



Osnovna šola Polje

Raziskovalna naloga

PROGRAMIRANJE Z DELČKI

Računalništvo in informatika

Mentorica:

mag. Polona Theuerschuh

Avtorica:

Natalija Žabjek, 9. a

Ljubljana, 2023

ZAHVALA

Zahvaljujem se mentorici mag. Poloni Theuerschuh za pomoč in podporo pri raziskovalni nalogi.

Zahvaljujem se tudi:

- učencem OŠ Polje, ki so sodelovali v raziskavi,
- doc. dr. Demšarju, predavatelju na Fakulteti za informatiko in računalništvo v Ljubljani in
- mag. Lokarju, višjemu predavatelju na Fakulteti za matematiko in fiziko v Ljubljani, ki sta odgovarjala na vprašanja,
- g. Marcelu Theuerschuh za pomoč pri izvedbi programiranja v razredu pri predmetu NIR,
- učiteljem, ki so rešili anketni vprašalnik in
- mojim staršem, ki so me podpirali pri delu.

Natalija

POVZETEK

Potreba po znanju programiranja vsak dan sorazmerno z napredovanjem tehnologije narašča in tako je prišlo do pomembnega priporočila, da se programiranja začnemo učiti že v osnovni šoli.

Otroci se za začetek začnejo učiti programiranja z delčki oziroma slikovnega programiranja. Tega se lahko naučijo v različnih slikovnih programih.

V osnovni šoli je le manjši delež učencev vključen v računalniške izbirne predmete (neobvezne ali obvezne) in tako le manjši del učencev začne programirati in razvijati računalniško mišljenje.

Opazila sem, da ima večina ljudi (ne samo učencev) težave že z osnovno uporabo računalnika, zato me je zanimalo, koliko se učenci lahko naučijo o tem v šoli.

Učiteljem računalništva na različnih osnovnih šolah sem poslala vprašalnik, kjer sem jih povprašala o njihovem mnenju o tem, kdaj naj bi se učenci začeli učiti programiranja in katere programe za programiranje uporabljajo pri pouku. Izvedla sem tudi dva intervjuja s profesorjema, ki programiranje učita na fakultetah, saj me je zanimalo tudi njuno mnenje. Prav vsi so menili, da je treba začeti programirati že v osnovni šoli.

Zanimalo me je znanje programiranja učencev Osnovne šole Polje, ki obiskujejo neobvezni izbirni predmet računalništvo ter nadarjene na področju naravoslovja iz 8. in 9. razreda. Učenci so reševali naloge v okolju Pišek ter kasneje za primerjavo naloge, ki sem jih sestavila sama, še v Blockly editorju. Moja pričakovanja, da bodo starejši učenci bolj uspešni pri nalogah programiranja, se niso uresničila.

Izkazalo se je, da učenci nimajo zadostnega znanja o programiranju, kar bo potrebno spremeniti, zato jim moramo nuditi možnost za učenje programiranja ter jih k temu tudi spodbujati.

KAZALO

ZAHVALA	2
POVZETEK.....	3
KAZALO	4
1 UVOD	6
2 PROGRAMIRANJE V OSNOVNI ŠOLI.....	7
2.1 Pomembnost programiranja v osnovni šoli.....	7
2.2 Programi, ki jih uporabljajo v osnovni šoli.....	8
2.3 Računalniška tekmovanja v osnovni šoli	9
2.3.1 Tekmovanje Pišek	9
2.3.2 Tekmovanje Bober	10
2.3.3 Računalniški pokal Logo za otroke v vrtcih in učence prvega triletja OŠ	11
2.3.4 Tekmovanje iz znanja računalništva – programiranje za osnovne šole.	12
3 SLIKOVNO PROGRAMIRANJE ALI PROGRAMIRANJE Z DELČKI	13
3.1 Programski koncepti.....	14
3.2 Vrste nalog	15
3.3 Pomen razvijanja programskih konceptov	16
4 IZHODIŠČE RAZISKAVE IN HIPOTEZE	17
4.1 Cilji raziskave	17
4.2 Postavitev hipotez	17
4.3 Raziskava	18
4.3.1 Raziskovalna metoda dela	18
4.3.2 Vzorec	18
4.3.3 Naloge.....	18
5 RAZISKAVA.....	19
5.1 Intervjuji	19
5.1.1 Intervju z gospodom dr. Janezom Demšarjem, profesorjem na Fakulteti za računalništvo in Pedagoški fakulteti v Ljubljani.....	19
5.1.2 Intervju z gospodom mag. Matijem Lokarjem, višjim predavateljem na Fakulteti za matematiko in fiziko v Ljubljani.....	21
5.2 Ankete	24
5.2.1 Anketa za učitelje.....	24
5.2.2 Anketa za učence	26
5.3 Spremljanje reševanja nalog učencev pri neobveznem izbirnem predmetu računalništvo.....	27
5.3.1 Rezultati učencev, ki prvo leto obiskujejo neobvezni izbirni predmet računalništvo	32

5.3.1	Rezultati učencev, ki že drugo leto obiskujejo neobvezni izbirni predmet računalništvo	34
5.4	Sestavljanje avtorskih nalog, ki preverjajo programerske koncepte.....	36
5.4.1	Reševanje nalog v Blockly editorju.....	37
5	REZULTATI RAZISKAVE	47
6.1	Rezultati anket.....	47
6.1.1	Rezultati ankete za učitelje	47
6.1.1	Rezultati ankete za učence	49
6.2	Rezultati reševanje nalog - učenci.....	52
7	RAZPRAVA.....	54
7.1	Potrjevanje hipotez	54
7.2	Ugotovitve	55
8	ZAKLJUČEK	56
9	LITERATURA IN VIRI.....	57
10	PRILOGE	60
10.1	Vprašalnik za učitelje računalništva	60
10.2	Vprašalnik za učence	66

1 UVOD

Tehnologija se razvija iz dneva v dan in z njo tudi samo programiranje. Posledično potreba po znanju programiranja narašča in to moramo sprejeti, ter se temu prilagoditi. Glede na to, da se otroški možgani lahko marsičesa naučijo, je tudi učenje programiranja priporočljivo že v osnovni šoli.

Učenci se v osnovni šoli za začetek začnejo učiti slikovnega programiranja oziroma programiranja z delčki. Ker so programi s slikovnim programiranjem (oz. kot jim tudi rečemo programi za programiranje z delčki) sestavljeni iz različnih barv ter delčkov, to otrokom ne predstavlja učenja, temveč to razumejo kot zanimivo in zabavno igranje igrice, kjer spoznavajo marsikaj novega.

Programiranje je pomembno, saj nam v vsakdanjem življenju omogoča reševanje problemov tudi na drugačen način, kar pa je marsikdaj uporabno.

Zanimalo me je, kakšne možnosti za učenje slikovnega programiranja učenci v osnovnih šolah dejansko imajo, ter kaj si o tem mislijo učitelji računalništva v osnovnih šolah.

Katera starost je najprimernejša za začetek programiranja? Odgovor na to vprašanje sem poiskala pri učiteljih računalništva na osnovnih šolah in pri profesorjih računalništva na fakultetah.

Odločila sem se, da znanje programiranja učencev tudi preverim. Učenci bodo najprej reševali že sestavljene naloge v programskem okolju Pišek, kjer je število delčkov omejeno in kjer lahko takoj preverijo svojo rešitev, in naloge, ki jih bom sestavila za namen raziskave, v programu Blockly (v blockly urejevalniku).

Delo učencev bom spremljala, da bi ugotovila, kako dobro programirajo in kaj jim dela največ težav.

2 PROGRAMIRANJE V OSNOVNI ŠOLI

2.1 Pomembnost programiranja v osnovni šoli

Včasih je veljalo prepričanje, da je programiranje le za strokovnjake in računalniške nadobudneže ali pa kot nekakšen nepomemben hobi. Z razvojem tehnologije pa se je pokazala potreba, da se otroci že v osnovni šoli naučijo nekaj osnov o programiranju. Preden se ustrašimo programiranja zaradi težavnosti, pa se moramo zavedati, kaj vse nam nudi znanje programiranja.

Za otroke, ki bi se radi naučili nekaj osnov programiranja obstajajo tečaji, spletne strani in tudi knjige. Če bi se posameznik želel naučiti osnov programiranja, bi si lahko brez težav priskrbel tako materialno kot tudi osebno pomoč.

Če bi se otroci v osnovni šoli naučili osnov programiranja, bi se marsikdo našel na tem področju. Ta ugotovitev lahko pri posamezniku vpliva na nadaljnjo šolanje in nato tudi na izbiro poklica (npr. programerja). Poklic programerja nam nudi več, kot si lahko predstavljamo. Delo je zanimivo, možnosti zaposlitve so velike, poleg tega lahko s takšnim znanjem zaslužimo veliko. Marsikje se iz leta v leto išče več in več programerjev. Znanje programiranja pa se ne obrestuje le v izbrani službi, temveč tudi v vsakdanjem življenju, saj lahko omogoči nov pogled na reševanje problemov. Z učenjem programiranja se naučimo logičnega reševanja različnih težav.¹



Slika 1: Programiranje v osnovni šoli.

¹ Zakaj se je pomembno naučiti osnov programiranja? Pridobljeno 12. 12. 2022 s <https://www.smartninja.si/blog/zakaj-se-je-pomembno-nauciti-osnov-programiranja>

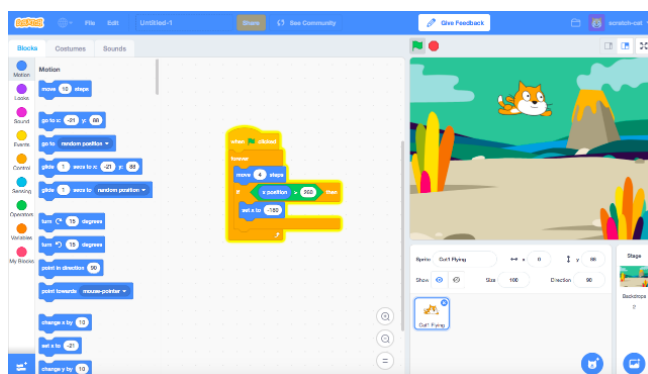
2.2 Programi, ki jih uporabljajo v osnovni šoli

Učenci, ki so zainteresirani za programiranje, se programiranja začnejo učiti že v osnovni šoli. Nekateri se prijavijo na razne tečaje, drugi pa poiščejo pomoč na internetu ali v knjigah. Tudi nekatere osnovne šole omogočajo nekaj malega v smislu učenja računalništva in programiranja kot izbirni predmet oziroma interesna dejavnost, vendar je tega izključno premalo, kar bo z vsakim letom bolj očitno.²

Učenci v osnovni šoli lahko v drugem triletju izberejo neobvezni izbirni predmet računalništvo (v četrtem, petem in/ali šestem razredu). V tretjem triletju pa lahko izbirajo med obveznimi izbirnimi predmeti: urejanje besedil, multimedija in računalniška omrežja.

Programirajo predvsem pri neobveznem izbirnem predmetu računalništvo. Pri obveznih izbirnih predmetih pa je programiranju namenjen le manjši del ur.

V osnovni šoli se učenci na začetku večinoma posvetijo programom kot so Scratch, Pišek, App Inventor, Blockly in Visual, saj vključujejo programiranje z delčki, kar je za začetek lažje.³



Slika 2: Programiranje v programu Scratch.

Ko se učenec že nauči programirati v teh programih, začne uporabljati programe kot so Python, C++, JBuilder, Delhi, Basic, JavaScript, saj v teh programih začne konkretno pisati program in je zato težje.⁴

² Drinovec, A. Računalniško opismenjevanje v osnovni šoli v Sloveniji. Pridobljeno 14. 12. 2022 s https://www.makspecar.si/application/files/7815/5253/8703/VIVID-2016-Racunalnisko_opismenjevanje_v_OS.pdf

³ Uporabniška programska oprema. Pridobljeno 8. 12. 2022 s http://egradivo.ecnm.si/KIT/uporabnika_programska_oprema.html

⁴ Wainwright, M. 2016. Osnove programiranja: Vodič po računalniškem programiranju. Tržič: Učila International d. o. o


```
4
5 while y <= len(x):
6     os.system("clear")
7     print(x[:y])
8     time.sleep(0.2)
9     y = y+1
10    time.sleep(2)
11    x = "You will stay here for as long as you can, you may
12    forfeit when you would like to."
13    y = 0
14
15 while y <= len(x):
16     os.system("clear")
17     print(x[:y])
18     time.sleep(0.2)
19     y = y+1
20    time.sleep(2)
```

Slika 3: Zaslonka slika - programiranje v programu Python.

2.3 Računalniška tekmovanja v osnovni šoli

V osnovni šoli učenci lahko tekmujejo na tekmovanjih iz računalništva, oziroma programiranja.

Trenutno lahko sodelujejo na tekmovanjih kot so Bobrček in Mladi Bober⁵, Pišek⁶, Računalniški pokal Logo za otroke v vrtcih in učence prvega triletnega OŠ in Tekmovanje iz znanja računalništva – programiranje za osnovne šole.⁷

Kot primer tekmovanja iz programiranja lahko navedemo tekmovanje Pišek. Tekmovanje Bober pa lahko uvrščamo med tekmovanja iz računalniškega razmišljanja.

2.3.1 Tekmovanje Pišek

Portal Pišek je nastal leta 2018 z namenom, da bi se več otrok zanimalo za programiranje z reševanjem nalog s slikovnim programiranjem oziroma programiranjem z delčki. Na portalu se nahajajo 4 učne poti oziroma metode (Zgodbe, Programski koncepti, Učbenik, Code Week), s katerimi lahko učenec pride do začetnega znanja programiranja.⁸

Pri tekmovanju Pišek je cilj tekmovanja napisati program z delčki, ki ustreza navodilom dane naloge. Portal Pišek omogoča tekmovanje, v katerem lahko sodelujejo učenci osnovne in srednje šole. Tekmovanje za osnovnošolce se deli na starostni kategoriji: 4. – 6. razred, 7. – 9. razred.

⁵ ACM Tekmovanja – Bober. Pridobljeno 11. 12. 2023 s <https://tekmovanja.acm.si/?q=node/693>

⁶ ACM Tekmovanja – Pišek. Pridobljeno 12. 12. 2022 s <https://tekmovanja.acm.si/?q=pisek/o-pisku>

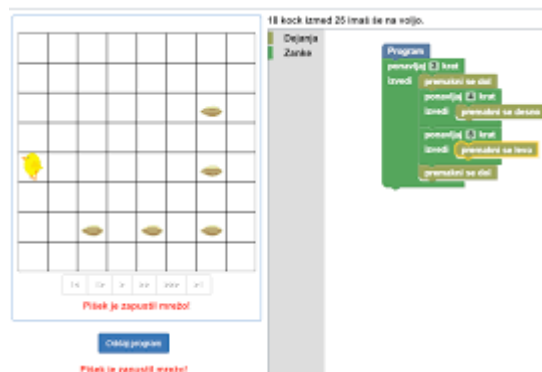
⁷ Programiranje. Pridobljeno 10. 1. 2023 s <https://www.zotks.si/programiranje/razpis>

⁸ Pišek. Pridobljeno 12. 12. 2022 s <https://pisek.acm.si/contents/4903/>

Vsaka starostna kategorija pa ima 2 težavnosti ravni (začetno in napredno kategorijo). Tako se lahko v programiranju preizkusijo tako začetniki kot tudi učenci, ki so že večji programiranja.⁹



Slika 4: Tekmovanje Pišek.



Slika 5: Program v okolju Pišek.

2.3.2 Tekmovanje Bober

Tekmovanje Bober je mednarodno računalniško tekmovanje za učence osnovne in srednje šole, ki je prvič potekalo leta 2004 v Litvi. Namen tekmovanja je povečati interes učencev za računalništvo.

Učenci naj bi spoznali, da računalnik ni le orodje za komuniciranje, brskanje po spletu, urejanje besedil ter poslušanje glasbe in gledanje filmov.

Računalnik je neizčrpen vir zanimivih logičnih problemov, zaradi katerih računalničarju ni nikoli dolgčas.

Vprašanja na tekmovanju se navezujejo na računalniško razmišljanje, logično sklepanje ter razvoj spretnosti za reševanje težav. Učenci spoznajo, da je takšno razmišljanje in produktivno reševanje problemov zanimivo in zabavno.¹⁰



ACM Tekmovanja - Bober
Mednarodno tekmovanje iz računalniškega razmišljanja

Slika 6: Tekmovanje Bober.

⁹ ACM tekmovanja - Pišek. Tekmovalne kategorije. Pridobljeno 13. 12. 2022 s <https://tekmovanja.acm.si/?q=pisek/tekmovalne-kategorije>

¹⁰ ACM tekmovanja – Bober. Pridobljeno 14. 12. 2022 s <https://tekmovanja.acm.si/?q=bober/o-bobru>

Bober se deli na več kategorij: Bobrček, Mladi bober, Bober in Starejši bober.

Tekmovanje Bobrček je namenjen učencem, ki obiskujejo 2. do 5. razred osnovne šole. Naloge rešujejo na papir.

Na tekmovanju Mladi bober lahko sodelujejo učenci šestega, sedmega, osmega in devetega razreda osnovne šole. Izkušenega Bobra se lahko udeležijo dijaki srednje šole. Naloge rešujejo na računalniku.

TEKMOVANJE BOBER 2019/20

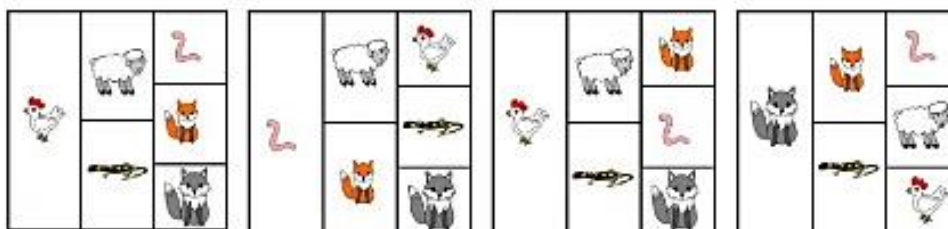
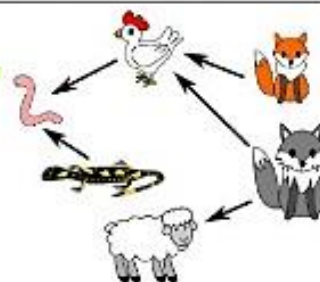
ČETRTI IN PETI RAZRED

IME IN PRIIMEK: _____

NA KMETIJI

Koko ima doma 6 živali. Za vsako je pripravil svojo ogrado. Poskrbeti mora, da so vse živali varne, zato pazi, da v sosednjih ogradah niso živali, ki bi se med seboj napadle. Kdo napada koga, kaže desna slika.

V katerem od spodnjih primerov je v nevarnosti vsaj ena žival?



Slika 7: Primer naloge za učence 4. in 5. razreda.

2.3.3 Računalniški pokal Logo za otroke v vrtcih in učence prvega triletja OŠ

Na Računalniškem pokalu LOGO lahko sodelujejo otroci iz vrtcev in prvega triletja osnovne šole.

Otroci tekmujejo v dvojicah v štirih starostnih skupinah: vrtec, 1. razred, 2. razred in 3. razred.¹¹

Tekmovanje poteka v učnem okolju Logo.

¹¹ Programiranje. Pridobljeno 10. 1. 2023 s <https://www.zotks.si/programiranje/razpis>

2.3.4 Tekmovanje iz znanja računalništva – programiranje za osnovne šole.

Tekmovanje poteka v programiranju s programi Logo, Pascal, Python in v programu Scratch.

Tekmovanje poteka v več kategorijah in več težavnostnih stopnjah.

Logo od 4. do 6. razreda se izvaja v dveh težavnostnih skupinah: A Logo skupina: 4. razred in B Logo skupina: 5. in 6. razred.

Tekmovanje iz programiranja v Pascalu ali C od 7. do 9. razreda se izvaja v treh težavnostnih stopnjah: 7. razred, 8. razred in 9. razred.

Scratch od 4. do 9. razreda se izvaja v treh težavnostnih stopnjah: 4. in 5. razred, 6. in 7. razred in 9. razred.

Tudi tekmovanje v programiranju s Pythonom za učence od 7. do 9. razreda poteka v treh skupinah: 7. razred, 8. razred in 9. razred.¹²



Slika 8: Tekmovanje za računalniški pokal LOGO.

¹² Programiranje. Pridobljeno 10. 1. 2023 s <https://www.zotks.si/programiranje/razpis>

3 SLIKOVNO PROGRAMIRANJE ALI PROGRAMIRANJE Z DELČKI

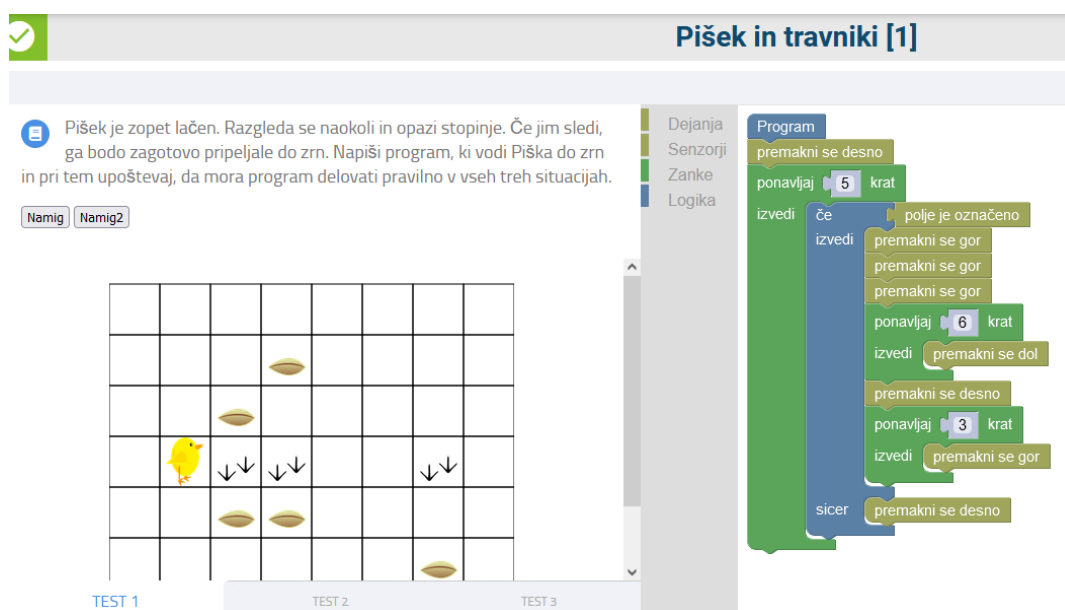
Programiranje je lahko težko, vendar se ga lahko vsak nauči. V pomoč učenju programiranja obstajajo programi, kjer se programira z delčki.

Zaenkrat sta v uporabi dva izraza – programiranje z delčki in slikovno programiranje, saj se strokovnjaki še niso dogovorili za en izraz.

Pri programiranju z delčki so ukazi že sestavljeni, potrebno pa jih je razporediti v smiselno zaporedje.

Taki programi so primernejši za začetnike, saj se lahko osredotočijo na zaporedje in postopek, namesto na sintakso oziroma slovnične napake, ki so lahko pri naprednejših programih težavnejše. S programiranjem z delčki se učenec nauči razmišljati kot računalnik in začne razumevati, kako sploh deluje računalnik.

S pomočjo programov Scratch, Pišek, App Inventor, Blockly, Visual itd. lahko to uresniči in se pripravi na napredovanje.¹³



Slika 9: Zaslonska slika programa Pišek.

¹³ Anželj, G. et al. 2018. Slikovno programiranje. E-učbenik. Pridobljeno 14. 12. 2022 s <https://lusy.fri.uni-lj.si/ucbenik/prog/1221/index.html>

3.1 Programski koncepti

Programski koncepti so procesi, s katerimi predstavimo sestavo in delovanje sistemov, ki jih s programom oblikujemo. Koncepti programskih jezikov so izpeljani v okviru določenih gradnikov programskih jezikov.¹⁴

Otrok se lahko začne učiti osnovnih programskih konceptov, takoj ko se nauči brati in logično sklepati.

Z malo pomoči lahko že predšolski otroci in prvošolci začnejo programirati v lažjih programih, kjer se programira z delčki, kot na primer Scratch. Otroci se lažje učijo preko igre, saj tako vedo, da je programiranje tudi zabavno in se z večjim veseljem odločijo za programiranje.¹⁵

Programskih konceptov je več in imajo različen pomen oz. namen.

Nekaj programskih konceptov in njihovi opisi so navedeni v spodnji tabeli.^{16 17}

Primer programskega koncepta	Opis
Zaporedje ukazov	Ukazovanje načina delovanja v določenem vrstnem redu.
Zanka FOR	Zanka, ki določi, kolikokrat se določen ukaz ponovi.
Zanka WHILE	Zanka, ki določa, da se nekaj izvaja dokler drug ukaz traja.
Napredne/ugnezdene zanke	Zanka v zanki.
Pogojni stavek (IF, ELIF, ELSE)	Izvrševanje ukazov pod pogoji.
Branje in izpis podatkov	Program prebere dano in izpiše zahtevano.
Spremenljivke	Definiranje danega in zahtevanega.
Upravljanje s seznammi	Preverjanje, brisanje, dodajanje in iskanje v seznamih.
Tabele in nizi	Izdelovanje seznamov iz danega.
Operator (senzor, logični, aritmetični, relacijski, sestavljeni)	Zaznavanje, prepoznavanje in ostanek.
Funkcije/Podprogrami in uporaba	Izvrševanje programskih ukazov kot enota.
Rekurzija	Funkcija, ki kliče sama sebe

¹⁴ Savnik, I. 2013. Programiranje II: Koncepti programskih jezikov. Pridobljeno 30. 12. 2022 s https://studentski.net/gradivo/upr_fmri1_pr2_sno_skripta_01

¹⁵ Lokar, M., Mujkić, M. Računalniško tekmovalstvo Pišek – oblika vzpodbujanje učenja programiranja za vse. Gradivo na izobraževanju Programiranje z delčki, UL Fakulteta za matematiko in fiziko (21. 10. 2022 – 13. 1. 2023)

¹⁶ Lokar, M. Programski koncepti. Gradivo na izobraževanju Programiranje z delčki, UL Fakulteta za matematiko in fiziko (21. 10. 2022 – 13. 1. 2023)

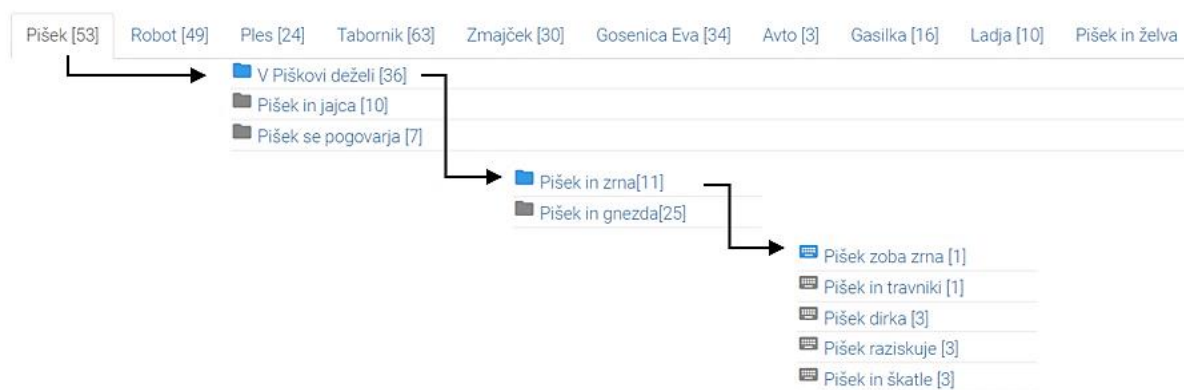
¹⁷ Lajovic, S. 2018. Python za otroke. Ljubljana: Pasadena

3.2 Vrste nalog

Naloge pri programiranju z delčki so različne. Lahko se delijo po težavnosti, starostni kategoriji, poglavjih in vsebini.

V okolju Pišek se naloge delijo na poglavja (*Pišek*, *Robot*, *Ples*, *Tabornik*, *Zmajček*, *Gosenica Eva*, *Avto*, *Gasilka*, *Ladja*, *Pišek in želva*), znotraj poglavja se naloge razdelijo v več obdobj, kjer posamezno obdobje predstavlja napredovanje, znotraj nekaterih obdobj pa se naloge lahko razdelijo še v oddelke, ki se nanašajo na določeno »zgodbo« oziroma določeno zaporedje nalog. Pri programu Pišek se naloge delijo tudi na začetne in napredne, da lahko vsi učenci utrjujejo naučeno.

Kategorij pri vrstah nalog je veliko in prav tako tudi samih nalog. Za uspešno učenje je najbolj priporočljivo iti postopoma, saj si naloge sledijo po težavnosti.¹⁸



Slika 10: Zaslonske slike - delitev nalog pri programu Pišek.

¹⁸ Pišek: Zgodbe. Pridobljeno 30. 12. 2022 s <https://pisek.acm.si/contents/4907-4902/>

3.3 Pomen razvijanja programskih konceptov

Začetni programerski koncepti so pomembni, ker se moramo zavedati nekaj splošnih dejstev.

Pri programiranju je zelo pomembna natančnost, saj se lahko v programu kaj kmalu najdejo napake, zaradi katerih program ne deluje pravilno, zato moramo biti pri pisanju programov zelo natančni in večkrat preveriti ali smo prav napisali program.

Tukaj nastopi dar vztrajnosti, saj je potrebno programe, še posebno daljše in zahtevnejše, večkrat preveriti in določene dele tudi znova napisati, oziroma najti problem ter se naučiti, kako ga rešiti. Za pomoč reševanja problemov v programu obstaja ukaz imenovan »razhroščevanje«, ki pomaga najti napako v programu, da se jo lahko potem popravi. Za izpolnjevanje tega moramo znati uporabljati določene ukaze, se zavedati podatkov in s tem razumeti programsko kodo ter znati izkoristiti veščine uporabe urejevalnika.

Marsikdo se kdaj znajde v situaciji, kjer ne zna rešiti problema, zato se mora naučiti, kako rešiti ta problem. Največkrat se takšnih stvari da naučiti preko interneta, vendar je potrebno znati to ustrezno poiskati. Tudi tega znanja primanjkuje današnji mladini, kar je vse bolj zaskrbljujoče.¹⁹

¹⁹ Lokar, M. Dodatni koncepti in veščine pri programiranju. Gradivo na izobraževanju Programiranje z delčki, UL Fakulteta za matematiko in fiziko (21. 10. 2022 – 13. 1. 2023)

4 IZHODIŠČE RAZISKAVE IN HIPOTEZE

4.1 Cilji raziskave

Cilj raziskave je ugotoviti, ali se učenci pri izbirnih predmetih s področja računalništva učijo programirati in v katerem programu programirajo.

Prav tako sem skušala ugotoviti, ali gre učencem programiranje lažje v okolju Pišek, ki uporablja Blockly, ali v Blockly editorju. Zanimalo me je, katere naloge bodo učencem povzročale največ težav.

4.2 Postavitev hipotez

1. hipoteza: »Učitelji računalništva na osnovni šoli pri neobveznih in obveznih izbirnih predmetih učijo programiranja pri svojih urah.«
2. hipoteza: »Profesorjem na fakulteti in učiteljem računalništva na osnovnih šolah se za začetek programiranja zdita najboljša slikovna programa Scratch ter Pišek in za nadaljevanje programski jezik oziroma Python.«
3. hipoteza: »Vsi učenci neobveznega izbirnega predmeta računalništvo (4., 5., 6. razred) bodo opazili velik preskok težavnosti iz 5. na 6. nalogo v Pišku, vendar pa jo bodo starejši učenci rešili bolje.«
4. hipoteza: »Nadarjeni učenci, ki obiskujejo pouk nadarjenih za naravoslovje v 9. razredu, bodo naloge v Blockly editorju reševali bolje od učencev v 8. razredu.«
5. hipoteza: »Učencem bo lažje reševati naloge v Pišku, kot v Blockly editorju, saj imajo v Pišku le določene ukaze ter omejeno število delčkov.«

4.3 Raziskava

4.3.1 Raziskovalna metoda dela

Raziskovalna metoda dela je analiza podatkov, ki sem jih dobila iz odgovorov intervjuvancev in analiza odgovorov, ki sem jih dobila iz anketnih vprašalnikov učiteljev, ki poučujejo računalniške izbirne predmete ter sestavljanje nalog in opazovanje učencev pri reševanju.

4.3.2 Vzorec

V raziskavi so sodelovali učenci 4., 5. in 6. razreda, ki obiskujejo neobvezni izbirni predmet računalništvo (v nadaljevanju NRA) in učenci 8. in 9. razreda, ki obiskujejo pouk za nadarjen za naravoslovje.

4.3.3 Naloge

Za učenci, ki obiskujejo NRA, sem najprej izbrala naloge, ki so objavljene na Piškovi strani: 1. šolsko tekmovanje – skupina OŠ 4. – 6. r. - začetniki. V nadaljevanju sem sestavila svoje naloge, ki so jih reševali v urejevalniku (editorju) programa Blockly.

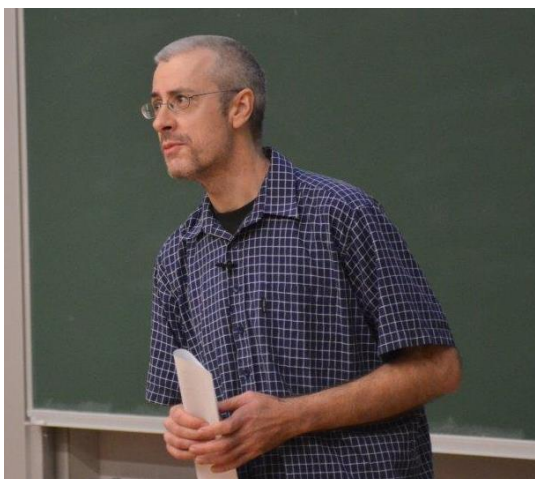
Učenci, ki obiskujejo pouk za nadarjene za naravoslovje (8. in 9. razred) so naloge, ki sem jih sestavila, programirali v urejevalniku (editorju) programa Blockly.

5 RAZISKAVA

5.1 Intervjuji

5.1.1 Intervju z gospodom dr. Janezom Demšarjem, profesorjem na Fakulteti za računalništvo in Pedagoški fakulteti v Ljubljani

Prof. dr. Janez Demšar je leta 1996 končal študij na Fakulteti za računalništvo in informatiko in tam je tudi delal kot mladi raziskovalec. Leta 2002 je na tej fakulteti opravil tudi doktorat. Od leta 1996 je zaposlen v laboratoriju za umetno inteligenco, na Fakulteti za računalništvo in informatiko pa je docent in predava več predmetov. Aplikativno se ukvarja tudi s problemi v medicini in genetiki.



Slika 11: Prof. dr. Janez Demšar.

Intervju

1. Prosim, če se najprej na kratko predstavite.

Janez Demšar, profesor na Fakulteti za računalništvo in Pedagoški fakulteti Univerze v Ljubljani. Učim programiranje ter algoritme in podatkovne strukture. Veliko programiram, moje osnovno raziskovalno področje pa je umetna inteligenca oz. analiza podatkov. Pripravljam tudi različne materiale za osnovne in srednje šole. Prevedel in priredil sem CS Unplugged (po slovensko Vidra), ki se uporablja za poučevanje računalništva brez računalnika, trenutno pa vodim razvoj materialov za vključevanje umetne inteligence v različne šolske predmete. Rad grem tudi v kako osnovno ali srednjo šolo in tam kakšno uro kaj "učim".

2. Ali menite, da je učenje programiranja v osnovni šoli pomembno? Zakaj?

Da. Ko se učimo programirati, se učimo razmišljati. Obstajajo različne vrste razmišljanja (o čemer si ne upam veliko pisati, saj nisem psiholog), ampak takrat, ko programiraš, moraš razmišljati sistematično, predvideti različne možne scenarije, dobro opisati postopke, ki jih mora izvajati računalnik... To so veščine, ki pridejo prav tako pravniku kot gozdarju, ne samo kasnejšemu profesionalnemu programerju.

3. Kdaj naj bi se učenci po vašem mnenju začeli učiti programiranja?

V nekem smislu kadarkoli. Tudi zlaganje lego kock je neka vrsta programiranja. Kocke gredo skupaj na tak in tak način, na nekega drugega pa ne. Če jih zložimo narobe, bo vse skupaj razpadlo (ali pa ne bo šlo skupaj). Torej: nekako pri dveh letih. :) Bolj formalne oblike, sploh take z računalnikom, pa seveda kasneje.

4. Kateri program se vam zdi najprimernejši za začetek programiranja?

Glej prejšnje vprašanje. :) Letom primerno. Zato bom vprašanje obrnil drugače: če vas zanima programiranje za računalnikom (ali tablico), to zahteva, da učenec zmore uporabiti računalnik in da imamo program, ki je primeren za ta leta. Najbolj zgođen, ki sem ga kdaj videl, je Scratch Jr., s katerim se lahko začnejo igrati že pred šolo. Kasneje pa seveda "pravi Scratch". Po mojih izkušnjah ta postane najkasneje v pol leta malo dolgočasen. Potem imam pa dilemo: Python ali JavaScript. Python je prikladnejši kot jezik, ker ima preprostejšo ... ehm, sintakso. JavaScript ima kup navlake, kot so razni zaviti oklepaji in podpičja, tako da se mora programer-začetnik ukvarjati s takšnimi detajli namesto s programom. Po drugi strani je JavaScript privlačnejši, ker so rezultati teh programov, ki tečejo v brskalniku, bolj "atraktivni". V Pythonu je težko zastavljati zabavne naloge za osnovnošolce. (V razne projekte, kjer programirajo igre v Pythonu, ne verjamem, ker je tam preveč že sprogramiranega, pripravljenega vnaprej, in morajo učenci potem stalno klicati neke "magične funkcije", ki jih ne razumejo. V vsakem trenutku, ko postanejo zadeve prezapletene, vstopi nek deus ex machina.)

5. V katerih programih učite programiranja?

V Pythonu. Vendar učim študente. Ko sem se to šel v osnovni šoli, pa sem poskušal Scratch, Python in celo C (z mikrokontrolerjem Arduino). Vsakega po pol leta. Izmed teh je bil, zanimivo, najmanj prijeten Python, prav zato, ker si nisem domislil dobrih nalog. C je bil, kot jezik, naporen, vendar je tam odtehtalo to, da smo imeli mikrokontrolerje, na katere smo lahko priklapljali lučke, zvočnike, mikrofone... Morda je prava kombinacija BBC-jev Micro:Bit ali kakšen drug mikrokontroler, ki mogoča programiranje v (Micro)Pythonu.

6. Kakšna je razlika med programom Scratch in programom Pišek?

Piška praktično ne poznam, zato ne morem komentirati.

7. Zakaj potreba o programu Pišek narašča?

Glej prejšnje vprašanje.

8. Kaj bi svetovali mladim programerjem?

Kaj bi svetoval mladim tekačem? Naj čim več tečejo, da se ustrezne mišice ustrezno okrepijo. In naj tečejo po naravi, ne na stadionu, tako da bodo uživali v teku. Mladim programerjem pa bi svetoval, naj čim več programirajo, da se ustrezne možganske mišice ustrezno okrepijo. In naj programirajo tam in takšne stvari, ki jim bodo v veselje.

5.1.2 Intervju z gospodom mag. Matijem Lokarjem, višjim predavateljem na Fakulteti za matematiko in fiziko v Ljubljani

Mag. Matija Lokar je višji predavatelj na Fakulteti za matematiko in fiziko, Univerze v Ljubljani. Po osnovni izobrazbi je matematik, magistriral pa je iz računalništva. Na fakulteti v zadnjih letih predava predvsem predmete s področja programiranja in podatkovnih struktur ter algoritmov, veliko pa se ukvarja z izobraževanjem učiteljev na študijskih programih izobraževanja iz računalništva in informatike, pri gradnji e-gradiv. V zadnjem času veliko časa posveča tudi vpeljavi programiranja z delčki (sistem Pišek) v osnovne in srednje šole.



Slika 12: Mag. Matija Lokar.

Intervju

1. Prosim, če se najprej na kratko predstavite.

Sem Matija Lokar, višji predavatelj na Fakulteti za matematiko in fiziko, Univerze v Ljubljani. Po osnovni izobrazbi sem matematik, magistriral pa sem iz računalništva. Na fakulteti v zadnjih letih predavam predvsem predmete s področja programiranja in podatkovnih struktur ter algoritmov. Pred tem pa sem poučeval tudi predmete o uporabi računalniških orodij v matematiki. Sodelujem pri različnih projektih s področja usposabljanja učiteljev in profesorjev z osnovnih in srednjih šol na področju računalništva. Sodelujem tudi pri pripravi različnih tekmovanj s področja računalništva. Sem pobudnik postavitve Sistema Pišek, spletne storitve, ki je namenjena pomoči pri prvih korakih v programiranju in vodja skupine, ki pripravlja ACM Tekmovanje v programiranju z delčki Pišek.

2. Ali menite, da je učenje programiranja v osnovni šoli pomembno? Zakaj?

Menim, da v sodobnem svetu poznavanje osnov algoritmičnega in računalniškega mišljenja nasploh spada (poleg računanja, branja in pisanja) med štiri osnovne sposobnosti, ki jim mora osvojiti prav vsak osnovnošolec. In učenje programiranja je zelo dober način spoznavanja in osvajanja teh osnov.

3. Kdaj naj bi se učenci po vašem mnenju začeli učiti programiranja?

Kot kažejo izkušnje, določene korake v programiranje lahko uspešno predstavimo že najmlajšim. Tako so v vrtcu, ki ga obiskujejo moje vnučke, na zanimiv način uporabljali Cubetto, robotka – kocko, katerega premikanje programiramo z zlaganjem ustreznih ploščic na "programsko desko". Obstajajo tudi različni programi, ki pa tečejo na računalnikih. Za slednje pa je po moje smiselno, da jih začno uporabljati "še" tam nekje v tretjem razredu.

4. Kateri program se vam zdi najprimernejši za začetek programiranja?

Kot lahko sklepate iz prejšnjega odgovora, menim, da ne obstaja tak jezik, ki bi bil najprimernejši. Odvisno je od starosti začetnika, pa tudi od tega, ali se s prvimi koraki »spopada« sam, ali v šoli, z vrstniki, kaj ga zanima tudi drugače. Tako npr. obstajajo programski jeziki s katerimi krmilite robote, jeziki s katerimi načrtujete videz oblačila, jeziki, s katerimi gradite 3D modele ... Za osnovnošolce pa so zagotovo za začetek zelo primerni jeziki, ki omogočajo programiranje z delčki. Med njimi sta tudi Scratch in jezik, s katerim delamo s Piškom.

5. V katerih programih učite programiranja?

V svoji praksi sem programiranje učil v številnih jezikih, verjetno za mnoge med vami niste še slišali (bodisi, ker so tako posebni, bodisi ker so že tako stari) : FORTRAN, Pascal, Deplhi, Prolog, LISP, C, Java, SQL, različne oblike BASICA, Logo, Snap, Scratch, Mathematica, GeoGebraScript, Blockly, ... V

zadnjem času pa so to predvsem C#, SQL, Blockly in Python ... Vendar kateri jezik predvsem za prve korake niti ni pomembno. Pomembno je razumeti osnovne koncepte: zaporedje ukazov, vejitev, zanka ...

6. Kakšna je razlika med programom Scratch in programom Pišek?

Scratch kot tak je programski jezik, med tem ko je Pišek okolje v katerem programiramo. Pri Pišku uporabljamo ukaze, ki nastajajo na osnovi Blocklya. No, tudi na Scratch lahko gledamo kot okolje, v katerem pišemo programe v Scratchu. Podobna sta si v toliko, da pri obeh za pisanje programov uporabljamo delčke. V svojem bistvu pa se precej razlikujeta. Scratch je predvsem okolje za izražanje.

Tam se učenci s pomočjo programiranja na določen način izražajo. Pri Pišku pa je osnovni namen reševanje različnih zabavnih (vsaj upamo tako) nalog, kjer nam Pišek tudi pove, če je naloga uspešno rešena.

7. Zakaj potreba o programu Pišek narašča?

No, tega podatka, da potreba po Pišku narašča, si ne bi upal trditi. Kot enega od začetnikov postavitev Piška bi me to seveda veselilo. Res je, da je vedno več učencev (in učiteljev), ki vedo za Sistem Pišek in ga tudi uporabljajo. Zato tudi poskušamo, da bi v Piška dodali nove naloge, nove možnosti in ga naredili še bolj zanimivega in uporabnega. Upamo, da bomo med poletnimi počitnicami zagnali prenovljenega Piška z novimi nalogami, novimi zmožnostmi in novim, bolj privlačnim videzom.

8. Kaj bi svetovali mladim programerjem?

Predvsem to, da je programiranje koristna, pa tudi zelo zabavna dejavnost. Predvsem naj jih programiranje zabava. Bolj ko ga bodo spoznali, številčnejše nove možnosti se jim bodo odpirale in omogočale početi zanimivejše stvari. Je pa programiranje tudi veščina, kot je na primer vožnja s kolesom. In zato tudi za programiranje velja: vaja dela mojstra!

Pri Pišku pa - največja nagrada, največji dosežek je to, da ti Pišek pove, da si nalogo uspešno rešil!

5.2 Ankete

5.2.1 Anketa za učitelje

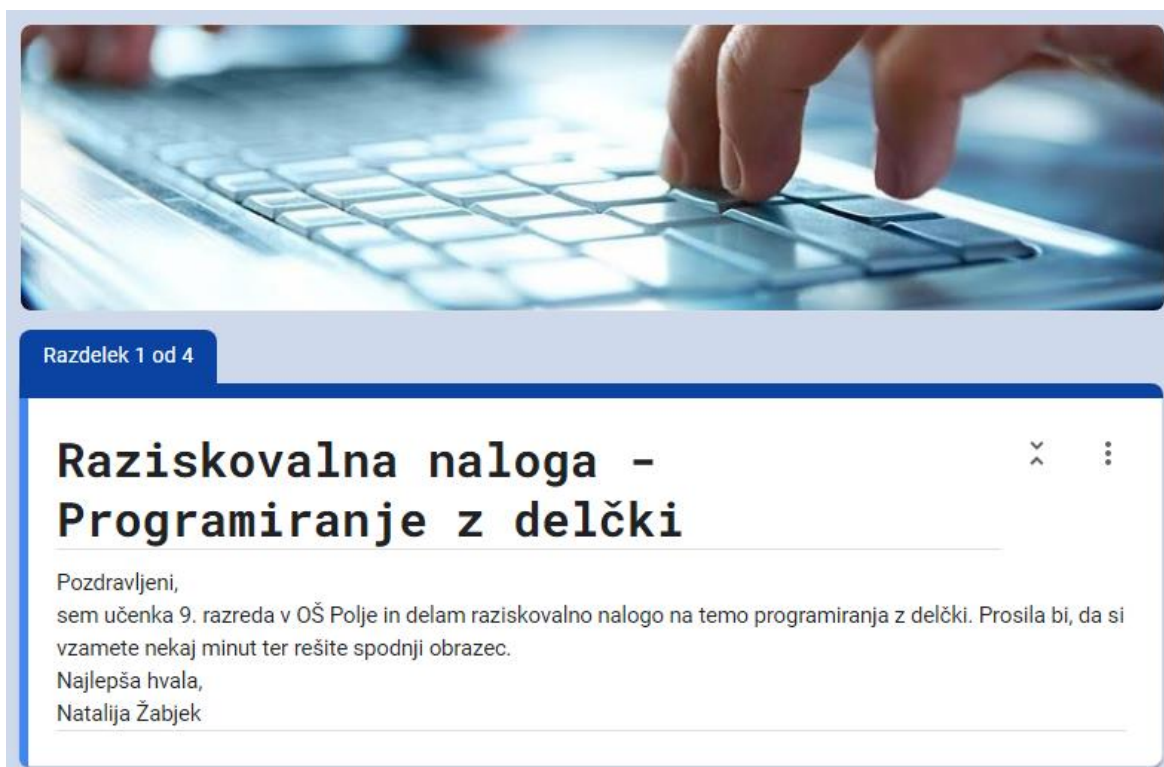
Z anketnim vprašalnikom sem želela ugotoviti, ali se učenci v osnovni šoli seznaniijo s programiranjem in ali pri pouku tudi programirajo z delčki.

Zanimalo me je:

- ali v osnovnih šolah učitelji poučujejo neobvezni izbirni predmet računalništvo ali obvezne izbirne predmete (UBE, MME, ROM),
- ali v osnovnih šolah učitelji pri svojih urah pouka računalniških vsebin učijo programiranja,
- katere programe pri tem uporabljajo,
- če se učiteljem računalništva zdi programiranje v osnovni šoli potrebno ter zakaj tako menijo,
- kako se po njihovem mnenju lahko otrok hitreje in učinkovitejše nauči programiranja,
- kateri program se jim zdi najprimernejši za začetek programiranja,
- kdaj naj bi se učenci začeli učiti programiranja ter
- kaj bi učitelji računalništva na osnovnih šolah svetovali mladim programerjem.

Glede na ta vprašanja, sem sestavila obrazec, ki sem ga razposlala med učitelje računalništva v osnovnih šolah. Vprašalnik so učitelji lahko reševali od 13. 1. 2023 do 4. 2. 2023. Kljub večkratnim pozivom, se je le malo učiteljev odzvalo in vprašalnik tudi rešilo.

Pridobila sem le 9 odgovorov.



Slika 13: Zaslonska slika obrazca za učitelje računalništva.

Vprašalnik je bil sestavljen tako, da so učitelji v prvem razdelku odgovorili na nekaj splošnih vprašanj:

- ime in priimek,
- starost,
- spol,
- smer izobrazbe,
- šolo, na kateri poučujejo,
- število let poučevanja ter
- katere računalniške predmete poučujejo (obvezne izbirne, neobvezni izbirni ali oboje).

(Če osebnih podatkov niso želeli izpolniti, to ni bilo potrebno.)

Glede na to, kaj so označili pri zadnjem vprašanju: »Katere računalniške predmete poučujete (obvezne izbirne, neobvezne izbirni ali oboje)?«, so nadaljevali na določen naslednji razdelek. Po osnovnem razdelku, sta sledila še dva. Eden je bil namenjen učiteljem, ki poučujejo obvezne izbirne predmete in eden tistim, ki poučujejo neobvezne izbirne predmete.

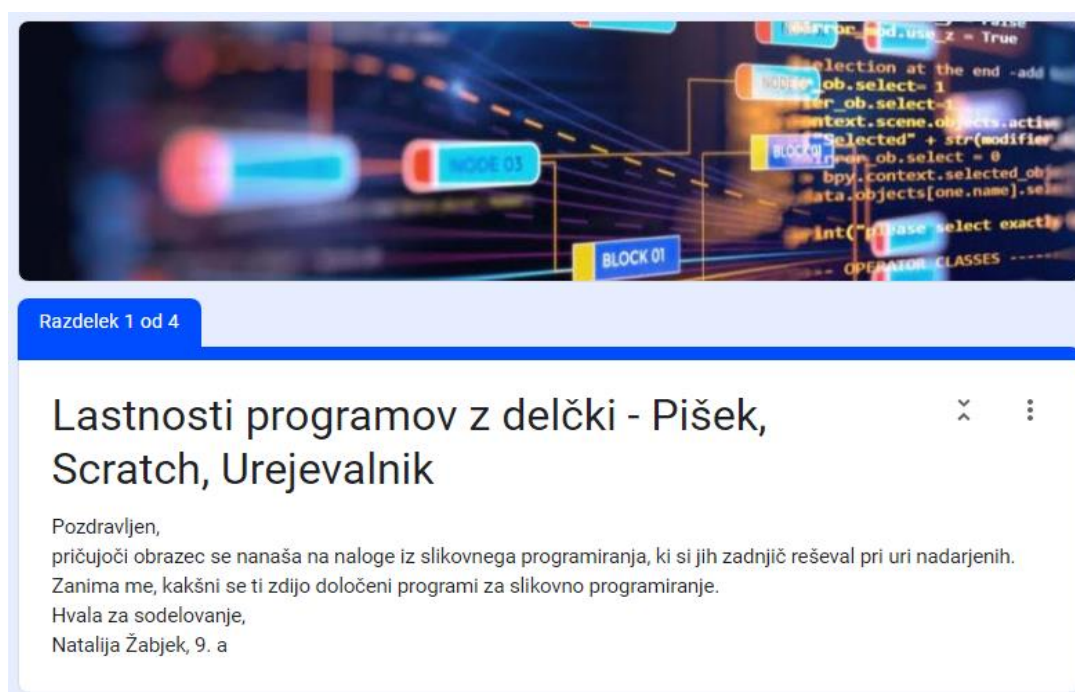
Kdor je označil »oboj« je odgovoril na oba razdelka.

5.2.2 Anketa za učence

Med opazovanjem nadarjenih učencev 8. in 9. razreda na področju naravoslovja, se mi je utrnila zamisel, da bi jih povprašala še o njihovem mnenju o določenih programih za slikovno programiranje. Do sedaj so bili seznanjeni s Scratchom, Piškom, v okviru moje raziskave pa tudi z Blockly editorjem.

Poslala sem jim obrazec, o prednostih in slabostih zgoraj naštetih programov.

Učenci so obrazec lahko reševali od 30. 1. 2023 do 8. 2. 2023, vendar kljub opominjanju nisem dobila več kot 8 odgovorov.



Slika 14: Zaslonska slika obrazca za učence.

Obrazec je bil sestavljen tako, da so v prvem razdelku učenci napisali osnovne podatke (ime in priimek ter razred).

Sledil je drugi razdelek v katerem sem jih povprašala, če kdaj programirajo in kako raje programirajo. V zadnjem razdelku pa sem jih prosila, naj napišejo prednosti in slabosti Scratch-a, Piška ter Blockly editor-ja.

5.3 Spremljanje reševanja nalog učencev pri neobveznem izbirnem predmetu računalništvo

V sklopu raziskave sem si zadala nalogo, da spremljam učence pri njihovem reševanju nalog. Odločila sem se, da bodo naloge reševali učenci, ki prvo leto obiskujejo neobvezni izbirni predmet računalništvo in tudi tisti, ki ga obiskujejo že drugo leto. Učenci pri pouku programirajo v Scratchu.

Za učence sem izbrala 1. šolsko tekmovanje – skupina OŠ 4. – 6. r. - začetniki.

TEKMOVANJE:



Slika 15: Zaslonske slike programa Pišek.

NALOGE:



Slika 16: Zaslonska slika nalog v Pišku.

NALOGE:

1. NALOGA: *Gusarji iščejo zaklad*

Učna pot > Stara tekmovanja > 1. šolsko tekmovanje > OŠ 4. do 6. r. začetniki > Gusarji iščejo zaklad

Gusarji iščejo zaklad

Gusarji so se znašli sredi morja, kjer je veliko otokov. A le na enem je zaklad. Sestavi program, ki jih bo pripeljal do pravega otoka, da bodo odšli domov z zakladom.



premakni se desno

premakni se levo

premakni se dol

premakni se gor

Program

Navigation controls: back, play, pause, forward, and a slider for speed.

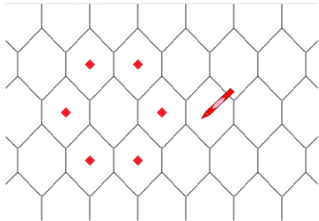
Slika 17: Zaslonska slika 1. naloge v Pišku.

2. NALOGA: *Neža barva*

Učna pot > Stara tekmovanja > 1. šolsko tekmovanje > OŠ 4. do 6. r. začetniki > Neža barva

Neža barva

Neža ima zelo rada pobarvanke, še posebno tiste, kjer so priložena navodila, ki točno povedo, kako naj barva. Žal pa so tokratna navodila pomešana. Pomagaj ji in zloži delčke tako, da bo pobarvala polja z rdečimi oznakami.



Dejanja

Program

pobarvaj rdeče

premakni voščenko levo

pobarvaj rdeče

premakni voščenko levo-dol

premakni voščenko levo

premakni voščenko levo-gor

pobarvaj rdeče

premakni voščenko desno

pobarvaj rdeče

pobarvaj rdeče

premakni voščenko desno-dol

pobarvaj rdeče

Navigation controls: back, play, pause, forward, and a slider for speed.

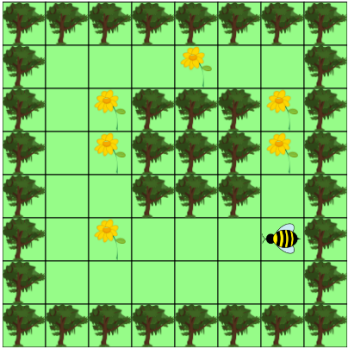
Slika 18: Zaslonska slika 2. naloge v Pišku.

3. NALOGA: Čebelica Rozi oprahuje

Učna pot > Stara tekmovanja > 1. šolsko tekmovanje > OŠ 4. do 6. r. začetniki > Čebelica Rozi oprahuje

Čebelica Rozi oprahuje

Čebelica Rozi mora oprahuje vse rože na travniku. Priskoči ji na pomoč. Čebelica oprahuje rožo tako, da poleti na njeno polje.



Dejanja
Zanke

- obrne se desno
- obrne se levo
- poleti naprej

Dejanja
Zanke

- ponavljaj 10 krat
- izvedi
- ponavljaj medtem ko
- izvedi
- za i od 1 do 10 s korakom 1
- izvedi
- zapusti zanko

Program

Delčki na voljo: 16

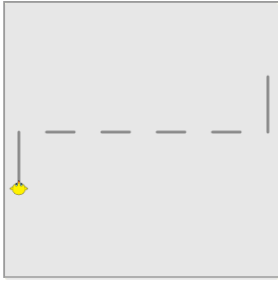
Slika 19: Zaslonska slika 3. naloge v Pišku.

4. NALOGA: Pišek riše

Učna pot > Stara tekmovanja > 1. šolsko tekmovanje > OŠ 4. do 6. r. začetniki > Pišek riše

Pišek riše

Pišek želi postati slikar. Pomagaj mu narisati preprost vzorec iz črt dolgih 10 oz. 20 enot. Na voljo imaš omejeno število delčkov, zato spretno uporabi zanko.



Pojdi naprej za 1

zavij levo 90

zavij desno 90

dvigni pisalo

spusti pisalo

ponavljaj 10 krat

izvedi

Program

Delčki na voljo: 16

Slika 20: Zaslonska slika 4. naloge v Pišku.

5. NALOGA: Zajček pobira korenčke

Zajček pobira korenčke

Zajček lahko na svojem sprehodu naleti na različne predmete. Ker je zajec, naj na svoji poti pobira samo korenčke. Bodi pozoren, naloga ima več testov.

```

    premakni se naprej
    poberi korenček
    je na polju korenček
    ponavljaj 10 krat
      izvedi
    če
      izvedi
  
```

TEST 1 TEST 2 TEST 3

Delčkov na voljo: 6

Slika 21: Zaslonska slika 5. naloge v Pišku.

6. NALOGA: Vrt sončnic

Vrt sončnic

Pišek obožuje sončnice, zato se je odločil, da si bo zasadil vrt sončnic v obliki ure. Stoji na sredini vrta. Popravi in dopolni program tako, da bo spretno zasadil dvanajst sončnic, ki bodo vse od središča vrta oddaljene enako (100 korakov). Stebla so dolga 60 korakov. Upoštevaj, da če pišek naredi naprej -10 korakov, da to pomeni, da je šel nazaj 10 korakov.



```

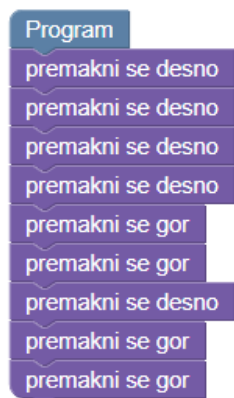
Program
nastavi barvo: [red]
ponavljaj 10 krat
izvedi
  dvigni pisalo
  posadi sončnico
  pojdí naprej za 15
  pojdí naprej za 15
  spusti pisalo
  zavij za 30° levo

```

Slika 22: Zaslonska slika 6. naloge v Pišku.

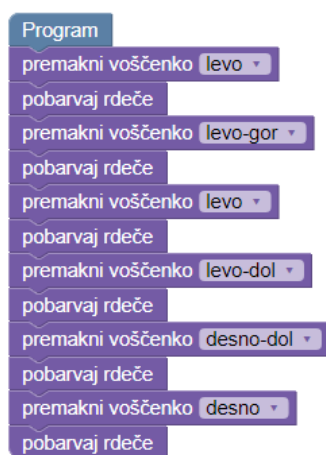
REŠITVE (moje):

1. NALOGA: Gusarji iščejo zaklad



Slika 23: Zaslonska slika rešitve 1. naloge.

2. NALOGA: Neža barva



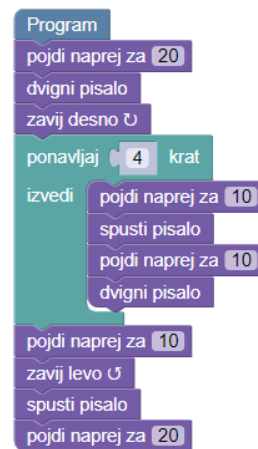
Slika 24: Zaslonska slika rešitve 2. naloge.

3. NALOGA: Čebelica Rozi oprahuje



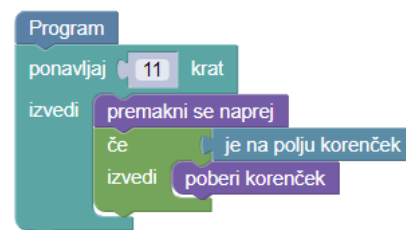
Slika 25: Zaslonska slika rešitve 3. naloge.

4. NALOGA: Pišek riše



Slika 26: Zaslonska slika rešitve 4. naloge.

5. NALOGA: Zajček pobira korenčke



Slika 27: Zaslonska slika rešitve 5. naloge.

6. NALOGA: Vrt sončnic



Slika 28: Zaslonska slika rešitve 6. naloge.

5.3.1 Rezultati učencev, ki prvo leto obiskujejo neobvezni izbirni predmet računalništvo

Sodelovali so učenci 4. razreda. Učenci pri pouku programirajo v Scratchu. V nadaljevanju so označeni s številkami od 1 do 3.

Merila sem čas reševanja, napake, ali so pri reševanju potrebovali pomoč in kako zahtevna se jim je zdela naloga (od 1 – najmanj zahtevna do 10 – najbolj zahtevna). Pred reševanjem nalog sem si zapisala predvidene rezultate.

1. NALOGA: *Gusarji iščejo zaklad*

- Čas: 1: 4min 2: 5min 3: 5min
- Težavnost 1-10: 1: 1 2: 2 3: 3
- Napake (v programu): 1: 3 2: 4 3: 4
- Pomoč (0, 1, 2): 1: 0 2: 0 3: 0

predvidevanje: 1-5min

predvidevanje: 1-4

predvidevanje: 0-4

predvidevanje: 0

2. NALOGA: *Neža barva*

- Čas: 1: 5min 2: 3min 3: 7min
- Težavnost 1-10: 1: 1 2: 1 3: 8
- Napake (v programu): 1: 4 2: 1 3: 5
- Pomoč (0, 1, 2): 1: 0 2: 0 3: 1

predvidevanje: 1-5min

predvidevanje: 1-4

predvidevanje: 0-5

predvidevanje: 0

3. NALOGA: *Čebelica Rozi oprahuje*

- Čas: 1: 6min 2: 7min 3: 6min
- Težavnost 1-10: 1: 2 2: 5 3: 5
- Napake (v programu): 1: 5 2: 4 3: 6
- Pomoč (0, 1, 2): 1: 0 2: 1 3: 0

predvidevanje: 3-7min

predvidevanje: 1-6

predvidevanje: 0-6

predvidevanje: 0-1

4. NALOGA: *Pišek riše*

- Čas: 1: 7min 2: 8min 3: 6min
- Težavnost 1-10: 1: 6 2: 7 3: 6
- Napake (v programu): 1: 7 2: 8 3: 5
- Pomoč (0, 1, 2): 1: 2 2: 2 3: 1

predvidevanje: 3-8min

predvidevanje: 2-6

predvidevanje: 1-8

predvidevanje: 0-2

5. NALOGA: *Zajček pobira korenčke*

- Čas: 1: 8min 2: 9min 3: 7min
- Težavnost 1-10: 1: 7 2: 7 3: 9
- Napake (v programu): 1: 6 2: 5 3: 7
- Pomoč (0, 1, 2): 1: 2 2: 2 3: 2

predvidevanje: 3-9min

predvidevanje: 3-7

predvidevanje: 0-7

predvidevanje: 0-2

6. NALOGA: *Vrt sončnic*

- Čas: 1: / 2: / 3: /
- Težavnost 1-10: 1: 9 2: 9 3: 10
- Napake (v programu): 1: 20+ 2: 20+ 3: 20+
- Pomoč (0, 1, 2): 1: 2 2: 2 3: 2

predvidevanje: 8-20min

predvidevanje: 6-10

predvidevanje: 3-20

predvidevanje: 0-2



Slika 29: Učenci – začetniki pri programiranju v računalniški učilnici.



Slika 30: Učenci – začetniki pri programiranju v računalniški učilnici.

5.3.1 Rezultati učencev, ki že drugo leto obiskujejo neobvezni izbirni predmet računalništvo

Dva učenca sta bila učenca 5. razreda, en pa šestošolec. Učenci pri pouku programirajo v Scratchu. Učenci so označeni s številkami od 1 do 3.

Merila sem čas reševanja, napake, ali so pri reševanju potrebovali pomoč in kako zahtevna se jim je zdela naloga (od 1 – najmanj zahtevna do 10 – najbolj zahtevna). Pred reševanjem nalog sem si zapisala predvidene rezultate.

NALOGA: *Gusarji iščejo zaklad*

- Čas: 1: 2min 2: 2 min 3: 1 min *predvidevanje: 1-3min*
- Težavnost 1-10: 1: 1 2: 1 3: 1 *predvidevanje: 1-3*
- Napake (v programu): 1: 0 2: 0 3: 0 *predvidevanje: 0-2*
- Pomoč (0, 1, 2): 1: 0 2: 0 3: 0 *predvidevanje: 0*

1. NALOGA: *Neža barva*

- Čas: 1: 2min 2: 3min 3: 1min *predvidevanje: 1-4min*
- Težavnost 1-10: 1: 1 2: 2 3: 2 *predvidevanje: 1-3*
- Napake (v programu): 1: 0 2: 2 3: 0 *predvidevanje: 0-3*
- Pomoč (0, 1, 2): 1: 0 2: 0 3: 0 *predvidevanje: 0*

2. NALOGA: *Čebelica Rozi oprašuje*

- Čas: 1: 5min 2: 7min 3: 3min *predvidevanje: 3-5min*
- Težavnost 1-10: 1: 2 2: 4 3: 2 *predvidevanje: 1-5*
- Napake (v programu): 1: 2 2: 3 3: 3 *predvidevanje: 0-4*
- Pomoč (0, 1, 2): 1: 0 2: 0 3: 0 *predvidevanje: 0-1*

3. NALOGA: *Pišek riše*

- Čas: 1: 6min 2: 8min 3: 4min *predvidevanje: 3-7min*
- Težavnost 1-10: 1: 3 2: 5 3: 2 *predvidevanje: 2-6*
- Napake (v programu): 1: 4 2: 6 3: 3 *predvidevanje: 1-6*
- Pomoč (0, 1, 2): 1: 0 2: 0 3: 0 *predvidevanje: 0-2*

4. NALOGA: *Zajček pobira korenčke*

- Čas: 1: 7min 2: 9min 3: 6min *predvidevanje: 3-8min*
- Težavnost 1-10: 1: 3 2: 4 3: 2 *predvidevanje: 2-6*
- Napake (v programu): 1: 5 2: 8 3: 3 *predvidevanje: 0-4*
- Pomoč (0, 1, 2): 1: 2 2: 0 3: 0 *predvidevanje: 0-2*

5. NALOGA: *Vrt sončnic*

- Čas: 1: 10min 2: / 3: / *predvidevanje: 6-15min*
- Težavnost 1-10: 1: 6 2: 10 3: 10 *predvidevanje: 4-8*
- Napake (v programu): 1: 15+ 2: 15+ 3: 15+ *predvidevanje: 3-15*
- Pomoč (0, 1, 2): 1: 2 2: 1 3: 2 *predvidevanje: 0-2*



Slika 31: Učenci – napredni pri programiranju v računalniški učilnici.



Slika 32: Učenci – napredni pri programiranju v računalniški učilnici.

5.4 Sestavljanje avtorskih nalog, ki preverjajo programerske koncepte

Odločila sem se, da bom tudi sama sestavila nekaj nalog iz slikovnega programiranja za določene učence.

Sestavila sem:

- 3 naloge za učence, ki obiskujejo računalništvo prvo leto,
- 3 naloge za učence, ki se učijo računalništvo drugo ali tretje leto,
- 4 naloge za nadarjene učence v 8. in 9. razredu na področju naravoslovja ter
- možno rešitev pri vsaki nalogi.

Pri tej raziskavi je sodelovalo:

- 7 učencev, ki obiskujejo računalništvo 1. leto,
- 20 učencev, ki se učijo računalništvo 2 ali 3 leta,
- 7 nadarjenih učencev na področju naravoslovja iz 8. razreda ter
- 8 nadarjenih učencev na področju naravoslovja iz 9. razreda.

Potek:

Med uro računalništva in nadarjenih za naravoslovje so učenci sedli za računalnike ter dobili vsak svoj list z naslovom »*NALOGE V BLOCKLY EDITOR + ZGLED*«. Odpri so Blockly editor, ki je dostopen na: <https://lusy.fri.uni-lj.si/ucbenik/prog/editor.html>.

Začeli so reševati naloge. Če so potrebovali pomoč, sem jim seveda pomagala oziroma jim razložila nalogo.

Na voljo so imeli 45 minut (1 šolsko uro).

Komur na koncu ni uspelo rešiti naloge, sem jim pokazala list z rešitvami ter jim pojasnila postopek, na list pa sem si zapisala »/«.

Po uspešno opravljeni nalogi sem učence povprašala po težavnosti (1 – 10) vsake naloge ter izpolnila tabelo. Načrtovala sem, da bom beležila še, koliko pomoči je potreboval posamezni učenec (0, 1 ali 2), vendar se je izkazalo, da so vsi potrebovali veliko pomoči, zato med delom tega nisem beležila, saj med njimi ni bilo bistvenih razlik.

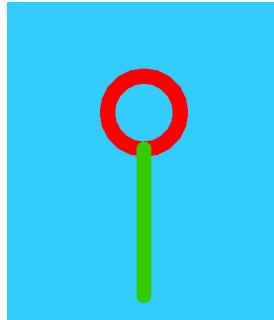
5.4.1 Reševanje nalog v Blockly editorju

Navodilo za začetnike

Naloge za začetnike (uporablaj samo ukaze *Risanje*):

1. Nastavi barvo ozadja na svetlo modro in nariši krog s polmerom 50 enot. Širina svinčnika naj bo 20 enot, barva pa rdeča. Na koncu nariši še steblo (navpično navzdol), ki naj bo zelene barve in dolgo 200 enot. (*Pomagaj si z zgledom*)

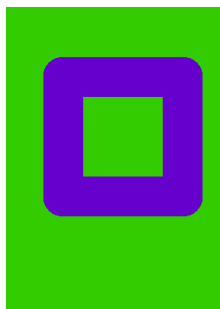
Zgled:



Slika 33: Zaslonska slika 1. naloge v Blockly editorju (začetniki).

2. Nastavi barvo ozadja na svetlo zeleno in nariši vijoličen kvadrat. Stranice naj bodo dolge 150 enot. Barva svinčnika naj bo vijolična, širina pa 50. (*Pomagaj si z zgledom*)

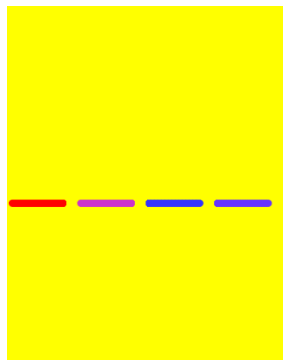
Zgled:



Slika 34: Zaslonska slika 2. naloge v Blockly editorju (začetniki).

3. Nariši spodnji vzorec na rumenem ozadju s svinčnikom, ki ima širino 10. Barve so (zaporedje ni pomembno): temno-modra, vijolična, roza, rdeča. Črte so dolge 70 enot, presledki pa 25 enot. Namig: obrneš se lahko za 90°, če želiš iti levo ali desno in 180°, če se želiš obrniti v nasprotno smer. (*Pomagaj si z zgledom*)

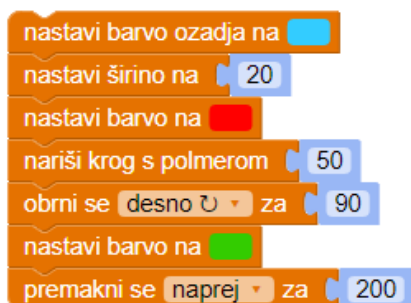
Zgled:



Slika 35: Zaslonska slika 3. naloge v Blockly editorju (začetniki).

Rešitve nalog za začetnike v Blockly editorju

NALOGA: Lahka roža



Slika 36: Zaslonska slika rešitve za 1. nalogo v Blockly editorju (začetniki).

2. NALOGA: Kvadrat



Slika 37: Zaslonska slika rešitve za 2. nalogo v Blockly editorju (začetniki).

3. NALOGA: 4 črte



Slika 38: Zaslonska slika rešitve za 3. nalogo v Blockly editorju (začetniki).

Ocena težavnosti nalog za začetnike v blockly editorju

Učenci so reševali 3 naloge (želvja grafika), ki so preverjale naslednje programerske koncepte:

1. naloga: Lahka roža – zaporedje ukazov

2. naloga: Kvadrat – zaporedje ukazov

3. naloga: 4 črte – zaporedje ukazov

Težavnost zastavljenih nalog so ocenjevali od 1 do 10. Poševnica (/) pomeni, da učenec naloge ni rešil.

Naloga	1.	2.	3.
	Lahka roža	Kvadrat	4 črte
Učenec	Težavnost	Težavnost	Težavnost
1.	8	8	10
2.	9	2	/
3.	7	2	/
4.	9	5	/
5.	7	6	/
6.	5	4	4
7.	6	4	5



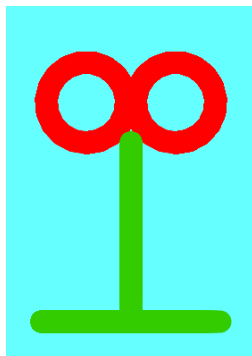
Slika 39: Učenci - začetniki pri programiranju v računalniški učilnici.

Navodilo za napredne

Naloge za napredne v Blockly editorju (uporablaj *Risanje, Zanka, Besedilo*):

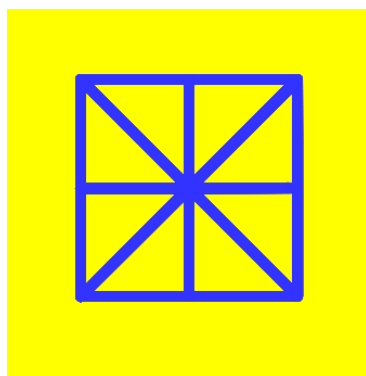
1. **RISANJE:** Nastavi svetlo-modro ozadje in nariši rožo z dvema rdečima krogoma s premerom 45 enot. Dodaj zeleno steblo in pravokotno še travo, ki naj bo dolgo 200 enot. Širina svinčnika naj bo 25 enot. (Namig: dolžina med začetkom enega kroga in začetkom drugega kroga je 100 enot) (*Pomagaj si z zgledom*)

Zgled:



Slika 40: Zaslonska slika 1. naloge v Blockly editorju (napredni).

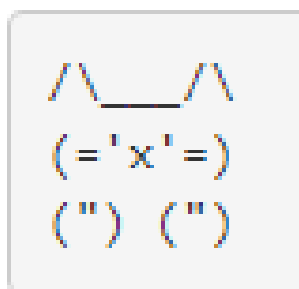
2. **RISANJE, ZANKE:** Nariši temno-modro kvadrat s snežinko v njej. Barva ozadja naj bo rumena, širina svinčnika pa 10 enot. Za kote uporablaj 90° in 135°; ena vodoravna/navpična črta je dolga 200 enot (polovica je 100), poševna pa 283 enot. (*Pomagaj si z zgledom*)



Slika 41: Zaslonska slika 2. naloge v Blockly editorju (napredni).

3. **BESEDILO:** Napiši kratek programček, ki ti bo izrisal mačko. (*Pomagaj si z zgledom*)

Zgled:



Slika 42: Zaslonska slika 3. naloge v Blockly editorju (napredni).

Rešitve nalog za napredne v Blockly editorju

1. NALOGA (RISANJE): Težja roža



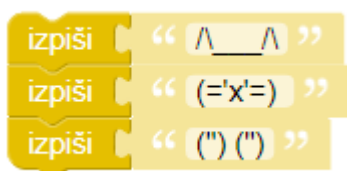
Slika 43: Zaslonska slika rešitve za 1. nalogo v Blockly editorju (napredni).

2. NALOGA (RISANJE, ZANKE): Kvadrat s snežinko



Slika 44: Zaslonska slika rešitve za 2. nalogo v Blockly editorju (napredni).

3. NALOGA (BESEDILO): Mačka



Slika 45: Zaslonska slika rešitve za 3. nalogo v Blockly editorju

Ocena težavnosti nalog za napredne v Blockly editorju

Pri nalogah so uporabljani ukaze iz nabora Risanje, Zanka in Besedilo.

1. naloga (risanje): težja roža – zaporedje ukazov
2. naloga (risanje, zanke): kvadrat s snežinko – zaporedje ukazov, ugnježena zanka
3. naloga (besedilo): mačka – izpis

Težavnost nalog so ocenjevali z ocenami 1 (naj manj težka) do 10 (najbolj težka). Poševnica (/) pomeni, da učenec naloge ni rešil.

Naloga	1.	2.	3.
	Težja roža	Kvadrat s snežinko	Mačka
Učenec	Težavnost	Težavnost	Težavnost
1.	/	/	/
2.	4	/	/
3.	8	/	/
4.	6	/	/
5.	/	/	/
6.	/	/	/
7.	/	/	/
8.	7	/	/
9.	/	/	/
10.	/	/	/
11.	2	/	/
12.	5	/	/
13.	8	/	/
14.	/	/	/
15.	/	/	/
16.	/	/	/
17.	/	/	/
18.	/	/	/
19.	/	/	/
20.	4	/	/

Navodilo za nadarjene za naravoslovje

Naloge za nadarjene za naravoslovje (8. in 9. razred) (*uporabljajo vse*)

1. S preprostim programom izpiši vsa števila do 1000.

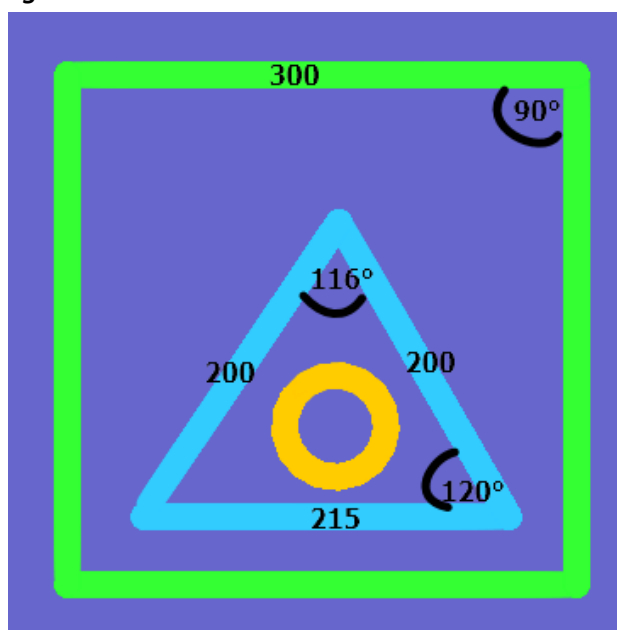
Zgled:



Slika 46: Zaslonska slika 1. naloge v Blockly editorju (nadarjeni).

2. Nariši oranžen krog s polmerom 30 enot, znotraj svetlomodelega trikotnika, znotraj zelenega kvadrata. Širina svinčnika je 15, ozadje naj bo svetlo-vijolično. Za dolžine stranic in kote si pomagaj z zgledom. (Namig: Uporabi koordinatni sistem.)

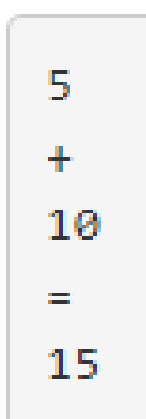
Zgled:



Slika 47: Zaslonska slika 2. naloge v Blockly editorju (nadarjeni).

3. Napiši kratek program, ki bo glede na to, kateri števili vpišemo, izračunal njuno vsoto. Program naj izpiše vse, torej: obe števili, +, = in vsoto (vsako v svojo vrstico).

Zgled:



Slika 48: Zaslonska slika 3. naloge v Blockly editorju (nadarjeni).

4. Zapiši program, ki bo glede na vpisano število, seštel vsa naravna števila do (vključno z) vpisanega števila.

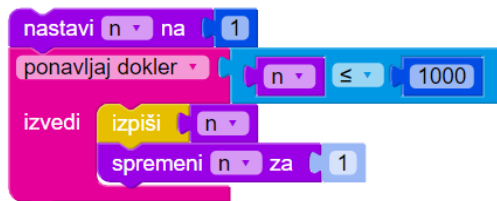
Zgled:

Vsota naravnih števil do vpisanega števila je:
15

Slika 49: Zaslonska slika 4. naloge v Blockly editorju (nadarjeni).

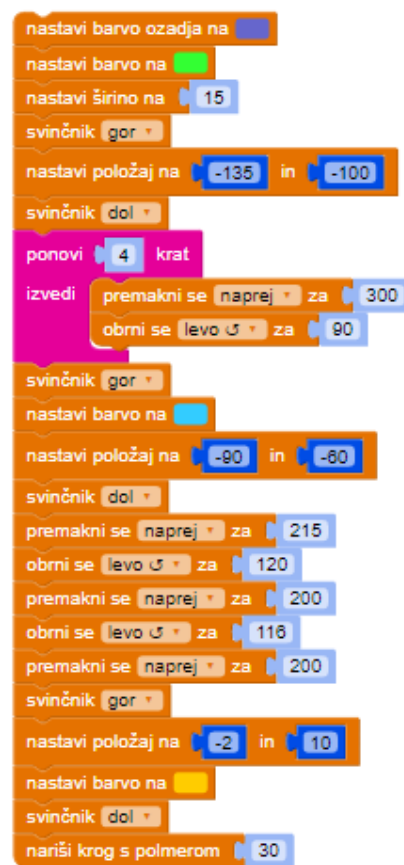
Rešitve nalog za nadarjene za naravoslovje (8. in 9. razred) (uporabljajo vse)

1. NALOGA: Števila do 1000



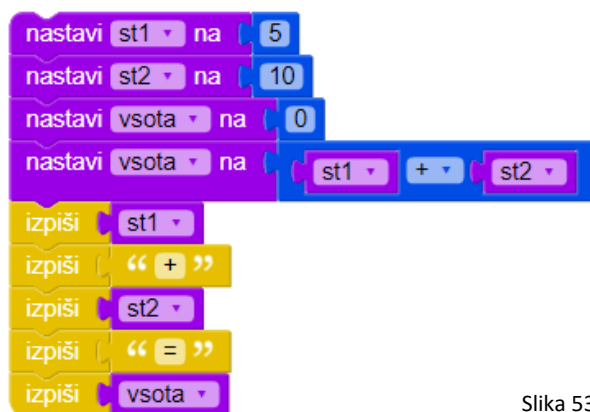
Slika 50: Zaslonska slika rešitve za 1. nalogo v Blockly editorju (nadarjeni).

2. NALOGA: Kvadrat, trikotnik, krog



Slika 51: Zaslonska slika rešitve za 2. nalogo v Blockly editorju (nadarjeni).

3. NALOGA: Vsota dveh števil



Slika 52: Zaslonska slika rešitve za 3. nalogo v Blockly editorju (nadarjeni).

4. NALOGA: Vsota naravnih števil



Slika 53: Zaslonska slika rešitve za 4. nalogo v Blockly editorju (nadarjeni).

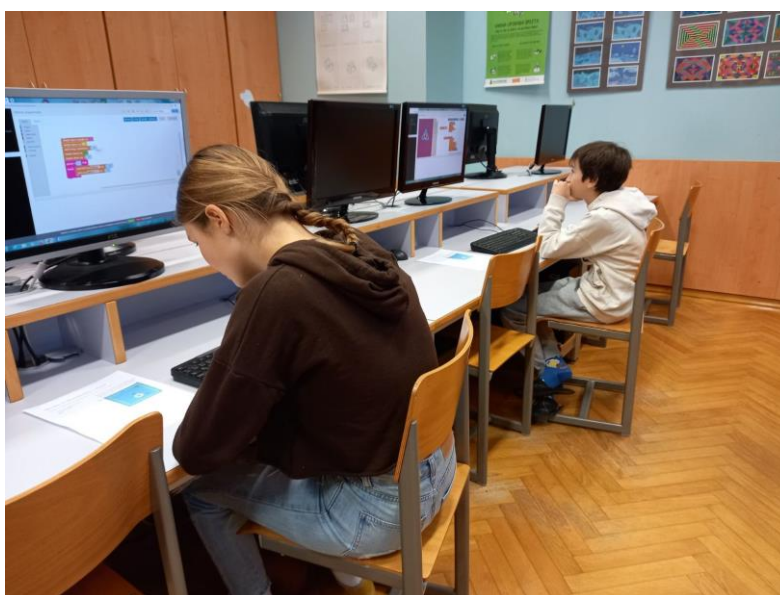
Ocena težavnosti nalog za nadarjene za naravoslovje (8. in 9. razred) (uporabljajo vse)

Učenci so reševali naloge, ki so preverjale naslednje programerske koncepte:

1. naloga: števila do 1000 – enostavna while zanka, spremenljivka, izpis podatkov
2. naloga: kvadrat, trikotnik, krog – zaporedje ukazov, for zanka (telo zanke se ponovi določeno število krat)
3. naloga: vsota dveh števil – spremenljivka, zaporedje ukazov, izpis podatkov
4. naloga: vsota naravnih števil – spremenljivka, enostavna while zanka (ponavlja dokler ...)

Težavnost nalog so ocenjevali z ocenami 1 (naj manj težka) do 10 (najbolj težka). Poševnica (/) pomeni, da učenec naloge ni rešil.

8. razred				
Naloga	1.	2.	3.	4.
	Števila do 1000	Kvadrat, trikotnik, krog	Vsota dveh števil	Vsota naravnih števil
Učenec	Težavnost	Težavnost	Težavnost	Težavnost
1.	4	7	/	/
2.	8	/	/	/
3.	5	4	/	/
4.	10	/	/	/
5.	1	8	5	/
6.	4	6	/	/
7.	10	/	/	/



Slika 54: Nadarjeni učenci 8. razreda za naravoslovje pri programiranju v računalniški učilnici.

9. razred				
Naloga	1.	2.	3.	4.
	Števila do 1000	Kvadrat, trikotnik, krog	Vsota dveh števil	Vsota naravnih števil
Učenec	Težavnost	Težavnost	Težavnost	Težavnost
1.	10	/	7	/
2.	9	8	6	/
3.	9	/	/	/
4.	9	5	/	/
5.	10	/	/	/
6.	8	5	/	/
7.	10	/	/	/
8.	8	7	/	/



Slika 55: Nadarjeni učenci 8. razreda za naravoslovje pri programiranju v računalniški učilnici.

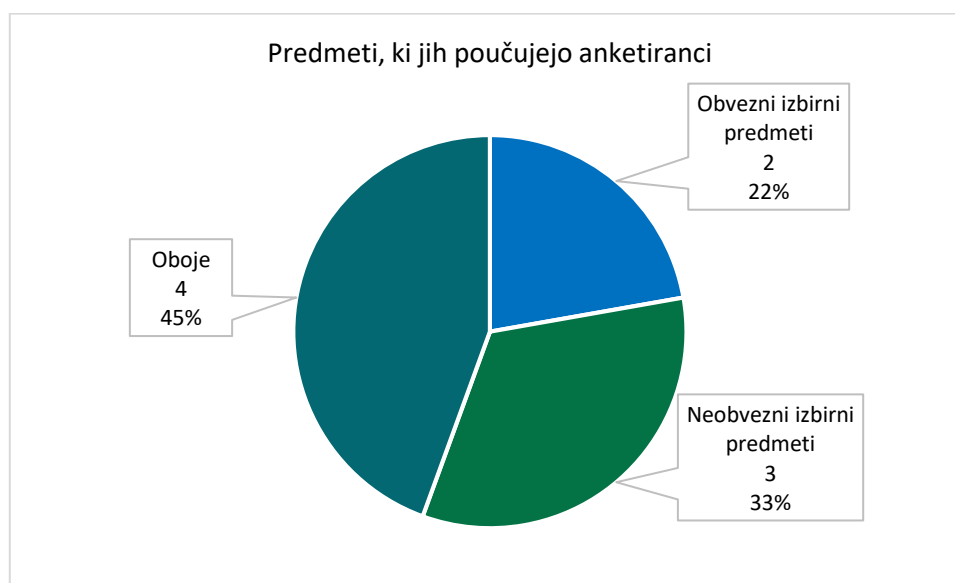
5 REZULTATI RAZISKAVE

6.1 Rezultati anket

6.1.1 Rezultati ankete za učitelje

Vprašalnik so rešili 4 učitelji in 5 učiteljic, ki so v povprečju stari od 36 do 60 let. Na osnovni šoli poučujejo računalništvo od 12 do 38 let.

Samo obvezne izbirne predmete poučujeta le dva anketiranca, medtem ko samo neobvezne izbirne predmete poučujejo trije anketiranci. Kar štirje pa poučujejo oboje.



Vsi anketiranci menijo, da je programiranje v osnovni šoli potrebno, zato ga tudi sami učijo pri pouku.

Utemeljitev anketirancev pa so bile dokaj podobne. Učitelji menijo, da se s programiranjem učenec nauči drugačnega oziroma računalniškega razmišljanja in s tem veščino bolj logičnega reševanja problemov.

Vsi učitelji v osnovnih šolah učijo programiranje z delčki, le redkokdaj omenijo pisanje ukazov (kot na primer v Pythonu).

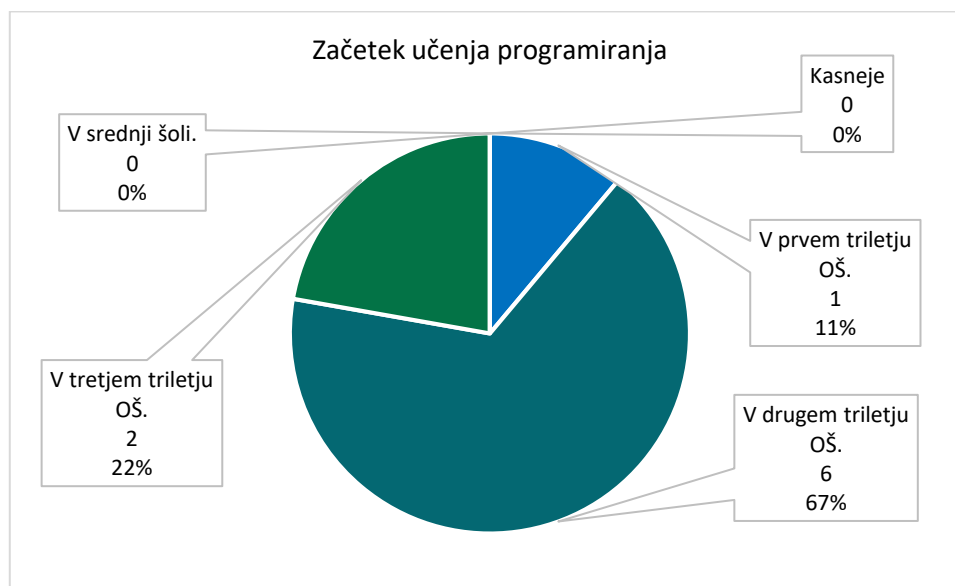
Za začetek učenja predlagajo program Scratch in Pišek oziroma kombinacijo obeh.

Pri neobveznih izbirnih predmetih učitelji večino učijo v Scratch-u, nekaj pa tudi v Pišku, mBlock-u in Micro:bit-u.

Pri obveznih izbirnih predmetih pa učitelji bolj kombinirajo programe, kot so Scratch, Pišek, HTML, Kodu, mBlock, Micro:bit ipd.

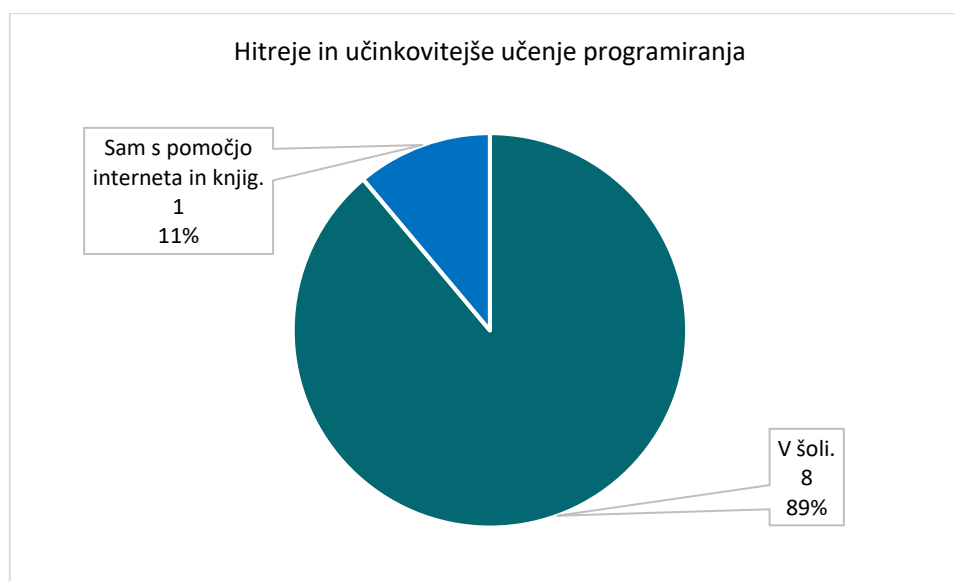
Vprašala sem jih tudi, kdaj naj bi se po njihovem mnenju učenci začeli učiti programiranja.

Vsi so odgovorili, da se morajo učenci začeti učiti programiranja v osnovni šoli. Od tega en anketiranec meni, da je to v prvem triletju, dva, da je to v tretjem triletju in ostalih šest v drugem triletju.



Zanimalo me je tudi mnenje učiteljev, kako se lahko otrok hitreje in učinkovitejše nauči programiranja.

Od vseh devetih anketirancev le eden meni, da se otrok programiranja lahko hitreje in učinkovitejše nauči sam s pomočjo interneta in knjig, drugi vztrajajo, da se v šoli naučijo več.



Preden sem se učiteljem zahvalila za sodelovanje, sem jih za zaključek povprašala še za nasvet mladim programerjem.

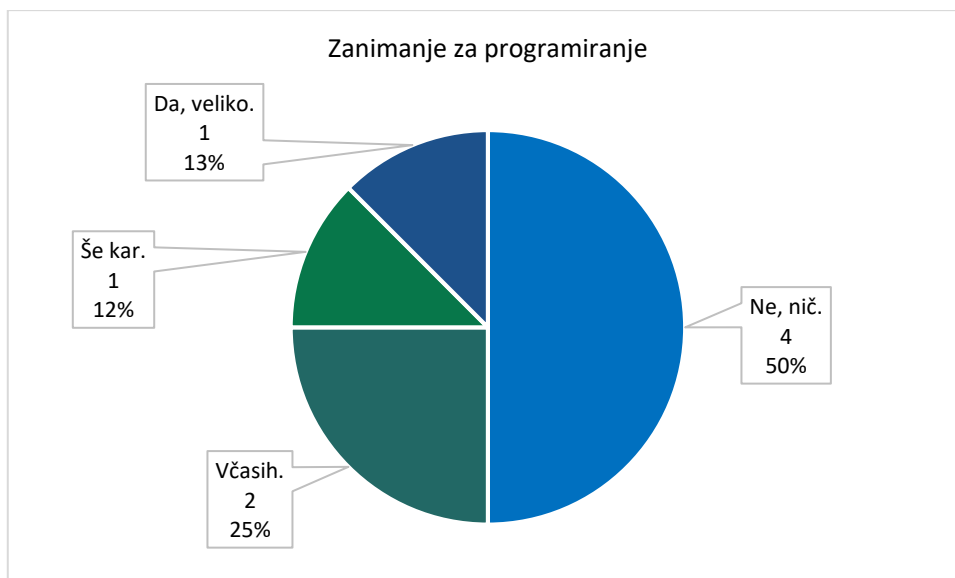
Tudi tukaj se odgovori niso dosti razlikovali. Vsi se zavedajo, da je pri programiranju močno potrebno vztrajanje, potrpežljivost in seveda veliko vaje. Svetujejo tudi gojenje iznajdljivosti, domišljije in z njo tudi radovednosti.

6.1.1 Rezultati ankete za učence

Anketni vprašalnik sem poslala 19 učencem, odgovore pa sem dobila od šestih učencev iz 9. razreda in dveh učencev iz 8. razreda.

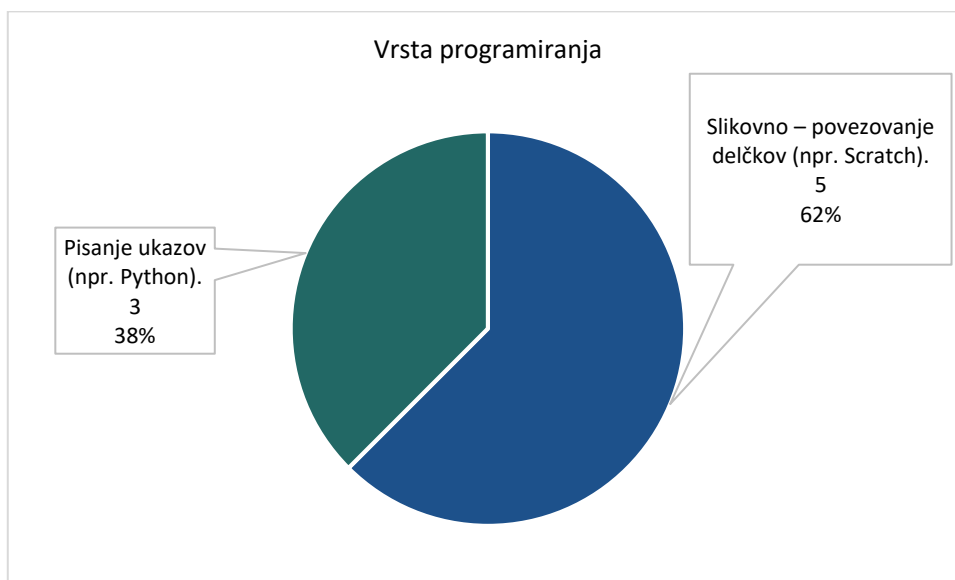
Pri vprašanju »Ali kdaj programiraš?« so imeli na voljo 4 odgovore, in sicer: *Da, veliko.*; *Še kar.*; *Včasih.*; *Ne, nič.*

Polovice od vseh anketirancev programiranje ne zanima niti najmanj. Dva učenca programirata le včasih, med tem ko eden veliko in drugi še kar.



Naslednje vprašanje »Kako raje programiraš?« je ponujalo dva odgovora, in sicer: *Slikovno – povezovanje delčkov (npr. Scratch).*; *Pisanje ukazov (npr. Python).*

Večina učencev raje programira slikovno, saj je lažje in, kot pravijo, »le povezuješ delčke v zaporedje«.



V tretjem razdelku so morali učenci opisati prednosti in slabosti 3 programov.

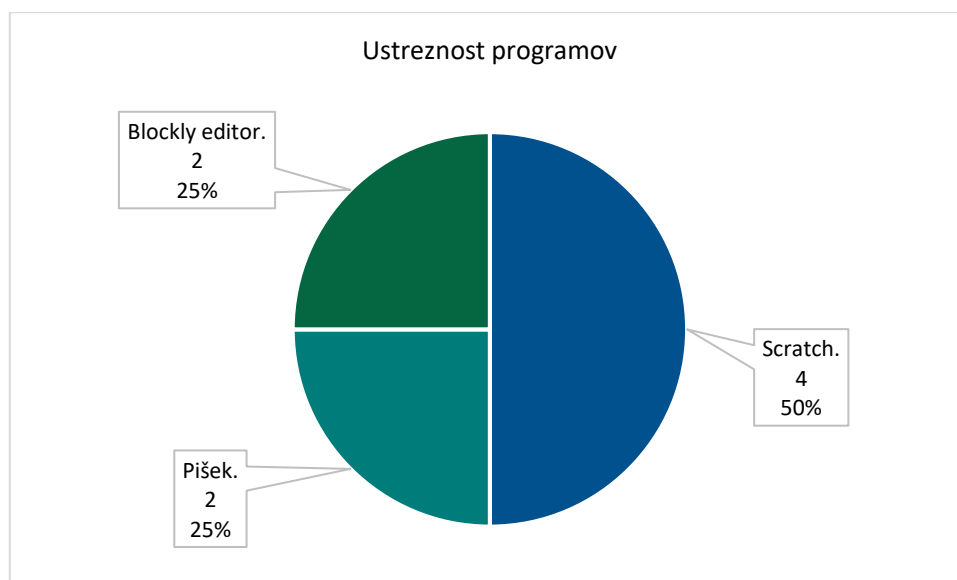
Iz odgovorov je jasno razvidno, da velika večina učencev sploh ne pozna teh programov oziroma jih ne razlikujejo.

Večina je pri Scratch-u napisala le, da ga je lahko uporabljati, saj so delčki že dani, vendar ni tako obsežen kot drugi programski jeziki.

Piška so poznali še manj, kar se vidi pri komaj polovičnih odgovorih. Napisali so, da je kopija Scratch-a in dober za začetnike, vendar da so ukazi le osnovni in da lahko programiraš le po navodilih.

Blockly editor pa so tisti dan spoznali prvič, zato so pod prednosti našli, da je izboljššan, vendar da ima premalo ukazov.

Za zaključek so še označili, kateri od zgornjih programov jim najbolj ustreza. Polovica je označila Scratch, četrtnina Pišek in druga četrtnina Blockly editor.



Po koncu raziskave med učenci sem ugotovila, da večina učencev nima niti osnovnega znanja o programiranju.

Tudi če učencev programiranje ne zanima, bi vsak moral znati vsaj nekaj praktičnih osnov, saj bi jim to tudi pomagalo izboljšati večino reševanja problemov, kar pa bi vsak posameznik potreboval v življenju.

6.2 Rezultati reševanje nalog - učenci

Učenci so reševali že pripravljene naloge v okolju Pišek in nato tudi naloge, ki sem jih sestavila sama v Blockly urejevalniku.

❖ Učenci – začetni:

Glede na njihovo mnenje o težavnosti, času, ki so ga porabili, številu napak ter potrebo po pomoči, lahko razberemo, da se težavnost pri nalogah v Pišku stopnjuje.

Vsem je uspelo uspešno zaključiti prvih 5 nalog, vendar za zadnjo je zmanjkalo časa, prav tako pa so menili, da je 6. naloga težja od 5.

Pri nalogah, ki so jih reševali v Blockly editor, je trem uspelo rešiti vse tri naloge, medtem ko je vsem ostalim uspelo rešiti le prvi 2. Večini se je 2. naloga (*»kvadrat«*) zdela lažja kot 1. naloga (*»Lahka roža«*).

❖ Učenci – napredni:

Pri izpiskih za naloge v Pišku za napredne učence, se vidi, da jim je šlo lažje in bolje od začetnih, kar je razumljivo.

Čeprav se vidi tudi pri njih stopnjevanje težavnosti pri nalogah, se tukaj zares vidi bistven prehod težavnosti med 5. in 6. nalogo. Le enemu je uspelo rešiti vseh 6 nalog.

Pri nalogah v Blockly editor je le 8 učencev od 20 rešilo 1. nalogo ter začelo z 2. Večina se ni prebila niti skozi prvo nalogo. Učenci niso upoštevali navodil, kljub mojim opozorilom in posledično jim program ni deloval pravilno.

❖ Nadarjeni učenci v 8. in 9. razredu na področju naravoslovja

Vsem nadarjenim učencem v 8. razredu za naravoslovje je v Blockly editorju uspelo rešiti 1. nalogo. Nekateri so potrebovali malo več napotkov, vendar so kaj kmalu ugotovili, kako v osnovi razmišlja računalnik. Čeprav je tudi vsem učencem iz 9. razreda uspelo rešiti 1. nalogo, so potrebovali bistveno več časa in pomoči. Do uspešno zaključene 2. naloge so se prebili le 4 učenci iz vsakega razreda. Težavnost 2. naloge se jim je zdela v obeh razredih enaka – naloga je lahka, le veliko delčkov je potrebno uporabiti. Tretjo nalogo sta v 9. razredu zaključila 2 učenca, v 8. pa le eden.

Do dobrega začetka zadnje naloge pa sta prišla le 2 učenca (po eden iz 8. in eden iz 9. razreda). Če primerjamo njun napredek pri reševanju naloge, je bil učenec iz 8. dosti bližje končni rešitvi kot učenec iz 9. razreda.

Če bi ugotovitve za 8. in 9. razred na kratko povzeli, bi lahko rekli, da so si rezultati dosti podobni, vendar so učenci v 9. razredu potrebovali veliko več časa in pomoči, medtem, ko so v 8. razredu učenci hitreje razumeli potek naloge ter pravilnejše povezali delčke v smiseln program.

Razlog je najverjetneje iskati v tem, da osmošolci vključeni v raziskavo več programirajo kot devetošolci.

7 RAZPRAVA

7.1 Potrjevanje hipotez

Glede na odzive na obrazec za učitelje računalništva v osnovni šoli, lahko **prvo hipotezo: »Učitelji računalništva na osnovni šoli pri neobveznih in obveznih izbirnih predmetih učijo programiranja pri svojih urah.«** potrdim, saj vsi učitelji menijo, da je programiranje v osnovni šoli pomembno in ga zato tudi izvajajo pri svojih učnih urah računalništva.

Drugo hipotezo: »Profesorjem na fakulteti in učiteljem računalništva na osnovnih šolah se za začetek programiranja zdita najboljša slikovna programa Scratch ter Pišek in za nadaljevanje programski jezik oziroma Python.« lahko potrdim, saj je iz intervjujev razvidno, da učitelji na začetku predvsem učijo v slikovnem programu Scratch in Pišek, v nadaljevanju pa profesorji učijo v programskih jeziki oziroma Pythonu.

Tudi **tretjo hipotezo: »Vsi učenci neobveznega izbirnega predmeta računalništvo (4., 5., 6. razred) bodo opazili velik preskok težavnosti iz 5. na 6. nalogo v Pišku, vendar pa jo bodo starejši učenci rešili bolje.«** potrdim, saj se iz izpiskov opazi preskok težavnosti iz 5. na 6. nalogo, tako po mnenju učencev kot tudi po opazovanju njihovega reševanja.

Četrto hipotezo: »Nadarjeni učenci, ki obiskujejo pouk nadarjenih za naravoslovje v 9. razredu, bodo naloge v Blockly editorju reševali bolje od učencev v 8. razredu.« moram zavrniti, saj se je izkazalo, da je šlo učencem iz 8. razreda bolje kot učencem iz 9., saj so učenci iz 8. razreda hitreje razumeli in rešili naloge, kljub manjši pomoči.

Zadnjo oziroma **peto hipotezo: »Učencem bo lažje reševati naloge v Pišku, kot v Blockly Editor, saj imajo v Pišku le določene ukaze ter omejeno število delčkov.«** lahko potrdim, saj se glede na uspešnost reševanja in mnenja učencev vidi, da hitreje razumejo nalogo, ko vidijo, katere ukaze imajo na voljo in jih le povežejo v pravilno zaporedje, kot je to v okolju Pišek, medtem ko imajo v Blockly Editorju večjo izbiro in potrebujejo več časa za izbiranje delčkov.

7.2 Ugotovitve

Skozi celotno raziskavo sem ugotovila marsikaj pričakovanega, vendar se me nekatere stvari kljub temu presenetile.

Z vpogledom in analiziranjem odzivov na obrazec, namenjen učiteljem računalništva v osnovni šoli, sem izvedela, da se učiteljem programiranje v osnovni šoli zdi pomembno in ga potemtakem tudi sami izvajajo pri svojih učnih urah. Pri pouku učijo le slikovnega programiranja, in sicer v programih, kot sta Scratch ter Pišek.

Na podlagi tega sem znanje učencev preverila in hkrati razbrala, kje jim je reševanje nalog lažje. Učenci so pokazali, da nimajo prav veliko znanja o programiranju, pa vendar jim je lažje programirati v okolju Pišek, kjer imajo omejeno število določenih ukazov.

Preko opazovanja učencev pri reševanju različnih nalog pa sem ugotovila, da niso navajeni delati po navodilih oziroma so potrebovali dosti pomoči. Učenci, ki obiskujejo neobvezni izbirni predmet računalništvo, so pri vseh nalogah opazili stopnjevanje težavnosti, vendar so to pri nalogah v Pišku še izrecno poudarili. Pri učencih, ki obiskujejo pouk nadarjenih na področju naravoslovja (8. in 9. razred), sem presenečeno odkrila, da učencem močno primanjkuje znanja o programiranju, kljub temu pa so bili učenci 8. razreda uspešnejši.

Za dodatek k raziskovalni nalogi sem izvedla tudi dva intervjuja, in sicer s prof. Janezom Demšarjem ter mag. Matijem Lokarjem. Izvedela sem veliko o začetku in nadaljevanju učenja programiranja (na primer, da se za nadaljevanje učenja programiranja uporabljajo različni programski jeziki kot na primer Python.) ter o samem programu Pišek in nekaj malega tudi o ostalih programih.

Dobila pa sem tudi potrditev, da je učenje programiranja zelo pomembno, a hkrati zabavno ter koristno v vsakdanjem življenju. Prav zato bi se osnov programiranja moral vsak naučiti že v osnovni šoli, saj se nam tako odpre drugačen način razmišljanja in s tem tudi povsem drug pogled na reševanje problemov.

8 ZAKLJUČEK

Tudi sama sem hodila tri leta na neobvezni izbirni predmet računalništvo in kasneje tudi na obvezne izbirne predmete računalništva. Doma sem sama ter s pomočjo interneta in knjig tudi veliko programirala in se naučila veliko več, kot sem se v šoli, ne le, ker sem programirala več, ampak ker sem dobila več različnih pojasnil iz več virov in ker sem lahko imela svoj ritem, kar se tiče obravnavanja snovi.

Tehnologija se z vsakim dnem bolj razvija in s tem raste tudi potreba po znanju njene uporabe, ki pa ga vsem, ne le učencem, primanjkuje. Razlog za to je tudi ta, da večina, tako učencev kot tudi odraslih, še vedno meni, da obvladajo delo z računalnikom, že samo z igranjem igrice ter iskanjem podatkov po svetovnem spletu.

Že v osnovnih šolah bi vse učence morali naučiti računalniških in programerskih osnov. Da bi to dosegli, bi morali v predmetnik osnovne šole uvrstiti obvezni predmet računalništvo za vse učence. Čeprav nekatere šole nudijo računalništvo kot izbirni predmet, se velikokrat izbirne računalniške vsebine sploh ne izvedejo zaradi premalo vpisov in tako se še tisti manjšini odvzame priložnost učenja računalniških vsebin. Učenci izbirne predmete največkrat izbirajo po nasvetu (izbiri) staršev ali pa po liniji najmanjšega truda, ki ga bo potrebno v predmet vložiti, da bodo pridobili oceno.

Tudi če učencev programiranje ne zanima, bi vsak moral znati vsaj nekaj praktičnih osnov, saj bi jim to tudi pomagalo izboljšati veččino reševanja problemov, kar pa bi vsak posameznik potreboval v življenju, zato jih moramo k tem spodbujati ter jim pomagati.

Vsega tega se zavedajo le redki, in morda to sedaj zares še ni zaskrbljujoče, vendar kaj kmalu bo in takrat bo treba ukrepati.

9 LITERATURA IN VIRI

VIRI

Anželj, G. et al. 2018. Slikovno programiranje. E-učbenik. Pridobljeno 14. 12. 2022 s
<https://lusy.fri.uni-lj.si/ucbenik/prog/1221/index.html>

Lajovic, S. 2018. Python za otroke. Ljubljana: Pasadena

Wainewright, M. 2016. Osnove programiranja: Vodič po računalniškem programiranju. Tržič:
Učila International d. o. o

Drinovec, A. Računalniško opismenjevanje v osnovni šoli v Sloveniji. Pridobljeno 14. 12. 2022 s
[https://www.makspecar.si/application/files/7815/5253/8703/VIVID-2016-](https://www.makspecar.si/application/files/7815/5253/8703/VIVID-2016-Racunalnisko_opismenjevanje_v_OS.pdf)
Racunalnisko_opismenjevanje_v_OS.pdf

Lokar, M. Dodatni koncepti in veščine pri programiranju. Gradivo na izobraževanju Programiranje
z delčki, UL Fakulteta za matematiko in fiziko (21. 10. 2022 – 13. 1. 2023)

Lokar, M. Programski koncepti. Gradivo na izobraževanju Programiranje z delčki, UL Fakulteta za
matematiko in fiziko (21. 10. 2022 – 13. 1. 2023)

Lokar, M., Mujkić, M. Računalniško tekmovanje Pišek – oblika vzpodbujanje učenja programiranja
za vse. Gradivo na izobraževanju Programiranje z delčki, UL Fakulteta za matematiko in fiziko (21.
10. 2022 – 13. 1. 2023)

Savnik, I. 2013. Programiranje II: Koncepti programskih jezikov. Pridobljeno 30. 12. 2022 s
https://studentski.net/gradivo/upr_fmn_ri1_pr2_sno_skripta_01

ACM tekmovanja - Pišek. Tekmovalne kategorije. Pridobljeno 13. 12. 2022 s
<https://tekmovanja.acm.si/?q=pisek/tekmovalne-kategorije>

ACM Tekmovanja – Bober. Pridobljeno 11. 12. 2023 s <https://tekmovanja.acm.si/?q=node/693>

ACM tekmovanja – Bober. Pridobljeno 14. 12. 2022 s [https://tekmovanja.acm.si/?q=bober/o-](https://tekmovanja.acm.si/?q=bober/o-bobru)
bobru

ACM Tekmovanja – Pišek. Pridobljeno 12. 12. 2022 s <https://tekmovanja.acm.si/?q=pisek/o-pisku>

Pišek. Pridobljeno 12. 12. 2022 s <https://pisek.acm.si/contents/4903/>

Pišek: Zgodbe. Pridobljeno 30. 12. 2022 s <https://pisek.acm.si/contents/4907-4902/>

Programiranje. Pridobljeno 10. 1. 2023 s <https://www.zotks.si/programiranje/razpis>

Uporabniška programska oprema. Pridobljeno 8. 12. 2022 s

http://egradivo.ecnm.si/KIT/uporabnika_programska_oprema.html

Zakaj se je pomembno naučiti osnov programiranja? Pridobljeno 12. 12. 2022 s

<https://www.smartninja.si/blog/zakaj-se-je-pomembno-nauciti-osnov-programiranja>

VIRI SLIK

Slika 1: <https://www.radio-odeon.com/site/assets/files/56442/Onip-rac-01.webp> (14. 12. 2022)

Slika 2: https://miro.medium.com/max/1400/0*356zWDYXZyGbmQxa. (14. 12. 2022)

Slika 3: https://upload.wikimedia.org/wikipedia/commons/b/bc/Python_Code.png (14. 12. 2022)

Slika 4:

https://video.arnes.si/attachments/video/eN/eNBHdTtwcWDS/images/HdNdWRMe7SNWQGnpB3BHq8lf.preview_image.jpg, pridobljeno (13. 12. 2022)

Slika 5: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ7m4BI6dvYYJL2f8-opSCkcdKgnIPmK8066vL8rHR7MvfkZB1NEuSdZiXpoQJfHlsCx-Y&usqp=CAU> (13. 12. 2022)

Slika 6: http://www.gimjes.si/files/2022/01/bober_ACM_1.jpg (14. 12. 2022)

Slika 7:

https://files.liveworksheets.com/def_files/2020/9/25/925081312687071/925081312687071001.jpg (14. 12. 2022)

Slika 8: <https://arhiv.mlad.si/wp-content/uploads/2017/04/DSCN05311-650x320.jpg> (19. 2. 2023)

Slika 9: <https://pisek.acm.si/contents/4907-4902-6586947264732270-337559782458156072-792990685659790508-703084601233116419/> (30. 12. 2022)

Slika 10: Zaslonska slika programa

- <https://pisek.acm.si/contents/4907-4902/>

- <https://pisek.acm.si/contents/4907-4902-6586947264732270-337559782458156072/>

- <https://pisek.acm.si/contents/4907-4902-6586947264732270-337559782458156072-792990685659790508/>

(14. 12. 2022)

Slika 11: <https://svet.fri.uni-lj.si/intervju-dr-janez-demsar/> (14. 1. 2023)

Slika 12: https://lokar.fmf.uni-lj.si/www/osebno/matija_lokar.htm (14. 1. 2023)

Slike 13 – 14: Zaslonske slike obrazcev

Slike 15 – 28: Zaslonske slike v okolju Pišek

- <https://pisek.acm.si/contents/4907/>
- <https://pisek.acm.si/contents/4907-319805995281415931/>
- <https://pisek.acm.si/contents/4907-319805995281415931-1468740812716735939/>
- <https://pisek.acm.si/contents/4907-319805995281415931-1468740812716735939-1065261577502713763/>
- <https://pisek.acm.si/contents/4907-319805995281415931-1468740812716735939-1065261577502713763-1001406523172202844/>
- <https://pisek.acm.si/contents/4907-319805995281415931-1468740812716735939-1065261577502713763-494973846763652249/>
- <https://pisek.acm.si/contents/4907-319805995281415931-1468740812716735939-1065261577502713763-78627486561669808/>
- <https://pisek.acm.si/contents/4907-319805995281415931-1468740812716735939-1065261577502713763-784057498778404964/>
- <https://pisek.acm.si/contents/4907-319805995281415931-1468740812716735939-1065261577502713763-1118370293325883022/>
- <https://pisek.acm.si/contents/4907-319805995281415931-1468740812716735939-1065261577502713763-249714026735994500/>

(14. 12. 2022)

Slike 29 – 32: Foto Natalija Žabjek

Slike 33 – 38 : Zaslonske slike Blockly editor

Slika 39: Foto Natalija Žabjek

Slika 40 - 53: Zaslonske slike Blockly editor

Slike 54 – 55: Foto Natalija Žabjek

10 PRILOGE

10.1 Vprašalnik za učitelje računalništva



Raziskovalna naloga - Programiranje z delčki

Pozdravljeni,
sem učenka 9. razreda v OŠ Polje in delam raziskovalno nalogo na temo programiranja z delčki. Prosila bi, da si vzamete nekaj minut ter rešite spodnji obrazec.
Najlepša hvala,
Natalija Žabjek

 [Preklopi med računi](#) (ni deljeno) 

* **Zahtevano**

Ime in priimek. *

Vaš odgovor

Starost. *

Vaš odgovor

Spol. *

☐ Ženski.

☐ Moški.

Smer izobrazbe. *

Vaš odgovor

Zapišite šolo, na kateri poučujete. *

Vaš odgovor

Koliko let poučujete? *

Vaš odgovor

Katere računalniške predmete poučujete? *

- ☐ Obvezne izbirne predmete (UBE, MME, ROM)..
- ☐ Neobvezni izbirni predmet računalništvo..
- ☐ Oboje.

Naprej

Počisti obrazec

Obvezni izbirni predmeti (UBE, MME, ROM)

Ali pri svojih urah pouka računalniških vsebin učite programiranja? *

- ☐ Da.
- ☐ Ne.

Katere programe uporabljate?

- ☐ Scratch.
- ☐ Pišek.
- ☐ Drugo: _____

Ali menite, da je učenje programiranja v osnovni šoli potrebno? *

- ☐ Da, zelo.
- ☐ Da.
- ☐ Ne.
- ☐ Nikakor.

Zakaj tako menite? *

Vaš odgovor _____

Kako se po vašem mnenju lahko otrok hitreje in učinkovitejše nauči programiranja? *

- ☐ V šoli.
- ☐ Sam s pomočjo interneta in knjig.

Kateri program se vam zdi najprimernejši za začetek programiranja? *

- ☐ Scratch.
- ☐ Pišek.
- ☐ Drugo: _____

Kdaj naj bi se učenci po vašem mnenju začeli učiti programiranja? *

- ☐ V prvem triletju OŠ.
- ☐ V drugem triletju OŠ.
- ☐ V tretjem triletju OŠ.
- ☐ V srednji šoli.
- ☐ Kasneje.

Kaj bi svetovali mladim programerjem? *

Vaš odgovor _____

Ali poučujete tudi neobvezni izbirni predmet? *

- ☐ Da.
- ☐ Ne.

Nazaj

Naprej

Počisti obrazec

Neobvezni izbirni predmet računalništvo

Ali pri svojih urah pouka računalniških vsebin učite programiranja? *

☐ Da.

☐ Ne.

Katere programe uporabljate?

☐ Scratch

☐ Pišek

☐ Drugo: _____

Ali menite, da je učenje programiranja v osnovni šoli potrebno? *

☐ Da, zelo.

☐ Da.

☐ Ne.

☐ Nikakor.

Zakaj tako menite? *

Vaš odgovor _____

Kako se po vašem mnenju lahko otrok hitreje in učinkovitejše nauči programiranja? *

☐ V šoli

☐ Sam s pomočjo interneta in knjig

Kateri program se vam zdi najprimernejši za začetek programiranja? *

- ☐ Scratch.
- ☐ Pišek.
- ☐ Drugo: _____

Kdaj naj bi se učenci po vašem mnenju začeli učiti programiranja? *

- ☐ V prvem triletju OŠ.
- ☐ V drugem triletju OŠ.
- ☐ V tretjem triletju OŠ.
- ☐ V srednji šoli.
- ☐ Kasneje.

Kaj bi svetovali mladim programerjem? *

Vaš odgovor _____

Nazaj

Naprej

Počisti obrazec

Hvala za vašo pomoč.

Če poznate še koga, ki uči računalništvo v osnovni šoli, bi prosila, da tudi njemu ta obrazec pošljete. Najlepša hvala.

Nazaj

Pošlji

Počisti obrazec

10.2 Vprašalnik za učence



Scratch code blocks showing a loop and a selection statement. The code includes comments in Slovenian and Python-like syntax for object selection in a game engine.

Lastnosti programov z delčki - Pišek, Scratch, Urejevalnik

Pozdravljen,
pričujoči obrazec se nanaša na naloge iz slikovnega programiranja, ki si jih zadnjič reševal pri uri nadarjenih. Zanima me, kakšni se ti zdijo določeni programi za slikovno programiranje.
Hvala za sodelovanje,
Natalija Žabjek, 9. a

 Preklopi med računi (ni deljeno) 

* Zahtevano

Ime in priimek *

Vaš odgovor

Razred *

Vaš odgovor

[Naprej](#) [Počisti obrazec](#)

Programiranje

Ali kdaj programiraš? *

- ☐ Da, veliko.
- ☐ Še kar.
- ☐ Včasih.
- ☐ Ne, nič.

Kako raje programiraš? *

- ☐ Slikovno - povezovanje delčkov (npr. Scratch).
- ☐ Pisanje ukazov (npr. Python).

Nazaj

Naprej

Počisti obrazec

Prednosti in slabosti programov z delčki

Naštej prednosti Scratcha. *

Vaš odgovor

Naštej slabosti Scratcha. *

Vaš odgovor

Naštej prednosti Piška. *

Vaš odgovor

Naštej slabosti Piška. *

Vaš odgovor

Naštej prednosti Urejevalnika. *

Vaš odgovor

Naštej slabosti Urejevalnika. *

Vaš odgovor

Kateri od spodnjih programov se ti zdi najboljši? *

☐ Scratch.

☐ Urejevalnik.

☐ Pišek.

[Nazaj](#) [Naprej](#) [Počisti obrazec](#)

Najlepša hvala za pomoč.

[Nazaj](#) [Pošlji](#) [Počisti obrazec](#)