

# Koncepti praga u nastavi informatike

---

**Jakovac, Gabriјela**

**Master's thesis / Diplomski rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka / Sveučilište u Rijeci**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:195:477317>

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

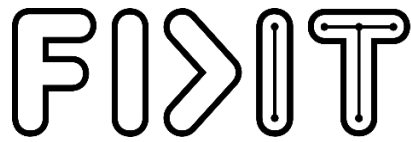
*Download date / Datum preuzimanja:* **2024-10-19**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)





Sveučilište u Rijeci

**Fakultet informatike  
i digitalnih tehnologija**

Sveučilišni diplomski studij Informatike - nastavnički smjer

Gabrijela Jakovac

# Koncepti praga u nastavi informatike

Diplomski rad

Mentor: Izv. prof. dr. sc. Martina Holenko Dlab

Rijeka, srpanj 2024.

Rijeka, 27. 5. 2024.

## Zadatak za diplomski rad

Pristupnik/ica: Gabrijela Jakovac

Naziv diplomskog rada: Koncepti praga u nastavi informatike

Naziv diplomskog rada na eng. jeziku: Threshold concepts in computer science teaching

Sadržaj zadatka: U radu je potrebno predstaviti karakteristike koncepata praga te navesti neke primjere. Potrebno je provesti istraživanje s ciljem identifikacije koncepata praga u području informatike temeljem stavova osnovnoškolskih i srednjoškolskih nastavnika. Također, potrebno je predložiti vlastite aktivnosti koje uključuju igrifikaciju ili obrazovne igre kojima se omogućuje usvajanje nekih koncepata praga.

Mentor/ica:  
Izv. prof. dr. sc. Martina Holenko Dlab

Voditeljica za diplomske radove:  
Doc. dr. sc. Lucija Načinović Prskalo

Komentor/ica:

Zadatak preuzet: 27. 5. 2024.

(potpis pristupnika/ce)

## **Sažetak**

Ovaj rad obuhvaća istraživanje koncepata praga u nastavi informatike pri čemu se koncept praga odnosi na ključne pojmove u obrazovanju koji su transformativni kada ih učenik razumije, dok su prije tog trenutka teški i nepoznati. Koncepti praga u nastavi informatike nisu dovoljno istraženi ni definirani, te o toj temi postoji malo dostupnih podataka. Definiranjem istih olakšati će se proces učenja i razumijevanja propisanih nastavnih sadržaja.

Cilj istraživanja bio je identificirati koncepte iz područja informatike temeljem stavova osnovnoškolskih i srednjoškolskih nastavnika. Kao temeljna metodologija istraživanja odabrana je tehnika nominalne grupe. Istraživanje je provedeno u OŠ Vežica u Rijeci, kao dio aktivnosti Županijskog stručnog vijeća učitelja informatike u osnovnim školama Primorsko-goranske županije, te na Fakultetu informatike i digitalnih tehnologija u Rijeci, kao dio aktivnosti Županijskog stručnog vijeća učitelja informatike u srednjim školama. Nakon detaljne analize ispitanika i analiza provedenih anketa u radu su prikazani rezultati. Identificirani su koncepti praga u nastavi informatike. Također, nakon analiziranja dobivenih rezultata u rad je priložen plan nastavnog sata sa aktivnostima za lakše savladavanje pojedinih koncepata praga temeljenih na elementima igrifikacije odnosno korištenja edukativnih igara. Rad naglašava izuzetnu važnost definiranja koncepata praga u nastavi informatike kako bi se unaprijedilo dosadašnje obrazovanje te olakšalo učenicima sam proces učenja i shvaćanja nastavnog sadržaja.

**Ključne riječi:** koncept praga; tehnika nominalne grupe; edukativne igre nastavne aktivnosti

## SADRŽAJ

|        |   |    |
|--------|---|----|
| 1.     | Uvod.....   | 1  |
| 2.     | Koncepti praga u obrazovanju.....   | 2  |
| 2.1.   | Karakteristike koncepta praga .....   | 4  |
| 3.     | Koncepti praga u matematici i informatici.....                                      | 6  |
| 3.1.   | Istraživanje koncepta praga u informatici .....                                     | 7  |
| 3.1.1. | Objektno orijentirano programiranje .....   | 7  |
| 3.1.2. | Pokazivači .....  | 8  |
| 3.2.   | Koncepti praga u matematici.....  | 9  |
| 3.2.1. | Limes.....  | 9  |
| 3.2.2. | Funkcije.....   | 10 |
| 4.     | Istraživanje stavova o konceptima praga .....                                       | 11 |
| 4.1.   | Metodologija.....   | 11 |
| 4.1.1. | Primjena tehnike nominalne grupe za identifikaciju koncepata praga .....            | 12 |
| 4.2.   | Ispitanici .....  | 13 |
| 4.2.1. | Podaci o ispitanicima u osnovnim školama .....                                      | 13 |
| 4.2.2. | Podaci o ispitanicima u srednjim školama.....                                       | 15 |
| 4.3.   | Rezultati istraživanja .....  | 18 |
| 4.3.1. | Rezultati provedeni nad ispitanicima zaposlenima u OŠ.....                          | 18 |
| 4.3.2. | Rezultati provedeni nad ispitanicima zaposlenim u SŠ .....                          | 21 |
| 4.4.   | Završni pregled rezultata istraživanja.....   | 24 |
| 5.     | Nastavne aktivnosti za usvajanje koncepata praga .....                              | 25 |
| 5.1.   | Obrazovne igre .....  | 25 |
| 5.2.   | Pristup temeljen na igri.....   | 26 |
| 5.3.   | Programski jezik Scratch.....   | 26 |
| 5.4.   | Primjer aktivnosti za lakše savladavanje koncepta praga - logički uvjeti .....      | 28 |
| 5.5.   | Primjer aktivnosti za lakše savladavanje koncepta praga - naredbe ponavljanja ..... | 37 |
| 6.     | Zaključak.....  | 46 |
|        | Literatura .....  | 47 |
|        | Popis tablica .....   | 49 |
|        | Popis slika .....   | 50 |
|        | Popis priloga.....  | 51 |

# 1. Uvod

U digitalnom društvu današnjice u kojem smo odrasli i njegovi smo aktivni članovi, jasno možemo doći do zaključka kako integracija informacijske tehnologije unutar obrazovnog sustava postaje sve značajnija (Bognar, 2016).

U kontekstu nastave informatike, baš kao i kod ostalih nastavnih predmeta, suočavamo se s izazovom postizanja određenih ciljeva učenja. Ključno pitanje koje se postavlja je kako učenici mogu preći prag razumijevanja i uspješno savladati temeljne informatičke koncepte.

Kao buduću nastavnicu informatike uvijek me je zanimalo kako su temeljni koncepti međusobno povezani i kako učenici mogu postići njihovo razumijevanje. Upravo radi toga sam pokazala veliki interes za pojam koncepta praga te sam iz istog razloga odabrala ovu temu za svoj diplomski rad. Smatram da će istraživanje u ovom radu uvelike pomoći budućim nastavnicima, a onima s više radnog iskustva zasigurno otkloniti poneke nedoumice.

Ciljevi istraživanja usmjereni su na razumijevanje percepcije i primjene koncepta praga u nastavi informatike u osnovnim i srednjim školama. Istraživanje ima za cilj identificirati zahtjevne koncepte koje nastavnici i učenici percipiraju kao izazovne te među njima identificirati same koncepte praga.

Ovaj diplomski rad usmjerava se na koncepte praga u nastavi informatike, istražujući njihove karakteristike, a ujedno i ključne prepreke s kojima se učenici suočavaju pri stjecanju znanja i vještina.

Kroz ovaj rad i istraživanje prikazat će se karakteristike koncepata koji utječu na postizanje pragova razumijevanja u nastavnom predmetu informatike te definirati sami pragovi za određene razrede. Cilj rada je ne samo razjasniti koncept praga, već i ponuditi preporuke za unapređenje nastavnih metoda s ciljem efikasnijeg savladavanja informatičkih sadržaja.

U radu ćemo razmotriti strukturu ključnih pojmova povezanih s konceptom praga, analizirati dosadašnja istraživanja u području informatičke nastave, te prezentirati rezultate vlastitog istraživanja s učiteljima i nastavnicima na području naše županije. Ovim istraživanjima nastojat ćemo doprijeti boljem razumijevanju koncepata praga u području informatike s ciljem poboljšanja kvalitete nastave u ovom ključnom području obrazovanja.

## 2. Koncepti praga u obrazovanju

U kontekstu područja informatike, koncepti praga igraju značajnu ulogu označivača granica područja, pružajući organizacijsko načelo za kurikulum.

Kurikulum se smatra završnim proizvodom procesa razvijanja obrazovnih programa, predstavljajući temeljni okvir za strukturiranje i izvođenje nastavnih jedinica. Ovaj plan obuhvaća jasne izjave o postavljenim ciljevima učenja, organizaciji i nadzoru procesa učenja, pružajući time učiteljima i učenicima sredstva za postizanje optimalnih rezultata u učenju (Klafki, Schulz, von Cube, Moller, Winkel i Blankertz, 1992).

Različite teorije učenja nude svoja gledišta o tome kako ljudi stječu nova znanja, odnosno kako ljudski mozak integrira nove informacije. Postoje razlike u stajalištima među znanstvenicima, a neke teorije su prihvaćene ili odbačene u određenim razdobljima ili od strane određenih znanstvenih zajednica.

U 20. stoljeću bihevioristička teorija dominirala je psihologijom, naglašavajući promjene u ponašanju kao pokazatelj usvojenog znanja. Praktične metode, poput nagrađivanja ili kažnjavanja, smatrale su se ključnima za poticanje učenja. S vremenom, popularnost biheviorizma opadala je, ustupajući mjesto kognitivizmu, koji tretira učenje kao proces u mozgu uz pomoć kojega sva novo usvojena znanja integriraju s onim koje smo već imali. Europski obrazovni sustavi često slijede kognitivistički model, fokusirajući se na apstraktno teorijsko znanje (Bognar, 2016).

Usprkos promjenama u teorijskim pristupima u psihologiji učenja, Britanska prosvjetna inspekcija se 1988. godine usredotočila se na dva nastavna segmenta, planiranje i pripreme nastave. Shodno navedenim segmentima postavljaju se i dva pitanja. Prvo pitanje je je li razumljiva svrha ovog nastavnog sata, te drugo, da li se vodilo dovoljno računa o zadovoljstvu potreba učenika. Prvo pitanje se odnosi na jasnoću ostvarenja pedagoških ciljeva, dok se drugim pitanje preispitivalo koliko su pedagoški ciljevi vodili računa o samoj sposobnosti učenika, odnosno njihovom predznanju koje označava prekretnicu u napretku ka budućim pedagoškim postignućima (Kyriacou, 1991).

Meyer i Land (2003) su identificirali koncept praga u području obrazovanja kao skup pojmova koji postaju transformirajući jednom kada ih pojedinac razumije, ali su prije toga teški i nepoznati. Bez obzira na to jesmo li orijentirani prema konstruktivističkom pristupu ili nekoj drugoj teoriji učenja, koncepti praga predstavljaju točke gdje će učenici vjerojatno naići na poteškoće u učenju. Dodatno definirajući koncepte praga, Meyer i Land navode da su oni:

1. *integrativni* jer otkrivaju prethodno nepoznat način povezivanja pojmova,
2. *nepovratni* jer kada ih učenik stvarno shvati taj novi način razmišljanja postaje dio učenika i
3. *označivači područja* jer određuju granicu dijela (ili čitavog) skupa pojmova. Cijela disciplina može imati svoju granicu na kojoj se može postaviti pojedini koncept praga čije savladavanje ukazuje na vladanje tom disciplinom.

Stoga, jedna od ključnih prepreka koju učenik može susresti u razvijanju potrebne vizije ili motivacije za istraživanje novog područja znanja može biti odbijanje nelogičnog, ali transformirajućeg koncepta. Naprimjer, prije nego što prihvate potencijalne koristi i inovativne ideje ili prakse učenja i poučavanja, pojedinci koji se obrazuju u području informatike moraju prihvatiti korisnost tog područja. Model vježbeništva, koji se koristi za razvoj novih učitelja u visokom obrazovanju, može biti teže prihvatiti nego što se čini na prvi pogled. Navedeni model ima za cilj razviti vještine i kompetencije budućih učitelja pružajući im priliku da primijene teoretsko znanje u stvarnom okruženju nastave. Ova integracija teži istaknuti važnost modela vježbeništva u stvaranju osnovnih vještina edukatora, čineći prihvaćanje tog koncepta još ključnijim u procesu obrazovanja (Meyer i Land, 2003).

Profesorica Katie King je na jednostavan način objasnila koncept praga, koji su razvili Land i Meyer. Po toj ideji, discipline ne posjeduju svoje bitno središte, već sadrže kompleksne ideje na svojim rubovima. Ove ideje mogu predstavljati izazov za one koji su novi u toj disciplini, no kada ih jednom shvate, otvara se potpuno novi način gledanja na tu disciplinu. Osim što obuhvaćaju temeljna znanja, ove ideje potiču specifične načine razmišljanja koji su karakteristični za tu disciplinu.

Profesorica je podijelila primjer suradnje s profesorom fotografije kako bi ilustrirala ove koncepte. Fotograf je izrazio zabrinutost jer studenti nisu razumjeli kako vrednuje njihove radove. Nisu shvaćali njegov pristup ocjenjivanju jer im nikada nije prenio ideju niti im objasnio na adekvatan način da je umjetnost u suštini rješavanje problema. Iako je to za njega bilo temeljno razmišljanje, studenti to nisu shvatili dok im nije izrijeком prenio tu ideju. Ovaj primjer jasno pokazuje kako prelazak preko Koncepta praga može biti jednostavan, ali transformirajući trenutak u razumijevanju.

Koncept praga ne svodi se samo na stjecanje temeljnog znanja, već i na poticanje specifičnih načina razmišljanja koji su svojstveni određenoj disciplini. Profesorica King naglašava važnost dijeljenja ključnih ideja s studentima kako bi im omogućila prelazak preko ovog praga i potpuno razumijevanje discipline koju studiraju (CELatElon, 2013).

Nedavno je u domeni visokog obrazovanja, posebno u kontekstu inženjerskog obrazovanja, primjetno sve učestalije korištenje koncepta praga. Koncepte praga možemo kategorizirati u dvije ključne grupe. Prva grupa pruža model ili okvir za kontekstualizaciju nastave učenika, integrirajući ih u širu društvenu zajednicu prakse unutar određene discipline. U svojoj suštini, koncepti praga predstavljaju neophodno promjenljive entitete. Kada učenik stekne razumijevanje koncepta praga, počinje razmišljati i djelovati više kao praktičar u određenom polju, naprimjer, počinje „razmišljati više kao informatičar“ (Roundtree & Roundtree, 2009). Shodno tome, usvajanje koncepta praga smatra se neophodnim iskustvom koje učenici moraju proći kako bi bili „prihvaćeni“ u zajednicu prakse.

Drugi pristup konceptima praga fokusira se na individualne koncepte, opisujući ih kao „barijere“ koje mogu ograničiti učenike u razvoju dubokog razumijevanja, sprečavajući ih da primijene svoje znanje izvan trenutnog konteksta (Blackmore, 2010).



## 2.1. Karakteristike koncepta praga

Dr. Viktorija navodi kako postoji više karakteristika koncepta praga te izdvaja pet temeljnih karakteristika: (SJSU School of Information, 2013).

- Transformativno
- Nepovratno
- Integrativno
- Problematično
- Omeđeno

Transformativnost se opisuje i povezuje s događajem koji nas je obilježio i nikada se neće zaboraviti, događajem koji nam je promijenio život. Primjer toga bilo bi polaganje vozačkog ispita.

Transformacija stava, vrijednosti ili razumijevanja često predstavlja ključnu točku u našem životu. Ova promjena ne samo da oblikuje naš identitet, već i duboko utječe na našu svakodnevicu. Kroz ovu evoluciju, novo razumijevanje postupno se asimilira u našu biografiju, postajući neizostavan dio onoga što jesmo. Ovaj proces nije trenutačan, već se odvija postepeno, prožimajući sve aspekte našeg života.

Nije samo puko prihvaćanje novog stava ili vrijednosti; umjesto toga, ta transformacija postaje dio našeg unutarnjeg bića, oblikujući način na koji percipiramo svijet oko sebe. Kroz ovu integraciju, novo razumijevanje postaje temeljni element našeg identiteta, utječući na naše misli, emocije i djelovanje. Ova promjena ne događa se izolirano; ona ima duboke implikacije na naše odnose, naš rad i naše životne odluke.

Važnost ovakvih transformacija leži i u tome što nam omogućavaju da rastemo i razvijamo se kao individue. Ona nas potiče da sagledavamo stvari iz različitih perspektiva, pružajući nam širu sliku života. Također, ove promjene često idu ruku pod ruku s osobnim rastom, čineći nas snažnijima i otpornijima pred izazovima života.

Nepovratnost, u kontekstu znanja, predstavlja dubok stupanj učenja gdje ono što smo jednom savladali postaje trajni dio našeg intelektualnog repertoara. To je kao vožnja bicikla ili plivanje, gdje jednom usvojena vještina postaje inherentna i neizbrisiva. Kroz proces nepovratnosti, znanje postaje ukorijenjeno u našem pamćenju na način koji ne podliježe zaboravu, čak i u izazovnim situacijama.

Ovaj fenomen možemo usporediti s vožnjom bicikla. Nakon što savladamo tehniku vožnje, to postaje dio našeg mišićnog pamćenja. Bez obzira koliko dugo nismo vozili bicikl, kada ponovno sjednemo na njega, taj proces vožnje automatski izranja iz naše podsvijesti. Slično tome, nepovratnost u učenju znači da jednom stečeno znanje postaje trajna vještina koja se aktivira bez obzira na dulji period nekorisćenja.

Integrativnost u kontekstu učenja, označava otvaranje nečega što je prije možda bilo skriveno ili nije bilo shvaćeno u svojoj punoj povezanosti. Ova karakteristika učenja ima snagu povezivanja odvojenih koncepata, što rezultira njihovim ujedinjenjem u cjelovitom razumijevanju. Ideje koje

su nekada bile odvojene sada postaju povezane, stvarajući šire razumijevanje koje obogaćuje perspektive pojedinaca.

Ovaj proces integracije možemo usporediti s povezivanjem različitih dijelova slagalice. Svaki pojedini komad može predstavljati određeni koncept, a kroz integraciju, ti pojedinačni dijelovi postaju sastavni dio većeg, potpunog slijetanja. Također, integrativnost nije samo spajanje različitih koncepata; ona ide korak dalje stvarajući prošireno razumijevanje koje obogaćuje naše percepcije svijeta oko nas.

Ova dimenzija integrativnosti ima značajan utjecaj na evoluciju razumijevanja pojedinca. Ideje koje su možda bile izolirane sada postaju dijelom šire mreže povezanosti, što rezultira bogatijim i dubljim doživljajem znanja.

Problematičnost u kontekstu učenja možemo povezati i opisati sa određenim konceptima koji mogu djelovati kontraintuitivno ili neugodno. No, ključno je suočiti se s tim izazovima kako bismo ih razumjeli. Često su ti koncepti povezani s situacijama koje izazivaju nelagodu ili su kontraintuitivne, a ta nelagoda može proizlaziti iz netočnih predrasuda. Posebno kada je riječ o rješavanju problema u fizici, početnici su često suočeni s različitim zabudama i kontradikcijama.

Međutim, kroz predanost u rješavanju tih izazova, oni ne samo da prevladavaju predrasude, već dosežu i nove razine razumijevanja. Hrvanje s kontraintuitivnim pojmovima postaje put prema dubljem razumijevanju, a taj proces omogućuje pojedincima da dosegnu nove visine u svojem učenju, eliminirajući pri tom nivo podcjenjivanja.

Omeđenost obilježava prisustvo terminalnih granica. Ove granice služe kao prijelaz između različitih konceptualnih područja, definirajući granice s drugim pragovima i uvodeći nas u nova područja razumijevanja. U specifičnom predmetnom području, specijalizirana terminologija dobiva novo značenje definirano upravo tim granicama.

Omeđenost označava ne samo prisutnost krajnjih točaka, već i mogućnost istraživanja i proširivanja tih granica kako bismo produbili svoje razumijevanje i otkrili nova značenja koja se pojavljuju unutar tih definiranih granica (SJSU School of Information, 2013).

### 3. Koncepti praga u matematici i informatici

U svijetu obrazovanja, koristimo smjernice koje nam pomažu razumjeti kako učimo i podučavamo u određenom predmetu. Razmišljamo o informatičkoj znanosti kao o posebnom predmetu. Ako je informatička znanost sama po sebi koncept praga, tada, po definiciji, sam pojam koncepta praga postaje koncept praga, objašnjavaju Falkner, Vivian i Falkner (2013). Ako učenik naiđe na izazovan i stran koncept, a pri tome ne može pronaći adekvatne riječi kojima bi ga opisao i objasnio nastavniku što ga muči, pa shodno tome nastavnik možda ne bi razumio zašto neki dijelovi gradiva uzrokuju poteškoće njemu ili ostalim učenicima. Uvođenjem teorije koncepta praga, postoji sve veće prihvaćanje unutar istraživačke zajednice obrazovanja da mnogi trajni problemi u učenju i poučavanju računalne znanosti proizlaze iz ovih izazovnih koncepata. Ipak, kako ćemo pokazati istražujući relevantnu literaturu, širenje ovih ideja događa se prilično sporo i snažno je ograničeno unutar zajednice istraživača obrazovanja (Falkner, Vivian i Falkner, 2013).

Poučavanje programiranja često predstavlja izazov za nastavnike koji se pitaju zašto učenici ne usvajaju gradivo i kako riješiti prepreke, posebno u prvim programskim tečajevima. Koncepti praga nude objašnjenje zašto određene prepreke uzrokuju poteškoće. Važno je napomenuti da su matematički i informatički koncepti praga ključni za razvoj logičkog razmišljanja i analitičkih sposobnosti učenika. Primjeri iz matematike, kao što su limes i funkcije, te primjeri iz informatike, poput objektno orijentiranog programiranja i pokazivača ilustriraju kako razumijevanje tih pojmova može značajno utjecati na napredovanje učenika. Matematika i informatika su posebno povezane kroz logičke i analitičke vještine koje se razvijaju kod učenika, a koje su ključne za uspjeh u STEM području. Osim navedenoga ove dvije discipline dijele slične izazove u poučavanju, što omogućuje bolje razumijevanje i usporedbu koncepta praga. Također, u brojnim istraživanjima i znanstvenim radovima spominje se povezanost informatike i matematike.

Koncepti praga postali su važna tema u svijetu računalne znanosti. Ljudi proučavaju kako primijeniti te koncepte kad podučavaju programiranje, oblikuju nastavne planove i biraju programski jezik. Razmatraju se i jednostavni načini rješavanja problema koje studenti često imaju, kao i kako upravljati frustrirajućim situacijama kada učimo nešto novo. Iako zvuči ozbiljno, "koncepti praga" zapravo mogu biti prilično jednostavni i svakodnevnici. Važno je prepoznati ove koncepte jer oni mogu značajno utjecati na način kako učimo. Ponekad je teško prenijeti ideju nekome tko nije upoznat s njom ili se nalazi u fazi učenja. S obzirom na izazove tradicionalnih metoda ocjenjivanja, nije iznenađujuće što tražimo načine primjene teorije koncepta praga kako bismo olakšali nastavnicima i studentima. Pitanje koje se postavlja je hoće li mehanizam koncepta praga sam na sebi imati utjecaj, otežavajući komunikaciju i prihvaćanje ove teorije kao nove metode razmišljanja o teškoćama u učenju. Kada zajednica nema jasnu viziju svojih praksi ili prihvaćene teorije, rasprave unutar nje mogu biti teže, a komunikacija s vanjskim zajednicama može dovesti do konfuzije i odbijanja. Budući da je važno imati jasnu viziju i informacije koje podržavaju praktičnu primjenu, obrazovna teorija koja nije dovoljno potkrijepljena ima malu vjerojatnost da će biti prihvaćena u zajednici praktičara. Ovo je posebno izraženo kad se suočavamo s izazovnim konceptima koji se brže odbacuju, jer nema dovoljno motivacije za prijelaz

preko praga. Primjer ćemo prikazati kroz analizu primjene konstrukcijske teorije i timskog rada u preddiplomskom obrazovanju (Falkner i ostali, 2013).

### **3.1. Istraživanje koncepta praga u informatici**

Istraživanje koncepta praga u informatici provedeno od strane Boustedt, Eckerdal, McCartney, Moström, Ratcliffe, Sanders, Zander (2007) fokusiralo se na pronalazak pojmova koji bi mogli odgovarati konceptima praga, te na njihovo potvrđivanje od strane učenika, a zatim i na projeru zadovoljavanja kriterija koncepta praga.

U lipnju 2005., na konferenciji o inovacijama i tehnologiji u nastavi računalnih znanosti ispitano je trideset i troje stručnjaka iz područja informatike iz ukupno devet zemalja o njihovom mišljenju za prijedlog pojma koji zadovoljava kriterije koncepta praga. Nakon toga, u studenom iste godine na konferenciji o nastavi računalnih znanosti u Finskoj (McCartney, Sanders, 2005) provedeno je isto istraživanje, a odgovori su bili prilično slični onima s prošle konferencije iz lipnja. Ispitanici su se usredotočili na "teško za naučiti" više nego na bilo koje druge aspekte koncepta o kojima su raspravljali.

Nakon toga su ispitani studenti s raznih sveučilišta iz nekoliko zemalja te im je postavljeno pitanje da odaberu koncept praga za koji smatraju da im je bio najteži za naučiti i shvatiti te koji je bio prekretnica u njihovom dotadašnjem učenju. Ispitanici su odabrali sljedeće: strukture kontrole, razmišljanje sekvencijalno, parametri, objekti, model memorije. Od koncepata koji su se detaljno razmatrali od strane studenata, za analizu su odabrana dva ključna pojma koja su u potpunosti zadovoljili kriterije koncepta praga: objektno orijentirane tehnologije i pokazivači. Za oba pojma pronađeni su dokazi koji ukazuju da zadovoljavaju kriterije za koncepte praga. (Boustedt, Eckerdal, McCartney, Moström, Ratcliffe, Sanders, Zander, 2007)

#### **3.1.1. Objektno orijentirano programiranje**

Objektno orijentirano programiranje (OOP) je pristup u kojem se računalni program zamišlja kao skup međusobno povezanih objekata. Umjesto razmišljanja o programu kao nizu uputa koje računalu govore što da radi, OOP se fokusira na male programe (objekte) koji reagiraju na određene događaje uzrokovane korisničkim akcijama.

Tradicionalni pristup se naziva procesno orijentirani model, gdje se program doživljava kao niz naredbi koje manipuliraju podacima. Ovaj model je korišten u proceduralnim jezicima poput C-a i Pascala. Međutim, s rastom veličine i složenosti programa, ovaj pristup postaje manje učinkovit.

Problem procesnog modela leži u tome što programer mora detaljno definirati rješenje, stvarajući poveznicu između računalnog modela i problema koji se rješava. Ovakav način rada bolje odgovara strojevima nego ljudskom razmišljanju, jer zahtijeva da programer preslika logiku rada računala u problem koji se rješava.

Kako programi postaju veći, javljaju se poteškoće poput:

1. Nemogućnosti praćenja toka izvršavanja

2. Otežanog ispravljanja grešaka
3. Neefikasnog dodavanja novih funkcija

OOP model omogućava fokusiranje programa na podatke (objekte). Program se sastoji od objekata koji surađuju prema unaprijed definiranim pravilima kako bi postigli određeni cilj. Prednosti ovog modela u odnosu na tradicionalni uključuju:

- Bolju razumljivost modela,
- Lakše pronalaženje i ispravljanje grešaka,
- Jednostavnije održavanje programa (Jovanović, 2012).

Objektno orijentirano programiranje (OOP) smatra se izazovnim za poučavanje i učenje. Boustedu i drugima u intervjuima sa studentima s prvih godina studija, studenti su izrazili poteškoće u razumijevanju objektno orijentiranih pojmova, unatoč uloženom trudu. Jedan student opisao je OOP kao teško za učenje, posebno u početku, ali je naglasio da mu je razumijevanje OOP-a pomoglo u shvaćanju drugih koncepta programiranja poput višenitnog programiranja.

Tijekom intervjua, istraživač je razgovarao sa studentima o integrativnosti OOP-a, pitajući ih jesu li razumjeli OOP i što su još razumjeli. Studenti su opisivali kako su primijenili koncepte OOP na druge aspekte, poput tečaja inženjeringa programske opreme, gdje su ih koristili za razumijevanje komunikacije između različitih dijelova sustava.

Studenti su naglasili da su osnovni koncepti uvijek prisutni, čak i kad se suoče s određenim sintaktičkim izazovima. To ukazuje na to da je objektno orijentirano programiranje, iako ponekad izazovno, duboko integrirano i teško zaboravljivo.

Vezano za transformaciju učenja OOP-a, studenti su izrazili promjenu u načinu razmišljanja o problemima. Znanje stečeno programiranjem na jednom modelu mogu primijeniti i prenijeti na druge modele, što pokazuje da je učenje OOP-a transformiralo njihov pristup programiranju.

Istraživanje ističe izazove u učenju objektno orijentiranog programiranja, ali također naglašava duboku transformaciju u razmišljanju studenata i sposobnost primjene stečenih znanja na različite situacije. (Boustedt i ostali, 2007)

### **3.1.2. Pokazivači**

Istraživanje se zatim fokusiralo na drugi identificirani koncept: pokazivače.

Koncept koji se prepoznaje kao potencijalna prepreka u učenju odnosi se na korištenje pokazivača, posebice kada se primjenjuju kao parametri u programiranju. Studenti su iznijeli izazove s kojima su se suočavali u povezivanju teorijskih pojmova s praktičnom primjenom pokazivača, posebno u kontekstu prosljeđivanja parametara u programskim jezicima.

Jedan student opisao je poteškoće u povezivanju teorijskih koncepata s praktičnom primjenom pokazivača. Drugi student potvrdio je da mnogi imaju problema s razumijevanjem pokazivača, no kada su shvatili da pokazivač jednostavno označava određeno mjesto u memoriji, koncept im je postao jasniji.

Razumijevanje pokazivača omogućilo je studentima da ga primijene u drugim područjima, poput klase strojne opreme i operativnih sustava. Tamo su pokazivači korišteni ne samo kao teorijski koncepti, već i u praksi, posebno u asemblerskom jeziku. Studenti su naglasili da, jednom kad savladaju koncept pokazivača, mogu jasnije razumjeti objekte i reference te ih uspješno primijeniti u programiranju.

U drugom dijelu intervjua, isti student opisao je kako je razumijevanje pokazivača pomoglo u drugim predmetima, poput hardverske klase i operativnih sustava. Student je naglasio da se, nakon shvaćanja kako pokazivači funkcioniraju, može primijeniti na različite aspekte računarstva s većom sigurnošću.

Najvažnije, studenti su istaknuli da su, nakon uspješnog savladavanja koncepta pokazivača, stekli duboko razumijevanje i sposobnost primjene, što im je omogućilo rješavanje kompleksnih problema u raznim područjima informatike (Boustedt i ostali, 2007).

## 3.2. Koncepti praga u matematici

Koncept praga u matematici odnosi se na ključne pojmove i metode koji omogućuju učenicima razumijevanje osnovnih principa i naprednijih matematičkih operacija. Pragovi u matematici često se odnose na kritične točke ili vrijednosti koje predstavljaju granice ili prijelaze između različitih stanja ili razina složenosti. Razumijevanje ovih pragova ključno je za daljnje napredovanje u matematici, jer omogućuje učenicima da prepoznaju i analiziraju promjene, prijelaze i granice u matematičkim funkcijama i procesima.

U nastavku ćemo istražiti nekoliko ključnih koncepata praga u matematici, uključujući limes, derivacije, integrale i druge srodne pojmove. Ovi koncepti su temelj za mnoge napredne matematičke operacije i razumijevanje njihovih pragova je esencijalno za uspješno savladavanje složenijih matematičkih izazova.

### 3.2.1. Limes

Razmišljajući o konceptu praga u matematici, često se suočavamo s izazovima koje učenici imaju s definicijom limes funkcije. Pojam limesa ključan je u matematičkoj analizi te čini temelj za razumijevanje drugih grana matematike poput diferencijalnog i integralnog računa. Razumijevanje definicije limesa pomaže u otvaranju vrata u polje analiza.

Limes, kao jedan od koncepata praga često predstavlja problem za većinu učenika, što i nije iznenađujuće s obzirom na to da se razvijao dugi niz godina. Istraživanja su pokazala da postoje dvije glavne kategorije problema s kojima se učenici suočavaju: oni koji proizlaze iz njihovih prethodnih predodžbi o limesima te oni koji proizlaze iz same definicije limesa.

Prikaz koje učenici imaju u vezi s pojmom limesa, kao i interpretacije riječi definicija čije tumačenje u svakodnevnom životu nikako nije u skladu s matematičkim značenjima, mogu utjecati na njihovo razumijevanje koncepta. Osim toga, struktura same definicije, s kvantifikatorima  $\forall$  i  $\exists$ ,

može zbuniti učenike. Često žele formulu ili algoritam kako bi izračunali limes, teško prihvaćajući korištenje definicije.

Mnogi učenici i matematičari opisuju trenutak kad im je postalo jasno značenje definicije. Činjenica da se studenti ne mogu sjetiti točnog trenutka kada se dogodilo shvaćanje definicije je zaista značajno. Ukazuje na to da je novo shvaćanje transformirajuće i nepovratno, što nam pokazuje da je riječ o dubokom razumijevanju (Meyer i Land, 2003).

### 3.2.2. Funkcije

Prije nego što studenti dođu u dodir s definicijom granice funkcije, već bi od prije trebali poznavati pojam funkcije. Ovaj koncept je temeljan u suvremenoj matematici te iako su učenici izloženi tom pojmu u školi, pokazalo se da mnogi studenti usprkos tome imaju poteškoća (Pettersson, 2011).

Zajednica za matematičko obrazovanje provela je mnoga istraživanja o razumijevanju studenata i poteškoćama s funkcijama. Ponovno, kao i u slučaju granica, otkrivamo da je jedan od glavnih problema s kojima se studenti suočavaju problem definicije. Studenti često smatraju funkcije formulom ili jednadžbom, i možda se ne žele prihvatiti funkcija koje nisu definirane jednim algebarskim izrazom (Vinner, Dreyfus, 1989). Također mogu očekivati da su sve funkcije kontinuirane. Ove poteškoće s definicijom funkcije slične su stadijima povijesnog razvoja tog pojma, što čini argument za opisivanje koncepta inherentno konceptualno teškim (ili problematičnim).

Možda kao posljedica gledanja funkcija definiranih algebarskim izrazom, studenti ih često doživljavaju u smislu radnji ili modela ulaz-izlaz. Na primjer, mogu vidjeti  $f(x) = 5x - 2$  kao recept za niz izračuna umjesto kao objekt sam po sebi. Da bi pravilno razumjeli funkcije i radili s njima u raznim područjima matematike, studenti bi trebali moći zamisliti funkciju kao radnju, kao proces i kao objekt (Breidenbach, Dubinsky, Hawks, Nichols, 1992).

Thurston je govorio o učenicima matematike koji korak po korak rade i muče se razumjeti pojam, ali tvrdi da, jednom kada zaista razumiju pojam, njihova perspektiva može se promijeniti kako bi ga mogli vidjeti kao cjelinu. Vjerovao je da takvo razumijevanje i mentalna kompresija mogu olakšati sjećanje na ideju i njezino korištenje kad je to potrebno u budućnosti (Thurston, 2005).

Pojam funkcije prožima mnoga područja matematike, i kao takvo, temeljito razumijevanje koncepta može otkriti prije nevidene veze između različitih tema. Studenti obično prvi put susreću formalnu definiciju funkcije u kontekstu analize, ali jednom kada je pravilno shvaćena, često znaju kako ju mogu povezati s linearnim sustavima i matricama s kojima su se susreli u algebri. U tom smislu, može se opisati kao integrativna.

## 4. Istraživanje stavova o konceptima praga

Ciljevi ovog istraživanja usmjereni su na razumijevanje percepcije i primjene koncepata praga u nastavi informatike u osnovnim i srednjim školama. Istraživanje ima za cilj identificirati zahtjevne koncepte koje nastavnici i učenici percipiraju kao izazovne te među njima identificirati koje nastavnici smatraju koncepte praga.

Identifikacija koncepata praga omogućuje unapređenje nastave informatike u smislu strukturiranja kurikuluma kojima će se osigurati da učenici temeljito razumiju ove koncepte prije nego što prijeđu na složenije teme. Nastavnici mogu koristiti specifične strategije poučavanja usmjerene na olakšavanje razumijevanja ovih koncepata učenicima. Također, nastavnici se mogu fokusirati na te koncepte prilikom praćenja napretka učenika.

### 4.1. Metodologija

Kao temeljnu metodologiju istraživanja odabrali smo tehniku nominalne grupe (NGT), smatrajući je prikladnom za postizanje dubljeg uvida u percepciju i razumijevanje koncepata praga među sudionicima.

- 1. Tiho generiranje ideja: sudionici pojedinačno pišu svoje ideje bez rasprave ili objašnjenja.
- 2. Diskusija ideja: facilitator (voditelj) grupne rasprave pojašnjava i razmatra ideje koje su studenti napisali.
- Glasovanje i rangiranje: Treći korak uključuje glasovanje i rangiranje ideja kako bi se odabrale najvažnije ili najkorisnije.
- Zaključivanje: U četvrtom koraku, najviše rangirane ideje se odabiru kao konačne odluke grupe. Ako postoji više ideja s jednakim brojem glasova, rasprava se ponavlja.
- Pisanje izvještaja: Na kraju, sudionik koji je facilitator piše izvještaj o cijelom procesu i rezultatima diskusije

Tehnika nominalne grupe (NGT) pokazala se vrlo učinkovitom metodom za poticanje kritičkog mišljenja kod studenata. Učinkovite diskusije trebaju uključivati mali broj sudionika, pružiti jasne i usmjerene upute te omogućiti konstruktivnu povratnu informaciju. NGT ispunjava sve te kriterije, a uz to osigurava punu participaciju svih članova grupe, što je posebno korisno u obrazovnom kontekstu.

NGT se često koristi u raznim disciplinama, uključujući medicinu, informacijsku tehnologiju, kreiranje politika, menadžment i obrazovanje, gdje služi kao procjenjivana metoda diskusije (Macphail, 2001). U obrazovnim postavkama, NGT se primjenjuje u dizajniranju i evaluaciji kurikuluma, ali i kao pedagoška metoda koja potiče sudionike na aktivno sudjelovanje (Chapple & Murphy, 1996). Studije su pokazale da NGT poboljšava produktivnost sudionike i njihovu sposobnost rješavanja problema kroz strukturirane diskusije (Madar, 1982).



#### 4.1.1. Primjena tehnike nominalne grupe za identifikaciju koncepata praga

Proces istraživanja proveden je u OS Vežica u Rijeci kao dio aktivnosti Županijskog stručnog vijeća učitelja informatike u osnovnim školama Primorsko-goranske županije te na Fakultetu informatike i digitalnih tehnologija Sveučilišta u Rijeci kao dio aktivnosti Županijskog stručnog vijeća nastavnika informatike u srednjim školama.

Proces primjene NGT-a (Slika1) sastoji se od šest koraka. Ovi koraci mogu se provoditi u sinkronim (istovremenim) i asinkronim (neistovremenim) okruženjima.

Prikaz koraka u procesu primjene tehnike nominalne grupe:



Slika 1. Proces primjene NGT-a

Proces istraživanja započeo je usmenom prezentacijom koja je obuhvatila navedene definicije i primjere. Pripremljena je prezentacija koja je sadržavala precizne definicije koncepata praga te primjere koncepata praga s obrazloženjima prema karakteristikama. Za demonstraciju koncepata praga odabrane su naredbe ponavljanja i potprogrami kao primjeri koji u potpunosti odgovaraju konceptu praga. Paralelno, kao primjer zahtjevnog koncepta, analizirana je rekurzija, te je izdvojena kao koncept koji zahtijeva dodatno razumijevanje, ali nije nužan preduvjet za daljnje učenje u programiranju.

Nakon toga, sudionici su ispunili upitnik koji je tražio identifikaciju koncepata kao i odgovor na pitanje jesu li ti koncepti temeljni, zahtjevni te odgovaraju li karakteristikama koncepata praga.

Nakon individualnog dijela, sudionici su se podijelili u manje grupe kako bi dodatno produbili svoje razumijevanje i raspravili predložene koncepte praga. U fazi grupnog odlučivanja, svaka grupa je odabrala jedan ili više koncepata za koje su smatrali da odgovaraju svim karakteristikama koncepata praga. Kroz konstruktivnu raspravu, sudionici su zajedno doprinijeli dubljem razumijevanju koncepata i njihovoj primjeni u nastavi informatike.

Kako bismo dobili jasne rezultate i rangiranje predloženih koncepata praga, provedeno je glasovanje putem online platforme Padlet.

Nakon glasovanja i pregleda rezultata dolazi se do zaključka te identifikacije mogućih koncepata praga.

## 4.2. Ispitanici

Istraživanje stavova nastavnika osnovnih i srednjih škola ima za cilj razumjeti percepciju i iskustva učitelja u vezi s implementacijom koncepata praga u nastavi. Kroz tehniku nominalne grupe provedenu među nastavnicima, prikupljeni su podaci koji pružaju uvid u raznolikost mišljenja i pristupa korištenju ovih koncepata u obrazovnom procesu. U istraživanju su sudjelovali nastavnici informatike iz osnovnih i srednjih škola. Ispitanici su odgovarali na pitanja koja se tiču njihovih osobnih stavova prema važnosti i praktičnosti primjene koncepata praga u svakodnevnoj nastavi informatike. Prikaz ispitanika prema njihovim demografskim podacima i radnom iskustvu omogućava dublji uvid u strukturu nastavničkog kadra.

### 4.2.1. Podaci o ispitanicima u osnovnim školama

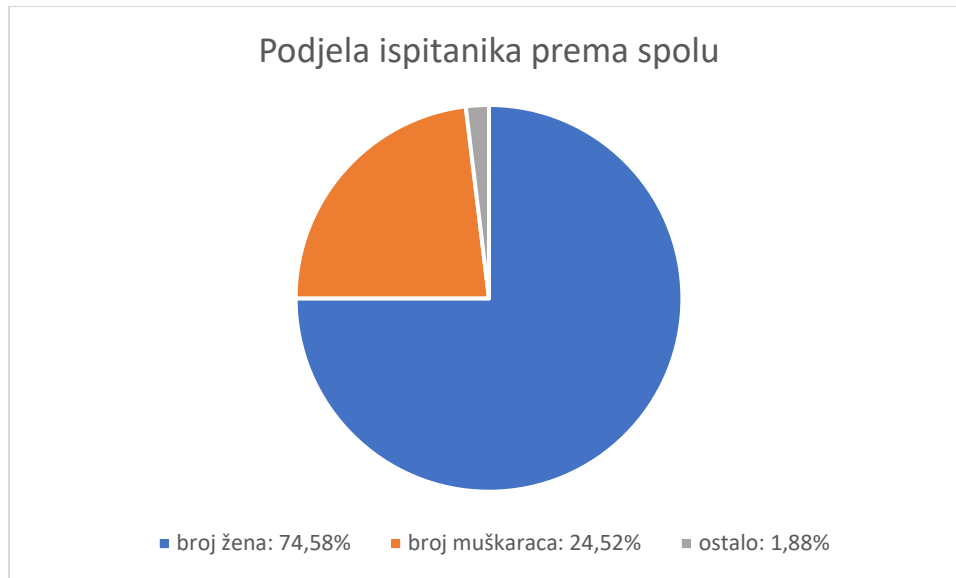
Ispitanici su osim o pitanjima vezanim za koncepte praga odgovarali na pitanja koja su omogućila prikupljanje podataka o njihovom obrazovnom profilu, godinama iskustva i predmetima koje predaju. Ovi podaci pružili su uvid u strukturu nastavničkog kadra te su poslužili kao osnova za daljnju analizu percepcije koncepata praga u nastavi informatike. Slijedi tablica s detaljnim popisom svih ispitanika.

Podaci o dobi ispitanika u OŠ

| Dob   | Broj ispitanika | Ž  | M | Ostalo |
|-------|-----------------|----|---|--------|
| 24-34 | 21              | 15 | 6 | 0      |
| 35-44 | 10              | 7  | 3 | 0      |
| 45-54 | 19              | 15 | 3 | 1      |
| 55+   | 3               | 2  | 1 | 0      |

U analizi ankete koja je provedena, proučeni su demografski podaci ispitanika. Rezultati su prilično zanimljivi i pružaju nam bolji uvid u ciljanu skupinu ispitanika, koji su u ovom slučaju nastavnici informatike u osnovnim školama. Analiza anketa se temelji na uzorku od 53 uredno ispunjena anketna listića.

Većina ispitanika su žene i to 39 ispitanica, dok je manji broj muškaraca te iznosi 13 ispitanika, dok se jedan ispitanik nije izjasnio, odnosno nije odabrao niti jednu od upravo spomenute kategorije. Ovi podaci ukazuju na značajnu rodnu nejednakost među nastavnicima informatike u osnovnim školama, pri čemu većina ispitanika čine žene (73,58%), dok je manji broj muškaraca (24,52%).



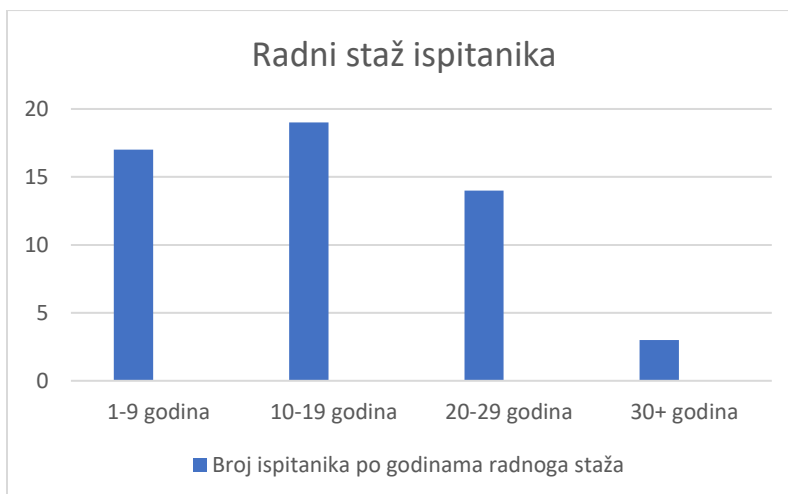
Slika 2. Prikaz ispitanika prema spolu (OŠ)

Imamo širok spektar starosti, od 24 do 63 godine, odnosno prosječnu starost od 39 godina. Iz ovih podataka možemo zaključiti da imamo ispitanike koji se nalaze u različitim fazama karijere i života. To bi svakako moglo imati utjecaj na njihove odgovore, s obzirom na raznolikost iskustva u radu među njima. Osobe koje su tek počele raditi kao nastavnici možda nemaju isto iskustvo kao i oni s dugogodišnjim iskustvom. Kao rezultat toga, njihove percepcije i razumijevanje koncepta praga u nastavi informatike mogu varirati.

Podaci o radnom stažu ispitanika u OŠ

| Radni staž | Broj ispitanika | Ž  | M | Ostalo |
|------------|-----------------|----|---|--------|
| 1-9        | 28              | 21 | 7 | 0      |
| 10-19      | 7               | 6  | 1 | 0      |
| 20-29      | 11              | 9  | 1 | 1      |
| 30+        | 2               | 1  | 1 | 0      |

Svi ispitanici imaju visoku stručnu spremu i zaposleni su kao nastavnici informatike u osnovnim školama. Prosječno vrijeme provedeno na radnome mjestu nastavnika iznosi 9,88, odnosno 10 godina. Ukupno je 17 ispitanika imalo radni staž od 1 do 9 godina, 19 ispitanika imalo je radni staž od 10 do 19 godina, 14 ispitanika imalo je radni staž od 20 do 29 godina, dok su 3 ispitanika imala više od 30 godina radnog staža (slika 2).



Slika 3 Radni staž ispitanika OŠ

Među ispitanicima postoji razlika u kombinacijama predmeta koje predaju, uključujući informatiku, matematiku, engleski jezik, fiziku i tehničku kulturu.

*Broj ispitanika po radnim mjestima*

| Broj ispitanika po radnim mjestima             |                |          |
|--|----------------|----------|
| Radno mjesto                                   | Broj sudionika | Postotak |
| nastavnik informatike u OŠ:                    | 39             | 75%      |
| nastavnik informatike i matematike u OŠ:       | 10             | 19,23%   |
| nastavnik informatike i tehničke kulture u OŠ: | 2              | 3,85%    |
| nastavnik informatike i engleskog jezika u OŠ: | 1              | 1,92%    |

#### 4.2.2. Podaci o ispitanicima u srednjim školama

Ispitanici su osim o pitanjima vezanim za koncepte praga odgovarali na pitanja koja su omogućila prikupljanje podataka o njihovom obrazovnom profilu, godinama iskustva i predmetima koje predaju. Ovi podaci pružili su uvid u strukturu nastavničkog kadra te su poslužili kao osnova za daljnju analizu percepcije konceptata praga u nastavi informatike. Slijedi tablica s detaljnim popisom svih ispitanika.

#### Podaci o dobi ispitanika u SŠ

| Dob   | Broj ispitanika | Ž | M |
|-------|-----------------|---|---|
| 24-34 | 1               | 0 | 1 |
| 35-44 | 5               | 3 | 2 |
| 45-54 | 3               | 3 | 0 |
| 55+   | 7               | 6 | 1 |

U analizi ankete koja je provedena, proučeni su demografski podaci ispitanika. Rezultati su prilično zanimljivi i pružaju nam bolji uvid u ciljanu skupinu ispitanika, koja su u ovom slučaju nastavnici informatike u srednjim školama. Analiza anketa se temelji na uzorku od 16 uredno ispunjenih anketnih listića.

Većina ispitanika su žene i to 12 ispitanica, dok je manji broj muškaraca te iznosi 4 ispitanika. Ovi podaci ukazuju na značajnu rodnu nejednakost među nastavnicima informatike u srednjim školama, pri čemu većina ispitanika čine žene (75%), dok je manji broj muškaraca (25%).



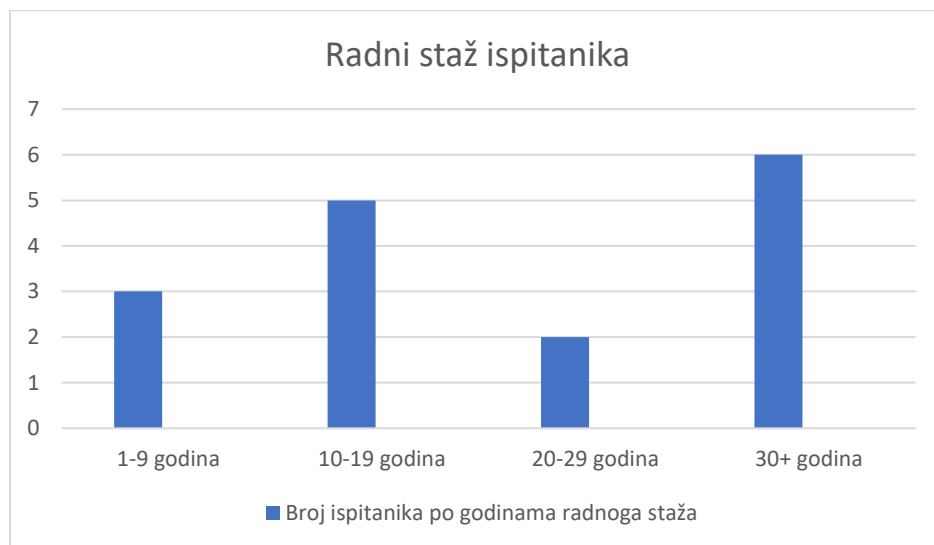
Slika 4 Prikaz ispitanika prema spolu (SŠ)

Ispitanici su različite životne dobi, od 27 do 63 godine, odnosno prosječnu starost od 47 godina. Iz ovih podataka možemo zaključiti da imamo ispitanike koji se nalaze u različitim fazama karijere i života. To bi svakako moglo imati utjecaj na njihove odgovore, s obzirom na raznolikost iskustva u radu među njima. Osobe koje su tek počele raditi kao nastavnici možda nemaju isto iskustvo kao i oni s dugogodišnjim iskustvom. Kao rezultat toga, njihove percepcije i razumijevanje koncepta praga u nastavi informatike mogu varirati.

Podaci o radnom stažu ispitanika u OŠ

| Radni staž | Broj ispitanika | Ž | M |
|------------|-----------------|---|---|
| 1-9        | 3               | 2 | 1 |
| 10-19      | 5               | 3 | 2 |
| 20-29      | 2               | 2 | 0 |
| 30+        | 5               | 4 | 1 |

Svi ispitanici imaju visoku stručnu spremu i zaposleni su kao nastavnici informatike u srednjim školama. Prosječno vrijeme provedeno na radnome mjestu nastavnika iznosi 17.6 godina. Ukupno su 3 ispitanika imali radni staž od 1 do 9 godina, 5 ispitanika imalo je radni staž od 10 do 19 godina, 2 ispitanika imalo je radni staž od 20 do 29 godina, dok je 5 ispitanika imala više od 30 godina radnog staža.



Slika 5 Radni staž ispitanika (SŠ)

Među ispitanicima postoji razlika u kombinacijama predmeta koje predaju, uključujući informatiku, matematiku i programiranje za web i baze podataka.

Broj ispitanika po radnim mjestima (SŠ)

| Broj ispitanika po radnim mjestima        |                |          |
|---|----------------|----------|
| Radno mjesto                              | Broj sudionika | Postotak |
| nastavnik informatike u SŠ:               | 10             | 62.5%    |
| nastavnik informatike i računalstva u SŠ: | 2              | 12.5%    |

|  |   |       |
|--|---|-------|
| nastavnik informatike i matematike u SŠ:             | 2 | 12.5% |
| Nastavnik primjene informatike i računalstva u SŠ:   | 1 | 6.25% |
| nastavnik programiranja za web i baze podataka u SŠ: | 1 | 6.25% |

### 4.3. Rezultati istraživanja

Istraživanja provedena u sklopu aktivnosti Županijskog stručnog vijeća učitelja informatike u osnovnim školama Primorsko-goranske županije u OŠ Vežica u Rijeci te na Fakultetu informatike i digitalnih tehnologija Sveučilišta u Rijeci kao dio aktivnosti Županijskog stručnog vijeća nastavnika informatike u srednjim školama. Ovaj istraživački rad dovelo je do značajnih rezultata u pronalasku i definiranju koncepata praga u nastavi informatike. U nastavku slijede detaljni rezultati analize provedene među navedenim ispitanicima.

#### 4.3.1. Rezultati provedeni nad ispitanicima zaposlenima u OŠ

Sljedeći rezultati otkrivaju temeljne i zahtjevne koncepte, kao i zadovoljavanje karakteristika koncepta praga u nastavi informatike prema mišljenju ispitanika zaposlenih u osnovnim školama.

*Prikaz predloženih temeljnih i zahtjevnih koncepata (OŠ)*

| br. | Koncept                | Br. Ispitanika | Temeljni koncept |    | Zahtjevni koncept |    | Karakteristike koncepta praga |    |                   |    |                 |    |              |    |
|-----|------------------------|----------------|------------------|----|-------------------|----|-------------------------------|----|-------------------|----|-----------------|----|--------------|----|
|     |                        |                | DA               | NE | DA                | NE | Povezanost                    |    | Transformativnost |    | Ireverzibilnost |    | Ograničenost |    |
|     |                        |                |                  |    |                   |    | DA                            | NE | DA                | NE | DA              | NE | DA           | NE |
| 1.  | Adresiranje            | 11             | 9                | 2  | 10                | 1  | 11                            |    | 11                |    | 11              |    | 11           |    |
| 2.  | Algoritam grananja     | 16             | 15               | 1  | 15                | 1  | 16                            |    | 16                |    | 12              | 4  | 16           |    |
| 3.  | Atributi               | 2              |                  | 2  | 2                 |    | 2                             |    |                   | 2  |                 | 2  | 2            |    |
| 4.  | Baza podataka          | 2              | 1                |    | 2                 |    | 2                             |    | 2                 |    | 2               |    | 2            |    |
| 5.  | Binarni brojevi        | 4              | 4                |    | 1                 | 4  | 4                             |    | 4                 |    | 1               | 3  | 4            |    |
| 6.  | Datoteka               | 2              | 2                |    | 1                 |    | 2                             |    | 2                 |    |                 |    |              | 1  |
| 7.  | Dijagram toka podataka | 5              | 5                |    | 1                 | 4  | 5                             |    | 5                 |    | 5               |    | 5            |    |
| 8.  | Dijelovi računala      | 4              | 4                |    |                   | 4  | 4                             |    | 4                 |    | 4               |    | 4            |    |
| 9.  | Elementi u Canvi       | 1              | 1                |    |                   | 1  | 1                             |    | 1                 |    | 1               |    | 1            |    |
| 11. | Funkcije               | 13             | 1                |    | 6                 | 7  | 13                            |    | 13                |    | 13              |    | 13           |    |
| 12. | HTML                   | 11             | 11               |    | 9                 | 2  | 11                            |    | 11                |    | 11              |    | 11           |    |
| 13. | Internet               | 3              | 3                |    |                   | 3  | 3                             |    | 3                 |    | 3               |    | 3            |    |
| 14. | Izlazne jedinice       | 1              | 1                |    |                   | 1  | 1                             |    | 1                 |    | 1               |    | 1            |    |
| 15. | Kopiranje              | 3              | 3                |    |                   | 3  | 3                             |    | 3                 |    | 3               |    | 3            |    |
| 16. | Korisnički podaci      | 3              | 3                |    |                   | 3  | 3                             |    | 3                 |    | 3               |    | 3            |    |
| 17. | Liste                  | 6              |                  | 6  | 5                 | 1  | 6                             |    | 6                 |    |                 | 6  | 6            |    |
| 18. | Logički uvjeti         | 13             | 13               |    | 13                |    | 13                            |    | 13                |    | 13              |    | 13           |    |
| 19. | Logički sklopovi       | 7              | 6                | 1  | 6                 | 1  | 7                             |    | 7                 |    | 7               |    | 7            |    |

|     |                                   |    |    |   |   |   |    |   |   |   |    |   |    |   |
|-----|-----------------------------------|----|----|---|---|---|----|---|---|---|----|---|----|---|
| 20. | Logičko razmišljanje              | 3  | 3  |   | 3 |   | 3  |   | 3 |   | 3  |   | 3  |   |
| 21. | Mape                              | 1  | 1  |   |   |   | 1  |   | 1 |   |    |   |    |   |
| 22. | Oblak                             | 5  | 5  |   | 5 |   | 5  |   |   | 5 |    | 5 | 5  |   |
| 23. | Organizacija podataka             | 1  | 1  |   | 1 |   | 1  |   | 1 |   | 1  |   | 1  |   |
| 24. | Osnovna obrada teksta             | 4  | 4  |   |   | 4 | 4  |   |   | 4 | 4  |   | 4  |   |
| 25. | Petlje                            | 3  | 3  |   | 3 |   | 3  |   | 3 |   | 3  |   | 3  |   |
| 26. | Pisanje e-pošte                   | 1  | 1  |   | 1 |   | 1  |   | 1 |   | 1  |   | 1  |   |
| 27. | Poveznice                         | 2  |    | 2 | 2 |   | 2  |   | 2 |   |    | 2 | 2  |   |
| 28. | Pretvorbe (brojevni sustavi)      | 2  | 2  |   | 2 |   | 2  |   | 2 |   | 2  |   | 2  |   |
| 29. | Primjena funkcija računalne mreže | 2  | 2  |   | 2 |   | 2  |   | 2 |   | 2  |   | 2  |   |
| 30. | Reference u Wordu                 | 1  |    | 1 | 1 |   | 1  |   |   | 1 |    | 1 | 1  |   |
| 31. | Relacije u bazi podataka          | 7  | 7  |   | 7 |   | 7  |   | 7 |   | 6  | 1 | 4  | 3 |
| 32. | Spremanje datoteke                | 11 | 11 |   | 7 | 4 | 10 | 1 | 5 | 6 | 11 |   | 11 |   |
| 33. | Struktura dokumenta               | 2  | 2  |   |   | 2 | 2  |   | 2 |   | 2  |   | 2  |   |
| 34. | Ulazno-izlazne jedinice           | 1  | 1  |   |   |   | 1  |   | 1 |   | 1  |   | 0  |   |
| 35. | Uporaba IKT-a                     | 3  | 3  |   |   | 3 | 3  |   | 3 |   | 3  |   |    | 3 |
| 37. | Varijable                         | 4  | 4  |   | 2 | 2 | 4  |   | 4 |   | 3  | 1 | 4  |   |

Najvažniji temeljni koncepti koje su ispitanici istaknuli uključuju "Algoritam Grananje", "Funkcije", "HTML", "Spremanje datoteke", "Adresiranje" i "Logički uvjet". Ti koncepti su prepoznati kao ključni za razumijevanje informatike, što znači da učenici moraju ovladati njima kako bi izgradili čvrste temelje u ovom području.

"Algoritam grananja" je identificiran kao temeljni koncept od strane 15 od 16 ispitanika, dok su "Funkcije" i "Logički uvjeti" dobili 13 od 13 glasova ispitanika koji su ih naveli kao temeljni koncept. "HTML" i "Spremanje datoteke" također su prepoznati od strane 11 ispitanika koji su ih označili kao temeljne. "Adresiranje" je također visoko rangirano, što ukazuje na njihovu važnost u kurikulumu informatike.

Zahtjevni koncepti, koji predstavljaju izazove u učenju, također su identificirani. "Algoritam Grananje" je ovdje također dominirao s 15 ispitanika koji su ga označili kao zahtjevan koncept. Slično tome, "Relacije u bazi podataka" i "Spremanje datoteke" su označeni kao zahtjevni od strane većine ispitanika. Ovi koncepti zahtijevaju dodatnu pažnju i resurse kako bi se osiguralo da učenici postignu adekvatno razumijevanje.



Povezanost koncepta je prema ispitanicima važan atribut jer ukazuje na to koliko dobro učenici mogu koristiti jedan koncept za razumijevanje drugih. "Algoritam Grananje", "Funkcije", "Logički uvjeti", "HTML" i "Spremanje datoteke" su prepoznati kao koncepti s visokom povezanošću, što znači da učenici koji ih savladaju lakše razumiju i druge aspekte informatike.

Transformativnost se pak odnosi na sposobnost koncepta da izazove značajnu promjenu u razumijevanju učenika. "Algoritam grananja", "Funkcije", "HTML" i "Logički uvjeti" su ovdje istaknuti kao koncepti koji, kada se jednom shvate, omogućuju učenicima da dožive dublje razumijevanje informatike. To ukazuje na njihov ključni utjecaj na obrazovni proces i važnost temeljitog učenja.

Ireverzibilnost koncepta odnosi se na to koliko je vjerojatno da će učenik zadržati svoje razumijevanje koncepta nakon što ga je savladao. Ispitanici smatraju da koncepti poput "Algoritam Grananje", "Funkcije", "HTML" i "Logički uvjeti" imaju visoku ireverzibilnost, što znači da učenici rijetko zaboravljaju ove koncepte nakon što ih nauče. Ovo je važan aspekt učenja jer sugerira dugoročne koristi od usvajanja tih koncepata.

Ograničenost je atribut koji ukazuje na fluktuirajuće razumijevanje koncepta među učenicima, gdje oni mogu oscilirati između razumijevanja i zaboravljanja. Ispitanici su "Relacije u bazi podataka" i "Algoritam Grananje" identificirali kao koncepte s određenom razinom ograničenosti, što sugerira potrebu za kontinuiranim osvježavanjem i ponavljanjem ovih tema u nastavi.

Rezultati istraživanja također ukazuju na nekoliko koncepata koji od strane ispitanika nisu prepoznati kao temeljni ili zahtjevni od strane većine ispitanika. Koncepti poput "Atributa", "Baza podataka", "Datoteka", "Elementi u Canvi", "Internet", "Izlazne jedinice", "Kopiranje" i "Reference u Wordu" nisu dobili visok broj glasova kao temeljni ili zahtjevni. Na primjer, "Atributi" su identificirani kao temeljni koncept od strane samo dva ispitanika, dok "Baza podataka" i "Datoteka" također imaju vrlo nizak broj prepoznavanja.

Nakon predloženih temeljnih i zahtjevnih koncepata uslijedila je rasprava između ispitanika.

Unutar rasprave ispitanici su zaključili kako veliki broj navedenih koncepata ne zadovoljava uvjet zahtjevnog koncepta što nam ukazuje na težinu prepoznavanja svojstava koncepata te na shvaćanje samoga koncepta praga.

Ispitanici su raspravljali o karakteristikama koncepta praga te o zadovoljavanju istih za predložene koncepte. Rasprava je rezultirala eliminacijom dijela predloženih koncepata koji nisu zadovoljili karakteristike.

Ispitanici su zaključili da "Algoritam grananja" i "Logički uvjet" mogu objediniti u jedan koncept s obzirom na to da je kod algoritma grananja formuliranje logičkog uvjeta dio koji je zahtjevan učenicima. Također, prijedlog "Organizacija podataka" i "Spremanje datoteke" objedinjen je u jedan jer se pri spremanju datoteke mislilo na organizaciju podataka kao dio koji je učenicima zahtjevan.

U sljedećoj tablici možemo vidljivi predložene koncepte praga te rezultate glasovanja za svaki predloženi koncept.

*Predloženi koncepti praga (OŠ)*

| PREDLOŽENI KONCEPTI             | DA | NE |
|---------------------------------|----|----|
| Logički uvjeti                  | 42 | 1  |
| HTML i slični jezici            | 29 | 10 |
| Korisnički račun                | 24 | 18 |
| Adresiranje ćelije              | 40 | 2  |
| Relacije (baze podataka)        | 31 | 11 |
| Varijable                       | 39 | 4  |
| Dijagram toka                   | 35 | 6  |
| Organizacija podataka (pohrana) | 39 | 6  |
| Zaštita osobnih podataka.       | 15 | 27 |

Analizirajući rezultate glasovanja, identificirani su koncepti praga na temelju značajne podrške dobivene od sudionika, a pri tom izdvajajući one koji su jasno dobili veću podršku kao ključne koncepte praga u nastavi informatike.

Logički uvjeti dobili su izuzetno visok broj glasova, s 42 glasa "ZA" i samo jednim glasom "NE".

Adresiranje ćelije također je dobilo velik broj glasova, s 40 glasova "ZA" i samo 2 glasa "NE".

Dijagram tijeka dobio je 35 glasova "ZA" i samo 6 glasova "NE". Organizacija podataka (pohrana) dobila je 39 glasova "ZA" i 6 glasova "NE".

Varijable, iako nisu dobile jednoglasnu podršku, s obzirom na pretežitu podršku, možemo smatrati konceptom praga.

#### **4.3.2. Rezultati provedeni nad ispitanicima zaposlenim u SŠ**

Sljedeći rezultati otkrivaju temeljne i zahtjevne koncepte, kao i zadovoljavanje karakteristika koncepta praga u nastavi informatike prema mišljenju ispitanika zaposlenih u srednjim školama.

*Prikaz predloženih temeljnih i zahtjevnih koncepata (SŠ)*

| br. | Koncept           | Br. Ispitanika | Temeljni koncept |    | Zahtjevni koncept |    | Karakteristike koncepta praga |    |                   |    |                 |    |              |    |
|-----|-------------------|----------------|------------------|----|-------------------|----|-------------------------------|----|-------------------|----|-----------------|----|--------------|----|
|     |                   |                | DA               | NE | DA                | NE | Povezanost                    |    | Transformativnost |    | Ireverzibilnost |    | Ograničenost |    |
|     |                   |                |                  |    |                   |    | DA                            | NE | DA                | NE | DA              | NE | DA           | NE |
| 1.  | Algoritam         | 4              | 2                | 2  | 3                 | 1  | 4                             |    | 4                 |    | 4               |    | 4            |    |
| 2.  | Dijelovi računala | 1              | 1                |    | 1                 |    | 1                             |    | 1                 |    | 1               |    |              | 1  |

|     |                               |   |   |   |   |   |   |   |   |   |   |   |   |  |
|-----|-------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|--|
| 3.  | Formule                       | 1 |   | 1 | 1 |   | 1 |   | 1 |   | 1 |   | 1 |  |
| 4.  | Funkcije                      | 4 | 3 | 1 | 4 |   | 4 |   | 4 |   | 4 |   | 4 |  |
| 5.  | Gotove aplikacije             | 1 |   | 1 |   | 1 |   | 1 |   | 1 |   | 1 | 1 |  |
| 6.  | Log. Uvjeti                   | 3 | 1 | 2 | 3 |   | 3 |   | 3 |   | 3 |   | 3 |  |
| 7.  | Matrica slajda                | 2 |   | 2 | 2 |   | 2 |   | 2 |   | 2 |   | 2 |  |
| 8.  | Objektno programiranje        | 4 | 4 |   | 4 |   | 4 |   | 4 |   | 4 |   | 4 |  |
| 9.  | Obrazac (Word)                | 2 |   | 2 | 2 |   | 2 |   | 2 |   | 2 |   | 2 |  |
| 10. | Operativni sustav             | 1 | 1 |   | 1 |   | 1 |   | 1 |   | 1 |   | 1 |  |
| 11. | Predložak                     | 2 |   | 2 | 2 |   | 2 |   | 2 |   | 2 |   | 2 |  |
| 12. | Skupno pismo                  | 2 |   | 2 | 2 |   | 2 |   | 2 |   | 2 |   | 2 |  |
| 13. | SQL upit                      | 2 | 2 |   | 2 |   | 2 |   | 2 |   | 2 |   | 2 |  |
| 14. | Tip podatka                   | 4 | 4 |   | 4 |   | 4 |   | 4 |   | 2 | 2 | 4 |  |
| 15. | Varijable                     | 5 | 5 |   | 4 | 1 | 5 |   | 5 |   | 5 |   | 5 |  |
| 16. | Vremenska složenost algoritma | 3 |   | 3 | 3 |   | 3 |   | 3 |   | 3 |   | 3 |  |

Među temeljnim konceptima koji su prepoznati od strane ispitanika su "Funkcije", "Objektno programiranje", "Tip podatka" i "Varijable". Ti koncepti su označeni kao ključni za razumijevanje informatike, s visokim brojem glasova za povezanost i transformativnost. Na primjer, "Funkcije" su istaknute kao temeljni koncept od strane 4 od 4 ispitanika, što sugerira na njihovu važnost u edukacijskom kontekstu.

Među zahtjevnim konceptima, "Algoritam", "Logički uvjeti" i "SQL upit" su identificirani kao izazovni za učenike. "Algoritam" je prepoznat kao zahtjevan koncept od strane 3 od 4 ispitanika, dok je "Logički uvjeti" prepoznat kao zahtjevan od strane 3 od 3 ispitanika. Ovi koncepti zahtijevaju dodatnu pažnju i podršku kako bi se učenicima olakšalo razumijevanje.

Koncepti kao što su "Funkcije", "Objektno programiranje" i "Tip podatka" su prepoznati kao visoko povezani s drugim informatičkim konceptima. To ukazuje na njihovu važnost u razvoju temeljnog razumijevanja informatičkih principa i tehnologija.

"Objektno programiranje" je identificirano kao koncept koji ima značajan utjecaj na transformaciju razumijevanja učenika. Kada učenici savladaju ovaj koncept, to često rezultira dubljim razumijevanjem programiranja i struktura podataka.

Koncepti kao što su "Funkcije" i "Tip podatka" imaju visoku ireverzibilnost, što znači da učenici rijetko zaboravljaju ove koncepte nakon što ih nauče. Ovo je važno za dugoročno usvajanje znanja i vještina u informatičkom obrazovanju.

"Vremenska složenost algoritma" je identificirana kao koncept s određenom razinom ograničenosti, što sugerira potrebu za redovitim ponavljanjem i praktičnim primjenama kako bi se učenicima olakšalo održavanje razumijevanja ovog koncepta.

Među konceptima koji nisu prepoznati kao temeljni ili zahtjevni su "Dijelovi računala", "Formule", "Gotove aplikacije", "Matrica slajda", "Obrazac (Word)", "Operativni sustav", "Predložak", "Skupno pismo" i "Vremenska složenost algoritma". Ovi koncepti nisu dobili visok broj glasova kao ključni za razumijevanje informatike ili kao izazovni za učenike, što sugerira da možda ne zahtijevaju jednaku razinu fokusa i resursa u obrazovnom programu kao oni koje su ispitanici prepoznali kao temeljne ili zahtjevne.

Nakon predstavljanja koncepta, uslijedila je rasprava među ispitanicima.

Tijekom rasprave, ispitanici su zaključili da mnogi od predloženih koncepta ne ispunjavaju kriterije za zahtjevne koncepte, što ukazuje na izazove u prepoznavanju karakteristika koncepta i razumijevanju samog pojma koncepta praga.

Rasprava je dovela do isključivanja dijela predloženih koncepta koji nisu zadovoljili potrebne kriterije.

U sljedećoj tablici možemo vidljivi predložene koncepte praga te rezultati glasovanja za svaki predloženi koncept.

*Predloženi koncepti praga (SŠ)*

| PREDLOŽENI KONCEPTI           | DA | NE |
|-------------------------------|----|----|
| Tipovi podataka               | 14 | 4  |
| Algoritam                     | 14 | 2  |
| Predložak                     | 11 | 6  |
| Varijabla                     | 18 | 0  |
| Vremenska složenost algoritma | 5  | 0  |
| Objektno programiranje        | 18 | 0  |
| Naredbe ponavljanja           | 19 | 0  |
| Potprogrami                   | 17 | 1  |
| Rekurzija                     | 5  | 12 |
| Logički uvjeti                | 19 | 0  |

Analizirajući rezultate glasovanja, identificirani su koncepti praga na temelju značajne podrške dobivene od sudionika, a pri tom izdvajajući one koji su jasno dobili veću podršku kao ključne koncepte praga u nastavi informatike.

Naredbe ponavljanja: Dobio je izuzetno visok broj glasova - 19 glasova "ZA" i nijedan glas "NE".

Logički uvjeti: Također je dobio velik broj glasova - 19 glasova "ZA" i nijedan glas "NE".

Varijable: Dobio je 18 glasova "ZA" i nijedan glas "NE".

Objektno programiranje: Dobio je 18 glasova "ZA" i nijedan glas "NE".

Potprogrami: Dobio je 17 glasova "ZA" i samo 1 glas "NE".

Provođenje ovoga istraživanja pomoglo je razumjeti kako nastavnici informatike srednjih škola percipiraju ključne koncepte praga u nastavi informatike. Analizirajući njihove odgovore dobivene su jasnije slike i širi uvid te zaključak kako je razumijevanje i identifikacija koncepata praga među nastavnicima ključni faktor za pridonosenje boljem obrazovanju u području informatike. Također, provođenjem ove ankete dovelo je do moguće identifikacije koncepata praga u nastavi informatike.

#### **4.4. Završni pregled rezultata istraživanja**

Istraživanje stavova o konceptima praga provedeno je među nastavnicima informatike u osnovnim i srednjim školama s ciljem identifikacije ključnih koncepata praga u nastavi informatike. U istraživanju je sudjelovalo ukupno 69 nastavnika, od kojih 53 iz osnovnih škola i 16 iz srednjih škola.

Analiza prikupljenih podataka pokazala je nekoliko ključnih nalaza. Demografska struktura ispitanika ukazuje nam na većinski broj ženskih ispitanika i to 73,58% u osnovnim školama, te 68,75% u srednjim školama. Prosječna starost ispitanika u osnovnim školama je 39 godina, dok je u srednjim školama 47 godina. Ispitanici imaju različite godine radnog staža, s prosječnim stažem od 9,88 godina u osnovnim školama i 17,6 godina u srednjim školama.

Što se tiče predmeta koje predaju, u osnovnim školama većina nastavnika predaje isključivo informatiku, dok neki predaju i matematiku, engleski jezik, fiziku ili tehničku kulturu. U srednjim školama, nastavnici predaju informatiku, matematiku, računalstvo, te programiranje za web i baze podataka.

Identifikacija ključnih koncepata praga pokazala je da su u osnovnim školama najviše ocijenjeni koncepti logičkih uvjeta, adresiranja ćelije, dijagrama tijeka, organizacije podataka i varijabli. U srednjim školama, najviše ocjene dobili su koncepti naredbi ponavljanja, logičkih uvjeta, varijabli, objektnog programiranja i potprograma.

Istraživanje je pokazalo da nastavnici informatike prepoznaju slične ključne koncepte praga bez obzira na razinu obrazovanja (osnovna ili srednja škola). Međutim, određene razlike u prioritetima ukazuju na specifične potrebe i izazove s kojima se nastavnici susreću na svakoj razini obrazovanja.

Ovo istraživanje predstavlja važan korak ka boljem razumijevanju i unaprjeđenju nastave informatike u osnovnim i srednjim školama, te može poslužiti kao temelj za buduće inicijative u obrazovnom sustavu. Također, predloženi koncepti jasno pokazuju da nastavnicima nije lako identificirati koncepte praga.

## 5. Nastavne aktivnosti za usvajanje koncepata praga

U ovom dijelu rada predstavljeni su prijedlozi nastavnih aktivnosti koje se mogu koristiti kako bi se učenicima pomoglo u razumijevanju i usvajanju koncepata praga iz informatike. Cilj ovih aktivnosti je ne samo usvajanje teorijskih znanja, već i razvijanje vještina primjene tih znanja u stvarnim situacijama. Učenici će kroz raznolike aktivnosti moći razvijati logičko razmišljanje, analitičke sposobnosti i kreativnost, što će im biti korisno ne samo u informatici, već i u drugim područjima.

Ove aktivnosti uključuju rad na računalima, grupne projekte, individualne zadatke i edukativne igre koje su prilagođene dobi i predznanju učenika. Posebna pažnja posvećena je osiguravanju da svaki učenik ima priliku aktivno sudjelovati i učiti kroz praktični rad, što pridonosi boljem razumijevanju i dugoročnom pamćenju naučenog gradiva. Kako bi sam proces učenja učenici što bolje prihvatili, te ostali motivirani i postigli što bolje rezultate izabran je pristup temeljen na igri. Obrazovne igre potiču intelektualni razvoj djeteta i objedinjuju učenje i zabavu. Kroz primjenu programskog jezika Scratch na edukativan i zabavna način provesti će se nastavne aktivnosti koje će učenicima olakšati usvajanje koncepata praga iz nastave informatike.

### 5.1. Obrazovne igre

Krajem 20. stoljeća znanstvenici su istaknuli potencijal računala i hipermedija kao alata za učenje. Fokusirali su se na važnost interaktivnosti u obrazovnom materijalu kako bi potaknuli proces učenja. Ideja je bila pružiti raznolike prezentacije znanja i omogućiti primjenu tog znanja u virtualnom okruženju. Interaktivnost je ključna, a može se postići kroz različite koncepte poput interaktivnosti preko objekata, linearnu interaktivnost te interaktivnost putem hiperlinkova.

Uvođenje digitalnih igara u proces učenja može biti korisno kao dodatak učenju u učionici. Cilj je prilagoditi nastavu s pomoću informacijske tehnologije i omogućiti učenicima stjecanje vještina relevantnih za poslovni svijet. Kroz digitalne igre, učenici bi trebali moći primijeniti svoje znanje, učiti po potrebi te stjecati iskustva u virtualnom okruženju koje može oblikovati njihovo ponašanje i razmišljanje (Pivec, 2006).

Stoga obrazovne igre predstavljaju važan aspekt u razvoju djece, pružajući im priliku da kroz igru usvoje ključne vještine i znanja. Osim što potiču intelektualni razvoj, igra ima važnu ulogu u emocionalnom i socijalnom napretku djeteta. Sve više istraživanja pokazuje da integracija igara u formalno obrazovanje pozitivno utječe na motivaciju učenika.

Igre ne omogućuju samo stjecanje znanja, već potiču razvoj različitih sposobnosti kao što su rješavanje problema, kritičko razmišljanje i kreativnost. Kroz igru, učenici postaju angažiraniji i motiviraniji za učenje, jer im se pruža prilika da istovremeno uče i zabavljaju se.

Pedagoške prednosti igre uključuju suradničko učenje, eksperimentalno učenje te učenje putem autentičnih situacija.

Kada je riječ o računalnim igrama, ključne značajke uključuju definirane ciljeve, strukturu pravila prilagođenu igri, korištenje virtualnog svijeta i umjetne inteligencije te interakciju između igrača. Važno je naglasiti da su obrazovne igre, za razliku od ostalih igara, usmjerene na postizanje određenih obrazovnih ishoda, čime se ističe njihova obrazovna komponenta.

Digitalne obrazovne igre postaju sve važnije s obzirom na uobičajenu upotrebu informacijsko-komunikacijske tehnologije u školama. Njihova primjena doprinosi povećanju motivacije učenika, poboljšanju raznih vještina te omogućuje ponavljanje i uvježbavanje gradiva na zanimljiv i interaktivan način (Hoić-Božić, Holenko Dlab, 2021.).

## **5.2. Pristup temeljen na igri**

Učenje putem igara, poznato i kao GBL (Game-Based Learning), koristi digitalne igre kako bi potaknulo motivaciju učenika i omogućilo postizanje obrazovnih ciljeva. Ključno je napomenuti da igre trebaju imati jasno definirane ciljeve učenja te poticati razvoj važnih vještina, ali istovremeno pružati zabavno iskustvo.

Studije su pokazale da su učenici izuzetno motivirani kada se obrazovni sadržaji prezentiraju kroz računalne igre. Međutim, važno je pažljivo planirati metodologije korištenja igara u nastavi. Neki od predloženih pristupa uključuju korištenje igara kao motivacije prije ili tijekom lekcije, kao grupne aktivnosti u razredu ili kao samostalne aktivnosti kod kuće.

Prije igranja igre, važno je provesti uvodne aktivnosti koje će učenicima predstaviti pravila, ciljeve i kontekst igre. Tijekom same igre, aktivnosti su manje učestale, ali mogu biti korisne kako bi se pomoglo učenicima koji naiđu na poteškoće. Nakon igranja igre, rasprava s ostalim učenicima i nastavnikom omogućuje refleksiju o iskustvima i naučenom te razmatranje primjene novih vještina i znanja.

Važno je razlikovati igre kao aktivnosti koju učenici koriste tijekom učenja i dizajniranja igara u kojem učenici aktivno sudjeluju u procesu razvoja igara. Primjena igara u obrazovanju može biti učinkovita, ali izrada obrazovnih igara zahtijeva vremenski i financijski angažman. Ipak, postoje alati koji olakšavaju proces dizajniranja igara, poput koncepta igrifikacije (Hoić-Božić, Holenko Dlab, 2021.).

Programski jezik Scratch objedinjuje igre kao aktivnost koju će učenici koristiti tijekom procesa učenja, ali i kao dizajniranje igara u kojima oni aktivno sudjeluju. Upravo radi navedenog, programski jezik Scratch može poslužiti kao odličan primjer za pristup učenju temeljenom na igri.

## **5.3. Programski jezik Scratch**

Scratch je programski jezik prilagođen djeci od 8 do 16 godina, stvoreno od strane Lifelong Kindergarten grupe. Osmišljen je kako bi učenje programiranja bilo zabavno i dostupno. Kroz Scratch, djeca mogu izrađivati interaktivne priče, igre, simulacije i animacije koristeći blokove koji se slažu poput slagalica. Ovi blokovi se organiziraju u grupe skripti, a korisnici ih povlače i slažu

na radno područje (Resnick, Maloney, Monroy-Hernández, Rusk, Eastmond, Brennan, Kafai, 2009.).

Jedna od značajnih prednosti Scratcha je mogućnost prilagodbe boja i zvuka. Djeca mogu stvarati vlastite vizualne elemente i zvučne efekte za svoje projekte. Osnovni princip Scratcha je "Programiraj i dijeli", koji potiče korisnike da dijele svoje radove s drugima u zajednici.

Scratch je napravljen koristeći HTML5 tehnologiju, što znači da se može koristiti na raznim preglednicima kao što su Google Chrome, MS Edge, Mozilla Firefox i Safari, ali nije kompatibilan s Internet Explorerom.

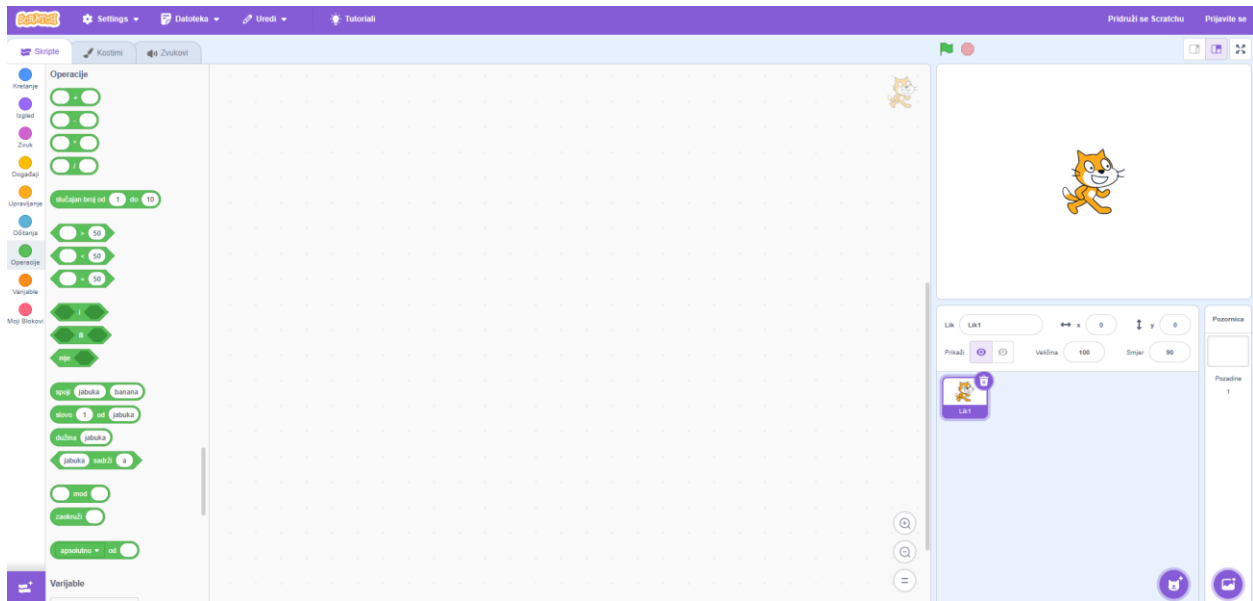
Za korištenje Scratcha, potreban je pristup samom programu, koji se može koristiti online ili preuzeti za offline rad. Programiranje u Scratchu uključuje osnovne grupe skripti koje pokrivaju razne aspekte programiranja kao što su kretanje, izgled, zvuk, događaji, kontrola, senzori, operacije i varijable.

Ovaj jednostavan i intuitivan alat omogućava djeci da razviju svoje programerske i kreativne vještine, dajući im priliku da izraze svoje ideje i stvaraju digitalne projekte (Budojević, Kanić, 2019).

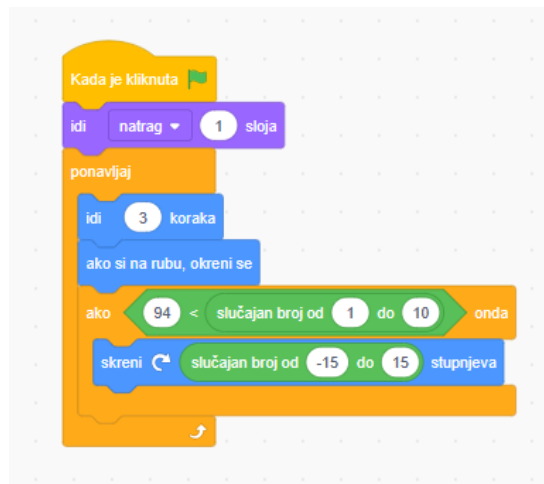
Programiranje u Scratchu je pogodno i za mlađu djecu, kao i za djecu s intelektualnim poteškoćama. Blokovi naredbi u Scratchu su dizajnirani da budu vizualno privlačni, izgledaju poput slagalica koje djeca mogu slagati kako bi stvorila razne programe. Kombiniranjem različitih blokova, učenici mogu kreirati likove koji izvode različite radnje. Svaki blok ima svoju boju, što dodatno olakšava programiranje, posebno za mlađu djecu i djecu s intelektualnim poteškoćama.

Ovakav način rada, gdje učenici slažu i kombiniraju blokove naredbi, pomaže u razvijanju viših kognitivnih sposobnosti, potičući logičko i kompleksno razmišljanje. Upravo na Slici 2 možemo pogledati kombinaciju bloka naredbi koje uključuju kombinaciju blokova Operacije (Slika 1.) te Upravljanja što nam je izuzetno zanimljivo prilikom rada sa logičkim uvjetima. Učenici mogu improvizirati i stvarati svoje priče ili igre, ali i rješavati zadane zadatke. Zbog svih ovih razloga, korištenje Scratcha u radu s mlađom djecom i djecom s intelektualnim poteškoćama je izuzetno korisno i važno (Budojević, Flisar Odorčić, Kanić, 2019).





Slika 6. Sučelje programa Scratch



Slika 7. Prikaz bloka naredbi (Scratch)

## 5.4. Primjer aktivnosti za lakše savladavanje koncepta praga - logički uvjeti

U sljedećem dijelu rada biti će predstavljen konkretan primjer iz nastave prilagođen za 4 nastavna sata. Aktivnosti su namijenjene učenicima četvrtog razreda osnovnih škola. Cilj je kroz ove aktivnosti učenike upoznati s konceptom praga u programiranju, koristeći praktične zadatke u kombinaciji sa igrama. Aktivnosti će uključivati korištenje jednostavnih uvjeta u programskom jeziku kako bi učenici shvatili kako logički uvjeti funkcioniraju. Za ove aktivnosti potrebni su računala.

### Odgojno-obrazovni ishodi:

B.4.2. Učenik rješava složenije logičke zadatke s računalom ili bez uporabe računala.

### Odgojno-obrazovni ishodi - razrada:

Učenik će moći:

- Rješavati složenije logičke zadatke.
- Primijeniti logičke uvjete.
- Analizirati i kritički razmišljati o različitim pristupima rješavanju logičkih problema.
- Demonstrirati vještine u rješavanju složenijih logičkih problema kroz praktične primjene.

### Očekivanja međupredmetnih tema:

#### *Učiti kako učiti*

uku B.2.4. Samovrednovanje/samoprocjena Na poticaj učitelja, ali i samostalno, učenik samovrednuje proces učenja i svoje rezultate te procjenjuje ostvareni napredak.

#### *IKT*

ikt A.2.1. Učenik prema savjetu odabire odgovarajuću digitalnu tehnologiju za obavljanje zadatka

ikt A.2.2. Učenik se samostalno koristi njemu poznatim uređajima i programima

### Povezanost s drugim nastavnim predmetima:

Matematika

#### ***Nastavna načela:***

Načelo individualizacije i socijalizacije: Rad se prilagođava napretku svakog učenika, nudeći dodatnu podršku onima koji je trebaju. Učenicima koji brže savladavaju gradivo pripremaju se dodatni izazovi. Potičemo suradnju umjesto konkurencije kako bismo razvili suradnički odnos među učenicima.

Načelo zornosti: Koristimo sliku i grafičke elemente kako bismo bolje vizualizirali koncepte i procese. Ovakav pristup omogućuje učenicima da bolje razumiju logičke uvjete i njihovu primjenu u programiranju.

Načelo postupnosti: Gradivo se prezentira postupno, počevši s osnovnim konceptima logičkih uvjeta te postupno prelazeći na složenije. Prilagođavamo tempo učenja tako da se osigura temeljito razumijevanje svake faze gradiva prije prelaska na sljedeću.

Načelo primjerenosti: Aktivnosti su prilagođene dobi i predznanju učenika četvrtog razreda. Strukturiramo zadatke i objašnjenja tako da budu primjereni njihovom razvojnom stupnju kako bi im omogućili uspješno savladavanje logičkih uvjeta u programiranju.

#### ***Nastavne metode:***

Metoda razgovora: Potičemo diskusiju između učenika kako bi razmijenili mišljenja i ideje te bolje razumjeli logičke uvjete.

Metoda demonstracije: Demonstriramo primjere i primjenu logičkih uvjeta kako bi učenici vizualno i praktično doživjeli koncepte i tehnike koje se uče.

Metoda usmenog izlaganja: Koristimo usmeno izlaganje kako bismo opisali i objasnili logičke uvjete, pružajući dodatne informacije i primjere učenicima. Ova metoda omogućuje bolje razumijevanje i interpretaciju logičkih uvjeta.

Učenici će ih povezati s matematičkim konceptima kod usporedbe s brojeva.

### ***Vrednovanje:***

#### Vrednovanje za učenje

Praćenje snalaženja u samostalnom i grupnom radu gdje nastavnik prati kako se pojedini učenik odnosi prema svojim razrednim kolegama u timu te općenito kako se učenici odnose jedni prema drugima.

Prilikom izrade i demonstracije igre u Scratchu nastavnik prati aktivnost učenika te im daje informacije što je dobro, a što bi mogli promijeniti ili popraviti. Ukazuje im na važnost korištenja logičkih uvjeta unutar same igre.

Prilikom rješavanja igre memorije nastavnik komentira parove koje su učenici međusobno spojili odnosno pridružili.

#### Vrednovanje kao učenje

Izlazna kartica u završnom djelu sata.

Vrednovanje naučenoga provoditi će se slijedeći sat.

### ***Oblici rada:***

Frontalna nastava, rad u paru, individualni rad.

### ***Nastavna sredstva:***

Računalo, projektor, programski editor (Scratch), ploča, kreda.

### **Razreda aktivnosti**

#### Aktivnost 1: Uvod u logičke uvjete (1 sat)

Učenik će moći:

- Raspravljati o logičkim uvjetima kroz jednostavne primjere iz svakodnevnog života

Nastavnik daje primjere logičkih uvjeta koristeći jednostavne primjere iz svakodnevnog života. Na primjer:

*"Ako je vani kiša, uzmi kišobran."*

*"Ako je u hladnjaku mlijeko, možemo napraviti doručak."*

*"Ako temperatura padne ispod nule, voda se smrzava."*

*"Ako imaš sve zadaće gotove, možeš se igrati na računalu."*

*"Ako auto ima dovoljno goriva, može voziti."*

Nakon tih primjera iz svakodnevnog života, nastavnik prelazi na primjere logičkih uvjeta u računalnim programima:

*"Ako korisnik unese ispravan lozinku, prijava je uspješna."*

*"Ako broj bodova prijeđe određeni prag, igrač prelazi na sljedeći nivo."*

*"Ako temperatura računala prijeđe kritičnu vrijednost, ventilator se uključuje."*

Nastavnik također može demonstrirati jednostavne primjere pomoću aplikacija ili online alata, pokazati kako logički uvjeti funkcioniraju u stvarnom vremenu.

Nastavnik zatim dijeli učenike u nekoliko manjih grupa. Svaka grupa se može sastojati od najviše četiri igrača. Također, svaka je grupa dobila svoj špil Uno karata. Nastavnik ukratko ponavlja pravila igre Uno te zadaje učenicima zadatak:

*"Pronađite logičke uvjete u toku igranja Una. Kada ih pronađete zabilježite ih."*

Učenici igraju igru Uno te prepoznaju i bilježe logičke uvjete. Na kraju sata grupe diskutiraju o prikupljenim podacima i primjerima logičkih uvjeta u igri. Nastavnik zatim pojašnjava učenicima kako su jednostavni logički uvjeti vezani za poteze koji su proizašli radi samo jednog specifičnog uvjeta te daje primjer:

*"Ako igrač nema kartu s određenom bojom ili brojem tada mora izvući novu kartu."*

Zatim učenicima pojašnjava kako se složeni logički uvjeti u igri Uno odnose na radnje koje ovise o više uvjeta. Nastavnik govori učenicima:

*"Ako igrač baci +4 kartu, tada i mijenja boju, na primjer u crvenu, pa sljedeći igrač mora izvući 4 karte i sljedeća bačena mora biti crvene boje, a ako igrač nema kartu crvene boje, tada mora izvući kartu iz špila sve dok ne dobije kartu crvene boje."*

Ovo je primjer složenog uvjeta u igri Uno. Nastavnik zatim sumira ključne točke ističući važnost razumijevanja logičkih uvjeta.

## Aktivnost 2: Lik koji mijenja boju (1 sat)

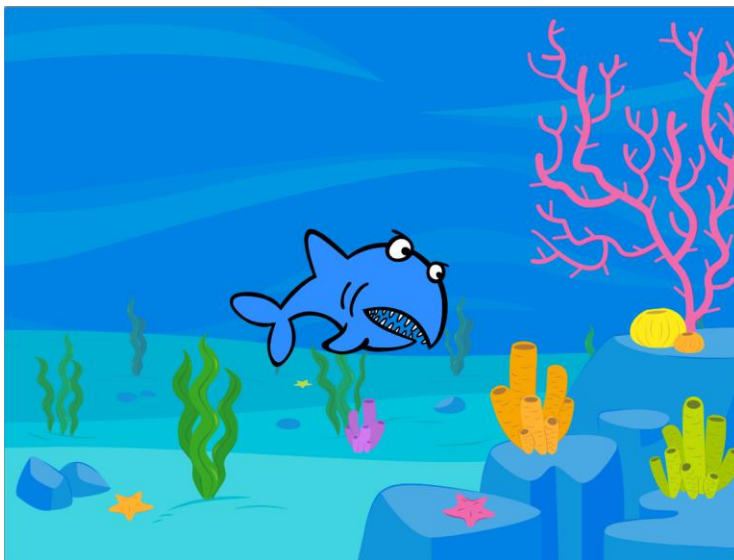
Učenik će moći:

- Primijeniti logičke uvjete za promjenu boje lika na ekranu

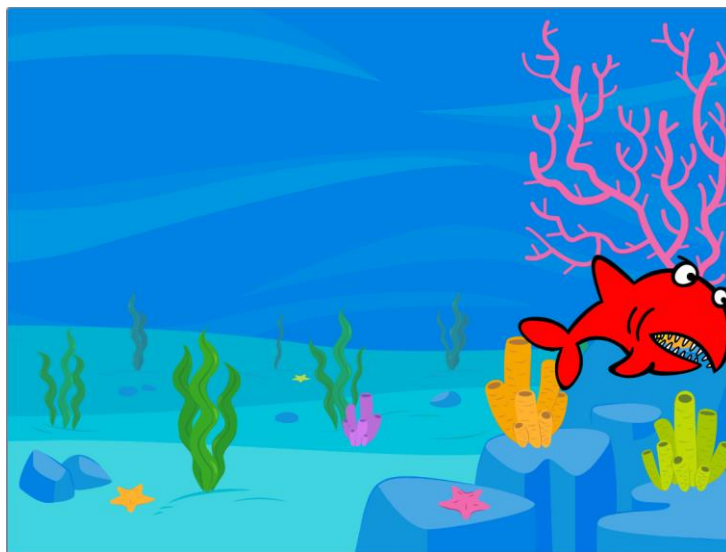
*Opis aktivnosti:* Učenici rade u paru ili malim grupama i koriste jednostavan program za programiranje (Scratch) kako bi izradili lika koji mijenja boju kada dođe do ruba ekrana. Nastavnik

započinje aktivnost demonstracijom osnovnih naredbi u odabranom programu, objašnjavajući kako se koriste uvjetne naredbe za promjenu boje lika.

Nastavnik će tada pokazati primjer učenicima. Lik morskog psa dolazi do ruba i mijenja boju. Onog trena kada dotakne rub, promijenit će se boja. Nastavnik pokazuje primjer igre u Scratchu učenicima (Slika 8 i 9).



Slika 8 Promjena boje u Scratchu - 1



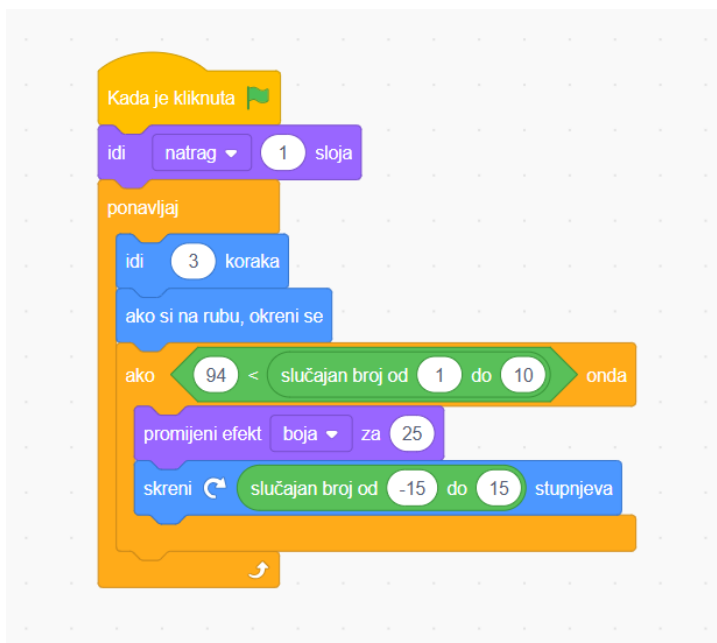
Slika 9 Promjena boje u Scratchu - 2

Učenici postavljaju početne uvjete za lika, npr. ako lik dodirne rub ekrana, boja se mijenja iz crvene u plavu. Nastavnik pomaže učenicima u pisanju i testiranju koda, osiguravajući da svi učenici razumiju kako i zašto lik mijenja boju. Učenici eksperimentiraju s različitim bojama i uvjetima, te dodaju dodatne uvjete ako žele. Naprimjer:

*"Ako lik dodirne lijevi rub, promijeni boju u zelenu."*

*"Ako lik dodirne desni rub, promijeni boju u žutu."*

Nastavnik pokazuje učenicima blok naredbi u Scratchu (Slika 10) u kojem mogu vidjeti prikaz promjene boju uz pomoć logičkih uvjeta.



Slika 10 Prikaz primjene logičkih uvjeta u Scratchu

Na kraju aktivnosti, učenici međusobno dijele svoja iskustva i donose zaključke o postavljanju uvjeta vezanih za promjenu izgleda lika. Nastavnik potiče učenike da razmišljaju o složenijim logičkim uvjetima koje bi mogli primijeniti na ovome primjeru.

### Aktivnost 3: Lik sa složenijim uvjetima (1 sat)

Učenik će moći:

- Primijeniti složenije logičke uvjete za ponašanje lika na ekranu

*Opis aktivnosti:* Učenici proširuju prethodni projekt tako da njihov lik reagira na složenije uvjete u okruženju. Nastavnik objašnjava nove uvjete koje će lik imati, koristeći primjere:

*"Ako lik morskog psa dođe do ribica i pojede ih, narast će za 50%."*

*"Ako dođe do grebena, vratit će se na prijašnju veličinu."*

*"Ako dođe do otpadaka, smanjit će se za 50%."*

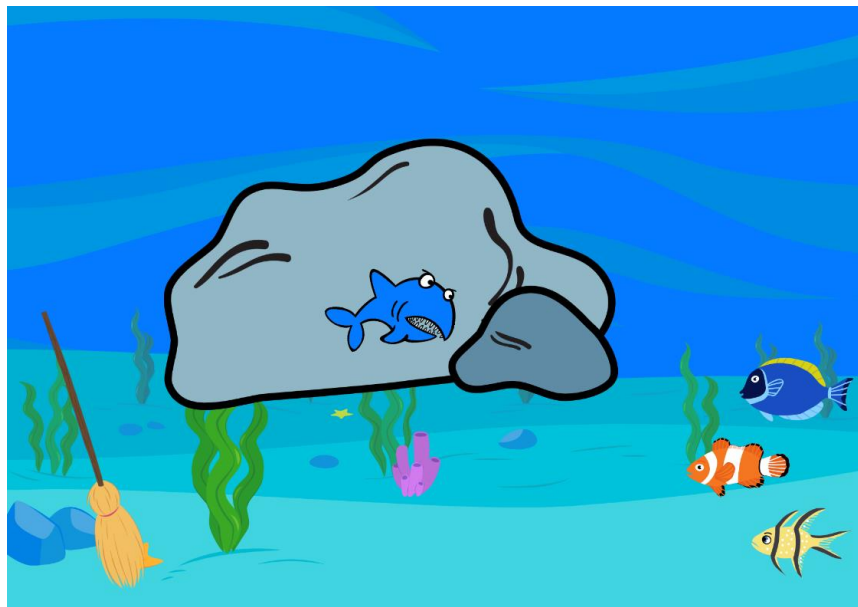
Nastavnik učenicima zadaje zadatak:

*"Otvorite prijašnji projekt. Liku kojega smo kreirali na prethodnoj aktivnosti dodati ćemo više uvjeta. Na pozadinu dodajte s desne strane lik ribica, s lijeve strane otpadak po želji i na sredini dodajmo greben. Ako lik dotakne ribice povećajte mu veličinu za 50%. Ako lik dotakne greben vraća se na prijašnju veličinu. Ako lik dotakne otpatke smanjit će se za 50%."*

Nastavnik pokazuje učenicima primjer koda za promjenu veličine (Slika 9) i primjer rasporeda likova na pozadini (Slika 11).



Slika 11 Prikaz promjene veličine u Scratchu uz pomoć logičkih uvjeta



Slika 12 Prikaz organizacije pozadine s objektima za promjenu veličine

Učenici koriste program za programiranje da bi implementirali ove uvjete. Nastavnik pomaže u definiranju novih uvjetnih naredbi i testiranju koda. Svaka grupa radi na tome da njihov lik pravilno reagira na različite uvjete u okruženju, koristeći različite scenarije unutar programa. Učenici mogu dodati i svoje uvjete, ovo su primjeri:

*"Ako lik dodirne morsku travu, smanji brzinu kretanja."*

Učenici dijele svoje projekte i demonstriraju kako lik reagira na različite uvjete, dok nastavnik i učenici daju povratne informacije. Nastavnik može postaviti dodatna pitanja kako bi potaknuo učenike na razmišljanje:

"Što bi se dogodilo ako dodamo još jedan uvjet?"

"Kako bismo mogli koristiti ove uvjete u stvarnim programima?"

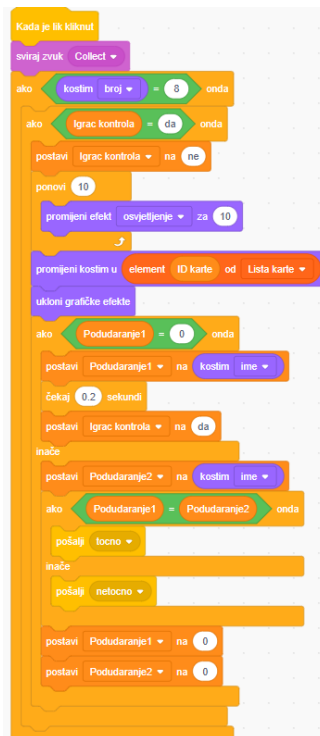
#### Aktivnost 4: Igra Memorije (1 sat)

Učenik će moći:

- Riješiti zadatak s logičkim uvjetima kroz igru memorije

*Opis aktivnosti:* Nastavnik demonstrira igru memorije izrađenu u Scratchu, objašnjavajući kako igra radi i koje logičke uvjete koristi. Objašnjava da se igra sastoji od kartica koje učenici trebaju upariti. Ako učenik odabere dvije iste slike, ispunjava se uvjet i slike nestaju, a broj bodova se povećava. Ako odabere dvije različite slike, one se vraćaju na početnu poziciju, a bodovi ostaju isti.

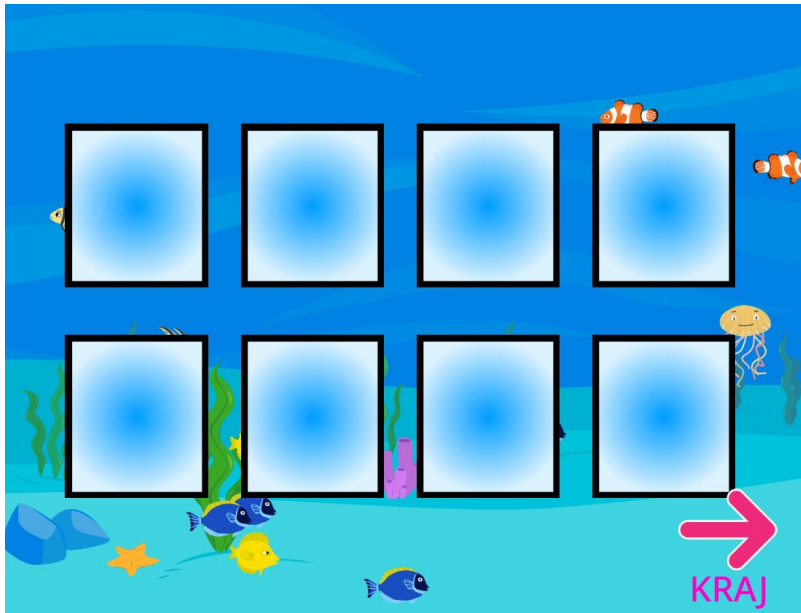
Nastavnik pokazuje primjer gotove igre (Slika 13) učenicima govoreći: "Pogledajte ovu igru. Kada odaberemo dvije kartice, one se okrenu i pokažu slike. Ako su slike iste, kartice nestaju i dobijemo bodove. Ako su slike različite, kartice se vraćaju na početnu poziciju i bodovi se ne mijenjaju."



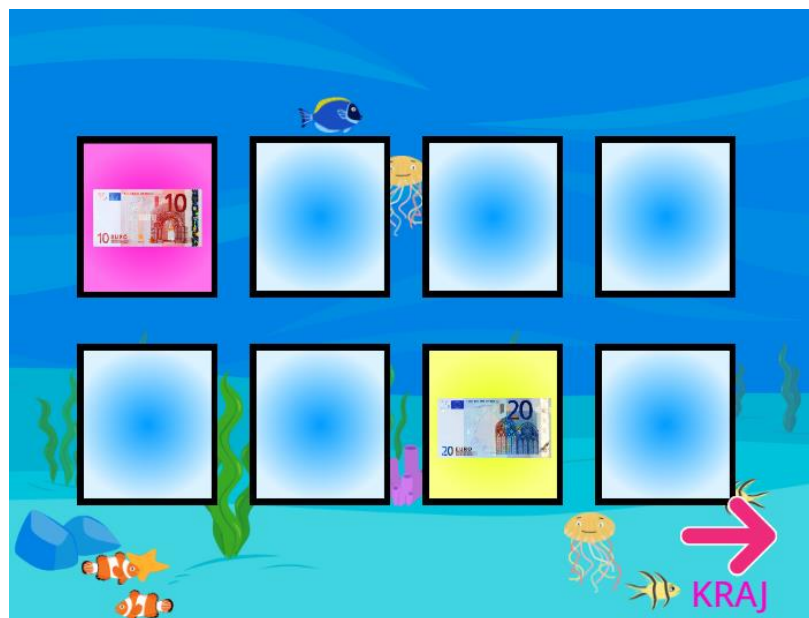
Slika 13 Prikaz korištenja složenih uvjeta u Scratchu

Učenici zatim igraju igru memorije, ponavljajući naučeno gradivo o logičkim uvjetima (Slika 14, 15 i 16). Igra im pomaže razumjeti kako se logički uvjeti primjenjuju u praktičnim situacijama.





Slika 14 Igra Memorije u Scratchu



Slika 15 Igra Memorije u Scratchu 2



Slika 16 Igra Memorije u Scratchu 3

Nakon što su odigrali igru, učenici raspravljaju o tome kako su logički uvjeti korišteni u igri i kako su utjecali na tijek igre. Nastavnik sumira naučeno i ističe važnost razumijevanja logičkih uvjeta u računalnim programima.

Nastavnik također postavlja dodatna pitanja kako bi produbio razumijevanje:

*"Kako bismo mogli promijeniti igru da bude teža?"*

*"Što bi se dogodilo ako dodamo više kartica?"*

*"Kako bismo koristili logičke uvjete da dodamo vrijeme kao ograničenje u igri?"*

Učenici raspravljaju o ovim pitanjima i dijele svoje ideje s razredom. Ove aktivnosti omogućuju učenicima da kroz praktične primjere i jednostavne projekte razumiju i primjene koncept logičkih uvjeta.

## 5.5. Primjer aktivnosti za lakše savladavanje koncepta praga - naredbe ponavljanja

U sljedećem dijelu rada biti će predstavljen konkretan primjer iz nastave prilagođen za 4 nastavna sata. Aktivnosti su namijenjene učenicima četvrtog razreda osnovnih škola. Cilj je kroz ove aktivnosti učenike upoznati s konceptom naredbi ponavljanja u programiranju, koristeći praktične zadatke u kombinaciji sa igrama. Aktivnosti će uključivati korištenje jednostavnih naredbi ponavljanja u programskom jeziku Scratch da bi učenici shvatili kako ove naredbe funkcioniraju. Za ove aktivnosti potrebna su računala.

Odgojno-obrazovni ishodi:

- B.4.2. Učenik rješava složenije logičke zadatke s računalom ili bez uporabe računala.

Odgojno-obrazovni ishodi - razrada:

Učenik će moći:

- Primijeniti naredbe ponavljanja.
- Rješavati logičke zadatke s naredbama ponavljanja koristeći programski jezik ili druge alate.
- Analizirati i kritički razmišljati o različitim pristupima rješavanju zadataka s naredbama ponavljanja.
- Demonstrirati vještine u rješavanju složenijih logičkih problema kroz praktične primjene.

Očekivanja međupredmetnih tema:

*Učiti kako učiti:* Učenik samovrednuje proces učenja i svoje rezultate te procjenjuje ostvareni napredak.

*IKT:* Učenik prema savjetu odabire odgovarajuću digitalnu tehnologiju za obavljanje zadatka; samostalno koristi njemu poznate uređaje i programe.

Povezanost s drugim nastavnim predmetima:

Matematika

***Nastavna načela:***

*Načelo individualizacije i socijalizacije:* Rad prilagođava se napretku svakog učenika, nudeći dodatnu podršku onima koji je trebaju. Učenicima koji brže savladavaju gradivo pripremaju se dodatni izazovi. Potičemo suradnju umjesto konkurencije kako bismo razvili suradnički odnos među učenicima.

*Načelo zornosti:* Koristimo sliku i grafičke elemente kako bismo bolje vizualizirali koncepte i procese. Ovakav pristup omogućuje učenicima da bolje razumiju naredbe ponavljanja i njihovu primjenu u programiranju.

*Načelo postupnosti:* Gradivo se prezentira postupno, počevši s osnovnim konceptima naredbi ponavljanja te postupno prelazeći na složenije. Prilagođavamo tempo učenja tako da se osigura temeljito razumijevanje svake faze gradiva prije prelaska na sljedeću.

*Načelo primjerenosti:* Aktivnosti su prilagođene dobi i predznanju učenika četvrtog razreda. Strukturiramo zadatke i objašnjenja tako da budu primjereni njihovom razvojnom stupnju kako bi im omogućili uspješno savladavanje naredbi ponavljanja u programiranju.

***Nastavne metode:***

*Metoda razgovora:* Potičemo diskusiju između učenika kako bi razmijenili mišljenja i ideje te bolje razumjeli naredbe ponavljanja.

*Metoda demonstracije:* Demonstriramo primjere i primjenu naredbi ponavljanja kako bi učenici vizualno i praktično doživjeli koncepte i tehnike koje se uče.

*Metoda usmenog izlaganja:* Koristimo usmeno izlaganje kako bismo opisali i objasnili naredbe ponavljanja, pružajući dodatne informacije i primjere učenicima. Ova metoda omogućuje bolje razumijevanje i interpretaciju naredbi ponavljanja.

### ***Vrednovanje:***

*Vrednovanje za učenje:* Praćenje snalaženja u samostalnom i grupnom radu gdje nastavnik prati kako se pojedini učenik odnosi prema svojim razrednim kolegama u timu te općenito kako se učenici odnose jedni prema drugima. Prilikom izrade i demonstracije igre u Scratchu nastavnik prati aktivnost učenika te im daje informacije što je dobro, a što bi mogli promijeniti ili popraviti. Ukazuje im na važnost korištenja naredbi ponavljanja unutar same igre.

*Vrednovanje kao učenje:* Izlazna kartica u završnom djelu sata. Vrednovanje naučenoga provodit će se sljedeći sat.

### ***Oblici rada:***

Frontalna nastava, rad u paru, individualni rad.

### ***Nastavna sredstva:***

Računalo, projektor, programski editor (Scratch), ploča, kreda.

### ***Razrada aktivnosti***

*Aktivnost 1: Uvod u naredbe ponavljanja (1 sat)*

Učenik će moći:

- Prepoznati naredbe ponavljanja kroz igru i primjere iz svakodnevnog života.

Opis aktivnosti: Nastavnik započinje sat kratkom igrom u kojoj sudjeluju svi učenici. Učenici stoje u krugu, a nastavnik objašnjava pravila igre: jedan učenik započinje jednostavnu radnju (npr. pljeskanje rukama), a zatim svaki sljedeći učenik ponavlja tu radnju dodajući novu (npr. skakanje na mjestu). Cilj je ponavljati sve prethodne radnje i dodavati nove dok svi učenici ne sudjeluju. Na taj način učenici će intuitivno shvatiti koncept ponavljanja kroz igru.

Nakon igre, nastavnik vodi diskusiju s učenicima o tome što su radili i kako se radnje ponavljaju. Nastavnik objašnjava da su na sličan način u programiranju naredbe ponavljanja korištene za izvršavanje istih radnji više puta.

Nastavnik pojašnjava učenicima: *"Pogledajte kako smo ponavljali radnje u našoj igri. U programiranju, koristimo naredbe ponavljanja da bismo više puta izvršili iste radnje. Na primjer, možemo koristiti naredbu 'ponovi 10 puta' da bismo više puta nacrtali kvadrat na ekranu."*

Nastavnik zatim prikazuje nekoliko jednostavnih primjera naredbi ponavljanja iz svakodnevnog života:

*"Ako želiš postići dobru kondiciju, trči svaki dan 10 minuta."*

*"Ako želiš naučiti pjesmicu, ponavljaj stihove dok ih ne zapamtiš."*

Nakon tih primjera, nastavnik demonstrira jednostavne primjere naredbi ponavljanja u računalnim programima:

*"Ako korisnik pritisne tipku 'Enter', ponovi unos podataka dok ne pritisne 'Escape'."*

*"Ako broj bodova prijeđe određeni prag, ponovi zvuk pobjede tri puta."*

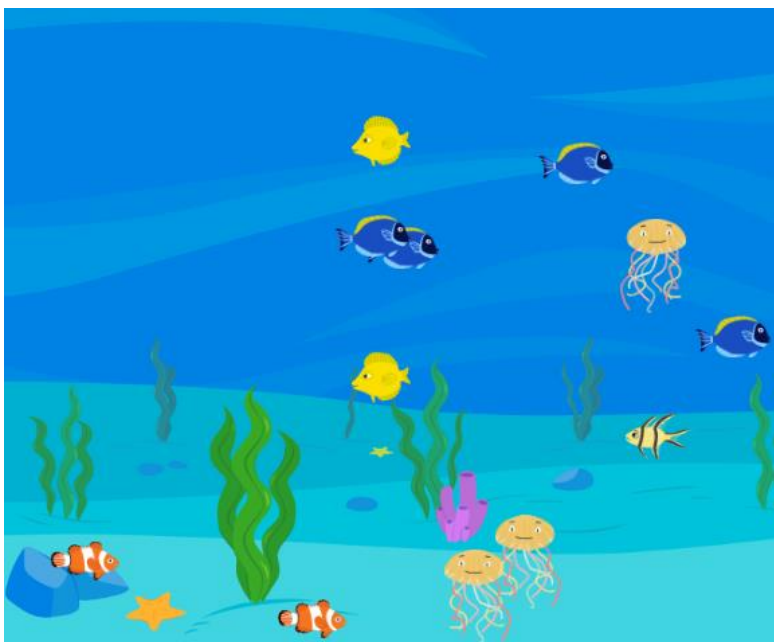
Učenici zatim u grupama raspravljaju o primjerima naredbi ponavljanja koje su vidjeli i kako se oni mogu primijeniti u različitim situacijama. Nastavnik zatim učenicima zadaje zadatak da izmisle vlastite primjere i podijele ih s razredom.

### Aktivnost 2: Lik koji se kreće u petlji (1 sat)

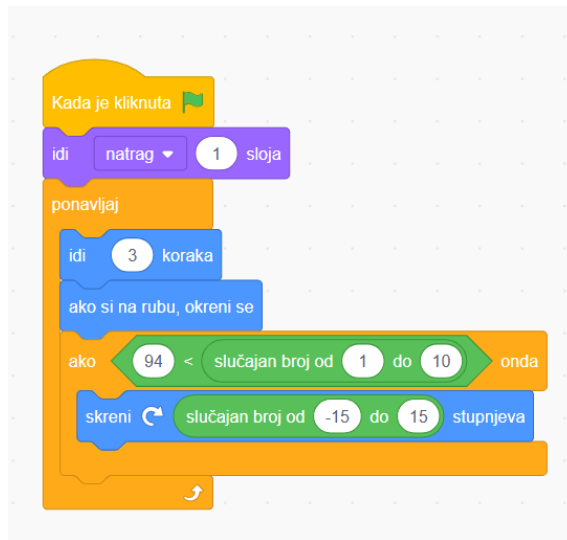
Učenik će moći:

- Primijeniti naredbe ponavljanja za kretanje lika na ekranu

*Opis aktivnosti:* Učenici rade u paru ili malim grupama i koriste jednostavan program za programiranje (Scratch) kako bi izradili lika koji se kreće u petlji. Nastavnik započinje aktivnost demonstracijom osnovnih naredbi u odabranom programu, objašnjavajući kako se koriste naredbe ponavljanja za kretanje lika. Nastavnik pokazuje učenicima primjer naredbe ponavljanja. Ribica pliva po vodi (slika 17) koristeći naredbu ponavljanja (slika 18).



Slika 17 Prikaz kretanja lika u Scratchu



Slika 18 Prikaz naredbe ponavljanja (Scratch)

Nastavnik govori učenicima: "Pogledajte, želimo da naša ribica pliva naprijed i natrag po ekranu. U Scratchu ćemo napisati kod koji kaže: Kada je kliknuta zastavica idi natrag jedan sloj (kako bi morska trava ostala na 1. sloju) te ponavljaj ovo: idi 3 koraka naprijed i ako dođeš do ruba, okreni se, ako je 94 manji od nasumičnog broja onda skreni. ,,

Učenici postavljaju početne uvjete za lika, npr. ako lik dodirne rub ekrana, ponovi kretanje. Nastavnik pomaže učenicima u pisanju i testiranju koda, osiguravajući da svi učenici razumiju kako i zašto lik ponavlja svoje kretanje. Učenici eksperimentiraju s različitim naredbama ponavljanja i dodaju dodatne uvjete ako žele. Ovo su primjeri uvjeta koji se mogu koristiti:

"Ponovi 5 puta: okreni se za 90 stupnjeva, idi naprijed 20 koraka."

"Ponovi dok lik ne dođe do ruba: idi naprijed 5 koraka."

Na kraju aktivnosti, učenici prezentiraju svoje radove ostatku razreda, objašnjavajući kako su postavili uvjete i zašto lik ponavlja kretanje.

### Aktivnost 3: Lik sa složenijim naredbama ponavljanja (1 sat)

Učenik će moći:

- Primijeniti složenije naredbe ponavljanja za ponašanje lika na ekranu

*Opis aktivnosti:* Učenici proširuju prethodni zadatak tako da njihov lik reagira na složenije uvjete u okruženju koristeći složenije naredbe ponavljanja. Nastavnik objašnjava nove uvjete koje će lik imati, koristeći primjere:

"Ako lik dođe do kuta ekrana, ponovi pokret 'skoči' 3 puta."

"Ako lik dodirne greben, ponovi pokret 'okreni se' dok ne prođe prepreku."

"Ako lik dođe do hobotnice, ponovi pokret 'ples' 5 puta."

Nastavnik zatim učenicima zadaje zadatak: *"Koristeći projekt sa posljednje aktivnosti, našem liku ribice ćemo dodati više složenih uvjeta. Ako naš lik dođe do kuta ekrana, skočiti će 3 puta. Ako dodirne greben okretati će se dok ne pređe prepreku. Ako ribica dotakne hobotnicu, ponoviti će pokret plesa 5 puta."*

Nastavnik zatim pokazuje primjer novoga koda gdje učenici mogu vidjeti kako se lik prikazuje kada se promjeni pozadine. Ako su bodovi manji od 19, onda se radnja ponavlja i lik klizi na nasumične pozicije. Ako su bodovi viši od 19 tada igra završava (slika 19).



Slika 19 Primjer složenijeg zadatka s naredbom ponavljanja

Učenici koriste programsku opremu za programiranje da bi implementirali ove uvjete. Nastavnik pomaže u definiranju novih naredbi ponavljanja i testiranju koda. Svaka grupa radi na tome da njihov lik pravilno reagira na različite uvjete u okruženju koristeći različite scenarije unutar programa. Učenici mogu dodati i svoje uvjete, a ovo su primjeri nekoliko uvjeta:

*"Ponovi dok lik ne završi krug: trči naprijed 10 koraka, okreni se za 45 stupnjeva."*

*"Ponovi dok ne sakupi sve predmete: idi naprijed 15 koraka, pokupi predmet."*

Učenici dijele svoje projekte i demonstriraju kako lik reagira na različite uvjete, dok nastavnik i učenici daju povratne informacije. Nastavnik može postaviti dodatna pitanja kako bi potaknuo učenike na razmišljanje:

*"Što bi se dogodilo ako dodamo još jedan uvjet?"*

*"Kako bismo mogli koristiti ove uvjete u stvarnim programima?"*

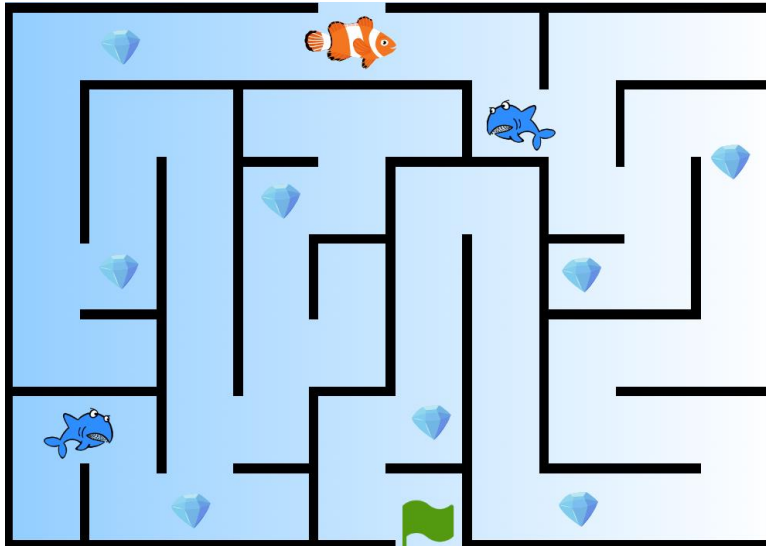
#### Aktivnost 4: Lik prolazi kroz labirint (1 sat)

Učenik će moći:

- Primijeniti naredbe ponavljanja za kretanje lika kroz labirint na ekranu.

Opis aktivnosti: Nastavnik započinje aktivnost objašnjavajući učenicima zadatak: njihov lik u Scratchu treba proći kroz labirint koristeći naredbe ponavljanja. Nastavnik daje učenicima sliku labirinta (Slika 20) sa preprekama koju trebaju učitati u Scratch kao pozadinu. Učenici rade u paru ili malim grupama kako bi programirali kretanje lika kroz labirint.

Nastavnik objašnjava učenicima: *"Vaš zadatak je pomoći našoj ribici da prođe kroz labirint. Koristit ćemo naredbe ponavljanja kako bismo olakšali kretanje kroz labirint. Učitat ćemo sliku labirinta u Scratch, a zatim ćemo programirati našeg lika da se kreće pomoću naredbi ponavljanja."*



Slika 20 Primjer igre labirinta u Scratchu

Kako bi učenici lakše razumjeli zadatak, nastavnik im zapisuje korake u rješavanju zadatka:

1. Učitavanje slike labirinta:

Učenici otvaraju Scratch i stvaraju novi projekt. Preuzimaju sliku i pohranjuju je u datoteku za izradu zadatka Labirint. Kliknu na "Backdrops" (pozadine) i učitaju sliku labirinta koju im je dao nastavnik.

2. Postavljanje lika:

Učenici dodaju lika (ribicu) koji će se kretati kroz labirint.

Postavljaju početnu poziciju lika na početak labirinta.

3. Programiranje kretanja lika:

Nastavnik objašnjava osnovne naredbe za kretanje lika i pri tome iznosi primjere: *"Idi naprijed 10 koraka"*, *"Okreni se za 45 stupnjeva"*.

Učenici programiraju kretanje lika koristeći naredbe ponavljanja.

Nastavnik pojašnjava naredbe kretanja i govori učenicima:



*"Ponavljaj dok ne dodirneš zid: idi naprijed 10 koraka."*

*"Ako dodirneš zid, okreni se za 90 stupnjeva."*

#### 4. Kraj igre:

Kada je zelena zastavica kliknuta, postavi lika na početnu poziciju.

Nakon što je nastavnik objasnio zadatak i korake u rješavanju istoga te pokazao primjer koda učenicima (Slika 21) još jednom sve sumira i govori:

Ponavljaj sljedeće korake dok lik ne stigne do cilja:

*"Idi naprijed 10 koraka."*

*"Ako dodirneš crnu boju (zid), okreni se za 90 stupnjeva."*

*"Ako dodirneš zelenu boju (cilj), zaustavi sve."*



Slika 21 Primjer koda za kretanje kroz labirint

Za testiranje i prilagodbu koda, učenici prvo testiraju svoj kod kako bi provjerili radi li ispravno. Ako lik zapne ili ne reagira kako treba, učenici prilagođavaju kod. Nastavnik pomaže učenicima u otklanjanju pogrešaka i optimizaciji koda.

Nakon što učenici završe programiranje, prezentiraju svoje radove ostatku razreda. Učenici objašnjavaju kako su postavili uvjete i naredbe ponavljanja za kretanje lika. Nastavnik i učenici raspravljaju o različitim rješenjima i pristupima rješavanju ovoga zadatka.

## 6. Zaključak

Na temelju provedenog istraživanja metodom nominalne grupe u OŠ Vežica u Rijeci te na Fakultetu informatike i digitalnih tehnologija u Rijeci, kao i analize rezultata glasovanja sudionika, možemo zaključiti da su određeni koncepti praga prepoznati kao ključni u nastavi informatike. Ovi koncepti praga, kao što su logički uvjeti, adresiranje ćelije, dijagram toka, organizacija podataka i varijable, dobili su visoku ocjenu od strane ispitanika, čime su istaknuti kao transformacijske točke u razumijevanju nastavnog gradiva.

Identifikacija i integracija ovih koncepata praga u nastavni plan i program informatike ima potencijal značajno unaprijediti metode poučavanja i učenja. Analizom odgovora ispitanika, nastavnika informatike osnovnih i srednjih škola, dobili smo širi uvid u njihovo razumijevanje i percepciju ključnih koncepata, što je od iznimne važnosti za razvoj i prilagodbu kurikuluma.

Provođenje ispitivanja pokazalo je da je razumijevanje i identifikacija koncepata praga među nastavnicima ključno za poboljšanje obrazovanja u području informatike. Integrativnost i transformativnost ovih koncepata omogućuju dublje povezivanje različitih ideja i trajnu promjenu u znanju učenika, što je neophodno za njihovo intelektualno napredovanje i aktivno sudjelovanje u zajednici prakse.

Rezultati istraživanja pružaju vrijedne uvide u ključne aspekte poučavanja informatike. Identifikacija temeljnih i zahtjevnih koncepata te razumijevanje njihovih karakteristika omogućuju nastavnicima da prilagode svoje nastavne metode kako bi kvalitetnije vodili učenike kroz proces učenja. Posebna pažnja trebala bi biti posvećena konceptima koji su prepoznati kao transformativni i povezani, jer oni imaju najveći potencijal za unapređenje cjelokupnog razumijevanja informatike. Također, koncepti s visokom ireverzibilnošću trebaju biti osnova kurikuluma, dok zahtjevni i ograničeni koncepti zahtijevaju dodatne resurse i pomoć u radu kako bi se osigurala njihova dugoročna usvojenost.

Daljnje istraživanje i primjenu koncepata praga u obrazovanju dodatno bi unaprijedilo proces učenja i poučavanja informatike. Korištenje interaktivnih i angažirajućih aktivnosti, poput igrafikacije i edukativnih igara, može dodatno motivirati učenike i olakšati im savladavanje složenih pojmova. Tako jasno definirani i integrirani koncepti praga mogu postati temelj za postizanje optimalnih rezultata u obrazovanju, što će dugoročno imati pozitivan utjecaj na stvaranje informatički pismenih i kompetentnih građana u digitalnom dobu.

## Literatura

- [1] Wolfgang Klafki, Wolfgang Schulz, Felix von Cube, Christine Moller, Rainer Winkel, Herwig Blankertz. Didaktičke teorije. Educa Zagreb, 1992.
- [2] J. H. F. Meyer and R. Land, "Threshold Concepts and Troublesome. Knowledge Linkages to Ways of Thinking and Practising," in *Improving Student Learning Ten Years On.*, C. Rust, Ed. Oxford, 2003.
- [3] N. J. G. Falkner, R. J. Vivian and K. E. Fal„ner, "Computer Science Education: The First Threshold Co“cept," 2013 Learning and Teaching in Computing and Engineering, Macau, Macao, 2013, pp. 39-46, doi: 10.1109/LaTiCE.2013.32.
- [4] SJSU School of Information. "The Expert Searcher and Threshold Concepts." Online izvor. Youtube. 10.12. 2023. dostupno na: <https://youtu.be/4I1Ue0vpcMw>, datum dohvaćanja: 10.12.2023.
- [5] CELatElon, Center for Engaged Learning. (2013, September 18). Decoding the disciplines and threshold concepts. Online video intervju, dostupno na: [https://youtu.be/Wqe\\_kKFoOq4](https://youtu.be/Wqe_kKFoOq4), datum dohvaćanja: 10.12.2023.
- [6] Chris Kyriacou. Nastavna umijeća, Metodički pristup za uspješno podučavanje i učenje. Božica Jakovlev, prevoditeljica. Zagreb: EDUCA, 1995.
- [7] Bognar, B. (2016.) Teorijska polazišta e-učenja, Hrvatski časopis za odgoj i obrazovanje, 2016., str. 225-256, dostupno na: <https://hrcak.srce.hr/file/229424>, preuzeto 28.12.2023.
- [8] Boustedt, J., Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., Sanders, K., & Zander, C. (2007). Threshold concepts in computer science: do they exist and are they useful?. *ACM Sigcse Bulletin*, 39(1), 504-508.
- [9] Pettersson, Kerstin. "Threshold concepts: A framework for research in university mathematics education." *Proceedings of the Seventh Congress of the European Society for Research in Mathematics Education*. 2011.
- [10] Vinner, S, and Dreyfus. T, "Images and definitions for the concept of function." *Journal for research in mathematics education* 20.4 (1989): 356-366.
- [11] Breidenbach, D., Dubinsky, E., Hawks, J., & Nichols, D. (1992). Development of the process conception of function. *Educational studies in mathematics*, 23(3), 247-285.
- [12] Thurston, W. P. (2005). Mathematical education. *arXiv preprint math/0503081*.
- [13] R. McCartney and K. Sanders. What are the "threshold concepts" in computer science? In T. Salakoski and T. Mäntylä, editors, *Proceedings of the Koli Calling 2005 Conference on Computer Science Education*, page 185, November, 2005.
- [14] Pivec, M. (2006). Igra i učenje: Potencijali učenja kroz igru. *Edupoint*, 6(49), 8-14.
- [15] Hoić-Božić, N., Holenko Dlab, M. (2021). Uvod u e-učenje: obrazovni izazovi digitalnog doba. Rijeka: Sveučilište u Rijeci, Odjel za informatiku
- [16] Budojević, A., Flisar Odorčić, A., Kanić, A. (2019). Scratch blokovi. Zagreb: Školska knjiga, d.d.
- [17] Budojević, A., Kanić, A. (2019). Istražite SCRATCH 3.0.. Zagreb: Školska knjiga, d.d.

- [18] Ministarstvo znanosti i obrazovanja [MZO]. (2018). Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije, dostupno na: [https://skolazazivot.hr/wp-content/uploads/2020/06/INF\\_kurikulum.pdf](https://skolazazivot.hr/wp-content/uploads/2020/06/INF_kurikulum.pdf), datum dohvaćanja: 12.4.2024.
- [19] Ministarstvo znanosti i obrazovanja [MZO]. (2019). Kurikulum međupredmetne teme Učiti kako učiti za osnovne i srednje škole, dostupno na: <https://tinyurl.com/45rv8bu4>, datum dohvaćanja: 12.4.2024.
- [20] Ministarstvo znanosti i obrazovanja [MZO]. (2019). Kurikulum međupredmetne teme Uporaba informacijske i komunikacijske tehnologije za osnovne i srednje škole, dostupno na: <https://tinyurl.com/432y3bjj>, datum dohvaćanja: 12.4.2024.
- [21] Jovanović, N. (2012) OBJEKTNO ORIJENTISANO PROGRAMIRANJE. SIGRAF, Kruševac, Visoka poslovna škola
- [22] Macphail, A. (2001). Nominal group technique: A useful method for working with young people. *British Educational Research Journal*, 27(2), 161-170. , dostupno na: <https://doi.org/10.1080/01411920120037117>, datum dohvaćanja: 1.5.2024.
- [23] Chapple, M., & Murphy, R. (1996). The nominal group technique: Extending the evaluation of students' teaching and learning experiences. *Assessment and Evaluation in Higher Education*, 21(2), 147-160. dostupno na: <https://doi.org/10.1080/0260293960210204>, datum dohvaćanja: 1.5.2024.
- [24] Madar, D. (1982). Using nominal group technique to foster productive behavior in group discussions. *Teaching Political Science*, 9(4), 185-189., dostupno na: <https://doi.org/10.1080/00922013.1982.9942918>, datum dohvaćanja: 1.5.2024.
- [25] Bognar, B. (2016). Teorijska polazišta e-učenja. *Croatian Journal of Education: Hrvatski časopis za odgoj i obrazovanje*, 18(1), 225-256.
- [26] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.

## Popis tablica

|  |    |
|--|----|
| Podaci o dobi ispitanika u OŠ .....                            | 13 |
| Podaci o radnom stažu ispitanika u OŠ .....                    | 14 |
| Broj ispitanika po radnim mjestima .....                       | 15 |
| Podaci o dobi ispitanika u SŠ .....                            | 16 |
| Podaci o radnom stažu ispitanika u OŠ .....                    | 17 |
| Broj ispitanika po radnim mjestima (SŠ) .....                  | 17 |
| Prikaz predloženih temeljnih i zahtjevnih koncepata (OŠ) ..... | 18 |
| Predloženi koncepti praga (OŠ) .....                           | 21 |
| Prikaz predloženih temeljnih i zahtjevnih koncepata (SŠ) ..... | 21 |
| Predloženi koncepti praga (SŠ) .....                           | 23 |

## Popis slika

|   |    |
|---|----|
| Slika 1. Proces primjene NGT-a .....  | 12 |
| Slika 2. Prikaz ispitanika prema spolu (OŠ) .....                           | 14 |
| Slika 3 Radni staž ispitanika OŠ .....                                      | 15 |
| Slika 4 Prikaz ispitanika prema spolu (SŠ) .....                            | 16 |
| Slika 5 Radni staž ispitanika (SŠ).....                                     | 17 |
| Slika 6. Sučelje programa Scratch .....                                     | 28 |
| Slika 7. Prikaz bloka naredbi (Scratch) .....                               | 28 |
| Slika 8 Promjena boje u Scratchu - 1.....                                   | 32 |
| Slika 9 Promjena boje u Scratchu - 2.....                                   | 32 |
| Slika 10 Prikaz primjene logičkih uvjeta u Scratchu .....                   | 33 |
| Slika 11 Prikaz promjene veličine u Scratchu uz pomoć logičkih uvjeta ..... | 34 |
| Slika 12 Prikaz organizacije pozadine s objektima za promjenu veličine..... | 34 |
| Slika 13 Prikaz korištenja složenih uvjeta u Scratchu.....                  | 35 |
| Slika 14 Igra Memorije u Scratchu.....                                      | 36 |
| Slika 15 Igra Memorije u Scratchu 2.....                                    | 36 |
| Slika 16 Igra Memorije u Scratchu 3.....                                    | 37 |
| Slika 17 Prikaz kretanja lika u Scratchu .....                              | 40 |
| Slika 18 Prikaz naredbe ponavljanja (Scratch) .....                         | 41 |
| Slika 19 Primjer složenijeg zadatka s naredbom ponavljanja .....            | 42 |
| Slika 20 Primjer igre labirinta u Scratchu .....                            | 43 |
| Slika 21 Primjer koda za kretanje kroz labirint.....                        | 44 |

# Popis priloga

## Prilog 1: Anketni list

Istraživanje je anonimno te se provodi u okviru projekta SciMaG - Science&Math educational games from preschool to university financiranog u okviru programa Erasmus+. Podaci se prikupljaju za potrebe znanstvenih i stručnih istraživanja.

\*\*\*\*\*

### PODACI O ISPITANIKU

- Dob: \_\_\_\_\_

- Spol: M Ž ostalo

- Stručna sprema: \_\_\_\_\_

- Status:

- nastavnik u osnovnoj školi:
  - predmet: \_\_\_\_\_
  - godine staža u obrazovanju: \_\_\_\_\_
- nastavnik u srednjoj školi:
  - predmet: \_\_\_\_\_
  - godine staža u obrazovanju: \_\_\_\_\_
- sveučilišni nastavnik:
  - predmet: \_\_\_\_\_
  - godine staža u obrazovanju: \_\_\_\_\_
- student:
  - studij: \_\_\_\_\_
  - godina: \_\_\_\_\_

U tablicu na sljedećoj stranici upišite temeljne koncepte u svojem području te ih obrazložite po kriterijima. Poželjno je tablicu popunjavati **grafitnom olovkom** tako da možete raditi promjene nakon diskusije.



| KONCEPT | Temeljni koncept (da/ne) | Zahtjevi koncept (da/ne) | Karakteristike koncepta praga |                           |                         |                      |
|---------|--------------------------|--------------------------|-------------------------------|---------------------------|-------------------------|----------------------|
|         |                          |                          | Povezanost (da/ne)            | Transformativnost (da/ne) | Ireverzibilnost (da/ne) | Ograničenost (da/ne) |
|         |                          |                          |                               |                           |                         |                      |
|         |                          |                          |                               |                           |                         |                      |
|         |                          |                          |                               |                           |                         |                      |
|         |                          |                          |                               |                           |                         |                      |
|         |                          |                          |                               |                           |                         |                      |
|         |                          |                          |                               |                           |                         |                      |

### Pomoć pri određivanju koncepta praga

1. Koje koncepte smatrate temeljnim za razumijevanje informatike?
2. Koje koncepte iz informatike ste smatrali teškima u svojem osnovnom/srednjoškolskom obrazovanju?
3. Među konceptima koje su učenici učili u osnovnoj školi izdvojite one koje po Vama učenici još uvijek smatraju teškima ili problematičnima u srednjoj školi? Među konceptima koje su učenici učili u srednjoj školi izdvojite one koje po Vama učenici još uvijek smatraju teškima ili problematičnima?
4. Kako ste percipirali promjene kod učenika nakon što su naučili određeni koncept praga? Koji je to koncept?

Identificirajmo koncepte kojima možemo pridružiti ostale karakteristike koncepta praga:

- **Povezanost:** Koji od teških koncepta koje smo upravo identificirali omogućuju učenicima da razumiju druge matematičke koncepte kada ih shvate?

- **Transformativnost:** Koji od teških koncepta koje smo upravo identificirali, kada ih učenici razumiju, pokreću značajnu promjenu u razumijevanju matematike kod pojedinca? Razumijevanje ovog koncepta stoga se može smatrati korakom prema stjecanju stručnog razumijevanja matematike.

- **Ireverzibilnost:** Za svaki od koncepta koje smo upravo definirali, je li sljedeća izjava istinita: "Vjerojatnost da učenik zaboravi koncept koji razumije je vrlo mala"?

- **Ograničenost:** Je li ovaj koncept karakteriziran fluktuirajućim razumijevanjem, odnosno učenik oscilira između razumijevanja koncepta i gubitka tog razumijevanja?

Prilog 2: Popis ispitanika za OŠ

| DOB | SPOL | STATUS  | STAŽ U<br>OBRAZOVANJU |
|-----|------|---|-----------------------|
| 46  | Ž    | nastavnik informatike u OŠ                    | 1                     |
| 40  | Ž    | nastavnik informatike u OŠ                    | 15                    |
| 26  | Ž    | nastavnik informatike u OŠ                    | 2                     |
| 50  | Ž    | nastavnik informatike i matematike u OŠ       | 25                    |
| 51  | Ž    | nastavnik informatike i matematike u OŠ       | 15                    |
| 45  | M    | nastavnik informatike u OŠ                    | 4                     |
| 29  | Ž    | nastavnik informatike u OŠ                    | 3                     |
| 52  | Ž    | nastavnik informatike i matematike u OŠ       | 22                    |
| 46  | Ž    | nastavnik informatike u OŠ                    | 20                    |
| 29  | Ž    | nastavnik informatike i matematike u OŠ       | 3                     |
| 30  | Ž    | nastavnik informatike u OŠ                    | 5                     |
| 49  | Ž    | nastavnik informatike u OŠ                    | 3                     |
| 30  | M    | nastavnik informatike u OŠ                    | 1                     |
| 38  | Ž    | nastavnik informatike i engleskog jezika u OŠ | 10                    |
| 28  | Ž    | nastavnik informatike u OŠ                    | 3                     |
| 28  | Ž    | nastavnik informatike u OŠ                    | 3                     |
| 35  | M    | nastavnik informatike i matematike u OŠ       | 4                     |
| 40  | Ž    | nastavnik informatike i fizike u OŠ           | 4                     |
| 38  | Ž    | nastavnik informatike u OŠ                    | 0                     |
| 32  | Ž    | nastavnik informatike u OŠ                    | 1                     |
| 40  | Ž    | nastavnik informatike u OŠ                    | 10                    |
| 27  | Ž    | nastavnik informatike u OŠ                    | 3                     |

|    |        |   |    |
|----|--------|---|----|
| 28 | Ž      | nastavnik informatike u OŠ                    | 3  |
| 30 | Ž      | nastavnik informatike u OŠ                    | 3  |
| 29 | Ž      | nastavnik informatike u OŠ                    | 4  |
| 44 | Ž      | nastavnik informatike i matematike u OŠ       | 12 |
| 35 | M      | nastavnik informatike u OŠ                    | 7  |
| 33 | Ž      | nastavnik informatike i tehničke kulture u OŠ | 8  |
| 51 | Ž      | nastavnik informatike u OŠ                    | 0  |
| 56 | Ž      | nastavnik informatike u OŠ                    | 26 |
| 27 | Ž      | nastavnik informatike u OŠ                    | 1  |
| 28 | Ž      | nastavnik informatike u OŠ                    | 5  |
| 48 | ostalo | nastavnik informatike u OŠ                    | 25 |
| 47 | M      | nastavnik informatike u OŠ                    | 7  |
| 27 | Ž      | nastavnik informatike u OŠ                    | 1  |
| 48 | Ž      | nastavnik informatike u OŠ                    | 5  |
| 27 | M      | nastavnik informatike u OŠ                    | 4  |
| 50 | Ž      | nastavnik informatike i matematike u OŠ       | 24 |
| 49 | Ž      | nastavnik informatike i matematike u OŠ       | 22 |
| 63 | Ž      | nastavnik informatike i tehničke kulture u OŠ | 37 |
| 44 | Ž      | nastavnik informatike u OŠ                    | 17 |
| 51 | M      | nastavnik informatike u OŠ                    | 25 |
| 46 | Ž      | nastavnik informatike u OŠ                    | 20 |
| 50 | Ž      | nastavnik informatike i matematike u OŠ       | 25 |
| 27 | M      | nastavnik informatike i matematike u OŠ       | 0  |
| 24 | M      | nastavnik informatike u OŠ                    | 0  |

|    |   |                            |    |
|----|---|----------------------------|----|
| 26 | M | nastavnik informatike u OŠ | 0  |
| 60 | M | nastavnik informatike u OŠ | 34 |
| 35 | Ž | nastavnik informatike u OŠ | 6  |
| 43 | M | nastavnik informatike u OŠ | 15 |
| 44 | Ž | nastavnik informatike u OŠ | 21 |
| 27 | M | nastavnik informatike u OŠ | 2  |
| 35 | Ž | nastavnik informatike u OŠ | 4  |

Prilog 3: Popis ispitanika za SŠ

| DOB | SPOL | STATUS  | STAŽ U<br>OBRAZOVANJU |
|-----|------|---|-----------------------|
| 27  | M    | nastavnik informatike u SS                          | 1                     |
| 50  | Ž    | nastavnik računalstva u SS                          | 6                     |
| 34  | ž    | nastavnik informatike i računalstva                 | 6                     |
| 37  | M    | nastavnik informatike i računalstva                 | 10                    |
| 37  | M    | nastavnik informatike u SS                          | 12                    |
| 39  | Ž    | nastavnik primjene informatike i računalstva        | 12                    |
| 41  | ž    | nastavnik njemačkog jezika i informatike u SS       | 15                    |
| 44  | Ž    | nastavnik računalstva u SS                          | 18                    |
| 58  | Ž    | nastavnik informatike u SS                          | 23                    |
| 49  | ž    | nastavnik informatike u SS                          | 25                    |
| 57  | Ž    | nastavnik matematike i informatike                  | 31                    |
| 55  | Ž    | nastavnik programiranja za web i baze podataka u SS | 31                    |
| 57  | Ž    | nastavnik računalstva u SS                          | 32                    |
| 60  | ž    | nastavnik informatike u SS                          | 33                    |
| 60  | M    | nastavnik informatike u SS                          | 34                    |
| 63  | ž    | nastavnik matematike i informatike                  | 40                    |