

# Primjena sustava Replit u osnovnoškolskoj nastavi informatike u domeni računalnog razmišljanja i programiranja

---

Mudrinić, Vedran

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:372914>

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

Download date / Datum preuzimanja: **2025-02-21**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



**Sveučilište Jurja Dobrile u Puli**

**Fakultet informatike**

**Diplomski sveučilišni studij Informatike**

**Nastavni smjer Informatika**

**VEDRAN MUDRINIĆ**

**PRIMJENA SUSTAVA REPLIT U OSNOVNOŠKOLSKOJ NASTAVI INFORMATIKE U  
DOMENI RAČUNALNOG RAZMIŠLJANJA I PROGRAMIRANJA**

**Diplomski rad**

Pula, lipanj 2024.

**Sveučilište Jurja Dobrile u Puli**

**Fakultet informatike**

**VEDRAN MUDRINIĆ**

**PRIMJENA SUSTAVA REPLIT U OSNOVNOŠKOLSKOJ NASTAVI INFORMATIKE U  
DOMENI RAČUNALNOG RAZMIŠLJANJA I PROGRAMIRANJA**

**Diplomski rad**

**JMBAG: 0125029907, izvanredni student**

**Studijski smjer: Informatika – nastavni smjer**

**Predmet: IT i edukacija**

**Znanstveno područje: Društvene znanosti**

**Znanstveno polje: Informacijske znanosti**

**Znanstvena grana: Informacijski sustavi i informatologija**

**Mentor: Izv. prof. dr. sc. Snježana Babić**

Pula, lipanj 2024.



## IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani **Vedran Mudrinić**, kandidat za **magistra edukacije informatike** ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

---

U Puli, lipanj 2024.



## IZJAVA

o korištenju autorskog djela

Ja, **Vedran Mudrinić** dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom **PRIMJENA SUSTAVA REPLIT U OSNOVNOŠKOLSKOJ NASTAVI INFORMATIKE U DOMENI RAČUNALNOG RAZMIŠLJANJA I PROGRAMIRANJA** koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, lipanj 2024.

Potpis

---

## ZAHVALA

*Na svesrdnoj i iskrenoj pomoći u savjetima, vremenu provedenom u razmišljanju i nadasve podršci u izboru teme kao doprinosu odgoju i obrazovanju djece osnovnih škola, jedno veliko hvala mojoj mentorici Izv. prof. dr. sc. Snježani Babić.*

*Vedran Mudrinić*

## SADRŽAJ

1. UVOD .....	1
2. RAČUNALNO RAZMIŠLJANJE I PROGRAMIRANJE	
2.1. OPĆENITO O POJMU RAČUNALNOG RAZMIŠLJANJA I PROGRAMIRANJA .....	2
2.2. SUSTAVI ZA UČENJE PROGRAMIRANJA U NASTAVI INFORMATIKE . .....	5
3. TEHNIČKI I PEDAGOŠKI POTENCIJAL SUSTAVA REPLIT U NASTAVI INFORMATIKE ZA UČENJE PROGRAMIRANJA U DOMENI RAČUNALNOG RAZMIŠLJANJA I PROGRAMIRANJA	
3.1. TEHNIČKE KARAKTERISTIKE SUSTAVA REPLIT .....	19
3.2. PEDAGOŠKI POTENCIJAL SUSTAVA REPLIT U NASTAVI INFORMATIKE UČENJE PROGRAMIRANJA .....	20
4. ANALIZA PRETHODNIH ISTRAŽIVANJA .....	22
5. ISTRAŽIVANJE MIŠLJENJA UČENIKA O PRIMJENI SUSTAVA REPLIT U NASTAVI INFORMATIKE U DOMENI RAČUNALNOG RAZMIŠLJANJA I PROGRAMIRANJA U VIŠIM RAZREDIMA OSNOVNE ŠKOLE	
5.1. Metodologija istraživanja .....	29
5.1.1. Cilj i metode istraživanja .....	29
5.1.2. Procedura istraživanja .....	29
5.1.3. Anketni upitnik .....	30
5.1.4. Uzorak ispitanika .....	32
5.2. Rezultati istraživanja .....	32
5.3. Diskusija rezultata istraživanja .....	56
6. ZAKLJUČAK .....	58
Popis grafikona .....	60
Popis slika .....	61

Popis tablica .....	62
Skraćenice .....	62
Prilozi .....	64
LITERATURA .....	66
SAŽETAK .....	76
SUMMARY .....	77



## 1. UVOD

„...Težište obrazovnog procesa u predmetu Informatika treba biti na rješavanju problema i programiranju kako bi se poticalo razvijanje računalnog načina razmišljanja koje omogućuje razumijevanje, analizu i rješavanje problema odabirom odgovarajućih strategija, algoritama i programskih rješenja. Takvi se načini razmišljanja trebaju prenositi i u druga područja posebice matematičko i prirodoslovno, kao i u svakodnevni život. Važnost učenja Informatike priprema učenika za mnoga područja djelovanja, osobna i poslovna. Učenja predmeta Informatika očituje se u razvoju računalnoga načina razmišljanja koje uključuje i tehnike rješavanja problema.“ (Kurikulum nastavnog predmeta Informatike, 2018).

Kod početnog učenja programiranja postoji niz problema koje učenike ometaju u učenju programiranja i daleko je od uspješnog učenja programiranja te su nužna istraživanja koja vode u razumijevanju poteškoća koja su uzrok navedenog. Vrlo je važno da se nastavnici upoznaju se problemima stjecanja znanja te poboljšaju učinkovitost učenja (L. Ma, J. Ferguson, M. Roper and M. Wood, 2011).

Kako bi se olakšalo usvajanje računalnog razmišljanja i programiranja učenicima, praksa je pokazala kako nastavnici u nastavi primjenjuju veći broj različitih alata za učenje: MicroWorlds JR, MSWLogo, Terrapin Logo, Pencilcode, Turtle Academy, Turtle Art, NetLogo, LCSI Microworlds environment, E-slate, Scratch, ScratchJr, Hopscotch, Blockly, Prolog, Racket, Squeak e-toys, Kodable, Tynke, LEGO Mindstorms, Robot Mesh, Code.org, CoderDojo (FESSAKIS, GEORGIOS, et al., 2019).

Na osnovu svega navedenog o ovom radu se smatralo opravdano istražiti mišljenje učenika o primjeni sustava Replit u nastavi informatike za učenje programiranja.

Glavni cilj empirijskog istraživanja u ovom radu bio je da istraži može li sustav za učenje programiranja Replit biti bolji oblik učenja programiranja od lokalno instaliranog prevoditelja za učenje programiranja u osnovno-školskoj nastavi informatike od 5. do 6. razreda u domeni računalnog razmišljanja i programiranja, te može li otkloniti negativne čimbenike koji ometaju uspješno učenje programiranja.

S tim u vezi provedeno je istraživanje mišljenja učenika viših razreda o primjeni sustava Replit u nastavi Informatike za učenje programiranja na primjeru odabrane osnovne

škole „Antuna Mihanovića Petropoljskog“ u Drnišu, s tri 5. i tri 6. razredna odjeljenja na uzorku od ukupno (N=57) učenika. Učenicima je bila dana mogućnost primjene dva modela za učenje programiranja kako bi mogli iskazati svoje mišljenje o primjeni sustava za učenje programiranja Replit s obzirom na njegove tehničke i pedagoške karakteristike. Jedan od alata je lokalno instaliran na školskim računalima IDLE, a drugi je sustav za učenje programiranja Replit. Učenici su jedan ciklus svog rada u učenju programiranja koristili lokalno instaliran prevoditelj IDLE, a drugi ciklus u sustavu za učenje programiranja Replit.

Rad se sastoji od šest poglavlja. U prvom poglavlju dan je uvod. U drugom poglavlju temeljito je razrađen pojam računalnog razmišljanja i programiranja kao i sustavi za učenje programiranja. U trećem poglavlju obrađen je tehnološki i pedagoški potencijal sustava Replit. U četvrtom poglavlju je provedena teorijska analiza prethodnih istraživanja. U petom poglavlju opisana je metodologija istraživanja te nakon učeničkog rada u školi i od kuće u sustavu Replit dani su rezultati provedene ankete među učenicima gdje su dobiveni podaci analizirani. U šestom ujedno zadnjem poglavlju je donesen zaključak istraživanja.

Svrha ovoga diplomskog rada je ukazati na pedagoški potencijal sustava Replit za učenje programiranja učenika u osnovnoj školi.

Doprinos ovog rada je dvostruk u odnosu na izbor alata za učenje programiranja u domeni računalnog razmišljanja i programiranja. Nastavnicima Informatike kao dobar izbor alata za rad u nastavi u smislu pedagoških i tehničkih karakteristika, a učenicima motivirajući, pojednostavljen, moderan i nadasve pristupačan način rada u odnosu na lokalno instaliran prevoditelj. Također izborom ovakvih sustava za učenje programiranja zalazimo u sferu inovativnog oblika učenja programiranja u obrazovne svrhe.

Doprinos ovog rada je važan za praktičare i znanstvenike. U praktičnom smislu, rezultati istraživanja mogu biti od pomoći za sve one koji imaju mogućnosti utjecati na uvođenje Replita u nastavni proces, posebno u školama u kojima je provedeno istraživanje. Također, dobiveni rezultati istraživanja mogu biti podloga za daljnja znanstvena i praktična istraživanja. Ovo istraživanje ima i neka ograničenja. U prvom

redu se to odnosi na mogućnost generaliziranja na sve škole, jer je ovo istraživanje provedeno samo u osnovnoj školi „Antuna Mihanovića Petropoljskog“ u Drnišu. Drugo ograničenje se odnosi na ograničen broj pitanja iz domene računalnog razmišljanja. U budućim radovima potrebno je uključiti veći broj škola u istraživanje, kao i proširiti anketni upitnik u domeni računalnog razmišljanja.

## **2. RAČUNALNO RAZMIŠLJANJE I PROGRAMIRANJE**

### **2.1. OPĆENITO O POJMU RAČUNALNOG RAZMIŠLJANJA I PROGRAMIRANJA**

Računalno razmišljanje smatra se neophodnom vještinom za 21. stoljeće koju je nužno poučavati od najranije dobi u korespondenciji s modernim teorijama poučavanja (Tomljenović, K., 2018). Izazov koji se danas učiteljima nameće je kako stvoriti digitalne tvorce programskih rješenja, a ne samo gledati pasivne konzumente. Pojam CT ima poprilično veliko područje primjene te nema jedinstvenu definiciju, ali ono što ga veže u jednom smislu razmišljanja je, da je to skup vještina. Generalno postoji konsenzus da je CT skup temeljnih vještina za svakog ne samo informatičare, koju veže jedna misaona aktivnost formuliranja i rješavanja problema na način koji to čine računala, a da pri tom postupak rješavanja problema ne uključuje nužno računala (Tomljenović, K., 2018). Proširena definicija CT kaže da je CT misaona aktivnost za rješavanje problema koja uključuje apstrakciju, dekompoziciju, algoritamski dizajn, evaluaciju i generalizaciju (Selby i Woollard, 2013). Djelovanjem velikog broja znanstvenika u definiciji CT došlo se do proširene definicije koja upotpunjuje i proširuje gornju, prema kojoj je CT proces rješavanja problema koji uključuje sljedeće karakteristike: „formuliranje problema tako da je moguće koristiti računala i druge alate pri rješavanju, logičko organiziranje i analiziranje podataka, reprezentiranje podataka kroz apstrakcije kao što su modeli i simulacije, izvođenje rješenja u algoritamskim koracima, identificiranje, analiziranje i primjena mogućih rješenja s ciljem postizanja najefikasnije kombinacije koraka i resursa, generalizacija i prijenos postupka na rješavanje drugih problema“ (Computational thinking toolkit, 2010). Dimenzija CT o odnosu na karakteristike i stavove je definirana sljedećim kognitivnim i misaonim stavovima pojedinaca: samouvjerenosti u rješavanju kompleksne problematike, upornosti pri rješavanju teških naizgled nedokučivih problema, prihvaćanje mogućih višestrukih rješenja problema, sposobnosti ili hvatanje ukoštac s istima, sposobnosti izmjene ideja i komunikacije tj. zajedničkog rada na istom problemu u svrhu postizanja rješenja (CSTA i ISTE, 2011). Usvajanje kompetencija i vještine CT može biti motivator učenicima ka izboru budućeg zanimanja. Često se vide poveznice u razmišljanju u kontekstu različitosti i sličnosti o odnosu računalno naspram matematičkog razmišljanja (Sneider, C. i dr., 2014). Moguće su i usporedbe računalnog naspram

inženjerskog oblika razmišljanja, dizajnerskog oblika razmišljanja, i svih drugih oblika razmišljanja u kojima bilo koji oblik razmišljanja ima preklapajuće elemente u razmišljanju (Wing, J. M., 2008) i (Razzouk, R., & Shute, V., 2012).

Što se tiče raščlanjivanja same domene CT na određeni način sagledavanja samih aspekata iste, ne postoji generalno suglasje oko raščlambe CT domene. U tablici 1 temeljem širokog pregleda literature, usporedbe i analize drugih autora, autori sumiraju u svoj aspekt – model CT-a, te prema njihovom modelu CT domena obuhvaća 6 glavnih aspekata: dekompozicija, apstrakcija, algoritmi, otklanjanje pogrešaka, iteracija i generalizacija (Shute, V.J., Sun, C., Asbell-Clarke, J., 2017).

Tablica 1. Model CT-a prema (Shute i dr., 2017.)

ASPEKT	DEFINICIJA
Dekompozicija	Rastavljanje kompleksnijih problema na manje cjeline jednostavnije za rješavanje.
Apstrakcija	Uočavanje problema koji se sastoje od: <ol style="list-style-type: none"> <li>1. prikupljanja i analize podatka: stvoriti što više veza i način povezanosti među njima,</li> <li>2. prepoznavanje uzorka tj. označavanje nepravilnosti u strukturi podataka,</li> <li>3. modeliranje ili izgradnja modela koji će opisati kako rješenje funkcionira.</li> </ol>
Algoritmi	Slijed logičko smislenih instrukcija za rješenje problema koje mogu biti napisane od strane čovjeka ili računala. Potpodjela je: <ol style="list-style-type: none"> <li>1. dizajn: konstrukcija adekvatnog niza koraka koji će zadovoljiti rješenje problema,</li> <li>2. usporednost: istovremeno izvršavanje niza koraka istovremeno,</li> </ol>

	<p>3. efikasnost: kreiranje što manji broj koraka za rješavanje problema, uklanjanje bespotrebnih koraka,</p> <p>4. automatizacija: sistematizacija u rješavanju sličnih problema.</p>
Otklanjanje pogrešaka	Označiti, identificirati i popraviti greške ako rješenje ne zadovoljava problem koji se rješava.
Iteracija	Repetitivnost procesa rješavanja problema do stanja idealnog rješenja koje zadovoljava dani problem.
Generalizacija	Prenošenje znanja u CT-u na druge slične probleme u sličnim situacijama koje se mogu poistovjetiti s danim problemom.

U tablici 2 autori CT model razlažu na tri dimenzije, istraživanja su bila bazirana na uključenim programskim aktivnostima u VPL – Scratch alatu (Resnick i sur., 2009):

Tablica 2. Model CT-a prema (Resnick i dr., 2007)

ASPEKT	DEFINICIJA
Koncepti	Znanja koja učenik koristi pri programiranju: naredbe, petlje, paralelnosti, odluke, algoritmi, uvjeti, operatori te rad s podacima.
Prakse	Koja se zbiva tijekom programiranja: eksperimentiranje, iteracije, testiranje, otklanjanje grešaka, višekratno korištenje i prevođenje, apstrahiranje i moduliranje.
Perspektive	Učeničko samorefleksivno razmišljanje i propitivanje sebe u odnosu na druge ljude i digitalni svijet oko sebe u smislu izražavanja i povezivanja važnih činjenica.

Još od početaka razvoja CT je istaknuta neupitna važnost aspekata i vještina CT generalno za čovječanstvo u današnjem modernom dobu, posebice vremenu koje dolazi, stoga bih glavni naglasak na istraživanje trebao biti na edukaciji mladih, što je

i razumljivo jer uspješno obrazovanje daje potporu mladim ljudima da budu pokretači nove budućnosti informatičkih revolucija, a to jedino mogu kreativni i inovativni umovi koji su pokretači gospodarstva i ekonomije (Shute i dr., 2017). Posebice je važna primjena obaveznog školovanja (K-12 obrazovanja koje u RH još formalno nije obavezno). K-12 je model a (K – kindergarten i 12 – twelfth grade), ovaj model obuhvaća obrazovanje od samoga vrtića do završetka osnovno školskog obrazovanja, tj. obaveznog školovanja. Namjera K-12 modela za poučavanje računalne znanosti, je model koji bi skicirao temeljne koncepte i dao prikladne temelje za svaki stupanj procesa učenja. Cilj je da učenici prestanu biti samo korisnici tehnologije, nego da postanu inovatori sposobni za poboljšanje kvalitete života (povećati znanje o „računalnom“). To znači potrebu uvođenja osnovnih koncepata računalne znanosti svim učenicima, počevši od učenika osnovnih škola. Također je bitno ponuditi dodatne predmete vezane uz računalnu znanost koji bi dozvolili zainteresiranim učenicima da to bolje nauče te ih pripremit za ulazak u srednjoškolsku sredinu ili na fakultet. K-12 za računalnu znanost trebao bi sadržavati: programiranje, hardverski dizajn, mreže, grafiku, baze podataka, računalnu sigurnost, softverski dizajn, programske jezike, logiku, programske paradigme, prevođenje između nivoa apstrakcije, umjetnu inteligenciju, ograničenja računala te aplikacije u tehnologiji informacije i informatičkih sustava. Smatra se kako prava revolucija u računarstvu neće nastupiti dok svi ne budemo obrazovani do te razine da novu tehnologiju možemo koristiti na nove, inovativne načine. Učenici kroz računalnu znanost uče: logičko zaključivanje, algoritamsko razmišljanje, skiciranje i rješavanje problema. Većina karijera 21. stoljeća će zahtijevati poznavanje računalne znanosti i nužno je razumjeti „računalno“ kako bi bili konkurentni u svom području zanimanja. Koncept a za K-12 računalnu znanost ima četiri razine:

1. razina – treba približiti osnovnoškolcima osnovne koncepte računalnih znanosti integrirajući osnovna znanja o tehnologiji s jednostavnim idejama o algoritamskom razmišljanju (rješavanje rutinskih hardverskih i softverskih problema). Trebaju se znati služiti softverom (razni alati) i hardverom, koristiti računalo za učenje... (razredi K-8)
2. razina – učenici bi trebali imati šire razumijevanje principa, metodologija i primjene računalne znanosti u modernom svijetu (razredi 9. - 10.)

3. razina – mogu izabrati učenici koji žele znati više o računalnoj znanosti. Stavlja naglasak na znanstvene i inženjerske aspekte računalne znanosti (matematički principi, rješavanje problema, mreže itd.). Priprema ih za naprednija predavanja. – može se preskočiti (razred 11.)

4. razina – pruža dubinu proučavanja jedne određene grane računalne znanosti (npr. programiranje, objektno orijentirani dizajn) – odabir neke određene grane omogućava lakši nastavak proučavanja iste teme na fakultetima ili pripremu za radni odnos (razred 12.)

Temeljni dokument kojim se iskazuju namjere povezane sa svrhom, ciljem, očekivanjem, ishodom, iskustvima djece, organizacijom odgojno-obrazovnog procesa i vrednovanjem naučenog, te povezanost s drugim predmetima i međupredmetnim temama je (Nacionalni kurikulum nastavnog predmeta Informatike u Republici Hrvatskoj, 2018). To je koncept u organizaciji znanja za podučavanje predmetnog kurikulumu kojeg sačinjavaju domene, unutar koje su određeni odgojno-obrazovni ishodi poučavanja. Odgojno-obrazovni ishodi predstavljaju očekivanja nakon perioda izobrazbe učenika od godine dana. Klasificirani su kao očekujuća znanja, vještine i stavovi koji se s vremenom poučavanja sistematiziraju s količinom usustavljenog gradiva. U par nekoliko zadnjih desetljeća razvoj računalne znanosti omogućio je razvoj informacijsko-komunikacijske tehnologije (skr. IKT dalje u tekstu) što je nametnulo nove uvijete na tržištu rada i života, pa samim se time propagiralo kao novi predmet u školama i to od sve ranije dobi. Primjena računala u svim sferama života danas je nametnula novu paradigmu pojednostavljeno-složenog života u ovisnosti o znanju pojedinca. Pojam digitalne pismenosti se pojavljuje kao posljedica korištenja IKT-a. Digitalna pismenost je pojam koji pojedincu omogućuje upotrebu računala i različitih računalnih sustava u svakodnevnom životu. Učenje temeljnih informatičkih koncepata programiranja, algoritama i struktura podataka je dio nužnosti kako ne bih u budućnosti postali samo pasivni korisnici IKT-a, već i aktivni stvaratelji novih izuma u području znanstvenog polja računalstva. Većina poslova današnjice zahtijeva razumijevanje i primjenu računalnih kompetencija koje su konkurent ljudskom razmišljanju (Kurikulum Informatike, 2018). Osobitosti učenja predmeta Informatike jest u razvoju računalnog načina razmišljanja koji uključuje sljedeće elemente nužne



za rješavanje problema: „apstraktni način mišljenja – apstraktnost informacija, logičko povezivanje, analizu podataka, automatizaciju rješavanja ponavljajućih problema - algoritmizacijom razmišljanja, prepoznavanje, analizu i primjenu mogućih rješenja s ciljem postizanja učinkovitoga rezultata vodeći računa o dostupnim resursima, formuliranje problema načinom primjerenim uporabi računala i računalnih alata, generalizaciju procesa rješavanja problema primjenjivog na niz sličnih problema“. Generičke kompetencije koje Informatika potiče jesu: „kreativnost i inovativnost stvaranjem digitalnih uradaka i algoritama, kritičko mišljenje i vrednovanje tehnologije i izvora znanja, rješavanje problema i donošenje odluka s pomoću IKT-a, informacijska i digitalna pismenost razumijevanjem i konstruktivnim razgovorom o pojmovima iz područja informatike, osobna i društvena odgovornost razmatranjem etičkih pitanja kao što su pitanja softverskih izuma ili krađe identiteta i vlasništva, odgovorno i učinkovito komuniciranje i suradnja u digitalnome okruženju, aktivno građanstvo kao spremnost i hrabrost za javno i odgovorno iskazivanje mišljenja i djelovanja uz međusobno poštovanje i uvažavanje u digitalnome okruženju, upravljanje obrazovnim i profesionalnim razvojem, učenjem s pomoću informacijske i komunikacijske tehnologije, učenjem na daljinu, videokonferencijama, virtualnim šetnjama, pristupom online bazama podataka i sl.“(Kurikulum Informatike, 2018). Pokraj pedagoški primjerenom praksom, nacionalni proces učenja definira pokraj navedenih kompetencija da učenici također trebaju steći: samostalnost, samopouzdanje, odgovornost i poduzetnost u radu. Nacionalni kurikulum ističe da su timski rad i suradnja značajni motivatori u procesu učenja. Sadržaj predmeta Informatika propisan je kao koncept cijelog školovanja od prvog razreda osnovne škole do završetka srednjoškolskog obrazovanja. Znanje je podijeljeno na module ili cikluse po načelu spiralnog modela (znanje koje se usvaja na nižim razinama te se proširuje i produbljuje na višima), pri čemu znanja, stavovi i vještine usvojeni u predmetu Informatike postaju podrška svim drugim predmetima. Moduli ili ciklusi su dvogodišnji: jedan ciklus učenja je dvije školske godine, tako da osnovno školsko obrazovanje se sastoji od četiri ciklusa. Također postoje definirane četiri domene kojima se realiziraju ciljevi učenja: e-Društvo, Digitalna pismenost i komunikacija, Informacije i digitalna tehnologija i Računalno razmišljanje i programiranje(Kurikulum Informatike, 2018).



Slika 1. Povezanost domena (Izvor: Kurikulum Informatike, 2018.)

Na slici 2 prikazane su četiri domene kojima se realiziraju ciljevi predmeta Informatika, to su: e-Društvo, Digitalna pismenost i komunikacija, Informacije i digitalna tehnologija, te Računalno razmišljanje i programiranje. Domene satnicom nisu proporcionalno zastupljene u svim razredima učenja Informatike i učitelji ih individualno formiraju Generalnim izvedbenim kurikulumom početkom školske godine, jer pojedini ishod ne zahtijeva jednako vrijeme učenja (Kurikulum Informatike, 2018). U domeni Računalnog razmišljanja i programiranja je istaknut značaj računalnog razmišljanja za pristup rješavanju problema koji se može riješiti uz pomoć računala, što u konačnici ima za rezultat da učenici izlaze iz sfere pasivnih korisnika stanovitih računalnih alata, već postaju aktivni dionici – stvaratelji nečega novoga. Primjenjivost računalnog razmišljanja i programiranja produbljuje druga znanja i povezuje nepovezano. Taj dio apstraktnosti dolazi kroz logičke zaključke, izradu modela i rješavanju postojećeg problema. Računalno razmišljanje jedinstvena je vještina koja potiče: točnost, preciznost, predikciju događaja i u konačnici sustavnost dolaska do pravog rješenja. Apstrakcija konceptualno potiče metakognitivne procese razdvajanja složenih problema na više jednostavnih u svrhu iznalaska rješenja. Rješavanje problema izradom računalnog programa potiče inovativnost i poduzetnost, a razvija: samopouzdanje, upornost, preciznost u ispravljanju pogrešaka, sposobnost komunikacije i zajedničkog rada u timu (Kurikulum Informatike, 2018).

## **2.2. SUSTAVI ZA UČENJE PROGRAMIRANJA U NASTAVI INFORMATIKE**

### **2.2.1. OPĆENITO O SUSTAVIMA ZA UČENJE PROGRAMIRANJA**

Najviša razina izgrađenih sustava za učenje programiranja su sustavi koji provjeravaju, nadopunjuju, ispravljaju, sugeriraju rješenje problema te obogaćuju znanje putem ugrađenih - implementiranih funkcionalnosti u sustav za učenje programiranja. Sustavi za poučavanje programiranja su komplementi klasične nastave koji mogu dodatno ili pak dijelom olakšati i unaprijediti nastavni proces. Jedna od najznačajnijih funkcionalnosti sustava za poučavanje programiranja je prepoznavanje sintaksnih i logičkih pogreški u kodu. Funkcioniranje ove značajke sustava za poučavanje programiranja olakšava razumijevanje ciljeva učenja programiranja kao i poteškoća koje početnici imaju u razumijevanju programskih pojmova i mentalnog procesa odgovornog za stjecanje - usvajanje novog znanja u domeni računalnog razmišljanja i programiranja. Sustavi učenja posjeduju ogromno znanje za područje koje se poučava i može se prikazati na način koji nije unaprijed definiran u vrijeme izgradnje sustava (Dadić, T., 2015).

### **2.2.2. KATEGORIZACIJA SUSTAVA ZA UČENJE PROGRAMIRANJA**

Sustavi za učenje programiranja građeni su prema svom primarnom cilju, u obrazovnom kontekstu da podučavaju programiranje i da programerima početnicima olakšaju učenje programiranja, te ih dijele u jedanaest kategorija i trinaest potkategorija (Kelleher i Pausch, 2005). U pregledu literature o sustavima za učenje programiranja nalazi se sužena podjela u četiri kategorije (Pears i dr., 2007):

1. vizualizacijski alati (najčešće VPL) - kroz vizualne prikaze prikazuju određene aspekte izvršavanja programa,
2. alati za automatsko ocjenjivanje (najčešće CPL) - omogućuju učenicima da predaju svoje uratke, a sustav im dodjeli ocjenu. Ovi sustavi koriste unaprijed pripremljene testne slučajeve s pomoću koji provjeravaju radi li program ispravno.,
3. programske okoline (IDE) - su različiti oblici integriranog razvojnog sučelja posebno prilagođenog edukacijskom kontekstu,

4. alati za podršku programiranju - su svi ostali alati koji ne spadaju i prije opisane grupe.

Druga vrsta podjele sustava za učenje programiranja (Dadić, T., 2015.) podijeljena je u 3 kategorije:

1. „mrežni klijentski alati za učenje programiranja“,
2. „mrežni prevoditeljski alati za programiranje“,
3. „mrežni i *desktop* prevoditeljski alati za programiranje“.

U mrežni klijentski alat za učenje programiranja spadali bi npr.: W3schools ili Programiz, u mrežni prevoditeljski alat za programiranje: OnlineGDB ili Geekforge, a u mrežni i *desktop* alat za programiranje koji imaju i mrežnu i *desktop* podršku: Jupyter ili Scratch.

### 2.2.3. KARAKTERISTIKE SUSTAVA ZA UČENJE PROGRAMIRANJA

Prema hrvatskom osnovnoškolskom Kurikulumu nastavnog predmeta Informatike (Kurikulum nastavnog predmeta Informatika, 2018) nije određeno kojim alatima u nastavi se učitelji trebaju koristiti, te koje nastavne metode za obradu stanovitih sadržaja u ovisnosti o temi primijeniti. U učenju i poučavanju izbor može biti raznovrstan po pitanju materijala, sadržaja i izvora učenja. Samostalno ih izabire učitelj u svrhu ostvarivanja odgojno-obrazovnih ishoda učenja, imajući u vidu da izbor podržava razvoj vještina i znanja. Učiteljev izbor treba biti onaj programski alat za učenje koji potiče kreativnost i motivaciju u učenika. Pri realizaciji osobitu važnost treba dati projektnom radu i suradničkom učenju, a pri tome treba odabrati sigurna komunikacijska i suradnička *online* okruženja. U nastavku ovog poglavlja navode se alati i njihove karakteristike koje se koriste za učenje programiranja u hrvatskim osnovnim školama:

*OnlineGDB.com* je online alat za prevođenje i ispravljanje pogrešaka za jezike C/C++ (*OnlineGDB.com*). To je prvi mrežni IDE na svijetu koji nudi mogućnost otklanjanja pogrešaka s ugrađenim gdb programom za ispravljanje pogrešaka. Ovo je pristupačna

web aplikacija za programere koji vole kodiranje u mrežnom IDE-u, ali se suočavaju s neočekivanim padovima i lukavim greškama u svom kodu. OnlineGDB pruža moć otklanjanja pogrešaka takvim korisnicima kako bi im pomogao. Dakle, sve u svemu, to je izvrstan mrežni IDE s uređivačem koda, prevoditeljem i programom za ispravljanje pogrešaka.

*PyCharm* je besplatna zajednica (PyCharm). PyCharm pruža pametno dovršavanje koda, inspekcije koda, isticanje pogrešaka u kodu i brze popravke, automatizirano refaktoriranje koda te bogatim mogućnostima navigacije. Nudi izvrsnu podršku specifičnu za razvoj web okvira kao što su Django, Flask, FastAPI, Pyramid i web2py. PyCharm se integrira s Jupyter Notebookom, ima interaktivnu Python konzolu i podržava Anacondu kao i više znanstvenih paketa uključujući Pandas i Polars DataFrames, matplotlib i NumPy. Osim Pythona, PyCharm podržava JavaScript, TypeScript, SQL, HTML/CSS, jezike predložaka, React, Vue, Node.js i još mnogo toga. Mogućnost daljinskog razvoja omogućuje: pokretanje, otklanjanje pogrešaka, testiranje i implementaciju aplikacija na udaljenim hostovima ili virtualnim strojevima, integriranim SSH terminalom i integracijom Dockera i Vagranta. PyCharm ima ogromnu kolekciju alata: integrirani program za ispravljanje pogrešaka i pokretač testova, Python profiler, ugrađeni terminal i integraciju s glavnim VCS (eng. Veritas Cluster Server) te ugrađenim alatima za baze podataka.

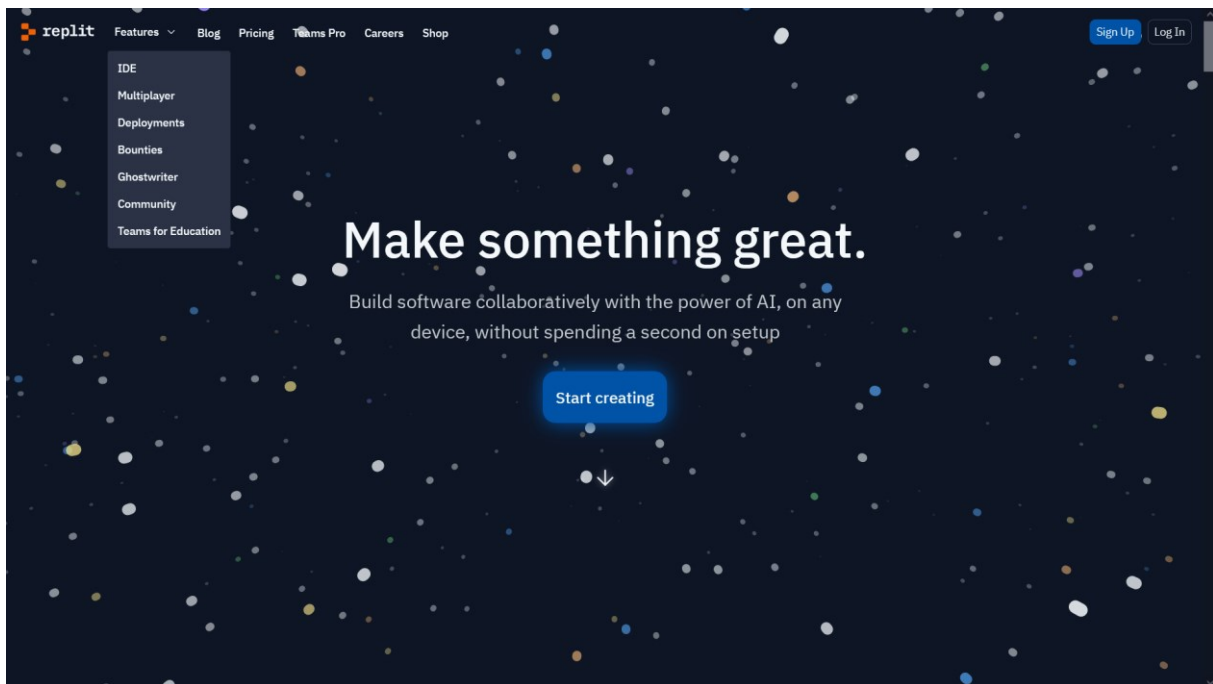
*Scratch* je najveća svjetska zajednica kodiranja za djecu s jezikom koji ima jednostavan pristup tj. jednostavno vizualno sučelje za izradu igara, animacija i priča (<https://scratch.mit.edu/about>). Neprofitna je organizacija, koja se financira isključivo dobrovoljnim donacijama korisnika istog programa tj. platforme za učenje kodiranja (Scratch Foundation). Ono što Scratch potiče, a cilj je edukacijskog procesa je: računalno razmišljanje i vještine rješavanja problema kodiranjem, kreativno učenje i poučavanje, samoizražavanje, suradnju i jednakost u programiranju. Scratch je besplatan te dostupan na više od 70 jezika i koristi se u više od 200 zemalja svijeta. Posebitost Scratcha je što je izrađen za dob od 8-16 godina, što je odličan alat u nastavi Informatike, a posebice u 1. razredima osnovne škole kada se učenici tek upoznaju uopće s predmetom Informatika. Pokraj toga je još VPL (vizualni programski jezik) što olakšava aplikativnu primjenu u razredu kod mlađe dobne skupine bez

ikakvih informatičkih predznanja. Učitelji vrlo lako integriraju Scratch u mnoga različita predmetna područja i dobne skupine. Kako edukatori koriste Scratch tako da dijele ideje i resurse Scratch tima i edukatora diljem svijeta. Repozitorij radova i ideja iz dana u dan raste. Svakog mjeseca web stranica ima novu temu za istraživanje i raspravu. Što se tiče resursa za edukatore prisutan je vodič za edukatore koji može uputiti nastavnike kako pripremiti i voditi Scratch satove i radionice (Vodič za edukatore). Postoji i hrvatska inačica radnog priručnika za rano učenje programiranja u izdanju Školske knjige (*Istražite SCRATCH, 2018*). Kreativni tim ScratchEd s Harvarda izdaje *Kreativno računarstvo* u kojemu su prisutne aktivnosti, planovi i strategije uvođenja kreativnog računarstva u učionice (Kreativno računarstvo). Što se tiče povezivanja s drugim edukatorima nastavnicima se nudi *Facebooku* grupa *Podučavanje sa Scratchom* (Podučavanje sa Scratchom). Mogući su i susreti edukatora s terminima susreta i mogućnosti pregleda po gradovima *ScratchEd susreti edukatora* (ScratchEd). Postoji dostupan i Googleov besplatni *CS First* kojeg koriste nastavnici diljem svijeta, ima preko 1000 video uputa i planova lekcija koje učenike uvodi u uspješan rad u Scratchu (CS First). U zajednici *Raspberry Pi Foundation* nastavnici imaju mogućnost izbora besplatnih modula za uključivanje učenika u učenje stvaranja interaktivnih priča, animacija i igara (Raspberry Pi Foundation). Za nastavnike postoji Scratchov račun za učitelje, koji olakšava kreiranje računa za učenike te upravljanje njihovim projektima i komentarima. Postoji *Vodič postavljanje računa učitelja* (Vodič postavljanje računa učitelja) i *Stranica s čestim pitanjima o računu učitelja* (Stranica s čestim pitanjima o računu učitelja).

Scratch je moguće instalirati na platforme Mac i Windows (E.R. Tee, 2013). Dostupan je besplatno za preuzimanje na [scratch.mit.edu](http://scratch.mit.edu) ([scratch.mit.edu](http://scratch.mit.edu)). Scratch ima licencu otvorenog koda od strane Open Source Initiative. Omogućuje korisnicima dijeljenje svojih radova na webu i forumima. Nedostatak je prilikom prevođenja gdje se nema uvid u greške izvođenja programa. Ima veliku brzinu izvođenja koda koja ovisi o snazi računala, ima tzv. *Set Single Stepping* koji omogućuje odabir brzine pojedinačnog koraka (od sporo do turbo brzine). Ima nisku razinu sigurnost u kodovima napisanim od strane korisnika, jer su dostupni za gledanje svima. Ima visok stupanj fleksibilnosti pri izmjenama i izdanjima novih verzija. Nadogradnje i ažuriranja mogu se jednostavno primijeniti na stare verzije. Ima relativno nisku modularnost, nema zasebne primjene

softverskih elemenata. Nije kompliciran softver za djecu, nudi video upute i često postavljana pitanja. Nove verzije su dostupne na [scratch.mit.edu](https://scratch.mit.edu).

*Replit* je poslužiteljska platforma koja ima vlastito okruženje koja omogućava učenicima online programiranje u školi i od kuće bez ikakvih posebnih konfiguracija (I S Zinovieva et al, 2021). Ima mogućnost izrade radnih timova – razreda. Promjene se ažuriraju u stvarnom vremenu. Sudionici imaju uvid u greške tijekom prevođenja programa što im olakšava razumijevanje kako program radi. Što se tiče komparativnih prednosti u odnosu na druge online kolaborativne alate, Replit ima sljedeće karakteristike: ima sintaksnu i semantičku provjeru napisanog koda prilikom pokretanja i prevođenja istog, algoritmizaciju koda, programsku logiku, rad s varijablama, nizovima, petljama, funkcijama i objektima. Također ima podršku za grupni ili suradnički tj. kolaborativni rad. Podršku za više od 50 programskih jezika.



Slika 3. Sustav za učenje programiranja Replit.com (Izvor:

<https://replit.com/site/teams-for-education>)

Na slici 3 prikazana je početna stranica sustava za učenje programiranja Replit tj. radni prostor tj. okruženje u kojem korisnik komunicira i uređuje svoj ili zajednički projekt ako je tim u pitanju. Služi kao integrirano razvojno okruženje (skrać. IDE), platforma za suradnju, pružatelj usluga u *Oblaku*, zajednica programera i edukacijska platforma.

Cijela Replit platforma i proizvod odnosi se na jedno okruženje ili projekt koji posjeduje korisnik ili tim. Replit je virtualni stroj (skrać. VM) koji radi u *Oblaku*.

Svi računalni resursi rade u *Oblaku*, a ne lokalno. Dostupni su svi alati, proširenja, implementacije i Replit AI.

Replit Mobile App (aplikacija za mobilne uređaje) dostupna je za Android i iOS korisnike. S ovom aplikacijom se može nositi projekte kodiranja kamo god se ide i koristiti kreativni potencijal kad god se želi.

Od dodataka u Replitu imaju sljedeće značajke: Replit Core je model pretplate za otključavanje potpunog Replit iskustva, a to je: neograničen pristup koji nudi Replit s Replitovim vodećim pomoćnikom za kodiranje s umjetnom inteligencijom koji pokreće GPT-4 (multimodalni model umjetne inteligencije). Inspirativan GPT-4 otklanja pogreške koda, priprema inteligentne prijedloge za automatsko dovršavanje koda i pretvara prirodni jezik u kod jednim klikom.

Svaki Replit korisnik sustava ima ponuđenu AI značajku. Dovršavanje koda i pomoć omogućeni su prema zadanim postavkama. Korisnici s besplatnim pretplatnim planom imaju pristup osnovnim AI značajkama, dok članovi Replit *Corea* imaju pristup najmoćnijim AI modelima i naprednim značajkama. Replit AI je alat za dovršavanje koda koji daje prijedloge na temelju koda u trenutnoj datoteci. Da bi se mogao koristiti, jednostavno se počne kodirati i prijedlozi se pojavljuju u tekstu. Replit AI za sada uključuje pet značajki: prva glavna značajka Replit AI je potpuni kod. Replit AI koristi kontekst za pružanje prijedloga dok se tipka. Pritiskom na *tab* moguće je prihvatiti prijedlog. Druga značajka je objasni kod. Označavanjem koda, Replit AI će ga proći korak po korak na jednostavnom engleskom jeziku. Treća značajka je uredi kod. Replit AI može dati – predlagati ponovno pisanje bloka koda. Četvrta značajka je generiranje koda. Davanjem upita na prirodnom jeziku Replit AI generira kod koji može pomoći u rješavanju problema. Peta značajka je chat. Replit AI pomoćnik koji korisnicima omogućuje brzo dobivanje odgovora na pitanja vezana uz kodiranje izravno u radnom prostoru, bez napuštanja stranice.

TIMOVI ZA OBRAZOVANJE



*Teams for Education* je značajka Replita koja ima ključne tijekove rada za učenike i nastavnike povrh osnovnih Replit značajki. *Teams for Education* je radni prostor za nastavnike i učenike. Ideja se temelji na projektima koji učenicima daju stvarno iskustvo. Svaki projekt može imati više verzija (često po jednu za svakog učenika ili grupu), tako da svi učenici mogu krenuti od iste točke, a proširiti osnovni projekt na različite načine. Postoji mogućnost kreiranja razreda. U tim se može pozvati nastavnike i učenike. Osim toga, mogu se promijeniti korisničke uloge za nastavnike kako bi oni imali administrativni pristup planu za stvaranje vlastitih timova, te izradu projekata. Postoji mogućnost dodavanja timskih nadimaka učenicima. Inače učenici koji se prijave s pomoću veze s omogućenom privatnošću dodjeljuje se nasumično korisničko ime.

Timski projekti su projekti i zadaci koje nastavnici mogu postaviti s početnim kodom, uputama i materijalima za svoje učenike. Kada započnu timski projekt, učenici dupliciraju vlastite, privatne verzije timskih projekata kako bi ih slobodno uređivali i prilagođavali.

Nastavnici mogu kreirati upute samo za čitanje koje će njihovi učenici slijediti, kao i planove lekcija samo za administratore koje će čuvati kao bilješke i dijeliti nastavni plan i program sa svojim kolegama s pomoću alata za izradu u stablu datoteke tima projekta. Planovi lekcija dostupni su samo administratorima. Namijenjene su pružanju bilješki govornika za pojedinačne nastavnike ili smjernica za nastavnike koji mogu dijeliti nastavni plan i program sa svojim kolegama.

*Teams for Education* je platforma za suradnju na kojoj nastavnici mogu kreirati zadatke za svoje učenike.

## INDIVIDUALNI, GRUPNI I SAMOGRUPIRAJUĆI PROJEKTI

Za pojedinačne projekte učenici mogu rastaviti projekt i izraditi vlastitu kopiju. Njihova će kopija biti privatna tako da samo učenici i administratori tima mogu vidjeti i uređivati njihov rad. Za grupne projekte se mogu dodijeliti učenike u grupe za predaju radi suradnje. Mogu se dodavati i uklanjati nove grupe, povlačiti i ispuštati učenike između grupa i tražiti učenike. Grupe se ne mogu ukloniti nakon spremanja, ali učenici se

kasnije mogu preraspodijeliti u druge grupe. Projekti samogrupiranja omogućuju učenicima stvaranje i pridruživanje vlastitim grupama.

#### ORGANIZIRANJE PROJEKATA U JEDINICE

Nakon što su timski projekti stvoreni, mogu se organizirati u jedinice. Projekti se povuku i ispuste u ispravne jedinice redoslijedom kojim se želim da se pojave za učenike. Nazivi jedinica se mogu mijenjati. Za promjenu redoslijeda jedinica strelicama gore dolje može se promijeniti redoslijed jedinica. Jedinice se mogu brisati.

#### KOPIRANJE PROJEKATA U DRUGE TIMOVE

Nastavnici mogu skupno kopirati i dijeliti projekte s bilo kojim timom. Ova je značajka korisna za jednog učitelja koji upravlja s više razreda, dijeli nastavne planove i programe s kolegama ili umnožava projekte unutar tima ili iz jednog semestra u drugi.

#### DUPLICIRANJE OSOBNIH ODGOVORA U TIMSKE PROJEKTE

Administratori *Teams for Education* mogu duplicirati odgovore sa svojih osobnih računa u *Teams for Education* kao timski projekt.

#### PREGLEDAVANJE DOMAĆE ZADAĆE

Nastavnici mogu ostaviti povratne informacije o projektima s pomoću komentara s nadzorne ploče svog tima klikom na *Prikaži prijave* pored naziva projekta. Prijave će pokazati sve učeničke radove koji su predani ili su u tijeku izrade. Za skupni pregled učeničkih radova pregled projekta nastavnik može otvoriti na početnoj stranici tima kako bih na prvi pogled vidio sve projekte i napredak učenika. Da bih nastavnik ostavio komentar o radu svojeg učenika može koristiti niti. Niti se mogu dodavati projektima i timskim odgovorima, te predstavljaju jednostavan način za ostavljanje komentara na određenu riječ ili redak koda u radu – projektu. Učenici mogu odgovoriti na komentar i u skladu s tim izmijeniti kod.

#### TESTIRANJE, PROCJENE I AUTOMATSKO OCJENJIVANJE

Replit ima brojne značajke koje se odnose na pomoć nastavnicima u automatskom ocjenjivanju zadatak učenika. Najjednostavniji od njih je testiranje ulaza/izlaza, koje

omogućuje da se provjeri rad ili domaća zadaća učenika za specifične rezultate, podudarajući se s točnim nizovima ili regularnim izrazima.

## ULAZNO/IZLAZNO TESTIRANJE

Replit input/output testiranje omogućuje učitelju stvaranje jednostavnih testova koji automatski usklađuju ulazne vrijednosti s očekivanim izlazom u učeničkim projektima. Učenici također mogu jednostavno testirati svoj kod prije podnošenja projekata.

## JEDINIČNO TESTIRANJE

Korak iznad ovoga je *Unit Testing* (Replit jedinično testiranje koje omogućuje autoru Replita stvaranje testova vođenih kodom koji uspoređuju stvarni izlaz funkcije s očekivanim izlazom) koji omogućuje pisanje potpunih jediničnih testova u Javi(JUnit), Pythonu(Unitest) ili JavaScriptu(Jest).

## TEST PODUDARANJA

Test podudaranja će proći ako je očekivani izlaz jednak stvarnom izlazu. Drugim riječima, stvarni *output* ne mora biti identičan očekivanom *outputu*, mora ga samo uključivati.

## INTEGRACIJA GOOGLE UČIONICA

Uz Replitovu integraciju *Googleove učionice* (online učionica, učionica u sklopu *Google Workspace for Education* usluge potpisane od strane Ministarstva znanosti i obrazovanja Republike Hrvatske u sklopu osnovnoškolskog obrazovanja nastala tijekom COVID-19), nastavnici mogu brzo dodati svakog učenika u svoju Google učionicu u tim i lako dijeliti Replit projekte kao zadatke Google učionice. Učenici i nastavnici mogu izraditi Replit račun i surađivati na njemu s Chromebooka, preuzeti ga iz trgovine Google Play na svoj uređaj ili ga pronaći na Chromebook App Hubu. Učitelji koji svoje učenike dodaju u Replit putem Google učionice također mogu integrirati svoje bilježnice Google učionica s pregledom projekata Replita. Učitelji mogu svoje učenike iz Google učionica dodati u Replit tim za obrazovanje.

## KURIKULUMI

Uz *Curriculum Hub*, administratori (nastavnici) na računima Teams for Education mogu pristupiti unaprijed izrađenim projektima, planovima lekcija, uputama i testovima za uvoz i korištenje u svojim timovima. Materijali *Curriculum Hub* dizajnirani su za rad kao cjelovite jedinice ili kao samostalni projekti. Korisnici (učenici) mogu pregledati planove lekcija i redoslijed za svaku nastavnu jedinicu. Projektni kod, upute i rješenja vidljivi su samo administratorima nakon što je nastavni plan i program uvezen u njihov tim.

## BESPLATNI NASTAVNI PLANOVI I PROGRAMI ZA NASTAVNIKE

U Replitu je misija učiniti programiranje dostupnijim svima. Taj cilj je i postignut kroz partnerstvo s talentiranim edukatorima u zajednici Replit. Zahvaljujući tim partnerstvima, osigurani su besplatni nastavni planovi i programi koji edukatore (nastavnike i učitelje) potiče da koriste, mijenjaju i dijele kada podučavaju programiranje.

### 2.2.4. OBRAZOVNO OKRUŽENJE

Odnos IKT-a i nastave sjedinjava pojmove koji čine temelj klasične nastave i tvore didaktički trokut kao suodnos između tri čimbenika nastave (nastavnik, učenik i nastavni sadržaj) uz dodatak tehnike u svrhu kvalitetnog izvođenja nastave (Radonić, 1997). Činjenica je da se sustav treba proširiti čimbenicima okruženje (prostor i tehnička oprema). Korelacijom gore navedenih čimbenika, te pravilnom optimizacijom u korištenju prostora i opreme, definiran je pojam didaktičkog peterokuta, s tim da je komunikacija nastavnika prema svim elementima, a da samo učenik ima povratnu informaciju. Moderniji pristup IKT predviđa da su svi elementi aktivni, te i korištenje suvremenih alata (računala, programska podrška, virtualna učionica) osigurava da i naizgled pasivni elementi (prostor, tehnička pomagala, nastavni sadržaj) postaju aktivni sudionici procesa obrazovanja. Obrazovni materijal postaje dostupan svim učenicima dionicima sustava za učenje programiranja (Radonić, 1997.).



Slika 2. Didaktički peterokut (Izvor: Radonić, 1997)

Na slici 2 vidljiv je suvremeni didaktički peterokut koji čine učenik, nastavnik, nastavni sadržaj, tehnička pomagala te prostor. U istraživanju UNESCO-a (Semenov, 2005) istaknuta je promjena paradigme - pristupa u kojemu nastavnik ne treba modificirati trenutno tj. „kako da ono što radi napravi uz pomoć IKT-a“, već da zađe u sferu razmišljanja „što da napraviti novo s IKT-om“. Ishodište razrade i primjene nastavnog oblika poučavanja uz korištenje IKT kao ključnog alata nužno zahtijeva odmak u razumijevanju nastavnika o „prostoru i vremenu“ koje se konstantno mijenja, digitalizira, a nastavnici jesu nužno dio toga. Potreba prilagodbe je nužnost, a ne opcija. Tehnologija zauzima sve veći dio života mladih, kao i sustavnosti obrazovanja, odnosno promjene u učionici. Time je sve veća potreba korištenjem sustava za učenje programiranja, a ne elementarnih alata za poučavanje programiranja samo da bih se zadovoljila forma obrazovanja. IKT nužno od strane nastavnika treba biti prihvaćena kao didaktički katalizator – nastavni akcelerator koji treba prihvatiti i iskoristiti u cilju unapređenja nastave. IKT omogućava raznolikost u kreiranu sadržaja, procesu komunikacije, pa time i u procesu učenja. IKT je ključ u procesu digitalnog opismenjavanja učenika i razvoju njihovih digitalnih kompetencija. Razumijevanje korištenja računala kao pomoćnika umjesto sebi nadređenog je proces koji puno daje ako se koristi u odabranim i primjenjivim obrascima učenja.

Moderno obrazovno okruženje, odnosno učenja i poučavanje, a koje se izvodi uz upotrebu nekog od oblika informacijske i komunikacijske tehnologije. Prednosti modernih obrazovnih okruženja su sljedeće:

- ✓ Iskustveno učenje (engl. Experience Learning),
- ✓ Učenje posvuda (engl. Everywhere Learning),
- ✓ Povećano ili bolje učenje (engl. Enhanced Learning),

- ✓ Prošireno učenje (engl. Extended Learning).

Cilj mu je unaprjeđenje i modernizacija procesa obrazovanja i ishoda obrazovanja. Ova definicija lako je primjenjiva na učenje programiranja pa bi sustavi za učenje programiranja podrazumijevali usvajanje znanja i vještina u svezi učenja programiranja posredstvom interneta te informacijske i komunikacijske tehnologije. Sustave za učenje programiranja, mogu se primijeniti u sljedećim oblicima:

- ✓ Klasična nastava – nastava u učionici (engl. face to face – F2F),
- ✓ Nastava uz pomoć ICT tehnologije u službi nadopune klasične nastave (engl. ICT supported teaching and learning),
- ✓ Hibridna ili mješovita nastava – kombinacija nastave u učionici i nastave uz pomoć tehnologije (engl. hybrid, mixed mode; blended learning),
- ✓ Online nastava – nastava koja je pomoći ICT tehnologije potpuno organizirana na daljinu (engl. Fully online).

Pokraj generalnih pristupa obrazovanja, znakovita je dinamičnost ovoga područja, kompleksnost samoga procesa razvoja sustava za učenje programiranja, varijabilnost promjena, kontinuirana praksa, presudna je uloga razmjene znanja i iskustva, kao i drugi elementi u smislu razvoja IKT-a (CARNET, 2024.).

### 3. TEHNIČKI I PEDAGOŠKI POTENCIJAL SUSTAVA REPLIT U NASTAVI INFORMATIKE ZA UČENJE PROGRAMIRANJA U DOMENI RAČUNALNOG RAZMIŠLJANJA I PROGRAMIRANJA

#### 3.1. TEHNIČKE KARAKTERISTIKE SUSTAVA REPLIT

**Prednosti** sustava Replit su detaljno opisane s aspekta obrazovanja u poglavlju 2.2.3. Sustavi za učenje programiranja.

**Nedostatci** sustava Replit koji su indirektno prirode vidljivi su tijekom postupka prve prijave učenika na sustav. Naime učenici prilikom prelaska iz 4. u 5. razred osnovne škole obično tada konkretno počinju koristiti mail račun koji imaju u sustavu školstva. AAI@EduHr (Autentikacijska i autorizacijska infrastruktura znanosti i visokog obrazovanja u Republici Hrvatsko) korisnički računi po osnovnim postavkama Carnet admin sustava Republike Hrvatske nisu računi koji uključuju Google Workspace for Education uslugu. Google Workspace for Education usluga dolazi s ugovorom Ministarstva znanosti i obrazovanja za osnovne škole, kao usluga koja se opcionalno mora aktivirati ponaosob svaki učenik za sebe, što učenicima iziskuje poprilično apstraktan pojam što moraju aktivirati. Kad se račun aktivira tada postoje određene administratorske zabrane za prijavu na sustave kao što su Replit.com pojavljuje se kao sporadičan problem, dok neki imaju roditeljski nadzor nad učeničkim računima (kao npr. *Google Family Link*) koji također zna vršiti zabranu prijave na sustav. U hrvatskim školama službeni račun koji učenici imaju se nalazi na školskoj domeni. I taj račun trebaju koristiti u svrhu obrazovanja. Također od kuće učenici su prepušteni sami sebi kako bih se s kućnog/roditeljskog računala prvi puta uspješno ulogirali u Replit.com. Nastavna praksa autora ovog diplomskog rada pokazala je da ova anomalija obično traje prva 2-3 tjedna uvođenja učenika u sustav Replit.com i nakon toga rad postaje nesmetan po pitanju ove problematike, kako u školi tako i kod kuće. Drugi direktni nedostatak sustava za učenje programiranja Replit bih bio nedostatak glasovne komunikacije među korisnicima sustava.

## **3.2. PEDAGOŠKI POTENCIJAL SUSTAVA REPLIT U NASTAVI INFORMATIKE ZA UČENJE PROGRAMIRANJA**

### **SURADNIČKI RAD**

Planiranje i primjena te izbor sustava za učenje utječe na djelotvornost izvođenja nastave i nastavnog procesa kao i eksperimentiranje s novim oblicima poučavanja i pedagoškim metodama (DigCompEdu, 2020.). S tog stajališta obrazovnog koncepta je vrlo uputno koristiti suradničko učenje kao uspješnu tehniku u području učenja programiranja. Suradničko učenje je oblik učenja u koje uključuje minimalno dva sudionika. Najčešće korištena metoda suradničkog rada je programiranje u paru (Beck, 2020). Suradnički rad – učenje počiva na konstruktivizmu gdje se znanje gradi i transformira od strane učenika. Suradničko učenje je potaknuto socijalnim vještinama poduprtim od strane učenika k samostalnom rješavanju problema u suradnji s drugim. Izvor koncepta suradničkog učenja leži u teoriji učenja prema Levu Vygotskyju u definiciji „zone proksimalnog razvoja“ koja je ključ za razvijanje složenih apstraktnih načina mišljenja. Složeni koncepti kao što je programiranje se razvijaju kroz zajedničke oblike suradničkog rada, prije nego što se razvijaju u vještine programiranja koje se mogu samostalno primjenjivati.

Kolaborativni (suradnički) rad uključuje isključene učenike, a posebice aktivira one koje tumači kao digitalne vođe posebice učenike koji su srasli s digitalnom tehnologijom (Sentance i Csizmadia, 2017). Rad u paru potiče bolje razumijevanje i općenito bolji uspjeh u rješavanju problema (Denner i sur., 2014). Suradnički rad u učenika potiče komunikaciju i raznovrsne oblike suradničkog učenja (Bjursten, EL., 2022). Također kada učenici rade u parovima ili radnim skupinama to im pomaže za pronalazak zajedničkog rješenja, kao što i podržava razvoj stručnog-tehničkog jezika i sposobnost komunikacije, koja je stalno u današnje vrijeme nametnuta kao posljedica informatičke demencije u pravome govoru koja se reflektira kao nedostatak komunikacije. Mnogo puta „...lakše je smisliti rješenje kada se o nečemu raspravlja zajedno“. Suradnički rad je definiran dvjema dimenzijama: kao jednostavan oblik suradnje na kojem učenici zajedno rade na rješavanju problema. Kada zadatak završi tj. kad se problem riješi, ne razvija se dalje. Druga dimenzija naprednijeg suradničkog rada je kada učenici sudjeluju u rješavanju problema gdje se zadatak razvija i ima duže



vremensko trajanje, a u rješavanju problema sudjeluje cijeli razred ili cijela škola. Hrvatski superobrazovateljski savjetnik (portal Carneta, hrvatske krovne organizacije za podršku obrazovnog sustava Republike Hrvatske) sugerira učiteljima da je Replit jednostavna i pristupačna platforma za početnike u programiranju (skole.hr). Ima podršku za mnogobrojne programske jezike. Korisnici mogu započeti s korištenjem odmah, bez kompliciranog postavljanja radne okoline na svome računalu. Ova značajka je posebno korisna za učenike koji su početnici u programiranju. Sustav za učenje programiranja Replit je besplatan, jedino što je učenicima potreban je Replit račun kojeg mogu stvoriti s pomoću svojeg AAI@EduHR školskog računa. Jedan od najznačajnijih specifičnosti Replit platforme je interaktivno okruženje za rad koje se izvodi u stvarnom vremenu koje u kombinaciji s pametnim pločama (projekt e-Skole, projekt koji potiče online kolaborativni rad) stvara konotaciju moderne učionice 21.stoljeća, a posebice u smislu obrnutih učionica. Isto tako Replit ima ugrađenu funkcionalnost ispravljanja i detektiranja pogrešaka u napisanom kodu. Dijeljenje poveznice za rad od iznimne je koristi za timski – suradnički rad, učenje u razredu i kod kuće, te razmjenu informacija tijekom rada što je od velike koristi. Jednostavan je, i ima intuitivan pristup datotekama što od velike je važnosti za rano učenje programiranja. Mape i datoteke su hijerarhijski posložene što olakšava pristup istima. Platforma Replit ima značajku stvaranja obrazovnih timova – razreda. Nakon postavljanja tima jednostavnom pozivnicom se dodaju članovi u razredni tim. Replit platforma je izvrsna, moderna i u svakom pogledu primamljiva platforma za one koji žele istraživati i sudjelovati u naprednim tehnologijama suradničkog učenja i danas sutra ostvariti san budućeg zanimanja – programera.

#### 4. ANALIZA PRETHODNIH ISTRAŽIVANJA

Cilj integracije alata u nastavi informatike u ranoj dobi poučavanja računalnog programiranja ima cilj razvoj kompetencija za rješavanje problema logičkog rasuđivanja. Postoje dva različita načina ili oblika programiranja što se tiču učenika osnovnih škola u Hrvatskoj. To su tekstualno i blokovsko kodiranje. Literatura koristi različite termine kako bi ih razlikovali. Kodiranje temeljeno na tekstu napisano s pomoću tipkovnice i spremljene kao tekstualne datoteke su tekstualni programski jezici (skr. CPL dalje u tekstu), dok programiranje kojima je temelj povlačenje slika koje nazivamo blokovi su vizualni programski jezici (skr. VPL dalje u tekstu) (Price CB, Price-Mohr RM, 2018). Neki od alata kao npr. Scratch prema studiji učenika 5. razreda osnovne škole ne sudjeluje u podizanju razine logičkog rasuđivanja, već samo podizanje srednje vrijednosti faktora samopouzdanja u rješavanju problema, (Kalelioglu i dr., 2019). CPL (tekstualni programski jezik) naspram VPL (vizualni programski jezik) računalnih programskih jezika, su poznati po svojoj kompleksnosti gdje učenici moraju znati sintaksu i semantiku programskog jezika, a i iznaći pravi put za rješavanje problema što je jednostavniji da se ne bih zapleli u problemu rješavanja istih. Rezultati istraživanja pokazuju da učenici imaju sposobnost učenja CPL-a ocjenom 71 %, također da su učenici uživali učenju CPL-a., te je preporuka dodati CPL u kurikulume osnovnih škola, kao i srednjim školama za učenje programiranja (WAKIL, K. i dr., 2019). Rezultati istraživanja pokazuju da su učenici dodatno izloženi izazovima u rješavanju problema što su morali naučiti okruženje temeljeno na CPL-u i jake koreliranosti matematičkog i računalnog razmišljanja što ide u prilog CPL načinu učenja programiranja (Cui, Z., i dr., 2021). Nastava informatike je u korelaciji s dvije varijable tehnološkim i situacijskim čimbenicima, te izbor alata nastavnika uvelike utječe na računalno zanimanje učenika i uspješnost samog programiranja (Sentance, S., Csizmadia, A., 2017). „...Na međunarodnoj razini, većina studija o alatima za poučavanje programiranja nije usmjerena na učenike od 4. do 6. razreda“ te tvrde „... da postoji praznina u postojećoj literaturi koja se odnosi na učenike koji pohađaju razrede 4. do 6.“ (Hill et al., 2015). Tim znanstvenika također kažu da nedostaje istraživanja koja su usmjerena na poučavanje programiranja učenika osnovnih škola (Elsawah, W i dr., 2023). „... Zaključci su da je važno zauzeti holistički pristup kako bi se istinski razumjele donesene prakse u učionici. Jedno od predloženih polazišta je

postavljanje četiri pitanja: Što podučavati? Zašto podučavati? Što je s učenicima (poteškoće i zablude)? Kako podučavati?“ (Gess-Newsome, J., 2015). Znanstvenici se nisu usuglasili koji je najbolji način za početak učenja programiranja, posebice u ranoj školskoj dobi (VLP naspram CPL) (Hillu i dr., 2015). Također nije jasno trebaju li učenici početi s VPL ili CPL (Salant i dr., 2011). Jedan od mnogih izazova i poteškoća s kojima se učenici u razredu se susreću su sljedeće: programiranje je sposobnost konceptualizacije realnog modela najčešće visoko korelirano s matematičkim znanjem gdje učenik uspješno primjenjuje i gaji sposobnost računalnog razmišljanja (Piteira, M i dr., 2018). U znanosti postoji konsenzus da je računalno razmišljanje mentalni proces koji koristi elemente mozgovne apstrakcije, generalizacije, razgradnju, algoritamsko rasuđivanje i ispravku pogreški (Webb, M i dr., 2017). Također postoji opći konsenzusu u znanstvenim krugovima kako programiranje učenika poboljšava opća učenička postignuća, te vlada mišljenje o težini prihvaćanja, razumijevanja i primjene stanovitih tema u programiranju (Alam A., 2022). Istraživanja kažu da se samoefikasnost i samoučinkovitost u programiranju podiže s praksom, a predlažu da bi trebalo poboljšati radno okruženje za učenje programiranja (Gunbatar MS, Karalar H., 2018), (Malaise, Y., & Signer, B., 2022) i (Tsai CY., 2019). Važnost stjecanja temeljnih kompetencija i vještina programiranja čine učenike sposobne da idu u korak s temeljima informacijskog društva kao osnovom za evoluciju učenja (Piteira, M i dr., 2018). Rano učenje programiranja kao i visok nivo znanja u učenicima stvara dobar osjećaj da učenik vlada s materijom što utječe na njegov odabir budućeg zanimanja (Noh J., Lee J., 2020) i (Zinovieva, IS i dr., 2021). Pa je preporuka započeti učenje programiranja u osnovnoj školi. Samoosnaživanje učenika svladavanjem programiranja konceptualizirano je kroz četiri komponente: smislenosti, utjecaja, kreativne samoučinkovitosti i samoučinkovitosti programiranja (S.C. Kong, M.M. Chiu, M. Lai, 2018). Na uzorku od 287 učenika osnovnih škola pokazano je da učenici s većim interesom za programiranje smatraju da je to značajnije, ima veći utjecaj, imaju veću kreativnu samoučinkovitost programiranja. Učenici čiji su stavovi prema suradnji bili pozitivniji od ostalih imali su veću kreativnu samoučinkovitost. Dječaci su pokazali veći interes za programiranje nego djevojčice. Učenici viših razreda smatrali su programiranje manje smislenim te su imali manju samoučinkovitost programiranja. Programiranje u parovima je korisno za poticanje računalnog razmišljanja (CT) u

učenika (Baichang Zhong, Qiyun Wang, Jie Chen, 2016). Prema ovome istraživanju promatrani su društveni čimbenici, i to: spol i partnerstvo kao programiranje u parovima u osnovnoj školi. Istraživanje je provedeno u četiri razreda osnovne škole učenika 6. razreda kineske osnovne škole. Rezultati istraživanja pokazali su da: nije bilo značajne razlike u kompatibilnosti među rodnim parovima, ali je bilo značajne razlike među partnerskim parovima, nije bilo značajne razlike u postignućima u programiranju i povjerenju među različitim parovima, ali su djevojčice postale produktivnije i pouzdanije u programiranju u parovima. Isto tako programiranje u parovima je pooštrilo partnerstvo unutar parova. Ovo istraživanje sugerira da bih učitelji trebali uzeti u obzir kao važan čimbenik u programiranju rad u parovima ili drugom obliku suradničkog učenja i usvojiti programiranje u parovima kao učinkovit pristup smanjenju rodnog jaza i poticanja komunikacije među učenicima. Programiranje u parovima ima očite prednosti u odnosu na samostalno programiranje, uključujući, ali ne ograničavajući se na sljedeće aspekte: programiranje u parovima može značajno poboljšati individualne vještine programiranja i promicati produktivnost ili kvalitetu programa jer učenicima pruža jasno definiranu metodu koja pomaže jedni drugima razumjeti kontekst problema i razmišljati o doprinosu kao oba programera (Braught i dr., 2008, Cliburn, 2003, Hannay i sur., 2009, Li i sur., 2013, Tomayko, 2002).

Jedan od pretežitih ili glavnih oblika programskih obrazovnih aktivnosti su isključene programske aktivnosti učenika u osnovnoj školi koje se provode u ranoj fazi osnovnog obrazovanja (Lihui Sun i dr., 2021). Istraživanje je pokazalo da je isključeno programiranje učinkovitiji način za poboljšanje CT vještina učenika 7 razreda osnovne škole. Znanje iz matematike i prethodno znanje iz programiranja imalo je značajan utjecaj na njihove CT vještine, ali taj se utjecaj ne odražava u procesu isključenih programskih aktivnosti. Zaključak istraživanja je da svi učenici mogu razviti svoje CT vještine kroz isključene programske aktivnosti, bez obzira na njihovu pozadinu učenja. Isključene programske aktivnosti mogu se koristiti kao učinkovit način za poboljšanje i razvoj CT vještina. Stoga isključene aktivnosti programiranja mogu riješiti probleme kao što su ograničenja hardverskih uređaja kao što su računala ili nedostatak rezervi znanja za rano učenje programiranja kod učenika. Isključene programske aktivnosti mogu podržati sudjelovanje učenika u uključenim programskim aktivnostima u budućnosti, te da je u ranim fazama osnovnog obrazovanja prikladnije provoditi CT

putem isključenih programskih aktivnosti prije uključenih programskih aktivnosti (Caelie i dr., 2020). Kroz pregled literature da isključene programske aktivnosti mogu pomoći učenicima i učiteljima da stimuliraju motivaciju za istraživanje Informatike na smislen i atraktivan način, a također mogu pomoći učenicima da provedu naknadno "uključeno" učenje (Bell i dr., 2016). Isključene programske aktivnosti mogu učinkovito razviti CT vještine učenika i pomoći im da prenesu CT vještine na druge scenarije rješavanja problema. Isključene programske aktivnosti su omogućile učenicima da imaju najvišu razinu razumijevanja algoritama, logičkih predviđanja i koncepata ispravljanja pogrešaka (Conte i dr., 2023). Na razvoj CT vještina u programskim aktivnostima utječu i ograničavaju ga različiti čimbenici, a metode poučavanja mogu utjecati na stjecanje CT vještina učenika. Dječaci i djevojčice podjednako mogu poboljšati CT vještine iz robotskih aktivnosti (Angeli i dr., 2020). Unutarnje psihološke karakteristike učenika također će utjecati na njihove CT vještine. Objavljene studije o uspješnosti učenika na Dabar zadacima pokazale su da je u većini CT zadataka učenici imaju znatno više rezultate od učenica (Kalelioğlu, 2015). Proučavanjem razvoja CT vještina učenika u programiranju učenja obrazovnih robota djevojčice trebaju više vremena za obuku od dječaka kako bi dosegle istu razinu CT vještina (Atmatzidou i dr., 2016). Metode poučavanja utjecat će na stjecanje CT vještina učenika (Lihui Sun i dr., 2021). Stavovi i stilovi razmišljanja učenika imaju značajan utjecaj na njihove CT vještine kao i stav učenja učenika osnovnih škola može značajno utjecati na njihove CT vještine. Spol predstavlja ključni čimbenik koji utječe na njihove CT vještine. Spol nema utjecaja na učenje programiranja (Akinola, S. O., 2015). Podučavanje programiranja korištenjem robota značajno poboljšava računalno razmišljanje i kreativnost. Računalno razmišljanje, međutim, nije značajno poboljšano u skupini koja je u početku pokazala visoke ocjene. Nadalje, kreativnost je poboljšana više kod djevojčica nego kod dječaka, a srednja razlika bila je statistički značajna, dok razlika u računalnom razmišljanju nije bilo. Implikacija ovog istraživanja je da je najbolji pristup osmisliti tečaj programiranja robota i primijeniti ga u stvarnim učionicama kako bi se raspravljalo o strategijama poučavanja i učenja prema prethodnim vještinama učenika i njihovom spolu (Noh, J., Lee, J., 2020). Čimbenik učenja programiranja jesu ekonomski i tehnički zahtjevi stvoreni računalnom tehnologijom (Chen i dr., 2017). Spol je važan čimbenik u programiranju obrazovanja (Carter i Jenkins, 1999). Korištenjem

robotu u učenju programiranja većina istraživanja pokazala je da su dječaci tradicionalno bolje upoznati s tehnologijom i da im je potrebno manje vremena da dosegnu određenu razinu računalnog razmišljanja nego djevojčice (Papastergiou, 2009) i (Atmatzidou i Demetriadis, 2016). Dječaci su pokazali značajna poboljšanja u apstrakciji, generalizaciji, modularnosti i razgradnji problema, dok su se djevojčice značajno poboljšale u apstrakciji, algoritmima, modularnosti i razgradnji problema. Dječaci općenito imaju više iskustva s računalima (Papastergiou, 2009). Dječaci više sudjeluju u radu na računalima (Rusk i dr., 2008). Dječaci imaju pozitivniji stav prema računalima nego djevojčice (Baser, 2013). Spol je važna varijabla u obrazovanju o programiranju, ali na sposobnost učenja programiranja mogu utjecati i druge varijable, poput rase i osobnosti (Akinola, 2015). Stav je veći za dječake nego za djevojčice. Srednja ocjena za stav bila je viša za dječake nego za djevojčice, a srednja razlika bila je statistički značajna. Srednja ocjena da je veća za dječake nego za djevojčice u samopouzdanju u programiranju, percipiranoj korisnosti programiranja i motivaciji za programiranje. Ta su područja pod faktori stava prema programiranju. Srednje razlike također su bile statistički značajne (Baser, 2013). Razine obrazovanja, učenički uspjeh na satovima Matematike i predmetima znanosti, kao i načini razmišljanja utjecali su na računalno razmišljanje, ali spol nije (Durak, H. Y., i Saritepeci, M., 2018). Nevažno je ako učenik ima visoko razvijene formalne sposobnosti rasuđivanja i rješavanja problema kada nema motivaciju za programiranje, ali bismo mogli pozitivno utjecati na afektivnu dimenziju odabirom odgovarajućeg programskog jezika, konteksta i pedagoškog pristupa (Mladenović, M. i dr., 2018). Također izbor programskog jezika je značajan za uspješnost u učenju programiranja, tako učenici s nižim sposobnostima rješavanja problema uspješniji su u Scratchu nego u tekstualnim programskim jezicima Logo i Python. Srednji i slabiji učenici imaju bolje rezultate pri programiranju u Scratchu od programiranja u Logu ili Pythonu. Biti uspješan u programiranju je biti sposoban za pretvaranje apstraktnih problema u programski kod kao radno rješenje na računalo (SitiRosminah MD Derus1, Ahmad ZamzuriMohamad Ali2, 2012). Za početnike da bi uspjeli u programiranju, potreban im je skup strategija za učenje koje će im pomoći da se nose s procesom programiranja kao i utjecati na način na koji početnik percipira proces programiranja. Zapravo, nedostatak vizualizacije stanja programa tijekom izvršavanja koda jedan je od činitelja poteškoće učenika u učenju programiranja. Stoga

su alati za vizualizaciju kao pomagala za učenje prikladni da bi pomogli učenicima razumjeti provedbu programskog koda, kao što su promjene na varijablama u memoriji računala. Međutim učenici bi trebali biti aktivno uključeni u alat za vizualizaciju kako bi se osiguralo da može imati pozitivan utjecaj na razumijevanje učenika o informacijama koje treba pratiti. Osim toga, korištenje alata za vizualizaciju može se koristiti tijekom praktične aktivnosti u laboratoriju kroz grupnu aktivnost o kojoj učenici mogu aktivnije raspravljati i istraživati o poduzetim aktivnostima. Prema teoriji multimedijalnog učenja za uspješne pedagoške smjernice za dizajniranje sustava za audio-vizualno učenje s aspekta obrazovnog stajališta upotreba obrazovnih tehnologija treba uzeti u obzir kako učenici uče. Smatra se da postoji sustav za procesuiranje vizualnih (animacije) i verbalnih (naracija) informacija. U multimedijalnom učenju učenik uključuje tri kognitivna procesa. Odabir nadolazećih verbalnih i vizualnih informacija u svrhu izrade tekstualne baze i baze slika. Organizacija tih baza u svrhu stvaranja modela sustava koje se treba objasniti. Integracija se događa kada učenik stvara veze između odgovarajućih dijelova između verbalnog i vizualnog modela informacija (Mayer i Moreno, 2002).

Učenje kao i poučavanje su složeni koncepti. Jedan traži usvajanje znanja, a drugi davanje ili prijenos znanja. To su dva ovisna procesa i ako ne rade u sinergiji i ako imaju međusobnih šumova u komunikaciji, rezultat transmisije znanja izostaje. Kako opada? Brzina prijenosa opada brzo, a naročito u trenutku kada inicijator znanja ili prenositelj znanja (učitelj) ne uoči prekid u transmisiji znanja između ova dva gornja objekta: primatelja i prenositelja znanja. Zato učitelj mora biti taj koji će na vrijeme znati kada treba provjeriti dali ta veza prijenosa i dalje funkcionalna. Najlakše je provjeriti pitanjima o sadržaju ili propitivanjima. Kao što učitelj može saznati jeli prekid prijenosa znanja nastupio dobivanjem krivog odgovora također potvrdu prekida prijenosa znanja učitelj može dobiti i postavljanjem krivog učeničkog pitanja. Oba upućuju na isto da je nastupila miskoncepcija. Predmet istraživanju u ovome radu nisu miskoncepcije već jedna druga priroda problema koja se javlja kod početnika učenja programiranja i činiteljima u nastavi koji ometaju učenike u radu. To su problemi koji se javljaju u nastavi programiranja, a odnose se na nekompetentnost učenika da svlada druge digitalne vještine kako bih mogao uspješno programirati, kao što je korištenje lokalno instaliranog prevoditelja naspram online sustava za učenje programiranja Replit, gdje

komplikiranosti ili nemogućnost pronalaska rješenja također izaziva prekid u prijenosu znanja.



## **5. ISTRAŽIVANJE MIŠLJENJA UČENIKA O PRIMJENI SUSTAVA REPLIT U NASTAVI INFORMATIKE U DOMENI RAČUNALNOG RAZMIŠLJANJA I PROGRAMIRANJA U VIŠIM RAZREDIMA OSNOVNE ŠKOLE**

### **5.1. Metodologija istraživanja**

#### **5.1.1. Cilj i metode istraživanja**

Cilj ovoga istraživanja je ispitati može li sustav za učenje programiranja Replit biti bolji oblik učenja programiranja u osnovnoškolskoj nastavi Informatike u domeni računalnog razmišljanja i programiranja naspram lokalno instaliranog prevoditelja IDLE. Replit je uzet kao paradigma vrste tehnološkog alata u nastavi kao oblik za učenje programiranja. Vodilja istraživanja su bili problemi koji se javljaju tijekom rada u učionici i kod kuće, tj. u učenju programiranja, a koje učenike neposredno ometaju u radu i učenju, te mogu li se negativni čimbenici otkloniti s pomoću sustava za učenje programiranja Replit. Iz tih razloga je izabrano 6 odjeljenja i učenika koji nikada nisu koristili sustav za učenje Replit u nastavi Informatike. Učenicima je dana suglasnost o sudjelovanju u istraživanju na znanje roditeljima. Imena učenika niti u jednom dijelu rada nisu predstavljena javnosti.

U istraživanju je korišten kvantitativni način prikupljanja informacija o učeničkoj problematici koja se javlja prilikom programiranja kako u školi tako i kod kuće u istraživanom sustavu za učenje programiranja Replit u školi i kod kuće.

#### **5.1.2. Procedura istraživanja**

Prva faza istraživanja započinje s istraživanjem postojeće literature o činiteljima uspješnosti programiranja u osnovnoj školi, bitnosti programiranja, kakav ona ima utjecaj na razvoj mišljenja učenika o primjeni sustava za učenje programiranja Replit i problema s kojim se učenici susreću u programiranju.

U razdoblju od 5. – 12. prosinca 2023. je provedena druga faza istraživanja te su provedeni polu strukturirani razgovori s učenicima tri 5. i tri 6. razreda osnovne škole u ukupnom trajanju od 6 školskih sati (svako odjeljenje po 1 školski sat) za cijeli

odabrani uzorak učenika. Cilj intervjua je bio prikupiti što više informacija od učenika koji su to problemi koji ih onemogućavaju u uspješnom učenju programiranja.

Mišljenja i svjedočanstva učenika jedan su od najvažnijih elemenata pristupu ovom dijelu kvalitativnog istraživanja i najbolja je opcija za dobivanje temeljnih informacija o problemima i iskustvima učenika koji se susreću tijekom programiranja. Polu strukturirani pristupni intervju je prva u nizu metoda koja se koristi u ovome radu kako bi se skupilo iskustvo, mišljenja i problemi učenika koji se susreću tijekom programiranja. Tijekom intervjua prisutan je otvoreni dijalog s učenicima. Intervju se vodio polu strukturiranim protokolom koji je omogućio postavljanje otvorenih pitanja kao i dodatnih pitanja na temelju odgovora učenika. Također pitanja osmišljena od strane ispitivača bila su usmjerena k tome da vode prema pitanjima koja su povezana s istraživačkim ciljem.

U trećoj fazi istraživanja učenicima je predstavljen sustav za učenje programiranja Replit svako odjeljenje po 1 školski sat, ukupno 6 školskih sati, gdje su učenici se uključili u rad i programirali, te od kuće sudjelovali u suradničkom - kolaborativnom radu.

U četvrtoj fazi istraživanja nakon rada u sustavu za učenje programiranja Replit u školi i samostalnog rada od kuće slijedila je anonimna istraživačka anketa među učenicima.

U završnoj fazi su obrađeni podaci i donesena zaključna studija istraživanja.

### **5.1.3. Anketni upitnik**

Pitanja koja su usvojena za istraživačku anketu dobivena su kao izvor polu strukturiranog razgovora s učenicima u trajanju od 45 min, dio pitanja je sastavljen na temelju nastavničke prakse, a neka usvojena iz literature (Suman, D. i Purković, D., 2018), (Režić, J., 2020) i (Marković, M., 2019) te prilagođena svrsi istraživačke studije, a ista su napravljena u Microsoft Forms obrascu kao anonimna anketa učenika, sastavljena po modelu Likertove ljestvice. Anketa sadrži 17 pitanja.

Prva skupina pitanja sastavljena na temelju izvora polu strukturiranih razgovora s učenicima: „Koliko vam je rad u Replitu jednostavniji u odnosu na prije s pomoću IDLE programskog rješenja?“; „U školi mi je lakše raditi u Replitu.“; „Lakše mi je pratiti nastavu u Replitu.“; „Kada bih birao raditi kod kuće i u školi u Replitu moj izbor bih bio Replit.“; „Lakše mi je izvršavati svoje školske obveze u Replitu.“

Druga skupina pitanja sastavljena je na temelju motivacijskih aspekata istraženih u literaturi koji ovise o izabranom sustavu za učenje programiranja: „Nastava u Replitu mi je zanimljivija.“; „Mislim da bolje napredujem u programiranju od kada koristim Replit.“; „Osjećam se modernije i više informatički obrazovan/obrazovana od kada koristim Replit“; „Preporučio bih i prijateljima učenicima koji ne idu s mnom u školu ili razred da koriste Replit.“

Treća skupina pitanja sastavljena je na temelju vlastite nastavničke prakse: „Lakše mi je doći do datoteke sve su na jednom mjestu.“; „Nemam problema s instalacijom i ne moram se brinuti odakle ću rješavati domaću zadaću.“;

Četvrta skupina pitanja sastavljena je temelju literature i prednosti sustava za učenje programiranja koji podržavaju suradnički rad: „Lakše mi je s prijateljima iz razreda skupa rješavati problemske zadatke iz programiranja.“; „Često skupa rješavamo domaću zadaću.“; „Drago mi je raditi u Replitu jer uvijek mogu postaviti pitanje prijateljima iz razreda i učitelju ako imam problem u rješavanju programskih zadataka.“; „Volim timski rad i kada možemo svi skupa raditi na programskom rješenju.“; „Kada bih bio programer jednog dana sigurno bih radio u sličnom alatu za suradnju sličnom Replitu.“

Anketa je provedena na uzorku od 57 učenika tri 5. razreda, i tri 6. razreda osnovne škole „Antuna Mihanovića Petropljskog“ u Drnišu. Istraživanje je provedeno krajem prosinca 2023. neposredno prije zimskih praznika. Prikupljeni podaci obrađeni su kvalitativno i kvantitativno.

Namjera rada je da će rad pridonijeti boljoj perspektivi razumijevanja problema s kojima se učenici osnovnih škola susreću prilikom učenja programiranja, te su postavljena sljedeća istraživačka pitanja:

**Istraživačko pitanje 1:** Učenje programiranja u sustavu za učenje programiranja Replit stvara osjećaj ugone i podrške u učenju programiranja, te podiže motivaciju u učenika.

**Istraživačko pitanje 2:** Učiteljev odabir za rad u sustavu za učenje programiranja Replit ima za rezultat pedagošku opravdanost za korištenje iste.

**Istraživačko pitanje 3:** Učiteljev odabir nastavnog alata za podučavanje programiranja ima značajan utjecaj na učenje programiranja.

#### **5.1.4. Uzorak ispitanika**

Okvir uzorka je odabran na temelju čimbenika koji su istraženi u literaturi, učenici 5. i 6. razreda osnovne škole, kao i poznavanju specifičnosti problema učenika te razvojne dobi. Učenici u Hrvatskoj s programiranjem u CPL obliku počinju u 5. razredu osnovne škole.

Uzorak se sastojao od učenika 5. i 6. razreda osnovne škole, ukupno 6 odjeljenja. Učenika 5. razreda je ukupno sudjelovalo 40 učenika, a učenika 6. razreda ukupno je sudjelovalo 44 učenika. Ukupno (N=84) učenika 5. i 6. razreda osnovne škole.

U drugoj fazi istraživanja sudjelovalo je N=79 učenika, od tog je 39 učenika 5. razreda, i 40 učenika 6. razreda. Učenika 5. razreda u intervju je sudjelovalo 15, dok je učenika 5. razreda sudjelovalo 24. Učenika 6. razreda u intervju je sudjelovalo 24, dok je učenika 6. razreda sudjelovalo 16.

U četvrtoj fazi istraživanja sudjelovalo je N=57 učenika, od tog je 37 učenika 5. razreda, i 20 učenika 6. razreda. Učenika 5. razreda u intervju je sudjelovalo 15, dok je učenika 5. razreda sudjelovalo 22. Učenika 6. razreda u intervju je sudjelovalo 14, dok je učenika 6. razreda sudjelovalo 6.

#### **5.2. Rezultati istraživanja**

U **dugoj fazi** polu strukturiranih razgovora rezultati istraživanja su sljedeći:

5. prosinca 2023. je proveden prvi intervju sa sva tri 5. razreda. U prvoj fazi istraživanja, i prvom ispitivanom odjeljenju (5.a., 13 učenika) probleme u programiranju koje učenici navode 10/13 njih su: „da im je programiranje teško“, nisu naveli specifične probleme što se tiče pisanja koda, također su naglasili 11/13 učenika: „da im je teško samostalno pisati kod“. Svi učenici su naveli: „u školi da programiraju u IDLE programskom rješenju“. Na upit kako doma rješavaju domaću zadaću što se tiče zadataka iz programiranja? Odgovor svih učenika je bio „nikako „. Na pitanje zašto, odgovor je bio: „zato što nemamo instaliran IDLE“ na kućnim računalima. Ponovno na postavljeno pitanje zašto nemate instaliran, odgovori su bili: „zato što nam je to teško“ 12/13, „to ne znamo sami“ 13/13, „strah nas je da nešto ne pokvarimo“ 1/13 učenika. U drugo ispitivanom razredu (5.b., 13 učenika), svi učenici navode „u školi programiramo u IDLE programskom rješenju“, da „kod kuće nemamo instaliran IDLE programsko rješenje“, na postavljeno pitanje kako od kuće programirate, učenici odgovaraju da „kod kuće ne možemo domaću zadaću iz programiranja uraditi“, na pitanje zašto kod kuće nemate instaliran IDLE programsko rješenje, odgovori su sljedeći: „nemam dovoljno znanja da to sam napravim“ (10/13), „moji roditelji to ne znaju, vjerojatno“ (8/13), „prekomplicirano mi je“ (12/13), „strah me je da ne napravim što krivo“ (8/13). U treće ispitivanom razredu (5.c., 13 učenika) svi učenici se slažu: „u školi koristimo IDLE programsko rješenje u kojemu programiramo“, na moje pitanje kako kod kuće rješavate svoju domaću zadaću iz programiranja odgovor je „nikako“, na moje pitanje zašto, odgovor je: „zato što kod kuće nemamo instaliran IDLE“, ponovno na pitanje zašto nemate instaliran IDLE, odgovori su sljedeći: „teško ga je instalirati vjerojatno“ (10/13), „nisam ni probao“ (9/13), „probao sam, ali ne znam zašto nije uspjelo“ (3/13), na moje pitanje jeste tražili pomoć roditelja, odgovori su sljedeći: „nismo“ (8/13), „oni to ne znaju“ (10/13).

8. prosinca 2023. je proveden polu strukturirani razgovor s učenicima 6-ih razreda. U četvrtom ispitivanom razredu (6.a., 14 učenika, sudjelovalo 12, 2 odsutna) svi učenici su se složila da u školi programiraju u „IDLE programskom rješenju“, „kod kuće nemaju instaliran IDLE programsko rješenje“, jedan dio učenika se složio „da je to komplicirano instaliranje“ (8/12), jedan dio učenika je izjavio da „ne bih znali to instalirati“ (7/12), na moje pitanje jeste li pokušali instalirati „ne nismo, mislimo da je komplicirano“ (10/12), dvoje se složilo „da misle da nije komplicirano, ali da nisu

probali“. U petom ispitivanom razredu (6.b., 15 učenika) svi su se učenici složili „da u školi rade u IDLE programskom rješenju“ jedan dio učenika se slaže „da im IDLE programsko rješenje zadaje problem u radu“ (4/15), na moje pitanje koji su to problemi su odgovori sljedeći: „ne mogu sami nekada riješiti zadan zadatak“ (6/15), „ne znaju gdje im je spremljena radna datoteka“ (12/15), na pitanje kako od kuće radite zadatke iz programiranja jedan dio učenika se složio „kod kuće imam instaliran IDLE programsko rješenje i u njemu programiram“ (5/15), na pitanje koji su vam problemi kada radite od kuće (misli se na učenike koji imaju instaliran IDLE) domaću zadaću tj. programirate jedan dio učenika se složio „da je teško jer im nema tko pomoći“ (5/15), drugi dio učenika „često ne znam odakle krenuti u rješavanju“ (4/15), te da „često odustanem jer izgubim volju kad mi ne radi“ (5/15). U šestom ispitivanom razredu (6.c., 15 učenika, dva učenika odsutna) svi se slažu „da u školi rade u IDLE programskom rješenju“, te „da kući nemaju instaliran IDLE programsko rješenje“. Na moje pitanje kako od kuće programirate i rješavate svoju domaću zadaću odgovor je bio „nikako“ (13/13), na pitanje: „Jeste li pokušali instalirati IDLE programsko rješenje“ odgovori su sljedeći: „nismo nikada pokušavali“ (13/13), „čini nam se to komplicirano“ (10/13), „to je teško“ (8/13) i „strah nas je da nešto ne napravimo krivo“ (5/13). Zaključak prve faze istraživanja bih bio: samo 5 učenika od istraživanog uzorka tj. 6,49 % učenika imaju kod kuće instaliran IDLE programsko rješenje i mogu samostalno programirati. Ostatak tj. većina smatra to kompliciranim, zahtjevnim i nisu sigurni kako bih to uspješno napravili. Također učenici su mišljenja da je programiranje zahtjevna i kompleksna kognitivna vještina koja im je vrlo teška kada nešto moraju sami napraviti bez tuđe pomoći.

U **četvrtoj fazi** nakon provedene istraživačke ankete rezultati su sljedeći:

U tablici 3 prikazani su mogući odgovori učenika s obzirom na tvrdnje i na mjernu skalu od 1 – 5 na Likertovoj skali:

Tablica 3. Mogući odgovori učenika s obzirom na tvrdnje (Izvor: Autor rada)

1	2	3	4	5
---	---	---	---	---

UOPĆE SE NE SLAŽEM	UGLAVNOM SE NE SLAŽEM	NITI SE SLAŽEM NITI SE NE SLAŽEM	UGLAVNOM SE SLAŽEM	U POTPUNOSTI SE SLAŽEM
-----------------------	-----------------------------	---	-----------------------	------------------------------

1 – uopće se ne slažem; 2 – uglavnom se ne slažem; 3 – niti se slažem niti se ne slažem; 4 – uglavnom se slažem; 5 – u potpunosti se slažem.

Sljedeća tablica 4 prikazuje mišljenje učenika vezanih uz odgovore na odabrane tvrdnje u anketnom upitniku o primjeni sustava Replit s obzirom na mjernu skalu od 1 – 5 na Likertovoj skali, nadalje u tablici su prikazane frekvencije i postotci ispitanika na skali od 1 – 5, vrijednost M – aritmetička sredina i  $\sigma$ - standardna devijacija.

Tablica 4: Mišljenje učenika vezanih uz odgovore na odabrane tvrdnje u anketnom upitniku o primjeni sustava Replit (Izvor: obrada autora)

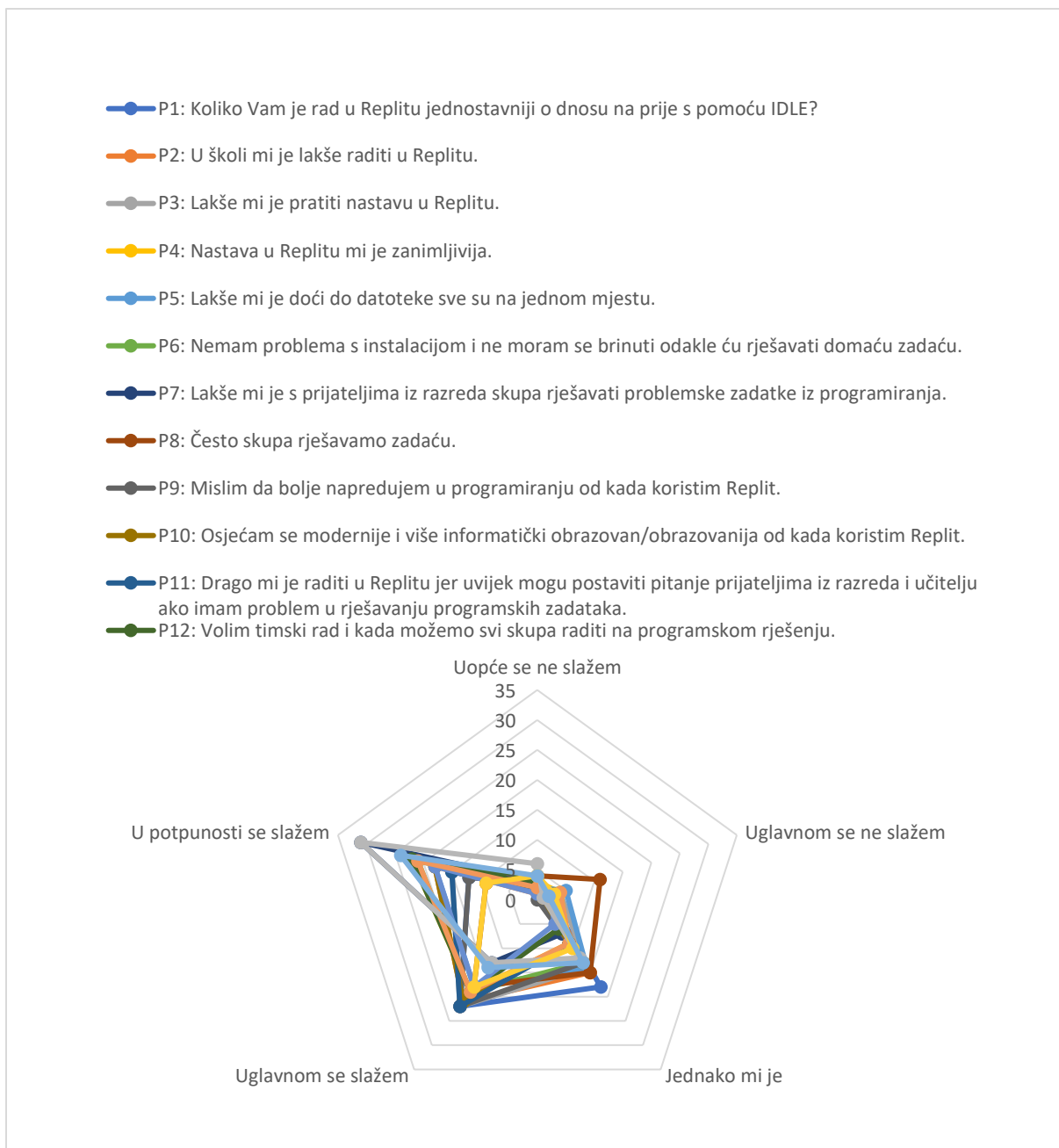
TVRDNJA	FREKVENCIJA I POSTOTCI (%) UČENIKA (N=57)						
	1	2	3	4	5	M	$\sigma$
T1: Koliko vam je rad u Replitu jednostavniji u odnosu na prije s pomoću IDLE programskog rješenja?	3	2	18	22	12	3.66	0.93
	5.3%	3.5%	31.6%	38.6%	21.1%		
T2: U školi mi je lakše raditi u Replitu.	1	2	15	21	18	3.93	0.94
	1.8%	3.5%	26.3%	36.8%	31.6%		
T3: Lakše mi je pratiti nastavu u Replitu.	2	4	14	22	15	3.77	1.03
	3.5%	7%	24.6%	38.6%	26.3%		
T4: Nastava u Replitu mi je zanimljivija.	3	4	9	19	22	3.93	1.20
	5.3%	7%	15.8%	33.3%	38.6%		

T5: Lakše mi je doći do datoteke sve su na jednom mjestu.	2	5	14	18	18	3.79	1.09
	3.5%	8.8%	24.6%	31.6%	31.6%		
T6: Nemam problema s instalacijom i ne moram se brinuti odakle ću rješavati domaću zadaću.	2	2	13	19	21	3.96	1.03
	3.5%	3.5%	22.8%	33.3%	36.8%		
T7: Lakše mi je s prijateljima iz razreda skupa rješavati problemske zadatke iz programiranja.	3	3	7	13	31	4.16	1.73
	5.3%	5.3%	12.3%	22.8%	54.4%		
T8: Često skupa rješavamo domaću zadaću.	4	11	15	18	9	3.29	1.15
	7%	19.3%	26.3%	31.6%	15.8%		
T9: Mislim da bolje napredujem u programiranju od kada koristim Replit.	3	0	13	33	8	3.75	1.09
	5.3%	0%	22.8%	57.9%	14%		
T10: Osjećam se modernije i više informatički obrazovan/obrazovana od kada koristim Replit.	4	1	9	23	20	3.95	1.10
	7%	1.8%	15.8%	40.4%	35.1%		
T11: Drago mi je raditi u Replitu jer uvijek mogu postaviti pitanje prijateljima iz razreda i učitelju ako imam problem u rješavanju programskih zadataka.	2	3	9	13	30	4.16	1.09
	3.5%	5.3%	15.8%	22.8%	52.6%		
T12: Volim timski rad i kada možemo svi skupa raditi na programskom rješenju.	3	2	6	17	29	4.18	1.09
	5.3%	3.5%	10.5%	29.8%	50.9%		



T13: Uz Replit mi je olakšan dolazak do vlastitog rješenja na temelju tuđih idejnih rješenja.	1	2	15	19	20	3.96	0.88
	1.8%	3.5%	26.3%	33.3%	35.10%		
T14: Kada bih birao raditi kod kuće i u školi u Replitu moj izbor bih bio Replit.	2	4	9	23	19	3.93	1.05
	3.5%	7%	15.8%	40.4%	33.3%		
T15: Kad bih bio programer jednog dana sigurno bih radio u sličnom alatu za suradnju sličnom Replitu.	6	1	12	19	19	3.68	1.24
	10.5%	1.8%	21.10%	33.3%	33.3%		
T16: Preporučio bih i prijateljima učenicima koji ne idu s mnom u školu ili razred da koriste Replit.	4	3	10	19	21	3.89	1.18
	7%	5.3%	17.5%	33.3%	36.8%		
T17: Lakše mi je izvršavati svoje školske obveze u Replitu.	4	2	13	14	24	3.91	1.20
	7%	3.5%	22.8%	24.6%	42.1%		

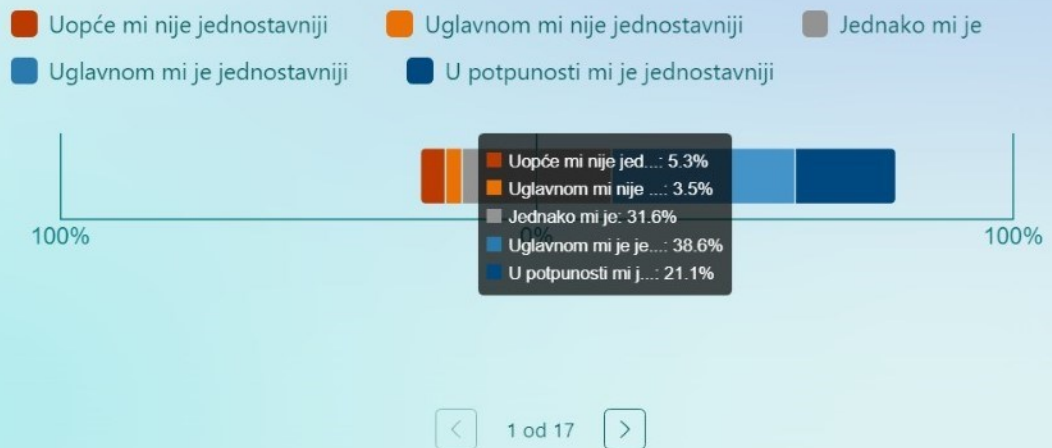
:



Slika 4. Grafički prikaz podataka (Izvor: obrada autora)

Na slici 4 dan je grafički prikaz rezultata istraživanja u plošnom grafikonu koja sadržava istraživačka pitanja i frekvencije pojavljivanja odgovora.

## Koliko vam je rad u Replit jednostavniji u odnosu na prije pomoću IDLE programskog rješenja?

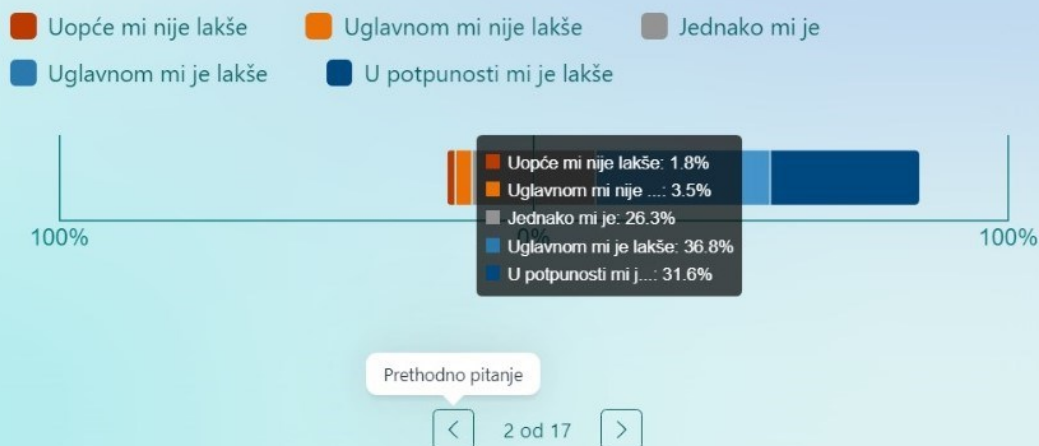


**Grafikon 1.** Koliko vam je rad u Replitu jednostavniji u odnosu na prije s pomoću IDLE programskog rješenja

Na grafikonu 1 prikazani su odgovori učenika na pitanje koje se odnosilo na jednostavnost rada u Replitu tj. procjenu učenika koliko im je radu Replitu jednostavniji nego prije s pomoću lokalno instaliranog prevoditelja IDLE. Odgovori učenika su sljedeći: 5.3 % učenika tvrdi rad u Replitu uopće nije jednostavniji nego prije. 3.5 % učenika tvrdi da im rad u Replitu uglavnom nije jednostavniji nego prije. 31.6 % učenika tvrdi da im se nije ništa promijenilo po pitanju jednostavnosti rada naspram prije. Vrlo velika većina učenika njih 38.6 % tvrdi da im je rad u Replitu uglavnom jednostavniji nego prije. Mali postotak učenika njih 21.1 % tvrdi da u potpunosti im je rad u Replitu jednostavniji nego prije. Aritmetička sredina iznosi 3.66, a standardna devijacija 0.93.

Poslano je ovoliko odgovora: 57

## U školi mi je lakše raditi u Replitu.



**Grafikon 2.** U školi mi je lakše raditi u Replitu

Na grafikonu 2 prikazani su odgovori učenika na pitanje jeli im je u školi lakše raditi u Replitu. Vrlo mali postotak svega 1.8 % učenika izjavljuje da im uopće nije lakše raditi u Replitu. 3.5 % učenika kaže da im uglavnom nije lakše raditi u Replitu. Učenici koji su nepromijenjenog mišljenja te kažu da im je jednako po pitanju lakoće rada ih je 26.3 %. Učenici koji izjavljuju da im je uglavnom lakše raditi u Replitu je njih 36.6 %, dok učenika koji kažu da im je u potpunosti rad u Replitu lakši je njih 31.6 %. Aritmetička sredina iznosi 3.93, a standardna devijacija 0.94.

Poslano je ovoliko odgovora: 57

## Lakše mi je pratiti nastavu u Replitu.

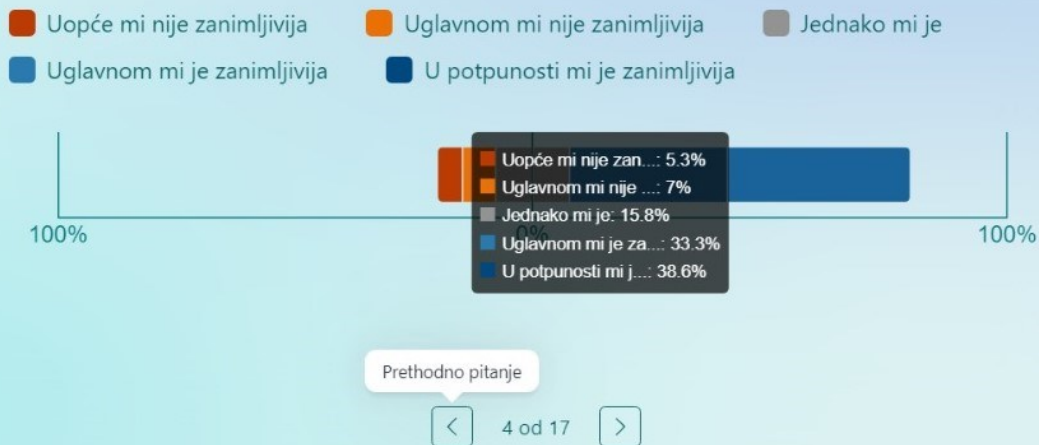


### Grafikon 3. Lakše mi je pratiti nastavu u Replitu

Na grafikonu 3 prikazani su odgovori učenika na pitanje koliko im je lakše pratiti nastavu u Replitu. Mišljenja da im uopće nije lakše pratiti nastavu u Replitu su 3.5 % učenika. Njih 7 % ih kaže da im uglavnom nije lakše pratiti nastavu u Replitu. Četvrtina učenika njih 24.6 % ima nepromijenjeno mišljenje po pitanju smanjenja opterećenja. Vrlo velika većina učenika ima pozitivan stav u smislu smanjenja opterećenja, njih 38.6 % je mišljenja da im je uglavnom lakše pratiti nastavu, a 26.3 % učenika izjavljuje da im je u potpunosti lakše pratiti nastavu u Replitu. Aritmetička sredina iznosi 3.77, a standardna devijacija 1.03.

Poslano je ovoliko odgovora: 57

## Nastava u Replitu mi je zanimljivija.

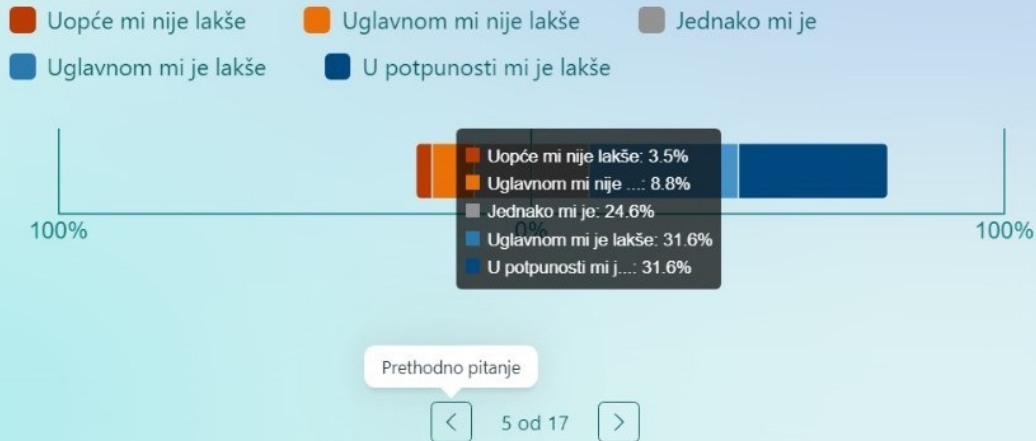


### Grafikon 4. Nastava u Replitu mi je zanimljivija

Na grafikonu 4 su prikazani odgovori učenika na pitanje koliko im je nastavu u Replitu zanimljiva. Na pitanje zanimljivosti nastave u Replitu odgovori su slijedeći: 5.3 % učenika kaže da im nastava u Replitu uopće nije zanimljiva, a 7 % njih da im nastava uglavnom nije zanimljiva u Replitu. 15.8 % učenika je nepromijenjenog mišljenja. 33.3 % učenika se slaže s tim da im je nastava u Replitu uglavnom zanimljivija, a 38.6 % učenika izjavljuje da im je nastava u Replitu u potpunosti zanimljivija. Aritmetička sredina iznosi 3.93, a standardna devijacija 1.20.

Poslano je ovoliko odgovora: 57

## Lakše mi je doći do datoteke sve su na jednom mjestu.



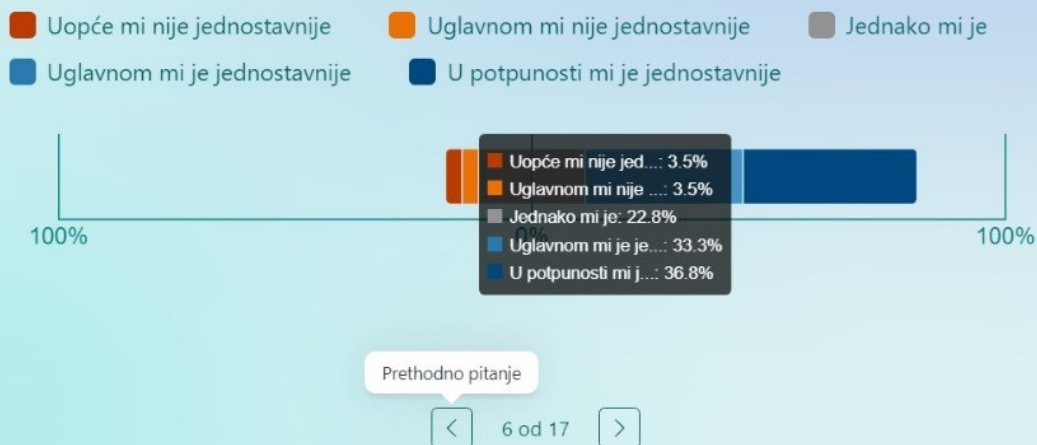
### **Grafikon 5:** Lakše mi je doći do datoteke sve su na jednom mjestu

Na grafikonu 5 prikazani su odgovori učenika na pitanje koliko im je olakšavan pronalazak i rad s datotekama u Replitu. Odgovori učenika su sljedeći:

Učenici koji su nepromijenjenog mišljenja tj. da i Replit uopće ne olakšava pronalazak datoteka je njih 3.5 %. 8.8 % ih je mišljenja da im uglavnom nije lakši put pronalaska datoteka za rad. Učenika s nepromijenjenim mišljenjem je 24.6 %. Dobar dio učenika 31.6 % je mišljenja da im Replit uglavnom olakšava put pronalaska datoteka, kao i 31.6 % učenika je mišljenja da im Replit u potpunosti olakšava pronalazak datoteka za rad. Aritmetička sredina iznosi 3.79, a standardna devijacija 1.09.

Poslano je ovoliko odgovora: 57

## Nemam problema s instalacijom i ne moram se brinuti odakle ću rješavati domaću zadaću.



**Grafikon 6:** Nemam problema s instalacijom i ne moram se brinuti odakle ću rješavati domaću zadaću

Na grafikonu 7 prikazani su odgovori učenika na pitanje s obzirom na problematiku instalacije programa i rješavanja domaće zadaće. Odgovori učenika su slijedeći: 3.5 % učenika se izjašnjava da im uopće nije jednostavnije po pitanju instalacije od kuće, također 3.5 % učenika se izjašnjava da im uglavnom nije lakše po pitanju instalacije, 22.8 % učenika ima nepromijenjeno mišljenje po pitanju instalacije. 33.8 % učenika se izjašnjava da im je uglavnom lakše po pitanju instalacije od kuće i da se ne moraju brinuti kako će domaću zadaću od doma rješavati, a 36.8 % učenika kaže da i je u potpunosti jednostavnije po pitanju instalacije i da se ne moraju uopće brinuti po pitanju rješavanja domaće zadaće od doma. Aritmetička sredina iznosi 3.96, a standardna devijacija 1.03.



Poslano je ovoliko odgovora: 57

## Lakše mi je s prijateljima iz razreda skupa rješavati problemske zadatke iz programiranja.

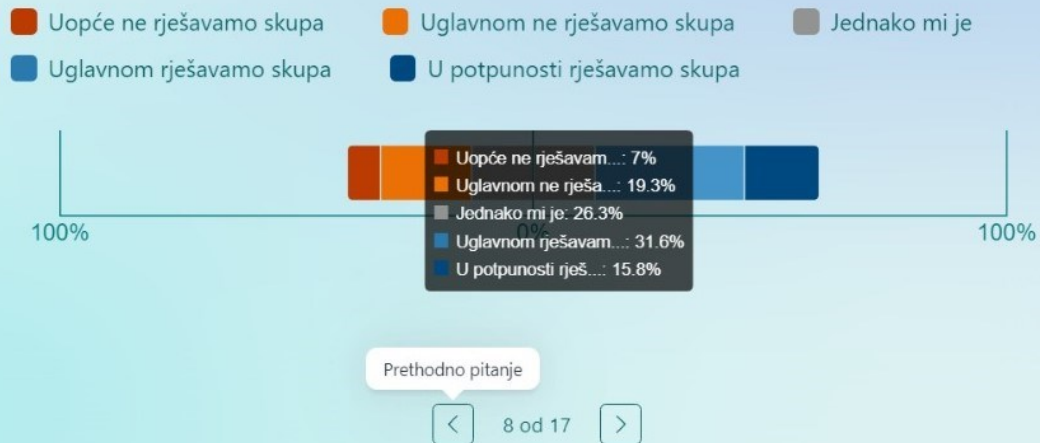


**Grafikon 7:** Lakše mi je s prijateljima iz razreda skupa rješavati problemske zadatke iz programiranja

Na grafikonu 7 prikazani su odgovori učenika na pitanje koliko im je lakše rješavati skupa problemske zadatke iz programiranja pomoću Replita. Odgovori učenika su slijedeći: 5.3 % učenika uopće ne vidi prednost u zajedničkom radu, 5.3 % ih se izjasnilo da im uglavnom nije lakše rješavati problemske zadatke iz programiranja skupa, 12.3 % ih je nepromijenjenog mišljenja. 22.8 % misli da im je uglavnom lakše rješavati zadatke skupa, a 54.4 % se slaže da im je u potpunosti lakše zadatke rješavati skupa. Aritmetička sredina iznosi 4.16, a standardna devijacija 1.73.

Poslano je ovoliko odgovora: 57

## Često skupa rješavamo domaću zadaću.



### Grafikon 8: Često skupa rješavamo domaću zadaću

Na grafikonu 8 prikazani su odgovori učenika na pitanje koliko često skupa rješavaju zadaću. Odgovori učenika su slijedeći: 7 % učenika kaže da uopće ne rješavaju domaću zadaću skupa. 19.3 % ih je mišljenja da uglavnom ne rješava zadaću skupa. 26.3 % ih je nepromijenjenog mišljenja po pitanju zajedničkog rada u rješavanju domaće zadaće. 31.6 % učenika je mišljenja da uglavnom rješava domaći rad uglavnom skupa, dok je 15.8 % učenika mišljenja da u potpunosti podržava skupno rješavanje domaće zadaće. Aritmetička sredina iznosi 3.29, a standardna devijacija 1.15.

Poslano je ovoliko odgovora: 57

## Mislim da bolje napredujem u programiranju od kada koristim Replit.



Prethodno pitanje

< 9 od 17 >

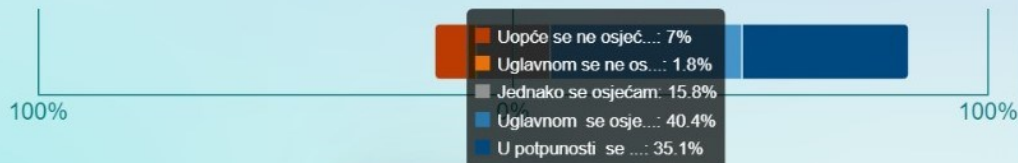
### **Grafikon 9:** Mislim da bolje napredujem u programiranju od kada koristim Replit

Na grafikonu 9 su prikazani odgovori učenika na pitanje o učeničkoj samoprocjeni napretka u učenju programiranja s pomoću Replit alata u odnosu na oblik učenja od prije s pomoću lokalno instaliranog prevoditelja IDLE. Odgovori učenika su sljedeći:

5.3 % učenika je negativnog mišljenja da uopće ne napreduju bolje od kad rade u Replitu. 22.8 % učenika je nepromijenjenog mišljenja da im je napredak jednak u odnosu na prije. Vrlo velika većina učenika ima promijenjeno mišljenje o napretku učenja na bolje i to njih 57.9 %, kao i 14 % učenika koji su mišljenja da u potpunosti napreduju bolje od kada koriste Replit u učenju programiranja. Aritmetička sredina iznosi 3.75, a standardna devijacija 1.09.

## Osjećam se modernije i više informatički obrazovan/obrazovana od kada koristim Replit.

- Uopće se ne osjećam modernije i više informatički obrazovan/na
- Uglavnom se ne osjećam modernije i više informatički obrazovan/na
- Jednako se osjećam
- Uglavnom se osjećam modernije i više informatički obrazovan/na
- U potpunosti se osjećam modernije i više informatički obrazovan/na



Prethodno pitanje

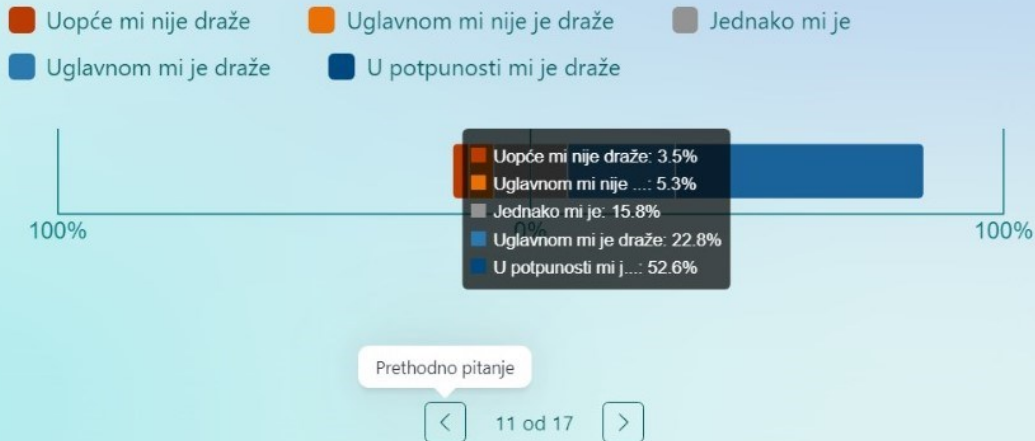
< 10 od 17 >

### **Grafikon 10:** Osjećam se modernije i više informatički obrazovan/obrazovanja od kada koristim Replit

Na grafikonu 10 prikazani su odgovori učenika na pitanje koliko se osjećaju modernije i više informatički obrazovanije od kada koriste Replit. Odgovori učenika su sljedeći: 7 % učenika nema nikakvih promjena i mišljenja su da se uopće ne osjećaju bolje i informatički obrazovanja od kada koriste Replit. 1.8 % ih je tako mišljenja da se uglavnom ne osjećaju bolje i više informatički obrazovanja oda kada koriste Replit. 15.8 % ih je nepromijenjenog mišljenja. Vrlo velika većina učenika se uglavnom osjeća bolje i informatički obrazovanja od kada koriste Replit, a učenici koji se u potpunosti osjećaju bolje i informatički obrazovaniji od kada koriste Replit je njih 35.1 %. Aritmetička sredina iznosi 3.95, a standardna devijacija 1.10.

Poslano je ovoliko odgovora: 57

Drago mi je raditi u Replitu jer uvijek mogu postaviti pitanje prijateljima iz razreda i učitelju ako imam problem u rješavanju pro-...

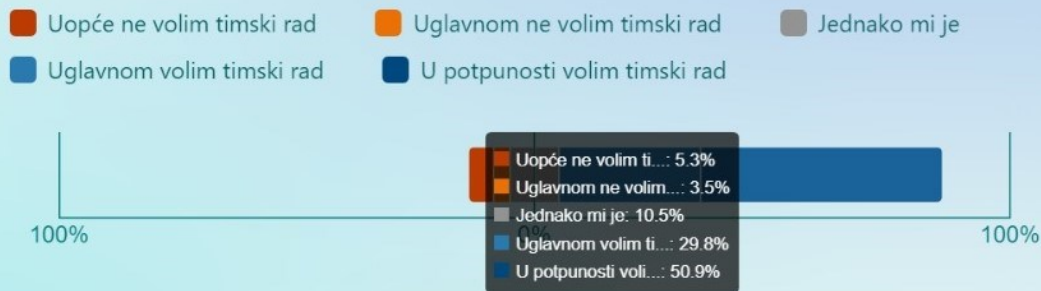


**Grafikon 11:** Drago mi je raditi u Replitu jer uvijek mogu postaviti pitanje prijateljima iz razreda i učitelju ako imam problem u rješavanju problemskih zadataka

Na grafikonu 11 prikazani su odgovori učenika na pitanje koliko im pomoć kolega i prijatelja iz razreda pomaže u učenju programiranja. Odgovori učenika su sljedeći: 3.5 % učenika je nepromijenjenog mišljenja da im uopće nije draže što imaju podršku, kao i 5.3 % učenika da im uglavnom nije draže što imaju podršku tijekom učenja. 15.8 % učenika imaju nepromijenjeno mišljenje u smislu podrške u učenju i stava su da im je svejedno. Učenika s promijenjenim mišljenjem i koji su stava da im je uglavnom lakše što imaju podršku tijekom učenja je 22.8 %. Učenika koji je bitna podrška i koji pridaju velik značaj da im je u potpunosti zadatke iz programiranja rješavati na takav način je njih 52.6 %. Aritmetička sredina iznosi 4.16, a standardna devijacija 1.09.

Poslano je ovoliko odgovora: 57

## Volim timski rad i kada možemo svi skupa raditi na programskom rješenju.



Prethodno pitanje



12 od 17

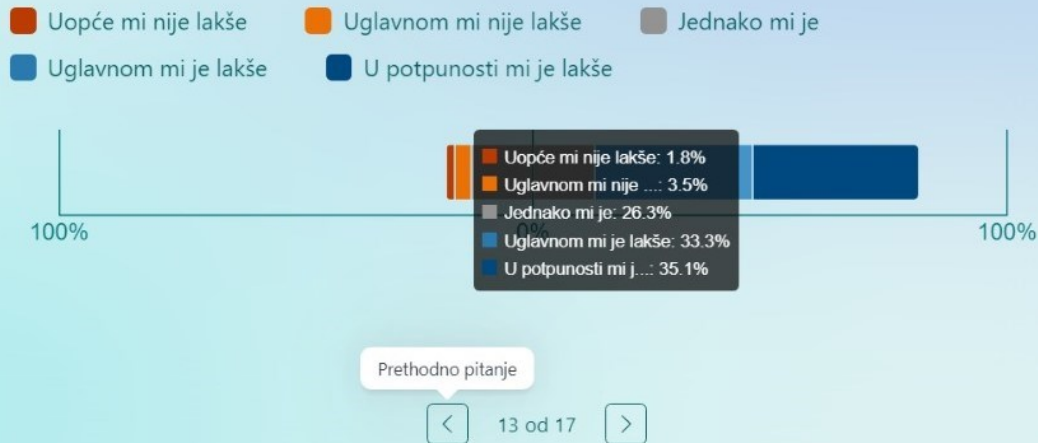


### **Grafikon 12:** Volim timski rad i kada možemo svi skupa raditi na programskom rješenju

Na grafikonu 12 prikazani su odgovori učenika na pitanje koliko vole timski rad i kada mogu svi skupa raditi na programskom rješenju. Odgovori učenika su sljedeći: 5.3 % učenika ima nepromijenjeno mišljenje u pogledu poticanja motivacije u učenju kroz timski oblik rada. 3.5 % učenika uglavnom ne voli timski rad također ima nepromijenjeno mišljenje. 10.5 % učenika također ima nepromijenjeno mišljenje po pitanju timskog rada u podizanju motivacije u učenju programiranja. 29.8 % učenika kaže da uglavnom voli timski rad i u zajedničkom rješavanju problema, dok vrlo velika većina učenika njih 50.9 % voli u potpunosti timski rad kada skupa mogu sudjelovati u rješavanju istog problema. Aritmetička sredina iznosi 4.18, a standardna devijacija 1.09.

Poslano je ovoliko odgovora: 57

### Uz Replit mi je olakšan dolazak do vlastitog rješenja na temelju tuđih idejnih rješenja.



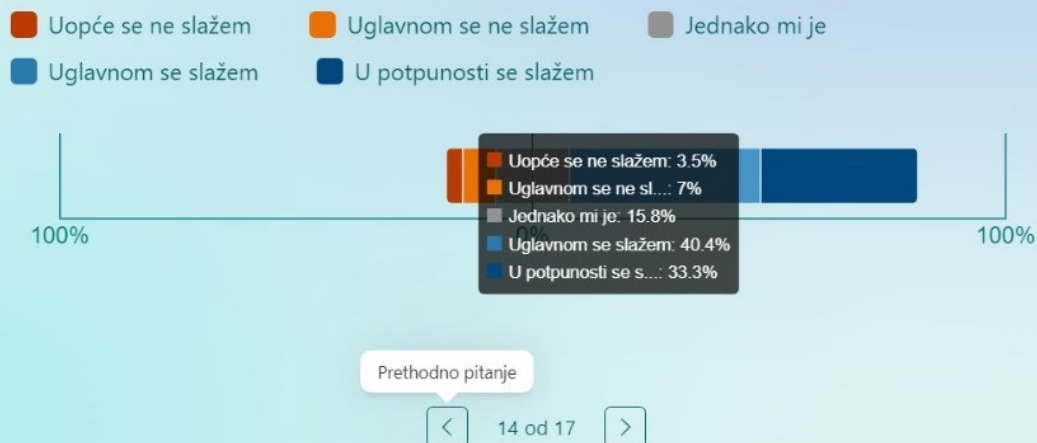
**Grafikon 13:** Uz Replit mi je olakšan dolazak do vlastitog rješenja na temelju tuđih idejnih rješenja

Na grafikonu 13 prikazani su odgovori učenika na pitanje koliko im je pronalazak vlastitog rješenja olakšan na temelju tuđih idejnih rješenja. Odgovori učenika su sljedeći: 1.8 % učenika je nepromijenjenog mišljenja da im uopće nije lakše doći do vlastitog rješenja na temelju tuđeg, 3.5 % učenika je mišljenja da im uglavnom nije lakše doći do vlastitog rješenja na temelju tuđeg, 26.3 % učenika je nepromijenjenog mišljenja da im je jednako po pitanju pronalaska svog rješenja na temelju tuđeg. 33.3 % učenika je mišljenja da im je uglavnom lakše doći do rješenja kada imaju uvid u tuđe rješenja pa mogu modifikacijom i prilagodbom doći do vlastitog. 35.1 % učenika se slaže da im je u potpunosti lakše doći do vlastitog rješenja kada imaju mogućnost modifikacije i prilagodbe na temelju tuđeg rješenja. Aritmetička sredina iznosi 3.96, a standardna devijacija 0.88.



Poslano je ovoliko odgovora: 57

## Kada bih birao raditi kod kuće i u školi u Replitu moj izbor bih bio Replit.

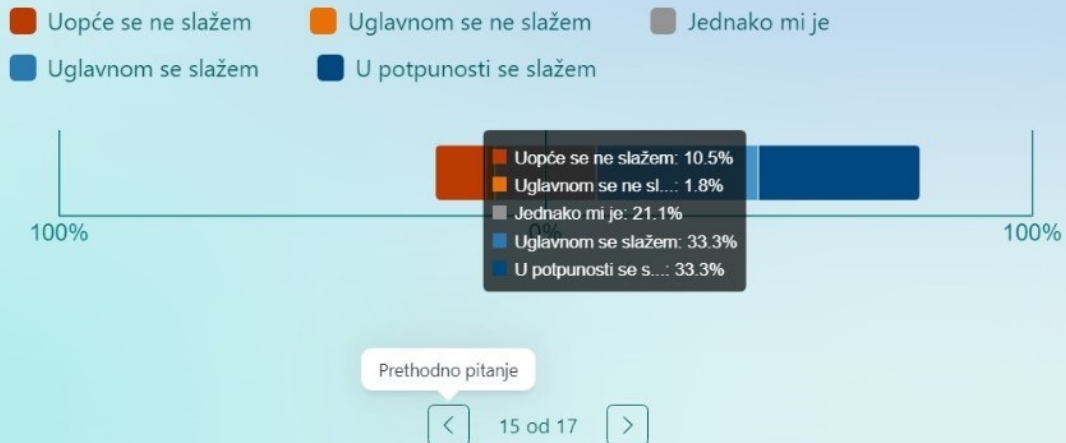


### **Grafikon 14:** Kad bih birao raditi kod kuće i u školi u Replitu moj izbor bih bio Replit

Na grafikonu 14 prikazani su odgovori učenika na pitanje kad bih radio kod kuće i u školi jeli bih njihov izbor bio Replit. Odgovori učenika su sljedeći: 3.5 % učenika kaže da njihov izbor ne bih bio Replit, kao i 7 % učenika koji kažu da se uglavnom ne slažu da bih njihov izbor bio Replit. 15.8 % učenika je nepromijenjenog mišljenja i kažu da im je jednako po pitanju izbora rada i učenja programiranja u alatu Replit. 40.4 % učenika bih uglavnom radilo u Replitu u školi i kod kuće, a 33.3 % učenika se u potpunosti slaže da bih radila kod kuće i u školi u Replitu. Aritmetička sredina iznosi 3.93, a standardna devijacija 1.05.



## Kad bih bio programer jednog dana sigurno bih radio u sličnom alatu za suradnju sličnom Replitu.

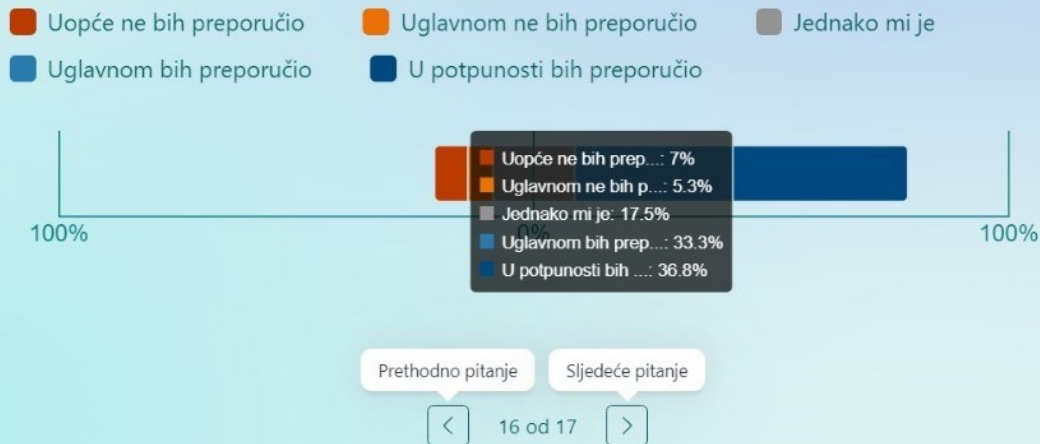


**Grafikon 15:** Kad bih bio programer jednog dana sigurno bih radio u sličnom alatu za suradnju sličnom Replitu

NA grafikonu 15 prikazani su odgovori učenika na pitanje kada bih jednog dana postali programeri bih li radili u sličnom alatu za suradnju sličnom Replit. Odgovori učenika su sljedeći: 10.5 % učenika se uopće ne slaže da bih Replit bio njihov izbor alata za rad u izabranom zanimanju, a 1.8 % učenika se ne uglavnom ne slaže da bih Replit bio njihov izbor za rad u budućem zvanju. Učenika koji imaju nepromijenjeno mišljenje po pitanju izbora alata za rad u budućem zvanju je njih 21.1 %. Učenici koji se uglavnom slažu da bih Replit bio izbor za rad u budućem zvanju je njih 33.3 %, a učenika koji se u potpunosti slažu da bih Replit bio izbor u budućem zvanju ih je 33.33 %. Aritmetička sredina iznosi 3.68, a standardna devijacija 1.24.

Poslano je ovoliko odgovora: 57

## Preporučio bih i prijateljima učenicima koji ne idu s menom u školu ili razred da koriste Replit.

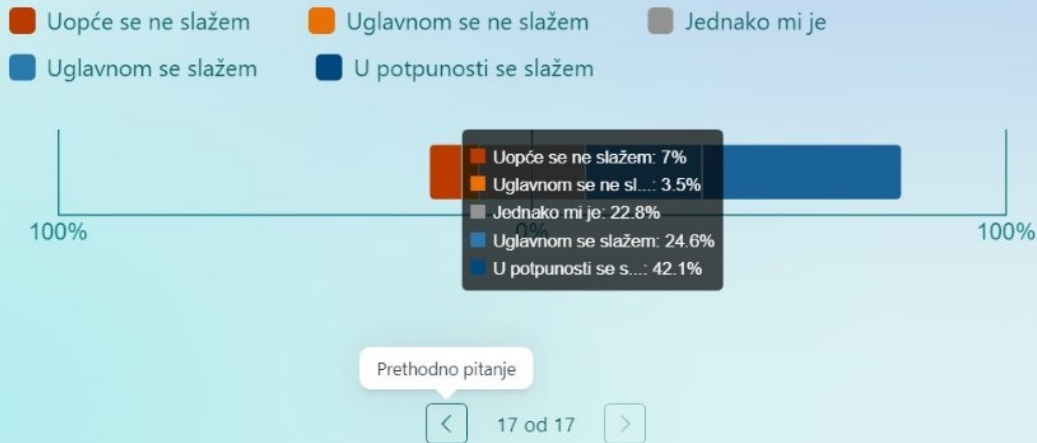


**Grafikon 16:** Preporučio bih i prijateljima učenicima koji ne idu s mnom u školu ili razred da koriste Replit

Na grafikonu 16 prikazani su odgovori učenika na pitanje o preporuci prijateljima da koriste Replit za učenje programiranja. Odgovori učenika su sljedeći: 7 % učenika je nepromijenjenog mišljenja i uopće ne bih svojim vršnjacima preporučilo korištenje Replita u obrazovne svrhe, učenika koji su mišljenja da uglavnom ne bih preporučili svojim vršnjacima korištenje Replita u obrazovne svrhe ih je 5.3 %. 17.5 % učenika je nepromijenjenog mišljenja i nemaju stav i preporuci vršnjacima za korištenje istoga u obrazovne svrhe. Učenika koji imaju promijenjeno mišljenje i koji bih uglavnom svojim vršnjacima preporučili korištenje Replita u obrazovne svrhe ih je 33.3 %, a učenika koji također imaju promijenjeno mišljenje i u potpunosti bi preporučili korištenje Replita svojim vršnjacima ih je 36.8 %. Aritmetička sredina iznosi 3.89, a standardna devijacija 1.18.

Poslano je ovoliko odgovora: 57

## Lakše mi je izvršavati svoje školske obveze u Replitu.



**Grafikon 17:** Lakše mi je izvršavati svoje školske obveze u Replitu

Na grafikonu 17 prikazani su odgovori učenika na pitanje koliko im je lakše izvršavati svoje školske obveze u Replitu. Odgovori učenika su sljedeći: 7 % učenika ima nepromijenjeno mišljenje u smislu ispunjava li Replit sva njihova očekivanja u smislu izvršavanja školskih obveza. 3.5 % učenika je mišljenja da ih Replit ne zadovoljava za izvršenje školskih obveza. 22.8 % učenika nema mišljenje dali ih Replit zadovoljava po pitanju izvršavanja školskih obveza. Dobra većina je mišljenja i uglavnom se slažu da ih Replit zadovoljava po pitanju izvršavanja školskih obveza, a 42.1 % učenika se u potpunosti slaže da ih Replit zadovoljava u izvršavanju školskih obveza. Aritmetička sredina iznosi 3.91, a standardna devijacija 1.20.

### 5.3. Diskusija rezultata istraživanja

Rezultati ovog istraživanja pokazuju da postoji značajna razlika u obliku izvođenja nastave učenja programiranja što se tiče odabira lokalno instaliranog prevoditelja IDLE na školskim računalima naspram sustava za učenje programiranja Replit u korist sustava za učenje programiranja Replit. Iako u ovoj populaciji djece osnovnoškolske dobi 5. i 6. razreda osnovne škole ima relativno malo istraživanja, postoje značajna uvjerenja učenika koja mogu ukazivati na niz prednosti odabira ovakvih alata u nastavi učenja programiranja. Većina učenika ima promijenjeno mišljenje njih 59.7 % te je stava da im je rad u Replitu jednostavniji u odnosu na prije kada su programirali u lokalno instaliranom prevoditelju IDLE. Velika većina učenika njih 68.4 % izjavljuje da im je rad u Replitu uglavnom i u potpunosti lakši u odnosu na prije s pomoću lokalno instaliranog prevoditelja IDLE. Većina učenika njih 64.9 % je mišljenja da im rad u Replitu smanjuje kognitivno opterećenje praćenja nastave u školi. Što se tiče motivacijskih aspekata učenika većina učenika izjavljuje njih 71.9 % da im je nastava u Replitu zanimljivija u odnosu na prije s pomoću lokalno instaliranog prevoditelja IDLE. Velika većina učenika njih 63.2 % izjavljuje da im je pronalazak i rad s datotekama olakšan u odnosu na lokalno instaliranog prevoditelja IDLE. Većina učenika njih 70.1 % je stava da im je olakšan rad po pitanju instalacije i rješavanja domaće zadaće od doma. Velika većina učenika njih 77.3 % izjavljuje da im je lakše s prijateljima iz razreda skupa rješavati problemske zadatke iz programiranja s pomoću kolaborativnog alata Replit, što nisu mogli do sad s pomoću lokalno instaliranog prevoditelja IDLE na školskim računalima. Manje od polovice ispitanih učenika 47.4 % je mišljenja da im kolaborativni alat Replit pomaže u suradničkom rješavanju domaće zadaće od kuće. Velika većina učenika 71.9 % je stava da bolje napreduju u učenju programiranja od kada koriste Replit. Velika većina učenika njih 75.5 % je stava da se osjećaju modernije i više informatički obrazovani od kada koriste Replit. Većina učenika 75.4 % je stava da im je od velikog značaja podrška koju mogu dobiti od učitelja i prijatelja iz razreda tijekom rješavanja problemskih zadataka. Većina učenika 50.9 % timski rad shvaća kao benefit u radu. Velika većina učenika njih 68.4 % je stava da im uvid u tuđe programsko rješenje može olakšati put pronalaska vlastitog rješenja tijekom učenja programiranja. Što se tiče mišljenja o izboru alata za rad u nastavi programiranja velika većina učenika njih 73.7 % je mišljenja da bih Replit bio njihov izbor u učenju

programiranja. U razmatranju stavova i mišljenja učenika na pitanje o izboru alata za programiranje kada bih i sami bili programeri odgovor većine učenika njih 66.6 % je bio Replit. Što se tiče preporuke i mišljenja učenika o preporuci vršnjacima 70.1 % ima pozitivno mišljenje o preporuci korištenja Replita u obrazovne svrhe svojim vršnjacima.

U prvoj skupini pitanja sastavljenih na temelju izvora polu strukturiranih razgovora s učenicima može se ustanoviti da je rad u Replitu učenicima jednostavniji, da im je lakše raditi u Replitu, da im je lakše pratiti nastavu u Replitu, kada bih birali da bih izabrali rad u Replitu kod kuće i u školi, te da im je lakše izvršavati svoje školske obveze u Replitu.

U drugoj skupini pitanja sastavljene na temelju motivacijskih aspekata istraženih u literaturi može se ustanoviti da u ovisnosti o izabranom sustavu za učenje programiranja Replit učenicima je nastava zanimljivija, mišljenja su da bolje napreduju u programiranju od kada koriste Replit, da se osjećaju modernije i obrazovanije, te da bih prijateljima također preporučili da koriste Replit.

U trećoj skupini pitanja sastavljenih na temelju vlastite nastavničke prakse može se ustanoviti da je učenicima lakši rad s datotekama u Replitu, da nemaju problema s instalacijom, te da se ne moraju brinuti kako će rješavati domaću zadaću.

U četvrtoj skupini pitanja sastavljenih na temelju literature i prednosti sustava za učenje programiranja koji podržavaju suradnički rad može se ustanoviti da je učenicima lakše s prijateljima skupa rješavati problemske zadatke iz programiranja, da im je lakše kada mogu postaviti pitanje prijateljima ili učitelju u rješavanju problemskih zadataka, da vole timski rad i kada mogu svi skupa raditi na programskom rješenju, te da kada bih jednog dana bili programeri da bih radili u sličnom alatu za suradnju Replit. Što se tiče odgovora učenika: „Često skupa rješavamo domaću zadaću“, stav učenika na Likertovoj skali je bio u rasponu od 1-3. Vjerojatno se radi o generalnom stavu o nerješavanju domaće zadaće što je bilo za očekivati.

## 6. ZAKLJUČAK

U današnjem dobu informatičke ere podučavanje i učenje programiranja ima dvostruki izazov. Za učitelja i učenika. Učitelj je taj koji odabire alat za rad u nastavi, a učenik koji bih trebao imati benefit u učiteljevom odabiru. Kad su uvjerenja nastavnika koji podučava programiranje, ista kao i uvjerenja učenika, tada možemo tvrditi da je odabran odgovarajući oblik za rad u nastavi programiranja, te ima svoju pedagošku opravdanost za korištenje iste tehnologije za rad u nastavi Informatike. Namjera ovog rada nije bila promicanje sustava za učenje programiranja Replit u nastavi programiranja, već paradigme oblika učenja i poučavanja programiranja u nastavi Informatike s pomoću sustava za učenje programiranja Replit naspram lokalno instaliranog prevoditelja. Ovakav način rada u prošlosti je bio nezamisliv, ne u smislu nepostojanja računala, već u smislu brzine prijenosa informacija mrežama. Danas je to postao standard, a lokalno instalirana bilo koja aplikacija stvar prošlosti. U današnje vrijeme se cijeli život preselio u *Oblak*, pa se tako i način učenja i poučavanja programiranja učenika osnovnih škola preselio u *Oblak*. Djeca u tome rado sudjeluju, imaju pozitivne stavove, mišljenja su da je to moderno, te je postalo opće prihvaćeno. Domaći rad danas više nije u bilježnici on je „tamo“, rješenja domaće zadaće su također „tamo“, komentari i pitanja su također „tamo“, sve je „tamo“ u *Oblaku*. Informatička era je uzela svoj danak, pa možemo kazati, „ako i kad“ se tehnologija koristi u pozitivne obrazovne svrhe, ima itekako svoju pedagošku opravdanost u korištenju iste. Ovaj oblik poučavanja djeluje motivirajuće i ima značajan utjecaj na učenje programiranja.

Stavovi učenika u pogledu učenja programiranja na ovakav način učenja su vrlo konzistentna te potvrđuju prvo postavljeno istraživačko pitanje: „Učenje programiranja u aplikaciji Replit stvara osjećaj ugone i podrške u učenju programiranja, te podiže motivaciju u učenika“. Također izbor tehnologije za podučavanje programiranja potvrđuje drugo postavljeno istraživačko pitanje: „Učiteljev odabir nastavnog alata za podučavanje programiranja ima značajan utjecaj na učenje programiranja“. Stavovi učenika o opravdanosti korištenja predmetne tehnologije su vrlo konzistentni i visoko natpolovični što upućuje u opravdanost i treće postavljenog istraživačkog pitanja:

„Učiteljev odabir za rad u školi u aplikaciji Replit ima za rezultat pedagošku opravdanost za korištenje iste aplikacije“.

Stoga rad i učenje s pomoću *cloud* tehnologije sustava za učenje programiranja Replit je bolji u odnosu na lokalno instaliranog prevoditelja na računalu.

## **Popis grafikona**

Grafikon 1. Koliko vam je rad u Replitu jednostavniji u odnosu na prije s pomoću IDLE programskog rješenja?

Grafikon 2. U školi mi je lakše raditi u Replitu.

Grafikon 3. Lakše mi je pratiti nastavu u Replitu.

Grafikon 4. Nastava u Replitu mi je zanimljivija.

Grafikon 5. Lakše mi je doći do datoteke sve su na jednom mjestu.

Grafikon 6. Nemam problema s instalacijom i ne moram se brinuti odakle ću rješavati domaću zadaću.

Grafikon 7. Lakše mi je s prijateljima iz razreda skupa rješavati problemske zadatke iz programiranja.

Grafikon 8. Često skupa rješavamo domaću zadaću.

Grafikon 9. Mislim da bolje napredujem u programiranju od kada koristim Replit.

Grafikon 10. Osjećam se modernije i više informatički obrazovan/obrazovanija od kada koristim Replit.

Grafikon 11. Drago mi je raditi u Replitu jer uvijek mogu postaviti pitanje prijateljima iz razreda i učitelju ako imam problem u rješavanju programskih zadataka.

Grafikon 12. Volim timski rad i kada možemo svi skupa raditi na programskom rješenju.

Grafikon 13. Kada bih birao raditi kod kuće i u školi u Replitu moj izbor bih bio Replit.

Grafikon 14. Kada bih bio programer jednog dana sigurno bih radio u sličnom alatu za suradnju sličnom Replitu.

Grafikon 15. Preporučio bih i prijateljima učenicima koji ne idu s mnom u školu ili razred da koriste Replit.

Grafikon 16. Lakše mi je izvršavati svoje školske obveze u Replitu.



## **Popis slika**

Slika 1. Povezanost domena, izvor: Kurikulum nastavnog predmeta Informatike, izvor: <https://mzo.gov.hr/UserDocsImages/dokumenti/Publikacije/Predmetni/%20nastavnog%20predmeta%20Informatika%20za%20osnovne%20skole%20i%20gimnazije.pdf>, datum pristupa: 05.02.2024.

Slika 2. Didaktički peterokut (Izvor: Radonić, 1997)

Slika 3. Sustav za učenje programiranja Replit.com (Izvor: <https://replit.com/site/teams-for-education> , datum pristupa: 14.05.2023.)

## **Popis tablica**

Tablica 1. Model CT-a prema (Shute i dr., 2017.)

Tablica 2. Model CT-a prema (Resnick i dr., 2007.)

Tablica 3. Mogući odgovori učenika s obzirom na tvrdnje (Izvor: autor rada)

Tablica 4. Mišljenje učenika vezanih uz odgovore na odabrane tvrdnje u anketnom upitniku o primjeni sustava Replit (Izvor: obrada autora)

## **SKRAĆENICE**

CT – eng. computational thinking, hrv. računalno razmišljanje,

VPL – vizualni programski jezik (Scratch, Alice, i dr...),

CPL – tekstualni programski jezik (IDLE, replit.com),

IKT – informacijsko komunikacijska tehnologija,

IDE – (eng. interface desktop environment, hrv. integrirano razvojno okruženje) je aplikacija je koja olakšava razvoj aplikacija. Općenito, IDE je radno mjesto zasnovano na grafičkom korisničkom sučelju,

VM – eng. virtual machine, hrv. virtualni stroj,

CLUI - eng. Command Line User Interface, hrv. korisničko sučelje naredbenog retka,

SSH - (eng. Secure Shell) je kriptografski mrežni protokol koji omogućuje sigurnu komunikaciju i upravljanje udaljenim računalima putem nesigurne mreže, kao što je Internet,

GPT-4 – (multimodalni model umjetne inteligencije, koji se nadograđuje na isključivo tekstualno razumijevanje prethodnih modela),

API – (eng. Application Programming Interface), skup definiranih pravila, protokola i alata koji omogućuju različitim softverskim aplikacijama komuniciranje i međusobnu integraciju,

VNC – (eng. Virtual Network Computing, hrv. virtualno mrežno računarstvo), tehnologija koja omogućava korisnicima pristup i kontrolu udaljenih računara preko mreže, kao da fizički sjede ispred tih računara. VNC omogućava udaljenom korisniku da vidi ekran udaljenog računara, upravlja mišem i tastaturom, te koristi aplikacije i datoteke na udaljenom računalu,

CSV - (eng. comma-separated values, hrv. zarezom odvojene vrijednosti),

VCS - (eng. Veritas Cluster Server),

AAI@EduHr - (Autentikacijska i autorizacijska infrastruktura znanosti i visokog obrazovanja u Republici Hrvatsko).

## Prilozi

Korištena pitanja na osnovu kojih je izrađen upitnik i gdje je provedena anketa među učenicima nakon aktivnosti koje su imale za cilj stjecanje učeničkih stavova korištenjem sustava Replit za učenje programiranja u domeni računalnog razmišljanja i programiranja su sljedeća:

1. Koliko vam je rad u Replitu jednostavniji u odnosu na prije s pomoću IDLE programskog rješenja? (Uopće mi nije jednostavniji, uglavnom mi nije jednostavniji, jednako mi je, uglavnom mi je jednostavniji, u potpunosti mi je jednostavniji)
2. U školi mi je lakše raditi u Replitu. (Uopće mi nije jednostavnije, uglavnom mi nije jednostavnije, jednako mi je, uglavnom mi je jednostavnije, u potpunosti mi je jednostavnije)
3. Lakše mi je pratiti nastavu u Replitu. (Uopće mi nije jednostavnije, uglavnom mi nije jednostavnije, jednako mi je, uglavnom mi je jednostavnije, u potpunosti mi je jednostavnije)
4. Nastava u Replitu mi je zanimljivija. (Uopće mi nije zanimljivija, uglavnom mi nije zanimljivija, jednako mi je, uglavnom mi je zanimljivija, u potpunosti mi je zanimljivija)
5. Lakše mi je doći do datoteke sve su na jednom mjestu. (Uopće mi nije lakše, uglavnom mi nije lakše, jednako mi je, uglavnom mi je lakše, u potpunosti mi je lakše)
6. Nemam problema s instalacijom i ne moram se brinuti odakle ću rješavati domaću zadaću. (Uopće mi nije jednostavnije, uglavnom mi nije jednostavnije, jednako mi je, uglavnom mi je jednostavnije, u potpunosti mi je jednostavnije)
7. Lakše mi je s prijateljima iz razreda skupa rješavati problemske zadatke iz programiranja. (Uopće mi nije lakše, uglavnom mi nije lakše, jednako mi je, uglavnom mi je lakše, u potpunosti mi je lakše)
8. Često skupa rješavamo domaću zadaću. (Uopće ne rješavamo skupa, uglavnom ne rješavamo skupa, jednako mi je, uglavnom rješavamo skupa, u potpunosti rješavamo skupa)

9. Mislim da bolje napredujem u programiranju od kada koristim Replit. (Uopće ne napredujem, uglavnom ne napredujem, jednako mi je, uglavnom napredujem, u potpunosti napredujem)
10. Osjećam se modernije i više informatički obrazovan/obrazovana od kada koristim Replit. (Uopće se ne slažem, uglavnom se ne slažem, jednako mi je, uglavnom se slažem, u potpunosti se slažem)
11. Drago mi je raditi u Replitu jer uvijek mogu postaviti pitanje prijateljima iz razreda i učitelju ako imam problem u rješavanju programskih zadataka. (Uopće se ne slažem, uglavnom se ne slažem, jednako mi je, uglavnom se slažem, u potpunosti se slažem)
12. Volim timski rad i kada možemo svi skupa raditi na programskom rješenju. (Uopće se ne slažem, uglavnom se ne slažem, jednako mi je, uglavnom se slažem, u potpunosti se slažem)
13. Uz Replit mi je olakšan dolazak do vlastitog rješenja na temelju tuđih idejnih rješenja. (Uopće mi nije lakše, uglavnom mi nije lakše, jednako mi je, uglavnom mi je lakše, u potpunosti mi je lakše)
14. Kada bih birao raditi kod kuće i u školi u Replitu moj izbor bih bio Replit. (Uopće se ne slažem, uglavnom se ne slažem, jednako mi je, uglavnom se slažem, u potpunosti se slažem)
15. Kad bih bio programer jednog dana sigurno bih radio u sličnom alatu za suradnju sličnom Replitu. (Uopće se ne slažem, uglavnom se ne slažem, jednako mi je, uglavnom se slažem, u potpunosti se slažem)
16. Preporučio bih i prijateljima učenicima koji ne idu sa mnom u školu ili razred da koriste Replit. (Uopće se ne slažem, uglavnom se ne slažem, jednako mi je, uglavnom se slažem, u potpunosti se slažem)
17. Lakše mi je izvršavati svoje školske obveze u Replitu. (Uopće se ne slažem, uglavnom se ne slažem, jednako mi je, uglavnom se slažem, u potpunosti se slažem)

## LITERATURA

1. Tomljenović, K. (2018). Računalno razmišljanje i uloga učenja pomoću igre na njegov razvoj. Učiteljski fakultet.  
[https://www.inf.uniri.hr/images/studiji/poslijediplomski/kvalifikacijski/Kvalifikacijski\\_rad\\_Kreso\\_Tomljenovic.pdf](https://www.inf.uniri.hr/images/studiji/poslijediplomski/kvalifikacijski/Kvalifikacijski_rad_Kreso_Tomljenovic.pdf), datum pristupa: 28.01.2024.
2. Selby, Cynthia and Woollard, John (2013) Computational thinking: the developing definition, University of Southampton (E-prints) 6pp., datum pristupa: 29.01.2024.
3. Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE), Computational thinking for all, 2011, [.https://www.iste.org/explore/articleDetail?articleid=152](https://www.iste.org/explore/articleDetail?articleid=152), datum pristupa: 29.01.2024.
4. Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE), Computational thinking for all, 2011, [.https://www.iste.org/explore/articleDetail?articleid=152](https://www.iste.org/explore/articleDetail?articleid=152), datum pristupa: 30.01.2024.
5. Shute, Valerie & Sun, Chen & Asbell-Clarke, Jodi. (2017). Demystifying computational thinking. Educational Research Review., datum pristupa: 30.01.2024.
6. Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60-67, datum pristupa: 30.01.2024.
7. Shute, Valerie & Sun, Chen & Asbell-Clarke, Jodi. (2017), Demystifying computational thinking. Educational Research Review, datum pristupa: 31.01.2024.
8. [https://narodne-novine.nn.hr/clanci/sluzbeni/2018\\_03\\_22\\_436.html](https://narodne-novine.nn.hr/clanci/sluzbeni/2018_03_22_436.html), datum pristupa: 04.02.2024.
9. [https://narodne-novine.nn.hr/clanci/sluzbeni/2018\\_03\\_22\\_436.html](https://narodne-novine.nn.hr/clanci/sluzbeni/2018_03_22_436.html), datum pristupa: 28.11.2023.

10. [https://www.onlinegdb.com/online\\_python\\_compiler](https://www.onlinegdb.com/online_python_compiler), datum pristupa: 23.02.2024.
11. <https://www.jetbrains.com/pycharm/promo/>, datum pristupa: 23.02.2024.
12. <https://scratch.mit.edu/about>, datum pristupa: 23.02.2024.
13. <https://www.scratchfoundation.org/>, datum pristupa: 23.02.2024.
14. <https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf>, datum pristupa: 23.02.2024.
15. <https://www.e-sfera.hr/prelistaj-udzbenik/46eb0ba9-475b-4d5f-ab77-a264ae54f6a7>, datum pristupa: 23.02.2024.
16. <https://scratched.gse.harvard.edu/guide/>, datum pristupa: 23.02.2024.
17. <https://www.facebook.com/groups/TeachingwithScratch/>, datum pristupa: 23.02.2024.
18. <https://www.meetup.com/pro/scratched/>, datum pristupa: 23.02.2024.
19. <https://csfirst.withgoogle.com/s/en/home>, datum pristupa: 23.02.2024.
20. <https://projects.raspberrypi.org/en/pathways/scratch-intro>, datum pristupa: 23.02.2024.
21. <https://resources.scratch.mit.edu/www/guides/en/scratch-teacher-accounts-guide.pdf>, datum pristupa: 23.02.2024.
22. <https://scratch.mit.edu/educators/faq>, datum pristupa: 23.02.2024.
23. [https://www.researchgate.net/profile/Ali-Ebrahimi-42/publication/269294042\\_Programming\\_For\\_Children\\_Alice\\_and\\_Scratch\\_Analysis/links/5486a7180cf268d28f0457ee/Programming-For-Children-Alice-and-Scratch-Analysis.pdf](https://www.researchgate.net/profile/Ali-Ebrahimi-42/publication/269294042_Programming_For_Children_Alice_and_Scratch_Analysis/links/5486a7180cf268d28f0457ee/Programming-For-Children-Alice-and-Scratch-Analysis.pdf), datum pristupa: 22.02.2024.
24. <https://scratch.mit.edu/>, datum pristupa: 22.02.2024.
25. <https://iopscience.iop.org/article/10.1088/1742-6596/1840/1/012029/pdf>, datum pristupa: 02.12.2023.
26. <https://docs.replit.com/>, datum pristupa: 08.02.2024.
27. [https://mako.cc/writing/to\\_fork\\_or\\_not\\_to\\_fork.html](https://mako.cc/writing/to_fork_or_not_to_fork.html), datum pristupa: 11.02.2024.
28. <https://replit.com/desktop>, datum pristupa: 16.02.2024.
29. <https://firewalledreplit.com/>, datum pristupa: 16.02.2024.

30. [https://www.carnet.hr/wp-content/uploads/2021/05/Google\\_Ucionica\\_osvrt\\_i\\_upute.pdf](https://www.carnet.hr/wp-content/uploads/2021/05/Google_Ucionica_osvrt_i_upute.pdf), datum pristupa: 22.02.2024.
31. Sentance, S., & Csizmadia, A. (2015). Perspektive nastavnika o uspješnim strategijama poučavanja računarstva u školi. In A. Brodnik & Lewin, Cathy (Eds.), Međunarodna federacija za obradu informacija, IFIP TCS 2015., datum pristupa: 24.11.2023., izvor:
32. Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Programiranje parova: Pod kojim je uvjetima povoljno za učenike srednjih škola? Časopis za istraživanje tehnologije u obrazovanju, 46 (3), 277–296., datum pristupa: 24.11.2023., izvor:
33. Bjursten, EL., Nilsson, T. & Gumaelius, L. Računalno programiranje u osnovnim školama: pedagoške strategije švedskih učitelja tehnologije. Int J Technol des Educ 33, 1345–1368 (2023)., datum pristupa: 24.11.2023., izvor: <https://doi.org/10.1007/s10798-022-09786-7>
34. <https://www.skole.hr/replit-online-platforma-za-jednostavan-i-pristupacan-razvoj-koda/>, datum pristupa: 28.11.2023.
35. Price CB, Price-Mohr RM. An evaluation of primary school children coding using a text-based language (Java). Comp Sch. 2018;35(4):284–301., datum pristupa: 25.11.2023.
36. Kalelioglu, Filiz, i Yasemin Gülbahar. "Učinci podučavanja programiranja putem nule na vještine rješavanja problema: rasprava iz perspektive učenika." Informatika u obrazovanju 13.1 (2014): 33-50., datum pristupa: 23.11.2023.
37. "Sposobnost učenika za učenje jezika računalnog programiranja u osnovnim školama." Međunarodni e-časopis za obrazovne studije 3.6 (2019): 109-115., datum pristupa: 23.11.2023.
38. Cui, Z., & Ng, O.-L. (2021). Međudjelovanje matematičkog i računalnog razmišljanja u matematičkom rješavanju problema učenika osnovnih škola unutar programskog okruženja. Časopis za obrazovna računalna istraživanja, 59 (5), 988-1012. <https://doi.org/10.1177/0735633120979930>, datum pristupa: 23.11.2023.



39. Bjursten, EL., Nilsson, T. & Gumaelius, L. Računalno programiranje u osnovnim školama: pedagoške strategije švedskih učitelja tehnologije. *Int J Technol des Educ* 33, 1345–1368 (2023). <https://doi.org/10.1007/s10798-022-09786-7>, datum pristupa: 24.11.2023.
40. Sentance, S., Csizmadia, A. Računalstvo: Izazovi i strategije iz perspektive učitelja. *Educ Inf Technol* 22, 469–495 (2017). <https://doi.org/10.1007/s10639-016-9482-0>, datum pristupa: 24.11.2023.
41. Hill, C., Dwyer, H. A., Martinez, T., Harlow, D., & Franklin, D. (2015). Podovi i fleksibilnost: Dizajniranje programskog okruženja za učionice od 4. do 6. razreda. U predmetu A. Decker, K. Eiselt, C. Alphonse & J. Tims (Eds.), *Zbornik radova 46. tehničkog simpozija ACM-a o obrazovanju u računalnim znanostima* (str. 546–551). Udruženje računalnih strojeva., datum pristupa: 24.11.2023.
42. Elsayah, W., Hill, C. Barriers to programming education in UAE primary schools: a qualitative review from ICT teachers' perspectives. *Discov Educ* 2, 20 (2023). <https://doi.org/10.1007/s44217-023-00043-0>, datum pristupa: 25.11.2023.
43. Gess-Newsome, J. (2015). Model profesionalnog znanja i vještine nastavnika, uključujući PCK: Rezultati razmišljanja sa sastanka na vrhu PCK-a. U A. Berry, P. Friedrichsen, & J. Loughran (Eds.), *Ponovno ispitivanje pedagoškog sadržaja, znanje u znanstvenom obrazovanju* (str. 38–52). New York: Routledge., datum pristupa: 24.11.2023.
44. Hill, C., Dwyer, H. A., Martinez, T., Harlow, D., & Franklin, D. (2015). Podovi i fleksibilnost: Dizajniranje programskog okruženja za učionice od 4. do 6. razreda. U predmetu A. Decker, K. Eiselt, C. Alphonse & J. Tims (Eds.), *Zbornik radova 46. tehničkog simpozija ACM-a o obrazovanju u računalnim znanostima* (str. 546–551). Udruženje računalnih strojeva., datum pristupa: 24.11.2023.
45. Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011). Navike programiranja u ogrebotini. U predmetu G. Rößling, T. Naps i C. Spannagel (Eds.), *Zbornik radova 16. godišnje zajedničke konferencije o inovacijama i tehnologiji u*

- obrazovanju u računalnim znanostima (str. 168–172). Udruženje računalnih strojeva., datum pristupa: 24.11.2023.
46. Piteira M, Costa CJ, Aparicio M. Computer Programming learning: how to apply gamification on online courses? *Journal of information systems engineering & management. international association for digital transformation and technological innovation*, 3(2). 2018., datum pristupa: 25.11.2023.
47. Webb M, Davis N, Bell T, Katz YJ, Reynolds N, Chambers DP, Sysło MM. Computer science in K-12 school curricula of the 21st century: why, what and when? *Educ Inf Technol*. 2017;22(2):445–68., datum pristupa: 25.11.2023.
48. Alam A. Educational robotics and computer programming in early childhood education: a conceptual framework for assessing elementary school students' computational thinking for designing powerful educational scenarios. In *2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)* (pp. 1–7). IEEE. 2022., datum pristupa: 26.11.2023.
49. Gunbatar MS, Karalar H. Gender differences in middle school students' attitudes and self-efficacy perceptions towards mBlock programming. *Europ J Educ Res*. 2018;7(4):925–33., datum pristupa: 26.11.2023.
50. Malaise, Y., & Signer, B. (2022, March). King's Scroll: An Educational Game to Practise Code Prediction. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education 2* 1367–1367., datum pristupa: 26.11.2023.
51. Tsai CY. Improving students' understanding of basic programming concepts through visual programming language: the role of self-efficacy. *Comput Hum Behav*. 2019;95:224–32., datum pristupa: 26.11.2023.
52. Piteira M, Costa CJ, Aparicio M. Computer Programming learning: how to apply gamification on online courses? *Journal of information systems engineering & management. international association for digital transformation and technological innovation*, 3(2). 2018., datum pristupa: 26.11.2023.
53. Noh J, Lee J. Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educ Tech Res Dev*. 2020;68(1):463–84., datum pristupa: 26.11.2023.

54. Siu-Cheung Kong, Ming Ming Chiu, Ming Lai, A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education, *Computers & Education*, Volume 127, 2018, Pages 178-189, ISSN 0360-1315, <https://doi.org/10.1016/j.compedu.2018.08.026>., izvor: <https://www.sciencedirect.com/science/article/pii/S0360131518302367>, datum pristupa: 14.05.2023.
55. (Baichang Zhong, Qiyun Wang, Jie Chen), The impact of social factors on pair programming in a primary school, *Computers in Human Behavior*, Volume 64, (2016), Pages 423-431, datum pristupa: 03.12.2023.
56. (G. Braught et al.), The effects of pair-programming on individual programming skill, *ACM SIGCSE Bulletin* (2008), datum pristupa: 03.12.2023.
57. D.C. Cliburn, Experiences with pair programming at a small college, *Journal of Computing Sciences in Colleges* (2003), datum pristupa: 03.12.2023.
58. J.E. Hannay et al., The effectiveness of pair programming: A meta-analysis, *Information and Software Technology* (2009), datum pristupa: 03.12.2023.
59. S.Y. Lye et al., Review on teaching and learning of computational thinking through programming: What is next for K-12?, *Computers in Human Behavior*, (2014), datum pristupa: 03.12.2023.
60. (Lihui Sun, Linlin Hu, Danhua Zhou, 2021.), Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors, *Thinking Skills and Creativity*, Volume 42, 100926, ISSN 1871-1871, <https://doi.org/10.1016/j.tsc.2021.100926>, datum pristupa: 05.12.2023.
61. (E.N. Caeli et al.), Unplugged approaches to computational thinking: A historical perspective *TechTrends*, (2020), datum pristupa: 05.12.2023.
62. (T. Bell et al.), Computational thinking is more about humans than computers *Set*, (2016), datum pristupa: 05.12.2023.
63. (M.Á. Conde et al.), Promoting computational thinking in K-12 students by applying unplugged methods and robotics, datum pristupa: 05.12.2023.

64. (C. Angeli et al.), Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy *Computers in Human Behavior*, (2020), datum pristupa: 05.12.2023.
65. (F. Kalelioğlu), A new way of teaching programming skills to K-12 students: Code.org *Computers in Human Behavior*, (2015), datum pristupa: 05.12.2023.
66. (S. Atmatzidou et al.), Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences *Robotics and Autonomous Systems*, (2016), datum pristupa: 05.12.2023.
67. (Lihui Sun, Linlin Hu, Danhua Zhou), Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors, *Thinking Skills and Creativity*, Volume 42, 2021, 100926, ISSN 1871-1871, <https://doi.org/10.1016/j.tsc.2021.100926>, datum pristupa: 06.12.2023.
68. (Akinola, S. O., 2015), Computer programming skill and gender difference: An empirical study. *American Journal of Scientific and Industrial Research*,7(1), 1–9., datum pristupa: 06.12.2023.
69. (Noh, J., Lee, J., 2020), Effects of robotics programming on the computational thinking and creativity of elementary school students. *Education Tech Research Dev* 68, 463–484 (2020). <https://doi.org/10.1007/s11423-019-09708-w>, datum pristupa: 06.12.2023.
70. (Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M., 2017), Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*,109, 162–175., datum pristupa: 06.12.2023.
71. (Carter, J., & Jenkins, T. , 1999), Gender and programming. *SIGCSE Bulletin*,31(3), 1–4. <https://doi.org/10.1145/384267.305824>., datum pristupa: 06.12.2023.
72. (Papastergiou, M., 2009), Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Computers & Education*,52(1), 1–12., datum pristupa: 06.12.2023.
73. (Atmatzidou, S., & Demetriadis, S., 2016), Advancing students' computational thinking skills through educational robotics: A study on age and gender

- relevant differences. *Robotics and Autonomous Systems*,75, 661–670., datum pristupa: 06.12.2023.
74. (Papastergiou, M., 2009), Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Computers & Education*,52(1), 1–12., datum pristupa: 06.12.2023.
75. (Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M., 2008), New pathways into robotics: Strategies for broadening participation. *Journal of Science Education and Technology*,17, 59–69., datum pristupa: 06.12.2023.
76. (Baser, M., 2013), Attitude, gender and achievement in computer programming. *Online Submission*,14(2), 248–255., datum pristupa: 06.12.2023.
77. (Akinola, S. O., 2015), Computer programming skill and gender difference: An empirical study. *American Journal of Scientific and Industrial Research*,7(1), 1–9., datum pristupa: 06.12.2023.
78. (Baser, M., 2013), Attitude, gender and achievement in computer programming. *Online Submission*,14(2), 248–255., datum pristupa: 06.12.2023.
79. (Durak, H. Y., & Saritepeci, M., 2018), Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*,116, 191–202., datum pristupa: 06.12.2023.
80. (Mladenović, M., Boljat, I. & Žanko, Ž.,2018), Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Educ Inf Technol* 23, 1483–1500 (2018). <https://doi.org/10.1007/s10639-017-9673-3>, datum pristupa: 07.12.2023.
81. (SitiRosminah MD Derus1 , Ahmad ZamzuriMohamad Ali2, 2012), DIFFICULTIES IN LEARNING PROGRAMMING: VIEWS OF STUDENTS, [ICCI2012\\_siti\\_rosminah-libre.pdf \(d1wqtxts1xzle7.cloudfront.net\)](https://www.researchgate.net/publication/312512512/DIFFICULTIES_IN_LEARNING_PROGRAMMING_VIEWS_OF_STUDENTS), datum pristupa: 07.12.2023.
82. (Suman, Darko, and Damir Purković.), "Preferencije učenika kao polazište za organizaciju i strukturiranje nastave općeg tehničkog odgoja i obrazovanja." *Politehnika: Časopis za tehnički odgoj i obrazovanje* 2.2 (2018): 29-42., datum pristupa: 28.11.2023.

83. (Režić, Jelena.), *Hibridno učenje nastavnog predmeta informatika u drugom razredu osnovne škole*. Diss. University of Split. Faculty of Humanities and Social Sciences, University of Split. Department of Teacher Education, 2020., datum pristupa: 28.11.2023.
84. (Marković, Monika.), *Uloga metodologije pedagoškog istraživanja u cilju poboljšanja vještina programiranja učenika srednje škole*. Diss. University of Pula. Faculty of Informatics in Pula, 2019., datum pristupa: 28.11.2023.
85. (Mayer, R.E., Moreno, R.), Animation as an Aid to Multimedia Learning. *Educational Psychology Review* **14**, 87–99 (2002).  
<https://doi.org/10.1023/A:1013184611077>
86. (Kelleher, C., & Pausch, R.), "Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers". *ACM Computing Surveys (CSUR)*, 37(2), 83-137. 2005.
87. (Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., ... & Paterson, J.), "A survey of literature on the teaching of introductory programming". *ACM SIGCSE Bulletin*, 39(4), 204-223. 2007.
88. (Dadić, T.), *Poučavanje početnoga programiranja temeljeno na semantičkom vrednovanju programa*, Zagreb, 2015.
89. (Radonić, F.), (1997), *Obrazovna tehnologija u nastavi i učenju*. Zagreb: Birotehnika 50
90. (Semenov, A.), (2005) *Information and communication technologies in schools – a handbook for teachers or How ICT Can Create New Open Learning Environments*. Paris: UNESCO
91. (DigCompEdu), (2020) [https://www.e-skole.hr/wp-content/uploads/2020/04/CARNET\\_digitalne\\_kompetencije\\_2020.pdf](https://www.e-skole.hr/wp-content/uploads/2020/04/CARNET_digitalne_kompetencije_2020.pdf)
92. (Beck, K.), (2020) *Extreme programming explained: embrace change*. addison-wesley professional
93. (Zuliani, Đ., Matić, M. i Keteleš, V.), (2015). POTICANJE KREATIVNOSTI U NASTAVI INFORMATIKE. *Život i škola*, LXI (1), 25-35. Preuzeto s <https://hrcak.srce.hr/152296>, datum pristupa: 16.06.2024.
94. (Sneider, C., Stephenson, C., Schafer, B., & Flick, L.), (2014). Computational thinking in high school science classrooms. *The Science Teacher*, 81(5), 53-59

95. (Wing, J. M.), (2008). Computational thinking and thinking about computing  
Philosophical Transactions of the Royal Society a: Mathematical, Physical and  
Engineering Sciences, 366, 3717-3725.
96. (Razzouk, R., & Shute, V.), (2012). What is design thinking and why is it  
important? Review of Educational Research, 82, 330-348
97. (L. Ma, J. Ferguson, M. Roper and M. Wood), (2011). "Investigating and  
Improving the Models of Programming Concepts Held by Novice  
Programmers," Computer Science Education, vol. 21, no. 1, pp. 57-80, 2011.
98. (FESSAKIS, GEORGIOS, et al.), (2019). "Overview of the Computer  
Programming Learning Environments for primary education." Review of  
Science, Mathematics and ICT Education 13.1: 7-33.
99. (Wakil, K., Khdir, S., Sabir, L., Nawzad, L. (2019). Student Ability for Learning  
Computer Programming Languages in Primary Schools. International E-  
Journal of Educational Studies, 3(6), 109-115.  
<https://doi.org/10.31458/iejes.591938>)

# PRIMJENA SUSTAVA REPLIT U OSNOVNOŠKOLSKOJ NASTAVI INFORMATIKE U DOMENI RAČUNALNOG RAZMIŠLJANJA I PROGRAMIRANJA

VEDRAN MUDRINIĆ

[vmudrinic@student.unipu.hr](mailto:vmudrinic@student.unipu.hr)

## SAŽETAK

Odabir sustava za poučavanje programiranja u osnovnoj školi u domeni računalnog razmišljanja i programiranja ima uvelike utjecaja na sam tijek usvajanja znanja. Odabirom različitih sustava imamo i različite učinke na sam tijek i količinu usvajanja sadržaja. U ovome radu je istraživana utjecaj i stavovi učenika koji su prvo bili radili u lokalno instaliranom prevoditelju IDLE, a nakon toga u sustavu za učenje programiranja Replit. Stavovi i uvjerenja učenika o prednostima, motivaciji, prihvatljivosti, dobroj preporuci za korištenjem iste su u značajnoj i prevladavajućoj prednosti u odnosu na lokalno instaliranog prevoditelja IDLE, te da je rad u Replit-u jednostavniji, lakši, zanimljiviji, da im je lakši rad s datotekama i da nemaju problema s instalacijom prevoditelja. Također učenici ističu benefite suradničkog i timskog rada, te da bih svojim vršnjacima sigurno preporučili učenje u Replit-u. Također učenici su stava da im Replit zadovoljava sve njihove potrebe za izvršavanje učeničkih obveza u smislu učenja programiranja u školi i kod kuće. S tim u svezi može se zaključiti da je rad i učenje programiranja u *cloud* tehnologiji sustava Replit bolje o odnosu na lokalno instaliranog prevoditelja na računalu.

Ključne riječi: osnovna škola, programiranje, sustavi za učenje programiranja, Replit.



# APPLICATION OF REPLIT SYSTEM IN ELEMENTARY SCHOOL TEACHING OF INFORMATICS IN DOMAIN OF COMPUTER THINKING AND PROGRAMMING

VEDRAN MUDRINIĆ

[vmudrinic@student.unipu.hr](mailto:vmudrinic@student.unipu.hr)

## SUMMARY

The choice of system for teaching programming in elementary school in the domain of computational thinking and programming has a great impact on the course of knowledge acquisition. By choosing different system, we have different effects on the flow and amount of content adoption. This paper investigated the influence and attitudes of students who first worked in the locally installed IDLE compiler, and then in Replit. Students' attitudes and beliefs about the benefits, motivation, acceptability, good recommendation for using it are in a significant and prevailing advantage compared to the locally installed IDLE compiler, and that working in Replit is simpler, easier, more interesting, that it is easier for them to work with files and that they have no problems installing the compiler. The students also point out the benefits of collaborative and team work, and that they would definitely recommend studying at Replit to they peers. Students are also of the opinion that Replit meets all their needs for fulfilling student obligations in terms of learning programming at school and at home. In this regard, it can be concluded that working and learning programming in the cloud technology of the Replit system is better comparing to locally installed compiler on the computer.

Keywords: elementary school, programming, systems for learning programming, Replit.