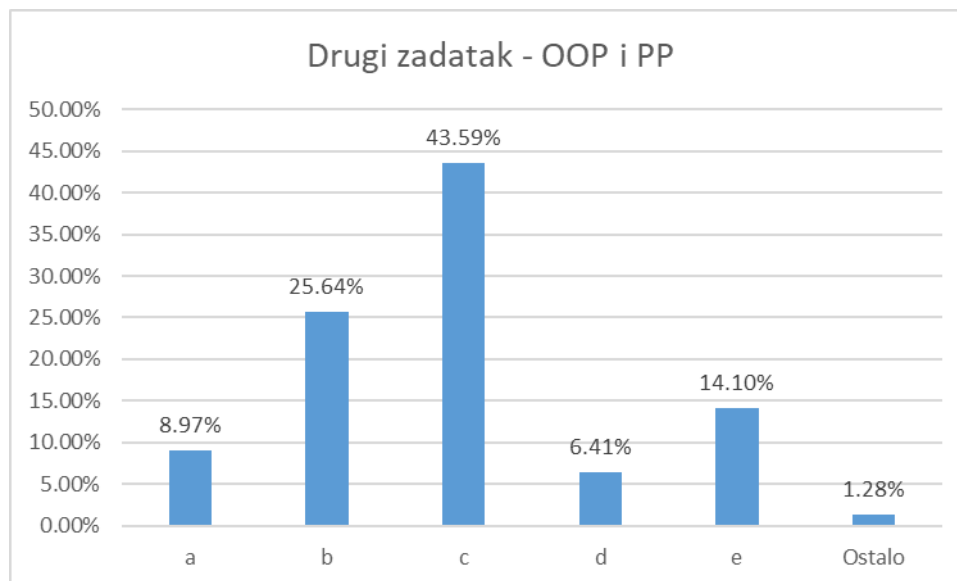
U drugom zadatku se također ispitivao pojam varijable. Tablica 8 prikazuje zadatak, ponuđene odgovore, odgovore koji se mogu smatrati miskoncepcijama i učestalost miskoncepcija za grupu studenata koji su pohađali OOP i PP.

Tablica 8. Drugi zadatak (za studente OOP-a i PP-a) - miskoncepcije

Kod zadatka	Ponuđeni odgovori	Miskoncepcije	Učestalost miskoncepcija	
			OOP i PP	P2
const predmet = "Programiranje"; var ECTS = 4; predmet = "Objektno-orijentirano programiranje"; ECTS = 6; console.log("Predmet je",predmet,"te vrijedi",ECTS,"ECTS boda.");	a) Predmet je Programiranje te vrijedi 4 ECTS boda.	b)	25.64%	/
	b) Predmet je Objektno-orijentirano programiranje te	c)	43.59%	/

	vrijedi 6 ECTS boda. c) Predmet je Programiranje te vrijedi 6 ECTS boda. d) Predmet je Objektno- orijentirano programiranje te vrijedi 4 ECTS boda. e) Program će javiti grešku pa se iz tog razloga ništa neće ispisati.			
--	---	--	--	--

U prethodnoj tablici je prikazano da se odgovori b) i c) mogu smatrati miskoncepcijama. Odgovor b) je odabralo 25.64% studenata OOP-a i PP-a, a odgovor c) je odabralo 43.59% studenata OOP-a i PP-a (ovaj zadatak je za studente P2 izbačen). Odgovor b) upućuje na miskoncepciju da studenti ne razumiju deklariranje varijabli, tj. što znači riječ *const* uz varijablu predmet, a učestalost pojavljivanja ove miskoncepcije bi se mogla smanjit tako da se studentima što više puta riječ *const* poveže s engleskom riječi *constant* te da se poveže i s matematičkim π -om jer je π konstanta koja se ne može nikada mijenjati jer je to zabranjeno (kao što se ne može pokušati promijeniti vrijednost varijable koja je označena ključnom riječi *const* jer će u protivnom program javiti grešku – odgovor e)). Odgovor c) upućuje na sličnu miskoncepciju kao i odgovor b) samo što je razlika što studenti vjerojatno razumiju da se vrijednost varijable koja je označena ključnom riječju *const* ne može mijenjati, ali nisu savladali činjenicu da će program javiti grešku ako se u bilo kojem trenutku vrijednost varijable, koja je označena ključnom riječju *const*, pokuša promijeniti. U ovom slučaju vidimo da studenti nisu odabrali odgovor u kojem se vrijednost varijable predmet promijenila u Objektno-orijentirano programiranje pa se može pretpostaviti da se njihova miskoncepcija može lakše ispraviti nego miskoncepcija studenata koji su označili odgovor b). Jedan od savjeta kako se učestalost ove miskoncepcija može pokušati smanjiti je da se studentima ispriča priča da ako bi netko išao promijeniti matematičku vrijednost π da bi velika većina ljudi rekla da je to greška kao što i program javlja grešku prilikom pokušaja promjene vrijednosti varijable definirane ključnom riječju *const*.



Slika 6. Raspodjela odgovora za studente OOP-a i PP-a u drugom zadatku

Na prethodnoj slici se može vidjeti da su odgovori (b i c) koji se mogu smatrati miskoncepcijama bili učestaliji od točnog odgovora (e).

Treći zadatak (za studente P2) / četvrti zadatak (za studente OOP-a i PP-a)

U ovome zadatku se također ispitivao pojam varijable. Tablica 9 prikazuje zadatak, ponuđene odgovore, odgovor koji se može smatrati miskoncepcijom i učestalost miskoncepcije za obje grupe studenata.

Tablica 9. Treći zadatak (za studente P2) / četvrti zadatak (za studente OOP-a i PP-a) - miskoncepcija

Kod zadatka	Ponuđeni odgovori	Miskoncepcija	Učestalost miskoncepcije	
			OOP i PP	P2
var br2 = 1; let br1 = 2; br2 = br1 + 3; br1 = br2 + 2; console.log(br2, 'br1');	a) 1 2 b) 2 1 c) 6 3 d) 3 5 e) Other __	e) 5 7	23.08%	23.61%

U tablici 9 je prikazano da je odgovor 5 7 ponudilo 23.08% studenata OOP-a i PP-a i 23.61% studenata P2. Ova miskoncepcija upućuje na problem referenciranja varijabli. Odgovor 5 7 studenti su dobili najvjerojatnije tako što su u zadnjoj naredbi koja služi za

ispis zanemarili navodnike kojima je omeđen br1 pa iz tog razloga napisali broj 7 ili su primijetili navodnike, ali ne znaju da se unutar navodnika stavlja ono što želimo doslovno tako ispisati (iako u programu postoji varijabla imena br1, ispisuje se riječ br1). Savjet za smanjivanje učestalosti ove miskoncepcije je da se što više primjera pokazuje studentima pomoću kojih će razlikovati kada se želi ispisati vrijednost neke varijable, a kada se želi ispisati neka riječ (ponajviše bazirano da ta riječ bude ime neke varijable koja postoji u programu kao u ovom slučaju).

Četvrti zadatak (za studente P2)

U ovome zadatku se ispitivalo grananje. Tablica 10 prikazuje zadatak, a tablica 11 ponuđene odgovore, odgovor koji se može smatrati miskoncepcijom i učestalost miskoncepcije za grupu studenata koji su pohađali P2.

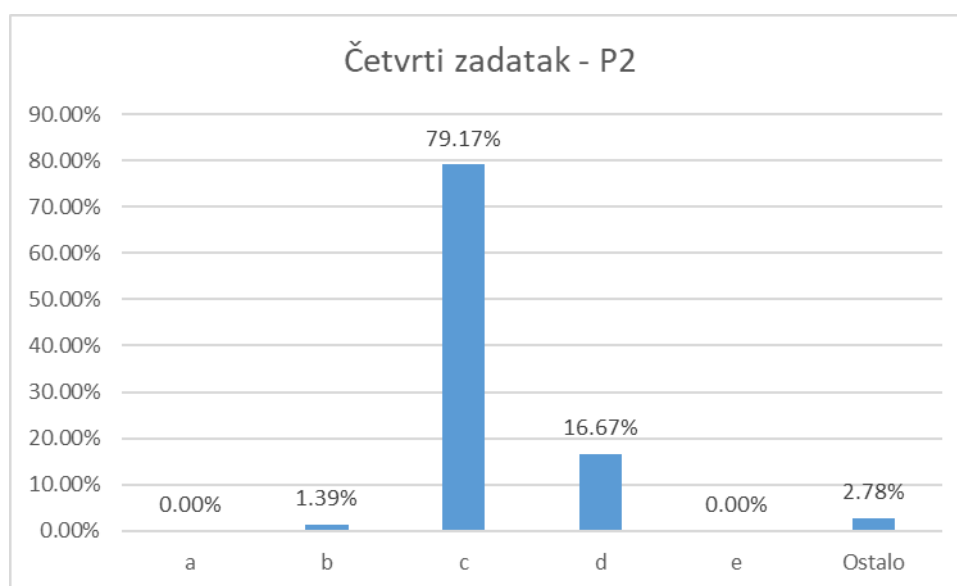
Tablica 10. Četvrti zadatak (za studente P2) - kod zadatka

Kod zadatka
<pre>var brzina = 60; var policija = 'policija'; if (brzina > 60) { policija = ("Prebrza vožnja!"); } console.log(policija);</pre>

Tablica 11. Četvrti zadatak (za studente P2) - miskoncepcija

Ponuđeni odgovori	Miskoncepcija	Učestalost miskoncepcije	
		OOP i PP	P2
a) brzina b) 60 c) policija d) Prebrza vožnja! e) Other _____	d)	/	16.67%

U prethodnim tablicama vidljivo je da se odgovor d) može smatrati miskoncepcijom jer ga je odgovorilo 16,67% studenata P2, a kod studenata OOP-a i PP-a odgovor d) nije miskoncepcija (kao ni drugi odgovori). Ova miskoncepcija proizlazi iz nerazumijevanja graničnog uvjeta. Vrijednost varijable koja se provjerava (brzina) u if uvjetu postavljena je na 60, a to je granični slučaj (u ovome primjeru) te je zbog toga odgovor Prebrza vožnja! toliko studenata odgovorilo. Pojavljivanje ove miskoncepcije bi se moglo uvelike smanjiti tako da se što više zadataka sa studentima prođe gdje se u uvjetima grananja nalaze granične vrijednosti te bi u tom slučaju studenti više pazili što se točno provjerava u uvjetu.



Slika 7. Raspodjela odgovora za studente P2 u četvrtom zadatku

Na prethodnoj slici je vidljivo da je točan odgovor izabralo najviše studenata (odgovor c), a da nitko nije smatrao da su odgovori a) i e) točni.

Peti zadatak (za studente P2)

U sljedećem zadatku se (također) ispitivalo grananje. Tablica 12 prikazuje zadatak, a tablica 13 ponuđene odgovore, odgovor koji se može smatrati miskoncepcijom i učestalost miskoncepcije za grupu studenata koji su pohađali P2.

Tablica 12. Peti zadatak (za studente P2) – kod zadatka

Kod zadatka
var brzina = 60; var policija = 'policija';


```

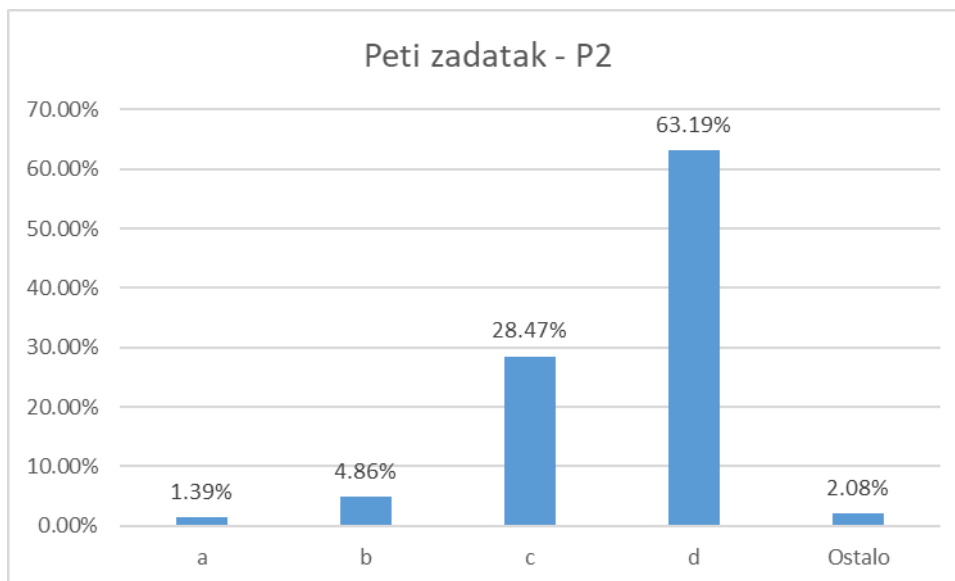
if (brzina >= 60) {
  brzina = 55;
  policija = "Prebrza vožnja!";
} else {
  policija = "Vozite unutar ograničenja, bravo!";
}
console.log(policija);

```

Tablica 13. Peti zadatak (za studente P2) – miskoncepcija

Ponudeni odgovori	Miskoncepcija	Učestalost miskoncepcije	
		OOP i PP	P2
a) 55 b) policija c) Vozite unutar ograničenja, bravo! d) Prebrza vožnja! e) Other _____	c)	/	28.47%

U tablicama 12 i 13 je vidljivo da odgovor c) smatra 28.47% studenata P2 ispravnim te se iz tog razloga taj odgovor može smatrati miskoncepcijom (kod studenata OOP-a i PP-a nijedan odgovor nije proglašen miskoncepcijom). Ova miskoncepcija proizlazi iz problema logičkog uvjeta. Studenti koji su odgovor c) označili kao točan odgovor smatraju (najvjerojatnije) da je potrebno izvršiti naredbe u else bloku jer je vrijednost varijable brzina u jednom trenutku promijenjena na 55 (55 sada ne zadovoljava if uvjet te se zato „ispisuje“ odgovor c)). Najbolji način za smanjiti učestalost ove miskoncepcije je taj da se što više ovakvih zadataka demonstrira studentima da bi se što bolje uvježbalo provjeravanje uvjeta.



Slika 8. Raspodjela odgovora za studente P2 u petom zadatku

Na prethodnoj slici je vidljivo da je točan odgovor (d) označilo najviše studenata te da je odgovore a i b smatralo malo studenata točnim odgovorima.

Šesti zadatak (za studente P2) / sedmi zadatak (za studente OOP-a i PP-a)

I u sljedećem zadatku se ispitalo grananje. Tablica 14 prikazuje zadatak, a tablica 15 ponuđene odgovore, odgovor koji se može smatrati miskoncepcijom i učestalost miskoncepcije za obje grupe studenata.

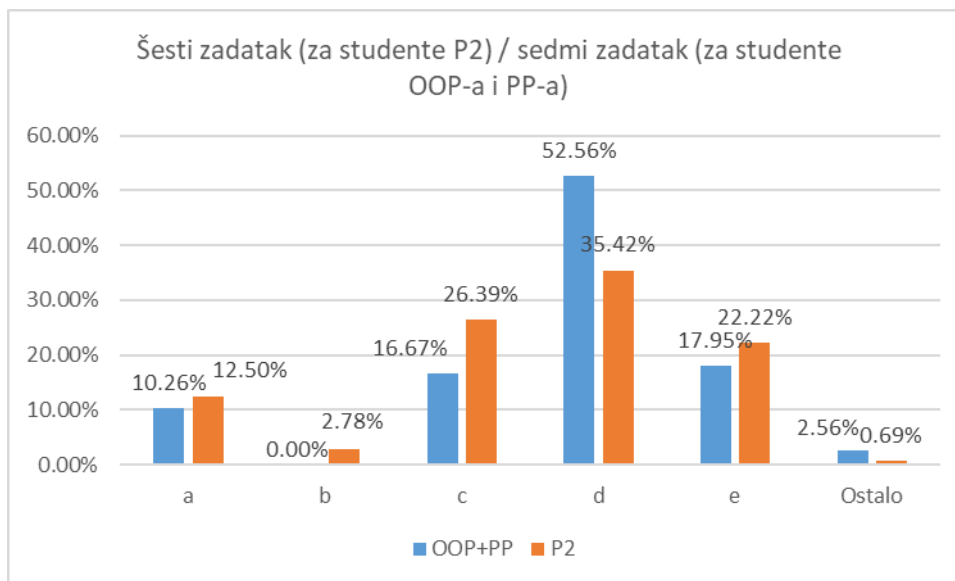
Tablica 14. Šesti zadatak (za studente P2) / sedmi zadatak (za studente OOP-a i PP-a) – kod zadatka

Kod zadatka
<pre> var broj = 20; var poruka = ""; if (broj < 100) { poruka = poruka + "Manji od 20!"; } if (broj < 50) { poruka = poruka + "Manji od 50!"; } if (broj === 20) { poruka = poruka + "20!"; } console.log(poruka); </pre>

Tablica 15. Šesti zadatak (za studente P2) / sedmi zadatak (za studente OOP-a i PP-a) –
miskoncepcija

Ponudeni odgovori	Miskoncepcija	Učestalost miskoncepcije	
		OOP i PP	P2
a) Manji od 20! b) Manji od 50! c) 20! d) Manji od 20!Manji od 50!20! e) Manji od 20! Manji od 50! 20!	c)	16.67%	26.39%

Kada je riječ o odgovoru c) koji je ponudilo 16.67% studenata OOP-a i PP-a te 26.39% studenata P2, on ukazuje na miskoncepciju koja se odnosi na nerazumijevanje razlika između više jednostrukih grananja i jednog višestrukog grananja. Kao što se vidi u kodu zadatka, postoje tri jednostruka grananja, a provjerava se svaki uvjet jednostrukog grananja i ovisno o tome je li uvjet zadovoljen ili ne se ulazi ili ne ulazi u if blok. Pojavljivanje ove miskoncepcije bi se moglo smanjiti tako da se par istih zadataka napravi s više jednostrukih i jednim višestrukim grananjem te da se usporede rezultati jer se najčešće neće isti rezultat dobiti (kao što se isti rezultat ne bi dobio ni u ovome zadatku da je korišteno jedno višestruko grananje umjesto tri jednostruka grananja).



Slika 9. Raspodjela odgovora za studente OOP-a i PP-a u sedmom zadatku

Na prethodnoj slici se može vidjeti da je najviše studenata smatralo da su odgovori d) i e) točni odgovori. U svrhu ovoga istraživanja oba odgovora (d i e) su se smatrali točnima jer se smatralo da je studente nepažnja (a ne miskoncepcija) „odvukla“ na odgovor d) umjesto na odgovor e). Odgovor c) se s prikazanim postotcima učestalosti smatra kao miskoncepcija.

Osmi zadatak (za studente P2)

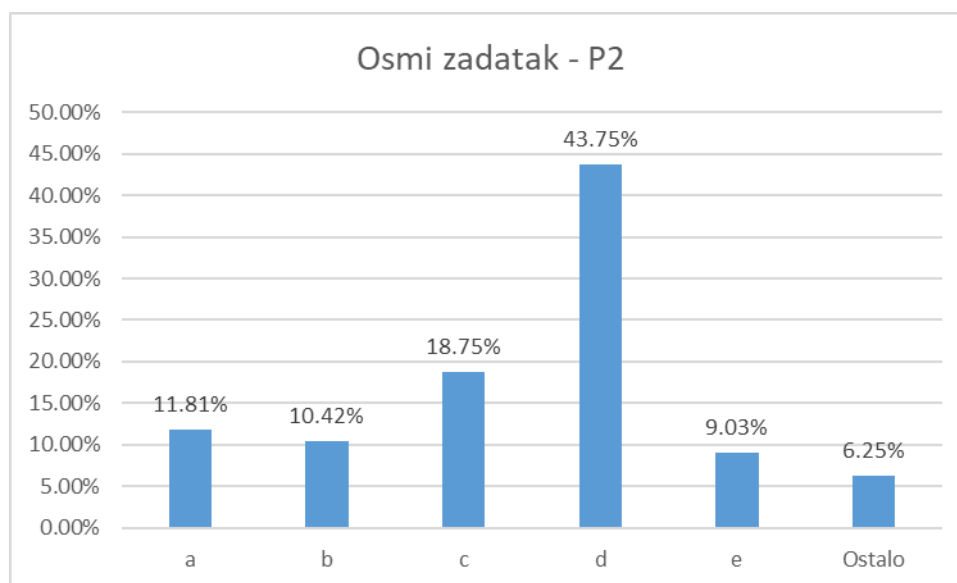
U sljedećem zadatku se ispitala *for* petlja. Tablica 16 prikazuje zadatak, ponuđene odgovore, odgovor koji se može smatrati miskoncepcijom i učestalost miskoncepcije za grupu studenata koji su pohađali P2.

Tablica 16. Osmi zadatak (za studente P2) - miskoncepcija

Kod zadatka	Ponuđeni odgovori	Miskoncepcija	Učestalost miskoncepcije	
			OOP i PP	P2

<pre> for (var i = 1; i<=3; i++) { console.log(i + 1); i = i + 1; console.log(i - 1); } </pre>	a) Brojeve: 2 3 4 1 3 b) Brojeve: 2 4 0 1 2 c) Brojeve: 1 2 3 4 d) Brojeve: 2 1 4 3 e) Brojeve: 2 0 4 2	c)	/	18.75%
---	---	----	---	--------

U prethodnoj tablici je prikazano da je odgovor c) odabralo 18.75% studenata P2. Među studentima OOP-a i PP-a nijedan odgovor nije proglašen miskoncepcijom. Ova miskoncepcija je vjerojatno nastala zbog nerazumijevanja ažuriranja vrijednosti unutar petlje. Također, ova miskoncepcija je možda nastala i zbog mišljenja da će program sam sortirati brojeve bez da mu se to navede u programu, tj. da je program dovoljno pametan da obavi nešto bez da mu se striktno navede (program ispisuje brojeve: 2, 1, 4, 3, a kada se sortiraju ti brojevi dobije se odgovor c)). Da bi se smanjila učestalost ove miskoncepcije, savjet je da se sa studentima prolazi što više zadataka gdje se vrijednosti varijabla mijenjaju „zbunjujuće“ i gdje ispis nije sortiran.



Slika 10. Raspodjela odgovora za studente P2 u osmom zadatku

Točan odgovor je odgovor d) i njega je najviše studenata označilo kao točan odgovor, a odgovor c) se može smatrati miskoncepcijom.

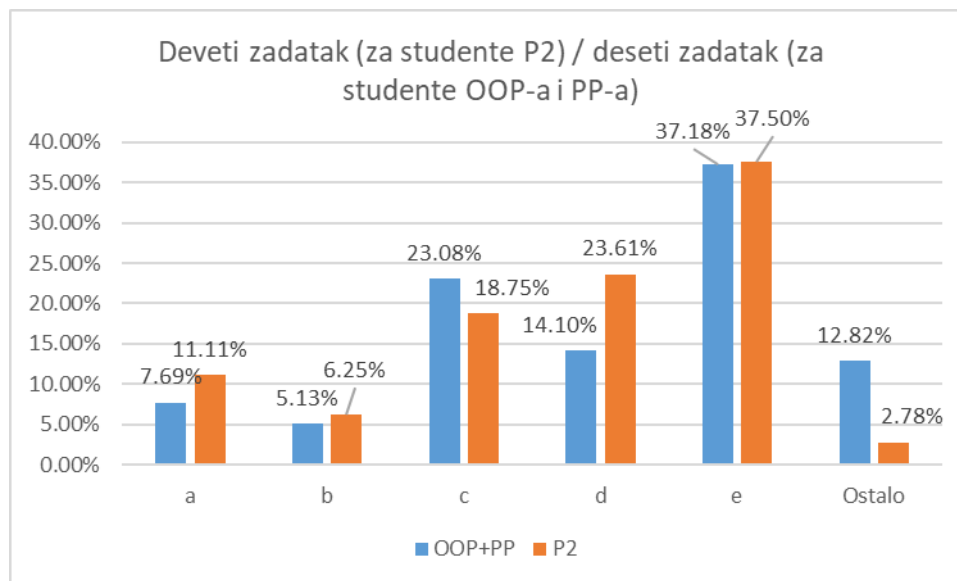
Deveti zadatak (za studente P2) / deseti zadatak (za studente OOP-a i PP-a)

I u sljedećem zadatku se, također, ispitivala *for* petlja. Tablica 17 prikazuje zadatak, ponuđene odgovore, odgovore koji se mogu smatrati miskoncepcijama i učestalost miskoncepcija za obje grupe studenata.

Tablica 17. Deveti zadatak (za studente P2) / deseti zadatak (za studente OOP-a i PP-a) - miskoncepcije

Kod zadatka	Ponuđeni odgovori	Miskoncepcije	Učestalost miskoncepcije	
			OOP i PP	P2
<pre>var a = 1; var b = 2; for (var i = 0; i<2; i++) { console.log(b); for (var j = 1; j<=2; j++) { console.log(a); } }</pre>	a) Brojeve: 1 1 2 2 b) Brojeve: 1 1 1 1 2 2 c) Brojeve: 2 1 2 1 d) Brojeve: 2 2 1 1 e) Brojeve: 2 1 1 2 1 1	c) d)	23.08% /	18.75% 23.61%

Kod što je prikazano u prethodnoj tablici odgovor c) je ponudilo 23.08% studenata OOP-a i PP-a i 18.75% studenata P2, a odgovor d) je ponudilo 23.61% studenata P2. Miskoncepcija koja se pojavljuje nastaje najvjerojatnije zbog nerazumijevanja redosljeda izvršavanja ugniježdenih petlji. Petlja s varijablom „i“ se izvršava prva, a zatim petlja s varijablom „j“. Također, unutar petlje s varijablom „j“, ispisuje se vrijednost „a“ koja je fiksirana na 1. Važno je napomenuti da se vrijednosti varijabli „a“ i „b“ ne mijenjaju unutar petlje, iako se petlje izvršavaju više puta. Također, treba naglasiti da se unutarnja petlja s varijablom „j“ izvršava svaki put kada se izvrši jedna iteracija vanjske petlje s varijablom „i“ (u svakoj iteraciji vanjske petlje će se ispisati vrijednost varijable „a“ dva puta unutar petlje s varijablom „j“). Savjet kako smanjiti učestalost ove miskoncepcije je da se pažljivo objasni redosljed izvršavanja petlji te da se naglasi da se vrijednosti varijabli ne mijenjaju uvijek tijekom izvršavanja petlji (osim ako se eksplicitno navede unutar petlje). Također, treba poticati studente na pisanje pseudokoda ili provjeru koraka izvršavanja petlji kako bi bolje razumjeli njihov redosljed izvršavanja te vježbati što više zadataka s ugniježdenim petljama.



Slika 11. Raspodjela odgovora za studente OOP-a i PP-a u desetom zadatku i za studente P2 u devetom zadatku

Odgovor e) je točan odgovor i njega je u obje grupe označilo najviše studenata. Odgovor c) se smatra miskoncepcijom u obje grupe studenata, a odgovor d) se smatra miskoncepcijom samo u grupi P2.

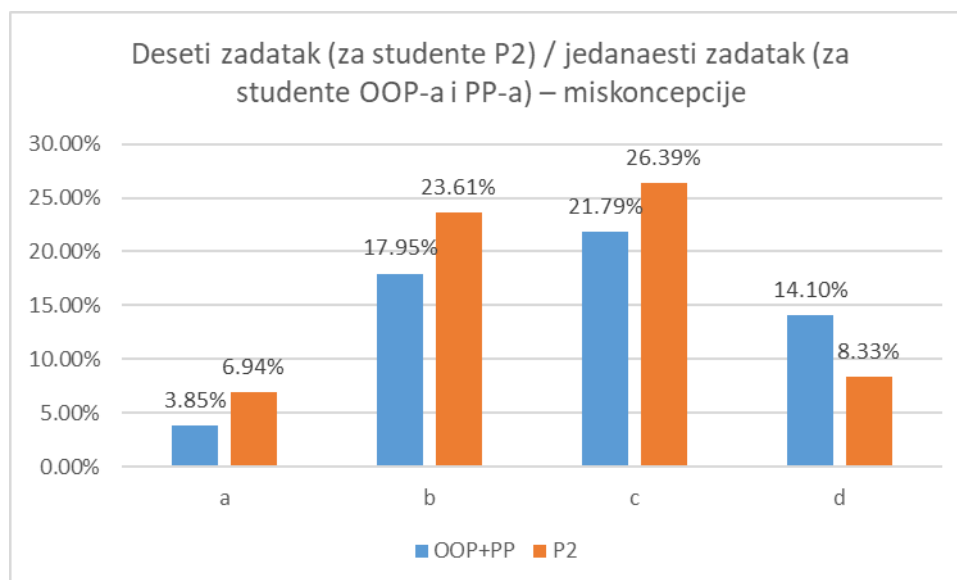
Deseti zadatak (za studente P2) / jedanaesti zadatak (za studente OOP-a i PP-a)

U idućem zadatku se ispitala petlja, ali ne više *for*, nego *while* petlja. Tablica 18 prikazuje zadatak, ponuđene odgovore, odgovore koji se mogu smatrati miskoncepcijama i učestalost miskoncepcija za obje grupe studenata.

Table 18. Deseti zadatak (za studente P2) / jedanaesti zadatak (za studente OOP-a i PP-a) – miskoncepcije

Kod zadatka	Ponuđeni odgovori	Miskoncepcije	Učestalost miskoncepcije	
			OOP i PP	P2
<pre>i = 1; while (i <= 5) { console.log('i'); };</pre>	a) Program će ispisati brojeve 1 2 3 4 5.	b)	17.95%	23.61%
	b) Program će ispisati pet puta slovo i.	c)	21.79%	26.39%
	c) i			
	d) Program neće ništa ispisati.			
	e) Other			

Kao što je prikazano u tablici 18 netočan odgovor b) odgovorilo je 17.95% studenata OOP-a i PP-a, a odgovor c) je odgovorilo 21.79% studenata OOP-a i PP-a. Odgovor „Program će ispisati pet puta slovo i.“ je miskoncepcija koja najvjerojatnije proizlazi iz pretpostavke da se vrijednost „i“ automatski povećava unutar petlje (vjerojatno utjecaj na ovo ima petlja for). Odgovor „i“ je miskoncepcija koju su studenti odgovorili jer nisu dobro interpretirali uvjet petlje. Da bi se smanjile učestalosti ovih dviju miskoncepcija bilo bi dobro što bolje razjasnite koncept petlje *while*, tj. detaljno objasniti: kako petlja radi, kako se uvjeti provjeravaju i kako tijelo petlje izvršava svoj kod. Također, dobro je uvijek potaknuti studente da izvrše ručnu provjeru koraka izvršavanja petlje kako bi razumjeli redoslijed izvršavanja i očekivane rezultate. Za ove miskoncepcije je dobro napraviti demonstraciju uživo, tj. izvesti interaktivne primjere ili demonstracije uživo.



Slika 12. Raspodjela odgovora za studente OOP-a i PP-a u jedanaestom zadatku i za studente P2 u desetom zadatku

U ovome zadatku točan odgovor nije bio ponuđen nego ga je trebalo napisati pod odgovor Other. Većina studenata koja je pisala svoj odgovor je točno napisala da će se beskonačno ispisivati slovo „i“ zbog uvjeta petlje, a bilo je i onih koji su pisali netočan odgovor (npr. broj 1 beskonačno puta).

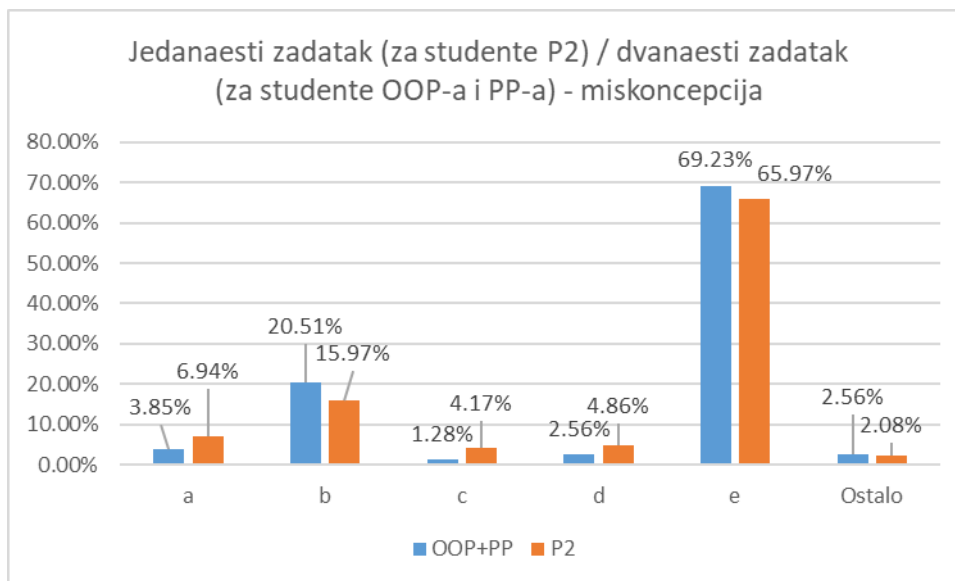
Jedanaesti zadatak (za studente P2) / dvanaesti zadatak (za studente OOP-a i PP-a)

While petlja se ispitivala u ovome zadatku. Tablica 19 prikazuje zadatak, ponuđene odgovore, odgovor koji se može smatrati miskoncepcijom i učestalost miskoncepcije za obje grupe studenata.

Tablica 19. Jedanaesti zadatak (za studente P2) / dvanaesti zadatak (za studente OOP-a i PP-a) - miskoncepcija

Kod zadatka	Ponudeni odgovori	Miskoncepcija	Učestalost miskoncepcije	
			OOP i PP	P2
<pre>i = 5; while (i < 5) { console.log(i); }</pre>	a) 5 b) Program će ispisati broj 5 beskonačan broj puta. c) i d) Program će ispisati slovo i beskonačan broj puta. e) Program neće ništa ispisati.	b)	20.51%	15.97%

U tablici 19 može se vidjeti da se odgovor „Program će ispisati broj 5 beskonačan broj puta“ smatra miskoncepcijom jer taj odgovor nije ispravan, a ponudilo ga je 20.51% studenata OOP-a i PP-a te 15.97% studenata P2. Ova miskoncepcija može se pojaviti iz različitih razloga, ali često je povezana s krivom interpretacijom uvjeta petlje. Početna vrijednost varijable „i“ je 5, što znači da uvjet odmah nije ispunjen i petlja se uopće ne izvršava. Kako bi se smanjila učestalost ove miskoncepcije preporučuje se: naglašavanje važnosti pažljivog čitanja i razumijevanja koda (studente treba potaknuti da čitaju svaku liniju koda pažljivo i da razumiju svaku naredbu i uvjet petlje), demonstriranje primjera i koraka izvršavanja (izvođenje primjera uživo ili koraka izvršavanja može pomoći studentima da vizualno vide kako se petlje izvršavaju i kako se uvjeti provjeravaju) i poticanje testiranja s različitim vrijednostima (studente treba potaknuti da testiraju svoj kod s različitim vrijednostima kako bi vidjeli kako se petlje ponašaju u različitim situacijama i uvjetima).



Slika 13. Raspodjela odgovora za studente OOP-a i PP-a u dvanaestom zadatku i za studente P2 u jedanaestom zadatku

Odgovor e) je točan odgovor za ovaj zadatak te je vidljivo da je taj odgovor odgovorilo najviše studenata iz obje grupe, a odgovor b) se može smatrati kao miskoncepcija u obje grupe.

Dvanaesti zadatak (za studente P2) / trinaesti zadatak (za studente OOP-a i PP-a)

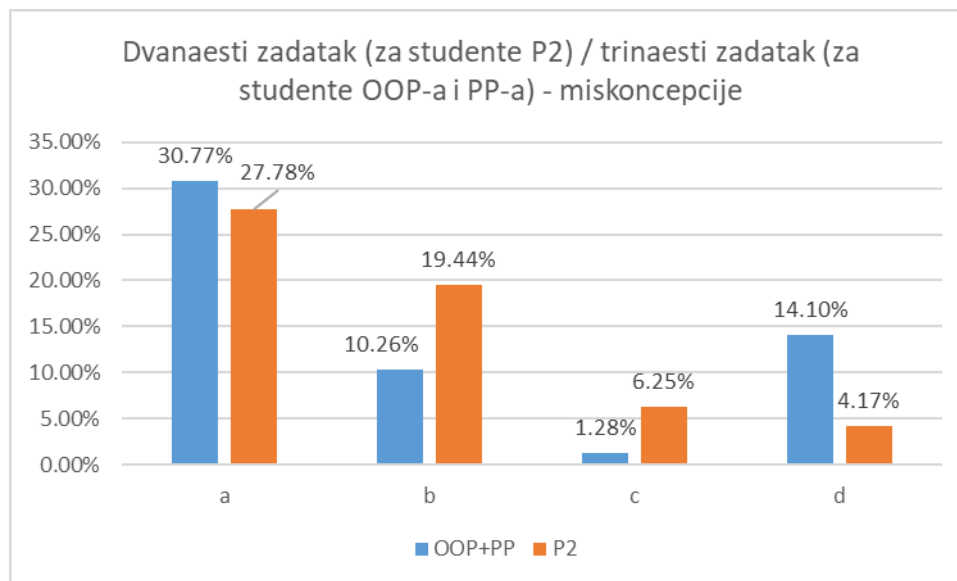
Idući osnovni koncept koji se ispitivao je niz. Tablica 20 prikazuje zadatak, ponuđene odgovore, odgovore koji se mogu smatrati miskoncepcijama i učestalost miskoncepcija za obje grupe studenata.

Tablica 20. Dvanaesti zadatak (za studente P2) / trinaesti zadatak (za studente OOP-a i PP-a) - miskoncepcije

Kod zadatka	Ponuđeni odgovori	Miskoncepcije	Učestalost miskoncepcije	
			OOP i PP	P2

let niz = [0, 5, 1, 3]; let broj = 2; niz.push(broj); niz.push(4); let niz1 = [6, 7]; let zavrzni = niz.concat(niz1); console.log(zavrzni);	a) [0, 1, 2, 3, 4, 5, 6, 7] b) [0, 5, 1, 3, 2, 4] c) [0, 5, 1, 3] d) [6, 7] e) Other _____	a) b)	30.77% /	27.78% 19.44%
---	--	--------------	-----------------	----------------------

Kao što je prikazano u prethodnoj tablici, odgovor a) se smatra miskoncepcijom jer je taj odgovor odgovorilo 30.77% studenata OOP-a i PP-a i 27.78% studenata P2. Odgovor b) je odgovorilo 19.44% studenata P2. Miskoncepcija koja proizlazi iz odgovora a) pojavljuje se zbog pogrešnog razumijevanja metode concat() i pretpostavke o sortiranju elemenata u rezultirajućem nizu (pretpostavka da je program dovoljno pametan da sam sortira elemente bez da mu se to naglasi naredbom u kodu). Studenti mogu pretpostaviti da će se broj 2 smjestiti između 1 i 3, stvarajući niz koji je sortiran u rastućem redoslijedu. Metoda concat() spaja dva ili više nizova, zadržavajući redoslijed elemenata iz svakog niza. U konkretnom zadatku, prvo se spaja niz niz s nizom niz1. Redoslijed elemenata ostaje isti kao u izvornim nizovima, bez ikakvog sortiranja ili promjene redoslijeda. Miskoncepcija koja proizlazi iz odgovora b) pojavljuje se zbog studentovog potpunog zanemarivanja naredbe (zavrzni = niz + niz1) za koju smatraju da se ne može izvršiti (pretpostavka je da studenti misle da se dva niza ne mogu zbrojiti – ciljajući na matematičko zbrajanje) kojom su nizovi konkatenirani korištenjem operatora +. Važno je napomenuti da izvor miskoncepcija može biti različit pa su i savjeti različiti poput: detaljnog objašnjavanja da je cilj metode concat() jednostavno spajanje nizova, bez ikakve promjene redoslijeda elemenata, demonstriranje primjera uživo (izvedba primjera s različitim nizovima kako bi studenti mogli vidjeti kako se elementi spajaju bez promjene redoslijeda) te poticanje logičkog razmišljanja (poticanje studenata da logički promisle o rezultatu koda i redoslijedu operacija).



Slika 14. Raspodjela odgovora za studente OOP-a i PP-a trinaestom zadatku i za studente P2 u dvanaestom zadatku

U ovome zadatku točan odgovor nije bio ponuđen nego ga je trebalo napisati pod odgovor Other. Većina studenata koja je pisala svoj odgovor je točno napisala da će se ispisati niz [0, 5, 1, 3, 2, 4, 6, 7], ali je bilo i onih koji nisu točno napisali odgovor (npr. greška).

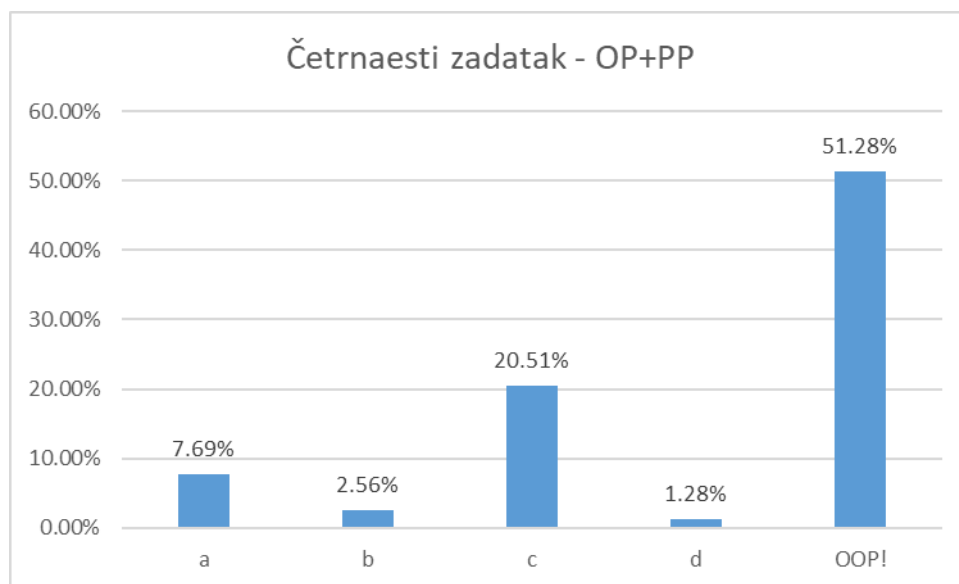
Četrnaesti zadatak (za studente OOP-a i PP-a)

Idući osnovni koncept koji se ispitivao su metode i objekti. Tablica 21 prikazuje zadatak, ponuđene odgovore, odgovor koji se može smatrati miskonceptijom i učestalost miskonceptije za studente OOP-a i PP-a.

Tablica 21. Četrnaesti zadatak - miskonceptija

Kod zadatka	Ponuđeni odgovori	Miskonceptija	Učestalost miskonceptije	
			OOP i PP	P2
<pre>let predmet = { ime: "OOP", ECTS: 6, ispis() { return this.ime + "!"; } }; console.log(predmet.ispis());</pre>	a) OOP b) OOP 6 c) OOP 6! d) 6! e) Other _____	c)	20.51%	/

Odgovor c) OOP 6! se smatra miskoncepcijom jer je taj odgovor ponudilo 20.51% studenata OOP-a i PP-a (studentima P2 ovaj zadatak je izbačen). Ova miskoncepcija proizlazi iz pretpostavke da u rezultatu moraju biti sva svojstva objekta. U ovom zadatku, metoda ispis() definirana je unutar objekta predmet. Metoda vraća vrijednost svojstva ime objekta predmet, s dodatnim znakom uzvika na kraju. Metoda ne uzima u obzir svojstvo ECTS prilikom formiranja rezultata. Treba promovirati čitanje i razumijevanje koda (potaknuti studente da pažljivo čitaju svaki redak koda i razumiju funkcionalnost svake linije). Treba potaknuti studente da eksperimentiraju s različitim vrijednostima svojstava objekta i provjere rezultate kako bi bolje razumjeli kako metode rade.



Slika 15. Raspodjela odgovora za student OOP-a i PP-a u četrnaestom zadatku

Na prethodnoj slici je vidljivo da je točan odgovor OOP! napisalo malo više od polovice studenata (točnije njih 51.28%).

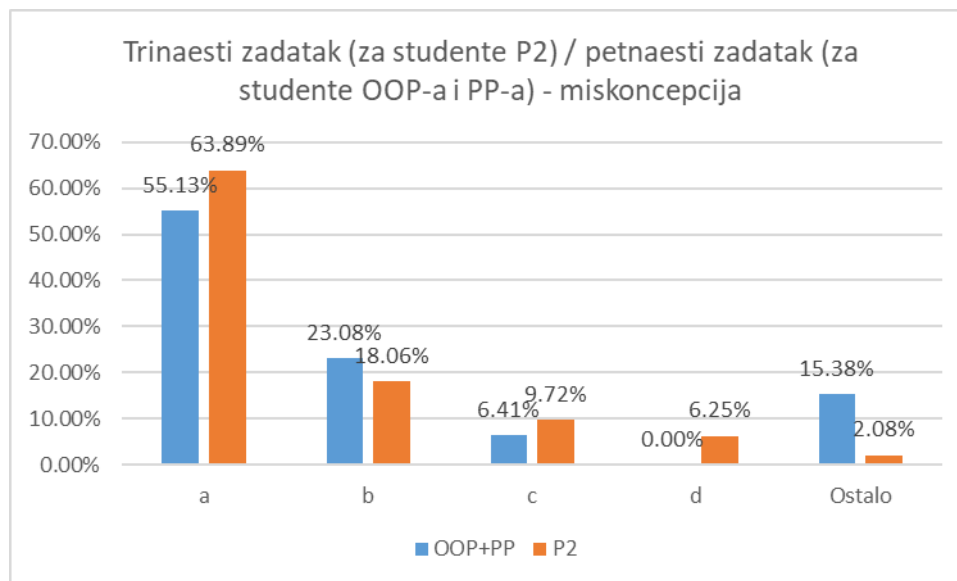
Trinaesti zadatak (za studente P2) / petnaesti zadatak (za studente OOP-a i PP-a)

Još jedan zadatak je služio za ispitivanje niza. Tablica 22 prikazuje zadatak, ponuđene odgovore, odgovor koji se smatra miskoncepcijom i učestalost miskoncepcije za obje grupe studenata.

Tablica 22. Trinaesti zadatak (za studente P2) / petnaesti zadatak (za studente OOP-a i PP-a) -
miskoncepcija

Kod zadatka	Ponudeni odgovori	Miskoncepcija	Učestalost miskoncepcije	
			OOP i PP	P2
<pre>let niz1 = [2,3,4]; let niz2 = niz1; niz2.push(5); console.log(niz1); console.log(niz2);</pre>	<p>a) niz1 će biti [2, 3, 4], a niz2 će biti [2, 3, 4, 5].</p> <p>b) niz2 će biti [2, 3, 4, 5], a niz1 će biti isti kao i niz2.</p> <p>c) niz1 će biti [2, 3, 4], a niz2 će biti isti kao i niz1.</p> <p>d) Program će javiti grešku.</p> <p>e) Other _____</p>	a)	55.13%	63.89%

Odgovor a) se smatra miskoncepcijom jer je taj odgovor označilo kao točan odgovor 55.13% studenata OOP-a i PP-a i 63.89% studenata P2. Ova miskoncepcija proizlazi iz pogrešnog razumijevanja referenciranja. Kada se niz1 dodijeli niz2 (let niz2 = niz1;), oba identifikatora referenciraju na isti niz u memoriji. Promjene koje se dogode na tom nizu odražavaju se na obje varijable jer one dijele istu referencu. Kada se metoda push(5) pozove nad niz2, element 5 se dodaje u taj niz. Budući da i niz1 i niz2 referenciraju na isti niz, promjena će se odraziti i na niz1. Kao rezultat toga, oba ispisivanja console.log(niz1) i console.log(niz2) prikazat će niz [2, 3, 4, 5]. Studentima treba objasniti da kada se jedan niz dodijeli drugoj varijabli, stvara se referenca na taj niz, a ne novi neovisni niz. Također, treba objasniti kako promjene utječu na oba niza (naglasiti da promjene koje se dogode na nizu putem jedne varijable odražavaju se i na druge varijable koje referenciraju na isti niz). Bilo bi poželjno da se izvedu primjeri s različitim nizovima te da se pokaže kako promjene utječu na sve varijable koje referenciraju na taj niz. Također, treba poticati praksu stvaranja kopija niza (kada je potrebno stvoriti neovisnu kopiju niza, treba upotrijebiti metode poput slice() ili concat() kako bi se osiguralo da se promjene na jednoj varijabli ne odraze na druge).



Slika 16. Raspodjela odgovora za student OOP-a i PP-a u petnaestom zadatku i za studente P2 u trinaestom zadatku

Odgovor a) se smatra miskoncepcijom za obje grupe s do sada rekordnom učestalosti, a točan odgovor (b) je u grupi OOP+PP odgovorilo 23.08% studenata, a u grupi P2 18.06% studenata.

Šesnaesti zadatak (za studente OOP-a i PP-a)

Još jedan zadatak je služio za ispitivanje metoda i objekata. Tablica 23 prikazuje zadatak, ponuđene odgovore, odgovore koji se mogu smatrati miskoncepcijama i učestalost miskoncepcija za studente OOP-a i PP-a.

Tablica 23. Šesnaesti zadatak (za studente OOP-a i PP-a) - miskoncepcije

Kod zadatka	Ponuđeni odgovori	Miskoncepcije	Učestalost miskoncepcije	
			OOP i PP	P2
<pre>class Pas{ constructor(v, i, g){ this.vrsta = v; this.ime = i; this.godine = 1; } pas(){</pre>	a) Maltezer Rio 2	a)	20.51%	/
	b) 1			
	c) 2	c)	32.05%	/
	d) Program će javiti grešku.	d)	19.23%	/
	e) Other _____			

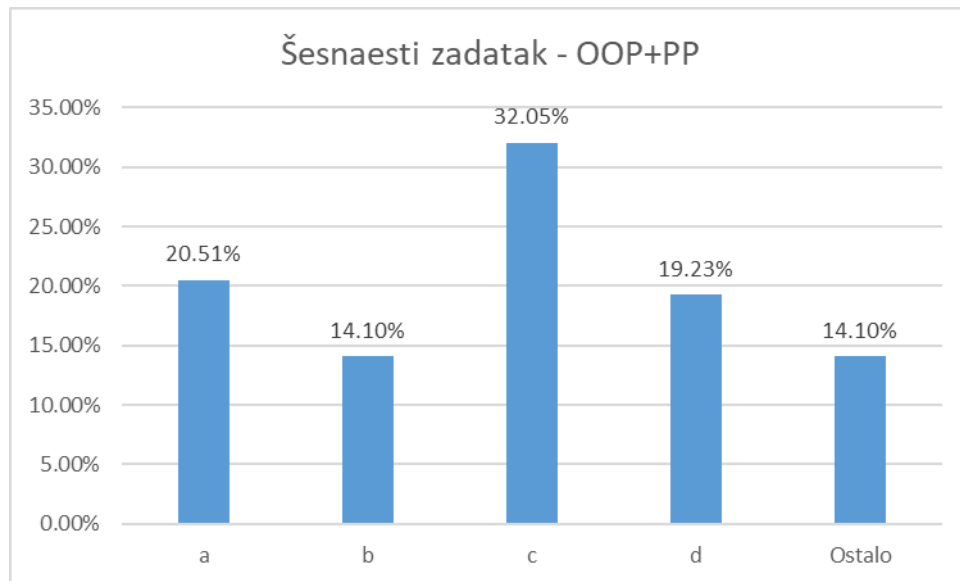
<pre> return this.godine; } } let prvi = new Pas("Maltezer", "Rio", 2); console.log(prvi.pas()); </pre>				
---	--	--	--	--

Odgovor a) je ponudilo 20.51%, odgovor c) je ponudilo 32.05% te odgovor d) je ponudilo 19.23% studenata OOP-a i PP-a (ovaj zadatak nije dan studentima P2). Zašto je trivijalni zadatak koji sadrži samo osnove objektno-orijentiranog programiranja riješilo točno samo 14.10% ?

Prva miskoncepcija: Maltezer Rio 2 → ova miskoncepcija proizlazi iz pretpostavke da će se prilikom ispisa prikazati cijeli objekt s vrijednostima. Metoda pas() (u ovom zadatku) vraća samo vrijednost svojstva godine objekta prvi.

Druga miskoncepcija: 2 → ova miskoncepcija proizlazi iz problema što studenti zanemaruju da su godine u konstruktoru inicijalizirane na 1, a to je povezano s logikom konstruktora klase i načinom postavljanja vrijednosti svojstava objekta. U zadanom kodu, konstruktor klase Pas postavlja svojstvo godine na vrijednost 1. Ovo je statička vrijednost koja će biti postavljena za svaki novi objekt koji se stvori pomoću konstruktora. U primjeru, objekt prvi ima vrijednost godine postavljenu na 1. Dakle, miskoncepcija je povezana s očekivanjem da će vrijednost godine biti 2, a to nije točno jer konstruktor postavio vrijednost na 1. Studenti bi trebali pravilno razumjeti da će se statičke vrijednosti postaviti za sve objekte koji se stvore pomoću konstruktora klase.

Treća miskoncepcija: Program će javiti grešku → ova miskoncepcija proizlazi iz pretpostavke da će kod izazvati grešku, vjerojatno, zbog vrijednosti koja je statički inicijalizirana u konstruktoru, a prilikom stvaranja objekta klase se ta vrijednost pokušava postaviti na 2.



Slika 17. Raspodjela odgovora za student OOP-a i PP-a u šesnaestom zadatku

Iz prethodne slike je vidljivo da su studenti sve tri miskoncepcije (a, c, d) češće označavali kao točan odgovor nego šta su označavali točan odgovor (b) kao točan odgovor.

4. Ograničenja istraživanja

Ograničenja istraživanja su neizbježna i važno ih je identificirati radi boljeg razumijevanja interpretacije rezultata. U ovom istraživanju, nekoliko ključnih ograničenja je prepoznato.

Prvo ograničenje se odnosi na sudionike istraživanja. Svi sudionici su bili s Prirodoslovno-matematičkog fakulteta u Splitu, što može ograničiti opću primjenjivost rezultata na širu populaciju ili druge kontekste izvan fakulteta.

Drugo ograničenje je to što je rješavanje zadataka bilo dobrovoljno pa je to moglo dovesti do manjka motivacije za precizno odgovaranje, a to onda može utjecati na valjanost rezultata.

Treće, korisno bi bilo provesti kvalitativno istraživanje za preciznije razumijevanje uzroka miskonceptija koje u ovom istraživanju nije provedeno (npr. intervjui sa sudionicima).

Na kraju, važno je istaknuti potrebu za dugoročnim praćenjem sudionika radi boljeg razumijevanja dugoročne zadrživosti stečenih znanja i vještina. Ponovno testiranje nakon određenog vremenskog razdoblja moglo bi pružiti uvid u ove aspekte.

Iako su identificirana ograničenja u ovom istraživanju, isto tako su ponuđeni korisni prijedlozi za buduće istraživanje, što može biti korisna osnova za daljnji razvoj u ovom području.

5. Zaključak

Cilj ovog istraživanja je bio odgovoriti na postavljena istraživačka pitanja. Tri istraživačka pitanja odnosila su se na razlike u uspješnosti studenata na testu iz programiranja u odnosu na studijsku grupu, spol i u odnosu na to je li student ponavlja kolegij ili ne. Također, u ovome istraživanju je postavljeno i istraživačko pitanje koje se odnosilo na postojanje miskoncepcije u razumijevanju osnovnih programskih koncepata među studentima.

Istraživanje se provelo tako da su se napravili testovi koji su služili za otkrivanje miskoncepcija među studentima. Testovi su dani studentima PMF-a te su se nakon toga analizirali rezultati korištenjem SPSS alata.

Analizom rezultata došlo se do zaključka da postoje statistički značajne razlike u uspješnosti studenata na testu iz programiranja u odnosu na studijske grupe za sve studente. Također, rezultati istraživanja su pokazala da postoji statistički značajna razlika u uspješnosti studenata na testu iz programiranja u odnosu na spol za studente kolegija P2 (studenti su bolje riješili test od studentica) te su rezultati istraživanja pokazala da postoji statistički značajna razlika u uspješnosti studenata na testu iz programiranja u odnosu na to je li student ponavlja kolegij ili ne za studente kolegija P2 (ponavljači su lošije riješili test od studenata koji ne ponavljaju kolegij).

Na temelju rezultata, jasno je da postoji potreba za dodatnom pažnjom i fokusom na razumijevanje ključnih koncepata programiranja. Miskoncepcije poput nerazumijevanja ključne riječi *const* uz ime varijable, korištenja ugniježdenih petlji, točnosti uvjeta prilikom izvršavanja petlji, netočnog shvaćanja povratnih vrijednosti funkcija i problema s referenciranjem nizova predstavljaju izazove s kojima se studenti susreću.

Ovaj istraživački rad je otvorio put za daljnje istraživanje na temu miskoncepcija u osnovnim (ključnim) konceptima programiranja te na temu razlika između studijskih grupa, spolova i studenata ponavljača i onih koji to nisu. Buduće studije mogu se usredotočiti na ispitivanje dodatnih koncepata programiranja poput polimorfizma i nasljeđivanja te se buduće studije mogu fokusirati na širu populaciju, a ne samo na jedan fakultet.

Uzimajući u obzir tempo tehnološkog napretka i sve veće zahtjeve suvremenog tržišta rada, važno je neprestano usklađivati kurikulum s najnovijim trendovima i tehnologijama i važna je percepcija nastavnika o miskoncepcijama učenika/studenata u programiranju [18].

Zaključno, uspješno prevladavanje miskoncepcija u programiranju ne samo da će omogućiti studentima da postanu dobri programeri, nego će im pružiti i osnovu za daljnje profesionalno usavršavanje i inovativno razmišljanje.

Literatura

- [1] Gomes, A., & Mendes, A. J. (2007, September). Learning to program—difficulties and solutions. In *International Conference on Engineering Education—ICEE* (Vol. 7).
- [2] Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2), 137-172.
- [3] Swidan, A., Hermans, F., & Smit, M. (2018, August). Programming misconceptions for school students. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (pp. 151-159).
- [4] Dehnadi, S. (2010). *A cognitive study of learning to program in introductory programming courses* (Doctoral dissertation, Middlesex University).
- [5] Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *Acm sigcse bulletin*, 37(3), 14-18.
- [6] Johnson, F., McQuistin, S., & O'Donnell, J. (2020, January). Analysis of student misconceptions using Python as an introductory programming language. In *Proceedings of the 4th Conference on Computing Education Practice* (pp. 1-4).
- [7] Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies*, 7, 55-66.
- [8] Bonar, J., & Soloway, E. (1983, January). Uncovering principles of novice programming. In *Proceedings of the 10th ACM SIGACT-SIGPLAN symposium on Principles of programming languages* (pp. 10-13).
- [9] Laura-Ochoa, L., Bedregal-Alpaca, N., & Vidal, E. (2022). Improving computational thinking in nursing students through learning computer programming. *International Journal of Advanced Computer Science and Applications*, 13(5).
- [10] Ma, L., Ferguson, J., Roper, M., & Wood, M. (2011). Investigating and improving the models of programming concepts held by novice programmers. *Computer Science Education*, 21(1), 57-80.
- [11] Kaczmarczyk, L. C., Petrick, E. R., East, J. P., & Herman, G. L. (2010, March). Identifying student misconceptions of programming. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 107-111).
- [12] Kallia, M., & Sentance, S. (2019, February). Learning to use functions: The relationship between misconceptions and self-efficacy. In *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 752-758).
- [13] McKenna, P. (2004). Gender and black boxes in the programming curriculum. *Journal on Educational Resources in Computing (JERIC)*, 4(1), 6-es.

- [14] Mladenović, M., Boljat, I., & Žanko, Ž. (2018). Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Education and Information Technologies*, 23, 1483-1500.
- [15] Perlis, A. J. (1962). *The Computer in the University*.
- [16] Tanielu, T., 'Akau'ola, R., Varoy, E., & Giacaman, N. (2019, May). Combining analogies and virtual reality for active and visual object-oriented programming. In *Proceedings of the acm conference on global computing education* (pp. 92-98).
- [17] Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 18(1), 1-24.
- [18] Qian, Y., Hambrusch, S., Yadav, A., Gretter, S., & Li, Y. (2020). Teachers' perceptions of student misconceptions in introductory programming. *Journal of Educational Computing Research*, 58(2), 364-397.
- [19] Žanko, Ž., Mladenović, M., & Krpan, D. (2022). Analysis of school students' misconceptions about basic programming concepts. *Journal of Computer Assisted Learning*, 38(3), 719-730.
- [20] Žanko, Ž., Mladenović, M., & Krpan, D. (2023). Mediated transfer: impact on programming misconceptions. *Journal of Computers in Education*, 10(1), 1-26.
- [21] Žanko, Ž., Mladenović, M., & Boljat, I. (2019). Misconceptions about variables at the K-12 level. *Education and Information Technologies*, 24, 1251-1268.

Popis slika

Slika 1. Indeks težine – OOP i PP	10
Slika 2. Indeks težine – P2	11
Slika 3. Usporedba indeksa težine za zajedničke zadatke.....	11
Slika 4. Razdioba sudionika u odnosu na spol i kolegij.....	12
Slika 5. Raspodjela odgovora kod studenata – prvi zadatak	19
Slika 6. Raspodjela odgovora za studente OOP-a i PP-a u drugom zadatku	21
Slika 7. Raspodjela odgovora za studente P2 u četvrtom zadatku	23
Slika 8. Raspodjela odgovora za studente P2 u petom zadatku	25
Slika 9. Raspodjela odgovora za studente OOP-a i PP-a u sedmom zadatku	27
Slika 10. Raspodjela odgovora za studente P2 u osmom zadatku	28
Slika 11. Raspodjela odgovora za studente OOP-a i PP-a u desetom zadatku i za studente P2 u devetom zadatku	30
Slika 12. Raspodjela odgovora za studente OOP-a i PP-a u jedanaestom zadatku i za studente P2 u desetom zadatku.....	31
Slika 13. Raspodjela odgovora za studente OOP-a i PP-a u dvanaestom zadatku i za studente P2 u jedanaestom zadatku	33
Slika 14. Raspodjela odgovora za studente OOP-a i PP-a trinaestom zadatku i za studente P2 u dvanaestom zadatku	35
Slika 15. Raspodjela odgovora za student OOP-a i PP-a u četrnaestom zadatku	36
Slika 16. Raspodjela odgovora za student OOP-a i PP-a u petnaestom zadatku i za studente P2 u trinaestom zadatku	38
Slika 17. Raspodjela odgovora za student OOP-a i PP-a u šesnaestom zadatku	40

Popis tablica

Tablica 1. <i>JavaScript vs. Python</i>	8
Tablica 2. Osnovne metrijske vrijednosti testova.....	9
Tablica 3. Deskriptivna statistika	14
Tablica 4. Deskriptivna statistika	15
Tablica 5. Deskriptivna statistika	16
Tablica 6. Prvi zadatak (za sve studente) - miskoncepcija.....	17
Tablica 7. Mogući mentalni model studenata za navedenu miskoncepciju	18
Tablica 8. Drugi zadatak (za studente OOP-a i PP-a) - miskoncepcije	19
Tablica 9. Treći zadatak (za studente P2) / četvrti zadatak (za studente OOP-a i PP-a) - miskoncepcija	21
Tablica 10. Četvrti zadatak (za studente P2) - kod zadatka	22
Tablica 11. Četvrti zadatak (za studente P2) - miskoncepcija	22
Tablica 12. Peti zadatak (za studente P2) – kod zadatka	23
Tablica 13. Peti zadatak (za studente P2) – miskoncepcija.....	24
Tablica 14. Šesti zadatak (za studente P2) / sedmi zadatak (za studente OOP-a i PP-a) – kod zadatka.....	25
Tablica 15. Šesti zadatak (za studente P2) / sedmi zadatak (za studente OOP-a i PP-a) – miskoncepcija	26
Tablica 16. Osmi zadatak (za studente P2) - miskoncepcija	27
Tablica 17. Deveti zadatak (za studente P2) / deseti zadatak (za studente OOP-a i PP-a) - miskoncepcije	29
Table 18. Deseti zadatak (za studente P2) / jedanaesti zadatak (za studente OOP-a i PP-a) – miskoncepcije	30
Tablica 19. Jedanaesti zadatak (za studente P2) / dvanaesti zadatak (za studente OOP-a i PP-a) - miskoncepcija	32

Tablica 20. Dvanaesti zadatak (za studente P2) / trinaesti zadatak (za studente OOP-a i PP-a) - miskoncepcije	33
Tablica 21. Četrnaesti zadatak - miskoncepcija	35
Tablica 22. Trinaesti zadatak (za studente P2) / petnaesti zadatak (za studente OOP-a i PP-a) - miskoncepcija	37
Tablica 23. Šesnaesti zadatak (za studente OOP-a i PP-a) - miskoncepcije	38

Prilozi

1. Zadaci koji su dani studentima Objektno-orijentiranog programiranja i studentima Programskih paradigmi s ponuđenim odgovorima.

1. Što će sljedeći program ispisati?

```
let najmanji = 3;
let najveći = 1;
let broj = 2;
najmanji = najveći;
najveći = najmanji - broj;
console.log(najmanji, najveći);
```

- a) -1 1
- b) 1 -1
- c) 1 1
- d) 1 3
- e) 3 1

2. Što će sljedeći program ispisati?

```
const predmet = "Programiranje";
var ECTS = 4;
predmet = "Objektno-orijentirano programiranje";
ECTS = 6;
console.log("Predmet je", predmet, "te vrijedi", ECTS, "ECTS boda.");
```

- a) Predmet je Programiranje te vrijedi 4 ECTS boda.
- b) Predmet je Objektno-orijentirano programiranje te vrijedi 6 ECTS boda.
- c) Predmet je Programiranje te vrijedi 6 ECTS boda.
- d) Predmet je Objektno-orijentirano programiranje te vrijedi 4 ECTS boda.
- e) Program će javiti grešku pa se iz tog razloga ništa neće ispisati.

3. Što će sljedeći program ispisati?

```
var zbroj = 1;
zbroj = 2;
zbroj = 3;
zbroj = 4;
console.log(zbroj);
```

- a) 10
- b) 4
- c) 3
- d) 2
- Other

4. Što će sljedeći program ispisati?

```
var br2 = 1;
let br1 = 2;
br2 = br1 + 3;
br1 = br2 + 2;
console.log(br2, 'br1');
```

- a) 1 2
- b) 2 1
- c) 6 3
- d) 3 5
- Other

5. Što će sljedeći program ispisati?

```
var brzina = 60;
var policija = 'policija';
if (brzina > 60) {
  policija = ("Prebrza vožnja!");
}
console.log(policija);
```

a) brzina

b) 60

c) policija

d) Prebrza vožnja!

Other

6. Što će sljedeći program ispisati?

```
var brzina = 60;
var policija = 'policija';
if (brzina >= 60) {
  brzina = 55;
  policija = "Prebrza vožnja!";
} else {
  policija = "Vozite unutar ograničenja, bravo!";
}
console.log(policija);
```

a) 55

b) policija

c) Vozite unutar ograničenja, bravo!

d) Prebrza vožnja!

Other

7. Što će sljedeći program ispisati?

```
var broj = 20;
var poruka = "";
if (broj < 100) {
  poruka = poruka + "Manji od 20!";
}if (broj < 50) {
  poruka = poruka + "Manji od 50!";
}if (broj === 20) {
  poruka = poruka + "20!";
}
console.log(poruka);
```

a) Manji od 20!

b) Manji od 50!

c) 20!

d) Manji od 20!Manji od 50!20!

e) Manji od 20! Manji od 50! 20!

8. Što će sljedeći program ispisati?

```
var broj = 70;
if (broj <= 100) {
  broj = 10;
}else if (broj < 50) {
  broj = 5;
}else {
  broj = 2;
}
console.log(broj);
```

a) 70

b) 2

c) 10

d) 5

Other

9. Koje brojeve će ispisati sljedeći program?

```
for (var i = 1; i<=3; i++) {  
  console.log(i + 1);  
  i = i + 1;  
  console.log(i - 1);  
}
```

- a) Brojeve: 2 3 4 1 3
- b) Brojeve: 2 4 0 1 2
- c) Brojeve: 1 2 3 4
- d) Brojeve: 2 1 4 3
- e) Brojeve: 2 0 4 2

10. Koje brojeve će ispisati sljedeći program?

```
var a = 1;  
var b = 2;  
for (var i = 0; i<2; i++) {  
  console.log(b);  
  for (var j = 1; j<=2; j++) {  
    console.log(a);  
  }  
}
```

- a) Brojeve: 1 1 2 2
- b) Brojeve: 1 1 1 1 2 2
- c) Brojeve: 2 1 2 1
- d) Brojeve: 2 2 1 1
- e) Brojeve: 2 1 1 2 1 1

11. Što će sljedeći program ispisati?

```
i = 1;  
while (i <= 5) {  
  console.log('i');  
};
```

- a) Program će ispisati brojeve 1 2 3 4 5.
- b) Program će ispisati pet puta slovo i.
- c) i
- d) Program neće ništa ispisati.
- Other

12. Što će sljedeći program ispisati?

```
i = 5;  
while (i < 5) {  
  console.log(i);  
}
```

- a) 5
- b) Program će ispisati broj 5 beskonačan broj puta.
- c) i
- d) Program će ispisati slovo i beskonačan broj puta.
- e) Program neće ništa ispisati.

13. Što će sljedeći program ispisati?

```
let niz = [0, 5, 1, 3];  
let broj = 2;  
niz.push(broj);  
niz.push(4);  
let niz1 = [6, 7];  
let završni = niz.concat(niz1);  
console.log(završni);
```

a) [0, 1, 2, 3, 4, 5, 6, 7]

b) [0, 5, 1, 3, 2, 4]

c) [0, 5, 1, 3]

d) [6, 7]

Other

14. Što će sljedeći program ispisati?

```
let predmet = {  
  ime: "OOP",  
  ECTS: 6,  
  ispis() {  
    return this.ime + "!";  
  }  
};  
console.log(predmet.ispis());
```

a) OOP

b) OOP 6

c) OOP 6!

d) 6!

Other

15. Kako će izgledati niz1 i niz2 nakon što se program izvrši?

```
let niz1 = [2,3,4];  
let niz2 = niz1;  
niz2.push(5);  
console.log(niz1);  
console.log(niz2);
```

a) niz1 će biti [2, 3, 4], a niz2 će biti [2, 3, 4, 5].

b) niz2 će biti [2, 3, 4, 5], a niz1 će biti isti kao i niz2.

c) niz1 će biti [2, 3, 4], a niz2 će biti isti kao i niz1.

d) Program će javiti grešku.

Other

16. Što će sljedeći program ispisati?

```
class Pas{  
  constructor(v, i, g){  
    this.vrsta = v;  
    this.ime = i;  
    this.godine = 1;  
  }  
  pas(){  
    return this.godine;  
  }  
}  
let prvi = new Pas("Maltezer", "Rio", 2);  
console.log(prvi.pas());
```

a) Maltezer Rio 2

b) 1

c) 2

d) Program će javiti grešku.

Other

2. Zadaci koji su dani studentima Programiranje 2 s ponuđenim odgovorima.

1. Što će sljedeći program ispisati?

```
najmanji = 3
najveci = 1
broj = 2
najmanji = najveci
najveci = najmanji - broj
print(najmanji, najveci)
```

- a) -1 1
- b) 1 -1
- c) 1 1
- d) 1 3
- e) 3 1

2. Što će sljedeći program ispisati?

```
zbroj = 1
zbroj = 2
zbroj = 3
zbroj = 4
print(zbroj)
```

- a) 10
- b) 4
- c) 3
- d) 2
- Other

3. Što će sljedeći program ispisati?

```
br2 = 1
br1 = 2
br2 = br1 + 3
br1 = br2 + 2
print(br2, 'br1')
```

- a) 1 2
- b) 2 1
- c) 6 3
- d) 3 5
- Other

4. Što će sljedeći program ispisati?

```
brzina = 60
policija = 'policija'
if (brzina > 60):
    policija = "Prebrza vožnja!"
print(policija)
```

- a) brzina
- b) 60
- c) policija
- d) Prebrza vožnja!
- Other

5. Što će sljedeći program ispisati?

```
brzina = 60
policija = 'policija'
if (brzina >= 60):
    brzina = 55
    policija = "Prebrza vožnja!"
else:
    policija = "Vozite unutar ograničenja, bravo!"
print(policija)
```

- a) 55
- b) policija
- c) Vozite unutar ograničenja, bravo!
- d) Prebrza vožnja!
- Other

6. Što će sljedeći program ispisati?

```
broj = 20
poruka = ""
if (broj < 100):
    poruka = poruka + "Manji od 20!"
if (broj < 50):
    poruka = poruka + "Manji od 50!"
if (broj == 20):
    poruka = poruka + "20!"
print(poruka)
```

- a) Manji od 20!
- b) Manji od 50!
- c) 20!
- d) Manji od 20!Manji od 50!20!
- e) Manji od 20! Manji od 50! 20!

7. Što će sljedeći program ispisati?

```
broj = 70
if (broj <= 100):
    broj = 10
elif (broj < 50):
    broj = 5
else:
    broj = 2
print(broj)
```

- a) 70
- b) 2
- c) 10
- d) 5
- Other

8. Koje brojeve će ispisati sljedeći program?

```
i = 1
for i in range(i, 4, i+1):
    print(i + 1)
    i = i + 1
    print(i - 1)
```

- a) Brojeve: 2 3 4 1 3
- b) Brojeve: 2 4 0 1 2
- c) Brojeve: 1 2 3 4
- d) Brojeve: 2 1 4 3
- e) Brojeve: 2 0 4 2

9. Koje brojeve će ispisati sljedeći program?

```
a = 1
b = 2
for i in range(2):
    print(b)
    for j in range(1, 3):
        print(a)
```

- a) Brojeve: 1 1 2 2
- b) Brojeve: 1 1 1 1 2 2
- c) Brojeve: 2 1 2 1
- d) Brojeve: 2 2 1 1
- e) Brojeve: 2 1 1 2 1 1

10. Što će sljedeći program ispisati?

```
i = 1
while (i <= 5):
    print('i')
```

- b) Program će ispisati pet puta slovo i.
- a) Program će ispisati brojeve 1 2 3 4 5.
- c) i
- d) Program neće ništa ispisati.
- Other

11. Što će sljedeći program ispisati?

```
i = 5
while (i < 5):
    print(i)
```

- a) 5
- b) Program će ispisati broj 5 beskonačan broj puta.
- c) i
- d) Program će ispisati slovo i beskonačan broj puta.
- e) Program neće ništa ispisati.

12. Što će sljedeći program ispisati?

```
niz = [0, 5, 1, 3]
broj = 2
niz.append(broj)
niz.append(4)
niz1 = [6, 7]
zavrzni = niz + niz1
print(zavrzni)
```

- a) [0, 1, 2, 3, 4, 5, 6, 7]
- b) [0, 5, 1, 3, 2, 4]
- c) [0, 5, 1, 3]
- d) [6, 7]
- Other

13. Kako će izgledati niz1 i niz2 nakon što se program izvrši?

```
niz1 = [2,3,4]
niz2 = niz1
niz2.append(5)
print(niz1)
print(niz2)
```

- a) niz1 će biti [2, 3, 4], a niz2 će biti [2, 3, 4, 5].
- b) niz2 će biti [2, 3, 4, 5], a niz1 će biti isti kao i niz2.
- c) niz1 će biti [2, 3, 4], a niz2 će biti isti kao i niz1.
- d) Program će javiti grešku.
- Other