

Prikaz metodičke obrade i prijelaza s programskog jezika Scratch u programski jezik Python

Latin, Mihael

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Teacher Education / Sveučilište u Zagrebu, Učiteljski fakultet**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:147:520499>

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

Download date / Datum preuzimanja: **2024-05-18**

Repository / Repozitorij:

[University of Zagreb Faculty of Teacher Education - Digital repository](#)



**SVEUČILIŠTE U ZAGREBU
UČITELJSKI FAKULTET
ODSJEK ZA UČITELJSKE STUDIJE**

MIHAEL LATIN

**PRIKAZ METODIČKE OBRADE I
PRIJELAZA S PROGRAMSKOG
JEZIKA SCRATCH U PROGRAMSKI
JEZIK PYTHON**

DIPLOMSKI RAD

Zagreb, rujan 2022.

SVEUČILIŠTE U ZAGREBU
UČITELJSKI FAKULTET
ODSJEK ZA UČITELJSKE STUDIJE

PREDMET: Metodika informatike

Ime i prezime pristupnika: Mihael Latin

**TEMA DIPLOMSKOG RADA: Prikaz metodičke obrade i
prijelaza s programskog jezika Scratch u programski jezik
Python**

DIPLOMSKI RAD

MENTOR:
prof. dr. sc. Mario Dumančić

Zagreb, rujan 2022.

Sadržaj

UVOD	1
1 Programiranje u osnovnoj školi	3
2 Programski jezik Scratch u osnovnoj školi.....	5
2.1 Programski jezik Scratch u udžbeniku Školske knjige.....	11
2.2 Programski jezik Scratch u Alfinom udžbeniku.....	13
3 Programski jezik Python u osnovnoj školi	14
3.1 Programski jezik Python u udžbeniku Školske knjige	16
3.2 Programski jezik Python u Alfinom udžbeniku	17
4 Aktivnosti.....	18
5 Anketni upitnik	20
5.1 Uzorak ispitanika.....	21
5.2 Opis istraživanja	22
5.3 Metode obrade podataka	22
6 Rezultati	23
6.1 Rezultati koji se odnose na programski jezik Scratch	23
6.2 Rezultati usporedbe programskog jezika Scratch i programskog jezika Python.....	26
6.3 Rezultati utjecaja učenja programskog jezika Scratch na prilagodbu u programskom jeziku Python.....	29
7 Rasprava.....	32
8 Zaključak	33
LITERATURA.....	34
POPIS SLIKA	35
POPIS TABLICA	36
Izjava o samostalnoj izradi rada.....	37

SAŽETAK

Razvijanje digitalne pismenosti postaje ključ za daljnji napredak, a sve kreće u osnovnoj školi. U svijetu globalizacija i automatizacije biti digitalno nepismen je isto kao što je prije par desetljeća bilo biti uistinu nepismen, a učitelji ponovno stoje ispred vrata znanja spremni da ih otvore učenicima željnim znanja. Jedno od tih korisnih znanje je programiranje. Programiranje je vještina na kojoj je potrebno raditi cijeli život, no potrebna je podloga koju može pružiti upravo osnovna škola. Razvojem programskog jezika Scratch učenicima se pružila jedinstvena prilika da razvijaju sposobnost logičnog i apstraktnog razmišljanja, sastavljanja algoritama i razumijevanja kompleksnih sintaksi, i to sve kroz igru. Učenje takvog programskog jezika može biti odlična podloga za kasnije učenje nešto kompleksnijeg jezika kao što je programski jezik Python.

U ovom diplomskom radu prikazani su programski jezik Scratch i programski jezik Python i njihova metodička obrada u udžbenicima Školske knjige i Alfe. Na osnovi anketnog upitnika provedeno je istraživanje u četiri različite škole s ukupno 229 učenika, s ciljem utvrđivanja lakoće tranzicije s programskog jezika Scratch u programski jezik Python kod učenika petih razreda.

Na temelju dobivenih rezultata je utvrđeno kako je učenje programskog jezika Scratch uistinu korisno u učenju programskog jezika Python te bi se trebalo poticati.

Ključne riječi: programiranje, informatika, programski jezik Scratch, programski jezik Python

SUMMARY

View of methodical process and transition from Scratch programming language to Python programming language

Developing digital literacy becomes the key to further progress, and everything starts in elementary school. In the world of globalization and automation, being digitally illiterate is the same as being truly illiterate a couple of decades ago, and teachers are once again standing in front of the door of knowledge ready to open it to students eager for knowledge. One of the useful pieces of knowledge is programming. Programming is a skill that you need to work on all your life, but you need a foundation that can be provided by elementary school. The development of the Scratch programming language gave students a unique opportunity to develop the ability to think logically and abstractly, compose algorithms and understand complex syntax, all through play. Learning such a programming language can be an excellent basis for later learning a more complex language such as Python programming language.

This thesis presents the programming language Scratch and the programming language Python and their methodical processing in the textbooks Školska knjiga and Alfa. Based on a questionnaire, research was conducted in four different schools with a total of 229 students, to determine the ease of transition from the Scratch programming language to the Python programming language among fifth graders.

Based on the obtained results, it was determined that learning the Scratch programming language is truly useful in learning the Python programming language and should be encouraged.

Keywords: programming, computer science, programming language Scratch, programming language Python

UVOD

Uvođenjem nastavnog predmeta Informatike kao obaveznog predmeta za učenike od prvog do četvrtog razreda osnovne škole otvorilo je mogućnost za širu i lakšu prilagodbu na tekstualno programiranje koje učenike čeka u predmetnoj nastavi. Nastavni predmet Informatike se u kurikulumu dijeli na četiri domene: A. Informacije i digitalna tehnologija, B. Računalno razmišljanje i programiranje, C. Digitalna pismenost i komunikacija i D. E-Društvo. Programiranje je dakle, uvođenjem novog kurikulumu, dobilo zasebnu domenu, zajedno s računalnim razmišljanjem. Za razvijanje mladih programera potrebno je razviti logičko i apstraktno mišljenje te analiza zadatka koji je stavljen pred njih. No programiranje razvija i neke korisne ljudske osobine kao što su samopouzdanje, upornost, preciznost i socijalne vještine izražene u radu u skupinama. Također, razvoj programa ponekad zahtjeva inovativnost, poduzetnost i preuzimanje inicijative. (MZO, 2018) Naravno, da bi smo razvili svaku od navedenih osobina kod učenika potrebno je od nekud krenuti, a programski jezik Scratch nudi odličnu platformu koja učeničko razmišljanje može odvesti na pravi put.

Programski jezik Scratch učenike kroz igru uvodi u svijet programiranja. Zamisao je da otprije definiranim blokovima učimo učenike logiku programa. Dakle, iz programiranja je izbačena sintaksa, ali algoritmi ostaju u programiranju kao nužan faktor u izradi programa. U teoriji to izgleda kao izbacivanje svega kompliciranog iz programiranja i uvođenja samo onoga što je učenicima već poznato – kockice. Zapravo učenici nesvjesno razvijaju logičko i apstraktno mišljenje.

Dolaskom učenika u predmetnu nastavu dolazi i, uz sve ostale brige koje im taj period donosi, učenje novog programskog jezika – Python. Programski jezik Python, za razliku od programskog jezika Scratch, uvodi sintaksu, uz nešto kompleksnije algoritme jer učitelji, po udžbenicima, s njima prolaze nastavno gradino predmeta Matematike. Učenici koji dolaze s podlogom učenja programskog jezika Scratch u novo okruženje programskog jezika Python bi, u teoriji, trebali imati određenu prednost nad onima koji se prvi put susreću s programskim jezikom. Iako je sintaksa nešto potpuno novo s čime se moraju suočiti, njihovo prethodno podsvjesno znanje algoritama i razvoja programa im može uvelike pomoći.

Programski jezici Scratch i Python, primjena tih jezika u udžbenicima Školske knjige i Alfe te istraživanje utjecaja učenja programskog jezika Scratch na učenje programskog jezika Python biti će predstavljene u sljedećim poglavljima rada.

1 Programiranje u osnovnoj školi

U skladu s razvojem tehnologije nameće se potreba savladavanja jezika računala, odnosno, principa njegova rada. Bubica i sur. (2014) navode kako u suvremeno vrijeme brzog tehnološkog razvoja valja i djecu poučavati programiranju i to od najranije dobi. No, kako je programiranje kompleksan i složen proces, s učenjem valja započeti od najjednostavnijih, djeci prirodnih i razumljivih načina. Na ideju da i djeca mogu programirati dolazi se od činjenice kako današnja djeca za ekranom provode mnogo svog slobodnog vremena. Učeći osnovne koncepte programiranja, djeca stječu nova znanja i vještine, a edukacijska vrijednost programiranja dokazana je brojnim istraživanjima (Đurđević, 2013).

Valja naglasiti kako programiranje ne podrazumijeva isključivo pisanje računalnih programa, već ono potiče širi način razmišljanja i razradu strategija. Bubica i sur. (2014) programiranje definiraju kao: „*rješavanje problema, otklanjanje grešaka, razvijanje logičkog razmišljanja i računalnog razmišljanja, a to podrazumijeva razvoj strategija za rješavanje problema koji se mogu odnositi i na neprogramerska područja.*“ Sukladno navedenoj definiciji, može se zaključiti kako je programiranje temelj budućeg obrazovanja i rada. Uzimajući u obzir činjenicu da živimo u digitalnom dobu, nameće se jasan zaključak kako je programiranje nužno uvesti već u osnovnoškolsko obrazovanje. Prisutnost informatičkih rješenja i upotreba informacijskih tehnologija u svakodnevnom životu u budućnosti će postajati još izraženija.

Programiranje se sve manje promatra kroz prizmu vještine koju je poželjno savladati, a sve se više promatra kao alat koji je implementiran u sve oblike učenja i rada. Činjenica je da se kroz programiranje razvijaju metakognitivne sposobnosti djece, a koje su temelj za daljnje školovanje, no postavlja se pitanje na koji način podučavati učenike vještini programiranja?

U osnovnoj se školi kao početni stupanj u učenju programiranja najčešće koriste programi Logo i BASIC, dok je Pascal rjeđe u upotrebi. U osnovnim školama u Republici Hrvatskoj učenici najčešće počinu s modelima programa Logo koji se počinje obrađivati u petom razredu osnovne škole u sklopu nastavnog predmeta Informatika. Program je osmišljen isključivo za djecu, kako bi na što jednostavniji način (crtanjem crteža po ekranu, dajući naredbe imaginarnom robotu kamo treba ići) djeca počela učiti i kako bi uopće mogla razumjeti na koji način program radi. Prva

verzija programa dizajnirana je 1968. godine u Cambridgeu, a razvoj programa nastavio se pod vodstvom W. Feurziga i S. Paperta. Među najpopularnijim verzijama programa Logo nalaze se Terrapin Logo, Berkeley Logo, MicroWorlds, MsWLogo i NetLogo. U osnovnim školama u Hrvatskoj najčešće se upotrebljava komercijalna verzija Logo programa pod nazivom Terrapin Logo (Đurđević, 2014, str. 95).

U hrvatskim se osnovnim školama za podučavanje programiranja koristi programski jezik BASIC (eng. *Beginner's All-purpose Symbolic Instruction Code*), koji je, kako i sam naziv kazuje, jedan od najjednostavnijih programskih jezika i idealan je za početnike. Program su kreirali John Kemeny i Thomas Kurtz godine 1963., a od samog početka do danas program je zbog svoje jednostavnosti u upotrebi diljem svijeta, što ga čini jednim od najpopularnijih programa. Do danas su razvijene brojne inačice programa, a najčešće korištena verzija je Quick Basic poznatija pod nazivom Qbasic. Program funkcionira na način da se za stvaranje jednostavnih programa koriste naredbe i varijable koje zajedno čine programski kod (Rouse, 2011).

Treći programski jezik koji se koristi u edukaciji osnovnoškolaca je Pascal, no u odnosu na navedena dva programa, program Pascal nije toliko jednostavan, stoga je njegova upotreba u osnovnim školama nešto slabija. Programski jezik Pascal pogodan je za strukturirano programiranje, što podrazumijeva da se problemi rješavaju na strukturiran način- odnosno, problem se razdijeli na manje dijelove, a potom se na sistematiziran se način rješava dio po dio problema. Pascal je osmislio švicarski znanstvenik Nikolaus Wirth godine 1970., a stvoren je na temeljima programskog jezika Algol. Programski jezik Pascal ime je dobio u čast matematičara i filozofa Blaisea Pascala.

Kako je programski jezik Pascal osmišljen ne samo za učenje programiranja, već i kao način lijepog stila u programiranju, tako je njegova uporaba nadišla obrazovnu razinu i koristi se čak i komercijalne svrhe. No, razvojem drugih, objektno orijentiranih programskih jezika, kao što su C++ i Java Pascal krajem dvadesetog i početkom dvadesetprvog stoljeća počinje gubiti na svojoj popularnosti u obrazovnom sustavu ali i u industriji. Neke inačice Pascala koriste se još i danas ne samo za učenje nego i za kreiranje i razvoj softvera (Borland Delphi) (CARNet, online tečaj „Programiranje u Pascal-u“, <https://tesla.carnet.hr/>). U posljednjih se nekoliko godina u obrazovnom sustavu sve više počinje primjenjivati programski

jezik Python, zamjenjujući navedene programske jezike. Prednosti njegove upotrebe ogledaju se u jednostavnijoj sintaksi i lakšem shvaćanju programiranja.

Iako je učenje programiranja i upotreba programskih jezika u hrvatskim školama poprilično zastupljena, još uvijek postoji problem koji se ogleda u nedovoljnoj zastupljenosti podučavanja programiranja. Razlozi za to očituju se u činjenici da je Informatika izborni predmet koji po novom kurikulumu postaje obavezan od petog do šestog razreda, ali ostaje izborni predmet od prvog razreda i u skladu s tim, velik broj učenika se ne odlučuje upisati Informatiku u nižim razredima osnovne škole. Uz to, postoji i određeni stupanj demotiviranosti i nedovoljnog stupnja edukacije kod nastavnika, što rezultira time da se programiranje uopće ne podučava u nekim osnovnim školama. Time se stvara nepravедno stvaranje negativan stav prema programiranju, uz popratno mišljenje kako je ono teško, komplicirano i dostupno samo odabranima.

2 Programski jezik Scratch u osnovnoj školi

Programiranje je pisanje uputa računalu što i kako učiniti, a izvodi se u nekom od programskih jezika. Programiranje je umjetnost i umijeće u stvaranju programa za računala. Zbog toga je programiranje potpuno apstraktno i teško za razumijevanje, ali je u isto vrijeme dobar alat za vježbu i razvoj apstraktnog razmišljanja. Djeca na početku osnovne škole nemaju mogućnost apstraktnog mišljenja, već ga stječu kasnije učenje programiranja čini im se teškim. Djeci u prvom razredu osnovne škole je problem još veći jer djeca u prvom razredu uglavnom ne znaju čitati ni pisati. Da bi djecu koja još nisu sposobna apstrakcijski razmišljati naučili programiranju, koristimo vizualne programske jezike. Vizualni programski jezici, osim uklanjanja problema sintakse, omogućuju učenje programiranja u konkretnom okruženju, gdje apstraktni pojmovi poput varijabli, petlji i sl., u vizualnom okruženju, pružaju konkretno iskustvo. (Bubica i sur., 2014). Programiranje u grafičkim okruženjima u kojima ne postoje sintaktičke pogreške ublažuje stresan početak učenja programiranja. Vizualno programiranje se koristi za početno učenje koncepata računalne znanosti i vještine kodiranja jer podupire koncept učenja putem istraživanja (Bubica i sur., 2014, str. 5).



Slika 1: Logo programskog jezika Scratch

Programi koji se uče u hrvatskim školama, kao što su Basic, Logo i Pascal, pomažu djeci sa prvim koracima učenja osnova programiranja i logičkog razmišljanja. No, zbog svoje složene sintakse nisu prilagođeni učenicima razredne nastave. Scratch se koristi za poučavanje djece programiranju. Prilagođen je djeci već od predškolske dobi. Programiranje u Scratchu temelji se na grafičkom povezivanju prethodno nacrtanih blokova, kao što je to slučaj sa LEGO kockicama koju većina djece poznaje. Različiti tipovi podataka su prikazani različitim oblicima blokova, a blokovi se slažu u cjeline tako da se spoje na sintaktički pravilan način. Takav pristup radu onemogućuje sintaktičke greške (greške nastale ne poštivanjem propisanih pravila pisanja programa- greške pri korištenju naredbi, tipfeleri, itd.) i fokusira pažnju učenika na problem koji bi trebali riješiti, a ne na samu mehaniku programiranja. Ova bi se vrsta programa u hrvatskim školama mogla primijeniti u nastavi informatike od drugog do četvrtog razreda. Pomoću njega učenici bi mogli raditi na raznim projektima i učiti širok spektar aktivnosti iz različitih predmeta te tako proširivati svoje znanje.

Scratch je programski jezik otvorenog koda razvijen 2003. godine na Tehnološkom institutu države Massachusetts (Massachusetts Institute of Technology – MIT) kao projekt Lifelong Kindergarten grupe. Scratch omogućuje lako stvaranje interaktivnih priča, igrice, animacija i projekata te njihovo dijeljenje s drugim korisnicima preko weba. (Mujačić, 2015). Grupa Lifelong Kindergarten surađivala je s tvrtkom Lego na izradi robota Lego Mindstorms koji se, također, koriste za poučavanje programiranja. Lifelong Kindergarten grupa i Lego su uočili da djeci u radu sa kockicama odmah počinju navirati ideje, mašta i kreativnost. Zato su udruženim snagama napravili vizualni programski jezik koji podsjeća na slaganje

kockica. Naredbe su napravljene u obliku slagalica, čime djeca lakše prate naredbe koje mogu složiti. Naredbe su grupirane tematski, a razlikuju se po obliku i bojama (Bubica i sur., 2014).

Projekt je razvijen u duhu proširenja vidika djece i poticaja djece na istraživanje, osmišljavanje i ostvarivanje svojih zamisli, koje zatim mogu pokazati drugima putem interneta, usput usvajajući osnove programiranja, matematike te vizualnog i interaktivnog dizajna (Valić, B., Radovan, A., Pavlović, D., 2013).

Scratch omogućuje rad u višejezičnom okruženju, uključujući sučelje i na hrvatskom jeziku, a može se instalirati na operacijske sustave Windows, MAC i Linux. Ima dobro razvijenu zajednicu korisnika koji postavljaju svoje projekte na Scratch-ove službene mrežne stranice te omogućuju njihovo besplatno preuzimanje i korištenje. Iako je Scratch nastao na idejama programskog jezika LOGO, programiranje u Scratchu razlikuje se od programiranja u drugim vizualnim programskim okolinama. U Scratchu se koriste naredbene strukture u obliku grafičkih programskih blokova, pa se na taj način uklanja mogućnost sintaksnih pogrešaka. Program omogućuje programiranje mišem povlačenjem i uklapanjem blokova koji se mogu spojiti samo ako to odgovara u određenom sintaksnom smislu i na taj način omogućuje učenicima da se fokusiraju na probleme koje oni žele riješiti. Nastao je kao pomoć za razvijanje osnovnih vještina koje autori programa smatraju iznimno bitnim za 21. stoljeće, kao što su kreativno razmišljanje, jasna komunikacija, sistematična analiza, efikasna kolaboracija i kontinuirano učenje (Đurđević, 2013).

Autori Scratch-a napominju da je Scratch namijenjen početnicima u programiranju u dobi od navršenih osam godina nadalje. S radom u Scratch-u se započinje lako, premda program omogućuje kreiranje i složenijih projekata.

Scratch učiteljima nudi priliku korištenja računala u svakodnevnim školskim aktivnostima, uzimajući u obzir razvojne osobine učenika kao pojedinca, ali i cijelog razreda. Učenje pomoću računala i Scratch-a ne može u potpunosti zamijeniti direktnu komunikaciju učenika i učitelja, ali mnoge aktivnosti može učiniti jednostavnijima i zanimljivijima. Ponavljanje uz pomoć edukativnih igara čini nastavnu aktivnost privlačnijom, stvaranje i prezentiranje vlastitih uradaka dobiva na novim mogućnostima izražavanja, a samostalno istraživanje potiče radoznalost i upornost kod učenika.

Sintaksa programa

Pomoću jednostavnih programskih jezika kao što je Scratch, naučiti najosnovnije elemente programiranja je lako, a isto može biti korisno kasnije u životu. Jednom kad je interes za programiranje otkriven i pokrenut, teško je reći gdje će završiti i u kojem će se smjeru razvijati. Jedna od najvažnijih prednosti programskog jezika Scratch u odnosu na ostale programske jezike koji uče programiranju je ta što se programi u Scratchu ne pišu, već se slažu (princip slaganja koda kao LEGO kockica), odnosno crtaju. Korisnik nije prisiljen učiti složene izraze kako bi opisao temeljne programske elemente već su mu ti elementi neposredno predstavljeni slikama i oblicima. Blokovi programa su označeni različitim bojama i organizirani u sekcije, ovisno o tome što ti blokovi izvode. Nema stroge sintakse već je vizualno jasno mogu li se neke naredbe spojiti ili ne. (Krpan, D., Mladenović, S., Zaharija, G., 2014). Blokovi programa se jednostavno mogu povući iz palete sa strane (unošenje jednostavnih ili naprednih naredbi), umjesto pisanja koda. Čak i ako kod nije valjan u nekom trenutku, projekt će svejedno raditi, dok u drugim programima program ne radi čim dođe do sintaktičke pogreške (npr. zaboravljen znak: točka-zarez i sl.). Još jedna prednost je ta da je Scratch potpuno besplatan program te ga mogu koristiti svi, za osobnu ili akademsku uporabu. Također, postoji prijevod na hrvatski jezik što dodatno olakšava rad, a Scratch ima i sjajnu zajednicu koja može pružiti podršku korisniku u stvaranju vlastitih projekata.

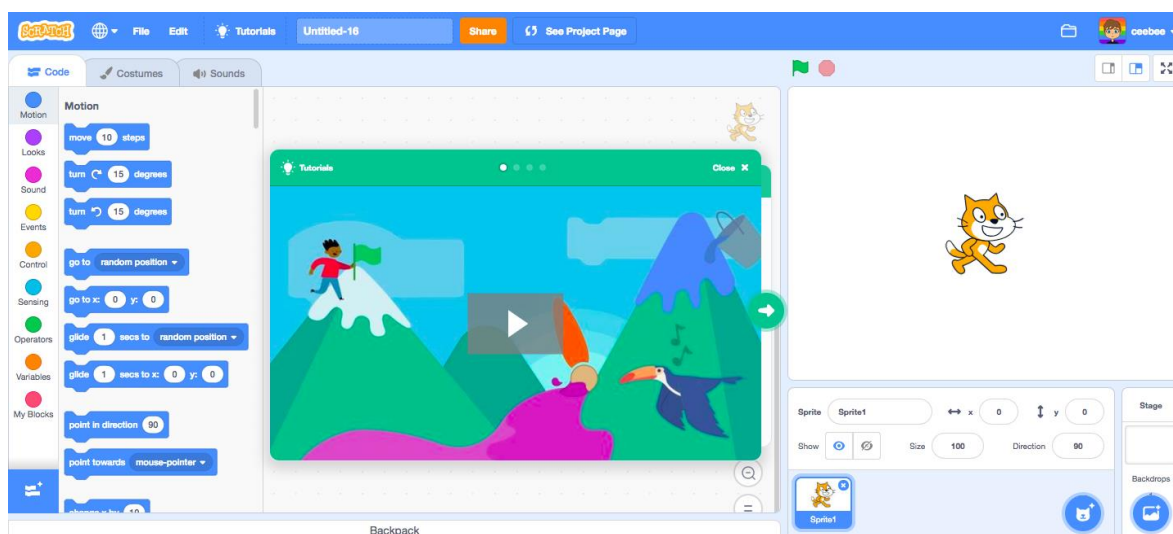
Scratch je dobar izbor prvog programskog jezika za djecu (niži razredi osnovne škole) jer pruža mogućnost učenja osnova rada na računalu (korištenje tipkovnice, miša, mikrofona, kamere i ostalih uređaja), podučava računalnoj grafici (crtanje jednostavnih likova i pozadina) i programiranju (izradi priča ili jednostavnih igara) (Buklijaš, 2010). Osnovna prednost vizualnog programskog jezika je ta da zahtijeva relativno malo inicijalnog znanja kako bi ga se moglo početi efikasno koristiti, dovoljno je znati koristiti se mišem i tipkovnicom. To je zbog toga što su svi elementi programskog jezika, njegove semantike i sintakse vizualno reprezentirani te se korištenjem kontekstne ovisnosti dinamički sužava izbor elemenata koji se mogu upotrijebiti i na taj način korisnika se vodi kroz postupak programiranja. (Vukotić, D., Tanković, N., 2011).

Nedostaci ovog programskog jezika su ti da online uređivač zahtjeva prijavu, odnosno izradu korisničkog računa za koji treba e-mail adresa, te stabilnu internetsku vezu. No, ovaj problem može se riješiti preuzimanjem programa na računalo. Sljedeći nedostatak je da drugi korisnici mogu pregledavati i uređivati tuđe projekte te ih spremi, preurediti ili preuzeti kao svoje i time pridobiti zasluge. Još jedan nedostatak je taj što potrebno mnogo „ručnog“ rada, npr. kopiranje skripti za svaki lik te mijenjanje naredbi u skriptama i njihovo ponovno dupliciranje te mnogo klikanja, povlačenja i ispuštanja blokova („drag and drop“). Nadalje, kako korisnici rade sve složenije skripte tako područje za izradu postaje premalo i nepregledno. Ozbiljniji korisnici (programeri) navode kako Scratch zaostaje u prikazivanju koliko moćno programiranje može biti.

Unatoč navedenim nedostacima i ograničenjima, osobe koje savladaju Scratch programski jezik puno će lakše shvatiti i razumjeti naprednije programske jezike, poput Pythona, jer će već razumjeti programske koncepte petlji, odlučivanja, varijabli, toka programa i slično.

Razvojno okruženje i osnovni elementi Scratcha

Scratch razvojno okruženje se nalazi na jednom prozoru. Elementi unutar prozora su fiksni i sadrže više dijelova. Na desnoj strani postoji mjesto gdje se izvodi program, ispod toga se nalazi popis objekata u programu. Na lijevoj strani se nalazi dio s pomoćnim blokovima i opcijama. U sredini je prostor za slaganje skripti. Raspored dijelova se može i obrnuti, tako da se mjesto gdje se izvodi program nalazi na lijevoj strani. (Valić, B., Radovan, A., Pavlović, D., 2013)



Slika 2: Razvojno okruženje za Scratch 3.0

Scratch projekt čine objekti koji se zovu likovi (eng. *sprites*). Njihov se izgled može mijenjati dodavanjem različitih kostima (eng. *costumes*). Lik može izgledati poput osobe, vlaka, leptira ili bilo čega drugoga. Za kostim je moguće koristiti bilo koju postojeću sliku: može se izraditi crtež u *Paint Editor*-u, može se učitati s računala ili metodom uhvati-povuci-pusti preuzeti s internetske stranice. Liku se mogu dati različite upute: da se kreće, razgovara, reproducira glazbu ili ostvaruje interakciju s drugim likovima. Za izdavanje naredbi korisnik mora složiti grafičke blokove u cjeline, koje se nazivaju skripte. (Otvoreno društvo za razmjenu ideja (ODRAZI), 2011).

Scratch blokovi organizirani su u osam kategorija različitih boja: Kretanje, Izgled, Zvuk, Olovka, Upravljanje, Očitavanja, Operacije i Varijable. Svaka vrsta blokova ima jedinstven izgled kojim pokazuje koji blok ide ispod, iznad ili unutar drugih blokova, na način da se blokovi spajaju kao slagalice. Svojim izgledom blokovi sugeriraju koja su ograničenja i mogućnosti te na taj način onemogućavaju javljanje pogrešaka. Iako nema dojave pogrešaka, sam program može funkcionirati na „krivi način“, ali eksperimentiranjem i iskustvom može se doći do željenih rezultata. Klikom na skriptu, Scratch prolazi kroz blokove od vrha do dna. Ako korisnik samo želi provjeriti kako djeluje određeni blok, dovoljno je kliknuti na njega i lik na ekranu će naredbu odmah izvesti. Skripte koje se grade uvijek se mogu izvesti te pojedini blokovi neće smetati ako se odvoje van skripte. (Valić i sur., 2013).

Tri su glavne vrste blokova u Paleti blokova:

- SLOG BLOKOVI: Ovi blokovi imaju ispupčenja na dnu i/ili udubljenja na vrhu poput puzzli. Ovi se blokovi mogu slagati u slogove. Neki slog-blokovi imaju predviđen prostor za unos podataka gdje se može unesti broj ili odabrati element iz padajućeg izbornika. Neki slog blokovi, , imaju „usta“ u obliku slova C, gdje je moguće umetnuti druge slog-blobove. (ODRAZI, 2011, str. 13).
- ŠEŠIRI: Ovakvi blokovi imaju zaobljene vrhove. Smještaju se na vrh slogova gdje čekaju neku radnju, primjerice pritiskanje određene tipke, nakon čega izvršavaju radnju bloka ispod njih. (ODRAZI, 2011, str. 13).

- REPORTERI: Ovakvi blokovi kreirani su kako bi stali u upisni dio drugih blokova. Reporteri sa zaobljenim krajevima dojavljuju brojeve ili nizove i pristaju u zaobljene i pravokutne praznine (ODRAZI, 2011, str. 14). Reporteri s oštrim rubovima dojavljuju boolean vrijednosti (točno ili netočno) i pristaju u blokove s oštrim ili pravokutnim prazninama. Neki reporteri pored sebe imaju tzv. checkbox, odnosno kutiju za označivanje. Klikom na checkbox, na pozornici se pojavljuje monitor koji prikazuje trenutnu vrijednost reportera. Monitor se automatski ažurira, ovisno o promjenama te vrijednosti.

Programska struktura je ustrojstvo programa, tj. način i redoslijed rješavanja pojedinih manjih radnji da bi se došlo do konačnog rješenja zadatka. Nekoliko je osnovnih programskih struktura koje se pojavljuju u mnogim računalnim programima pa tako i u Scratchu. To su:

- pravocrtna programska struktura (slijed/sekvencija)
- struktura grananja (odabir/selekcija)
- struktura petlje (ponavljanje/iteracija)

2.1 Programski jezik Scratch u udžbeniku Školske knjige

Scratch kao programski jezik se pojavljuje u seriji udžbenika *e-Svijet* od nakladnika Školska knjiga, od udžbenika za drugi razred osnovne škole. Scratch se koristi kao programski jezik za rješavanje problema koje udžbenici opisuju. Sam jezik se sustavno ne obrađuje, već svaki udžbenik za pojedini razred donosi neku novinu iz Scratcha. Primjeri problema riješenih kroz Scratch se nastoje približiti učenicima na zanimljiv i pristupačan način, primjeren dobi učenika u pojedinom razredu, ali bez pomoći učitelja, učenici ne mogu samostalno savladavati Scratch.



Slika 3: Naslovnica udžbenika iz informatike za 2. razred osnovne škole, e-Svijet 2

Scratch se samostalno obrađuje kroz priručnik koji izdaje Školska knjiga: Istražite SCRATCH 3.0, autora Ane Budojević i Anite Kanić. Priručnik je namijenjen svima onima koji žele naučiti kako kreirati animacije, interaktivne priče ili igre programirajući u Scratchovoj 3.0 inačici. Uz ovaj priručnik, učenici od 1. razreda osnovne škole mogu samostalno savladavati Scratch.

Priručnik metodički obrađuje Scratch, učeći djecu vještinama: učenja, kritičkog razmišljanja, rješavanja problema, komunikacije, suradnje, kreativnosti i inovacija. Te vještine uključuju jasno komuniciranje, analiziranje, sustavno i učinkovito surađivanje, kreativno razmišljanje te kontinuirano učenje. Ovakav pristup učenju i poučavanju kroz primjere inspiriran je konstruktivističkom teorijom učenja i obrazovanja (učenje temeljeno na iskustvu).

Priručnik se sastoji od 7 poglavlja. Kroz prva dva poglavlja, priručnik na zanimljiv i pristupačan način, primjeren djeci u dobi od 7 do 13 godina, daje detaljan pregled sastavnih dijelova Scratcha. Kroz sljedeća 4 poglavlja, priručnik daje zanimljive projekte, vježbe sa primjerima rješenja i zadatke za samostalno rješavanje. Projekti u priručniku se ne moraju rješavati redom navedenim u sadržaju.



Slika 4: Naslovnica priručnika za učenje Scratcha nakladnika Školska knjiga, Istražite Scratch 3.0

2.2 Programski jezik Scratch u Alfinom udžbeniku

Iako se isječci programskog koda Scratcha pojavljuju, kod izdavača Alfa, od udžbenika za prvi razred osnovne škole, tek se u udžbeniku za peti razred Scratch sustavno i detaljnije obrađuje. Iako je pristup obradi Scratcha sustavan i obrađen na 20-ak stranica, sam pregled Scratcha i primjeri u njemu nisu dovoljni za samostalni nastavak učenja Scratcha izvan nastave. U udžbeniku se nalazi premalo primjera i zadataka za samostalni rad.



Slika 5: Naslovnica udžbenika iz informatike za 5. razred nakladnika Alfa, LikeIT

Za razliku od udžbenika Školske knjige, koja je izdala samostalni priručnik za Scratch, Alfini udžbenici od petog do osmog razreda sadrže poglavalja o Scratchu. U Alfinim udžbenicima nastavnici viših razreda osnovne škole mogu birati hoće li podučavati djecu Scratchu ili Pythonu. S obzirom da ne pokazuju svi učenici afinitet, želju i talent za savladavanje gradiva programiranja, dva programska jezika na biranje mogu biti bolji izbor i za nastavnike i za učitelje.

3 Programski jezik Python u osnovnoj školi

Programski jezik Python počeo je zamjenjivati ostale programske jezike u informatici za više razrede osnovne škole. Python je stvoren krajem 1980-ih. Osmislio ga je Guido van Rossum koji je imao viziju stvaranja novog jezika koji će biti jednostavan i ugodan za korištenje. Upravo ova činjenica omogućava korištenje Pythona u edukacijske svrhe u osnovnoj školi. Za razliku od Scratcha, koji je prvenstveno programski jezik za korištenje u edukacijske svrhe, Python je programski jezik opće namjene. Python je danas ujedno i jedan od popularnijih programskih jezika koji koriste programeri diljem svijeta, za izradu svih vrsta računalnih programa.

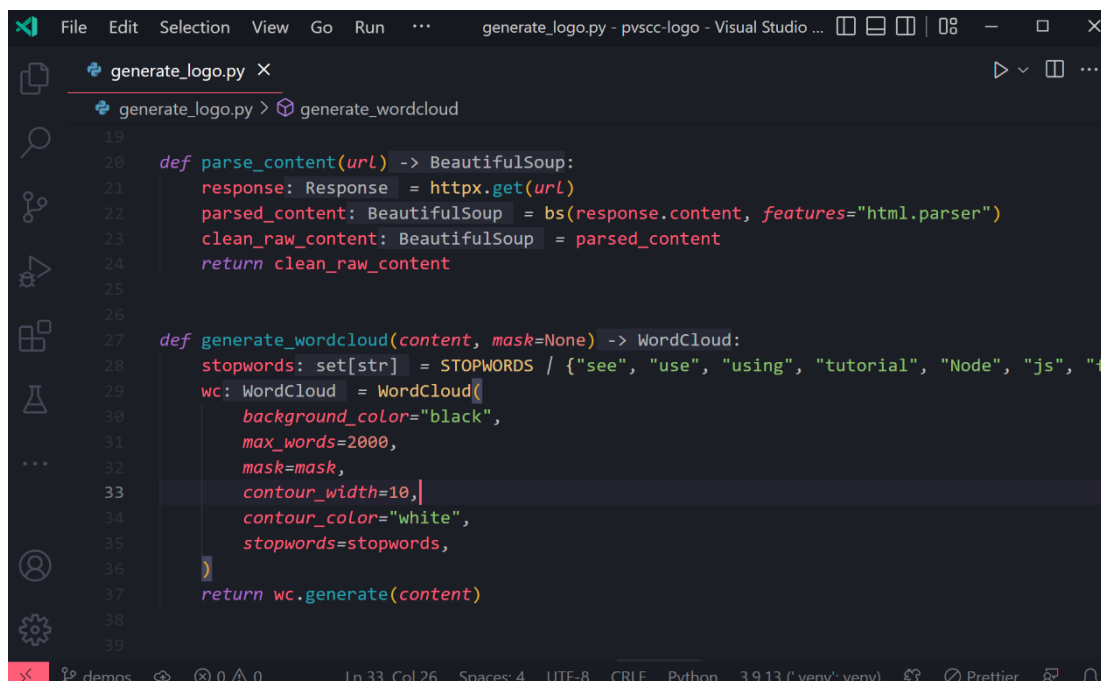


Slika 6: Logo programskog jezika Python

Učenje programskog jezika Python kreće u 5. razredu osnovne škole, kada informatika postaje obavezni predmet za sve učenike. Učenici koji su do tada bili upoznati sa programskim jezikom Scratch će lakše savladavati Python, dok će ostali imati ponešto problema jer Python, kao "viši" programski jezik, unosi nivo apstrakcije koji učenici, odnosno djeca, teže savladavaju.

Razvojno okruženje i osnovni elementi Pythona

Za razliku od Scratcha koji ima jedno razvojno okruženje, za programski jezik Python postoji više razvojnih okruženja, od Notepada do *Visual Studio Code*-a, koji je trenutno najpopularniji.



```
File Edit Selection View Go Run ... generate_logo.py - pvscc-logo - Visual Studio ...
generate_logo.py > generate_wordcloud
19
20 def parse_content(url) -> BeautifulSoup:
21     response: Response = httpx.get(url)
22     parsed_content: BeautifulSoup = bs(response.content, features="html.parser")
23     clean_raw_content: BeautifulSoup = parsed_content
24     return clean_raw_content
25
26
27 def generate_wordcloud(content, mask=None) -> WordCloud:
28     stopwords: set[str] = STOPWORDS / {"see", "use", "using", "tutorial", "Node", "js", "f
29     wc: WordCloud = WordCloud(
30         background_color="black",
31         max_words=2000,
32         mask=mask,
33         contour_width=10,
34         contour_color="white",
35         stopwords=stopwords,
36     )
37     return wc.generate(content)
38
39
```

Slika 7: Visual Studio Code, razvojno okruženje za programski jezik Python

Python programi se sastoje od više različitih dijelova, a najosnovniji dio je varijabla. Varijable su dijelovi programa koji se koriste kako bi se spremile vrijednosti. Tim vrijednostima (informacijama) pristupa se iz raznih dijelova računalnih programa. Svaka varijabla određena je svojim imenom i memorijskom lokacijom na kojoj je zapisana određena vrijednost.

Najjednostavniji opis pojma varijable bio bi da je varijabla "kutija" u koju se spremaju stvari (u našem slučaju vrijednosti) te se prema potrebi te stvari mogu uzeti (koristiti, pridruživati drugim varijablama), mijenjati s drugim stvarima (s drugim vrijednostima) itd. U varijable je moguće pospremiti svakakve vrijednosti, od

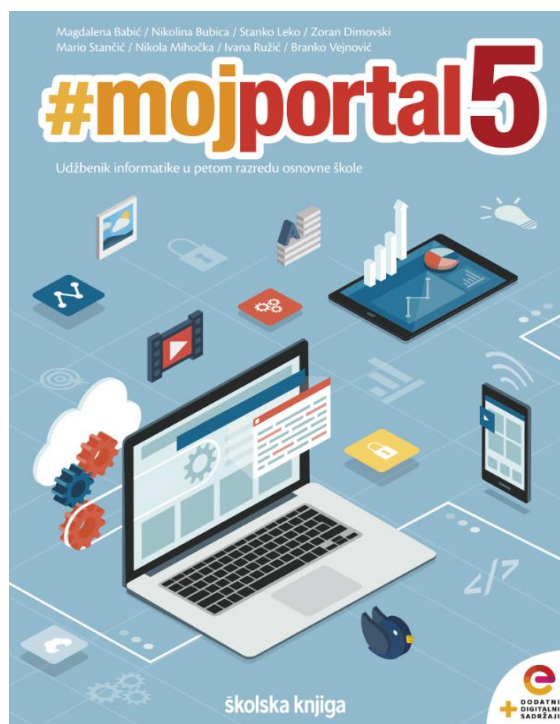
jednostavnih kao što su brojevi, slova, riječi, do složenih kao što su rječnici, liste ili objekti.

Nekoliko je osnovnih programskih struktura koje se pojavljuju u mnogim računalnim programima pa tako i u Pythonu. To su:

- pravocrtna programska struktura (slijed/sekvencija)
- struktura grananja (odabir/selekcija)
- struktura petlje (ponavljanje/iteracija)

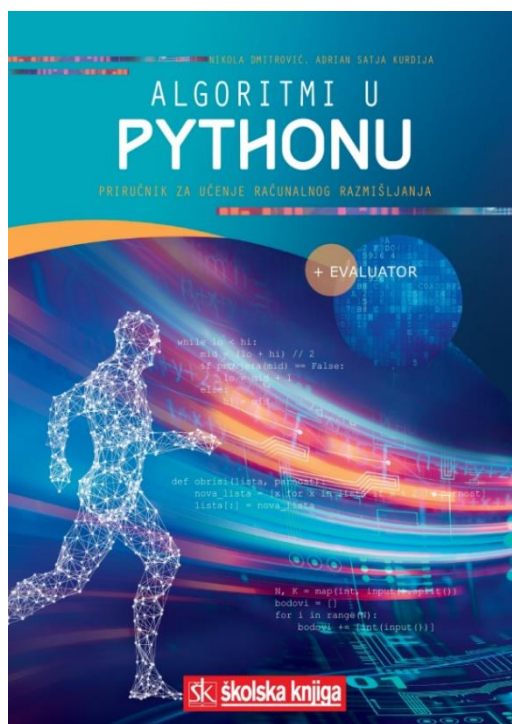
3.1 Programski jezik Python u udžbeniku Školske knjige

Python se počinje koristiti u nastavi informatike u 5. razredu osnovne škole. U udžbeniku *#mojPortal5*, Pythonu je posvećeno 30-ak stranica, što čini najveću zasebnu cjelinu u udžbeniku. Python je objašnjen kroz primjere, čime se maksimalno smanjuje nivo apstrakcije potreban za savladavanje novih pojmova. Primjeri problema riješenih kroz Python se nastoje približiti učenicima na zanimljiv i pristupačan način, primjeren dobi učenika u pojedinom razredu. Prateći udžbenik i zadatke u njemu, učenici su sposobni samostalno upoznavati Python. Python se koristi i u ostalim razredima informatike do kraja osnovnoškolskog obrazovanja te se kroz njega uvode novi pojmovi iz svijeta programiranja.



Slika 8: Udžbenik iz informatike za 5. razred osnovne škole, nakladnika Alfa, #mojPortal5

Uz udžbenik, Python se obrađuje i u priručniku za učenje računalnog razmišljanja *Algoritmi u Pythonu*, u izdanju Školske knjige. Priručnik uvodi i razlaže važne i široko primjenjive algoritme ilustrirajući ih na primjerima zadataka s informatičkih natjecanja. Algoritmi u priručniku su ostvareni u Pythonu, koji je široko rasprostranjen u softverskoj industriji. Priručnik počinje osnovama Pythona, a potom prati program informatičkih natjecanja u kategoriji Algoritmi i stoga najviše koristi darovitim osnovnoškolcima i njihovim mentorima, ali i srednjoškolcima i studentima koji svoje tehničko znanje žele dopuniti vrlo važnim i traženim znanjem o algoritmima. Priručnik obrađuje Python mnogo detaljnije i dublje od redovnog udžbenika za nastavu informatike u višim razredima osnovne škole.



Slika 9: Priručnik za učenje programskog jezika Python, nakladnik Školska knjiga, *Algoritmi u Pythonu*

3.2 Programski jezik Python u Alfinom udžbeniku

Programski jezik Python se u udžbenicima nakladnika Alfe pojavljuje u udžbeniku za peti razred osnovne škole, kao jedan od dva programska jezika koji se mogu izabrati i kroz koje se onda obrađuje gradivo programiranja iz predmeta informatika. Svi udžbenici do osmog razreda osnovne škole daju takav izbor, tako da ako nastavnik u petom razredu odluči pokazivati gradivo sa Pythonom, to može činiti

do kraja osnovne škole. Sam jezik je obrađen sustavno i detaljno, na učenicima zanimljiv i privlačan način. Jezik je dobro opisan, sa dobrim primjerima programskog koda te učenici mogu samostalno nastaviti istraživati i učiti nakon obrađenih nastavnih cjelina u školi.

5. PROGRAMIRANJE	
A Scratch	
5.1. Prisjetimo se.....	76
5.2. Grananje <i>ako-onda</i>	80
5.3. Grananje <i>ako-onda-inače</i>	84
5.4. Ponavljanja s uvjetom.....	86
5.5. Programiranjem do rješenja	89
B Python	
5.1. Prisjetimo se	94
5.2. Uvjet u programiranju.....	98
5.3. Grananje <i>if-else</i>	101
5.4. Grananje <i>if-elif-else</i>	104
5.5. Ponavljanje s uvjetom	107
5.6. Programiranjem do rješenja	110

Slika 10: Gradivo nastavne cjeline Programiranje, informatika za 5. razred osnovne škole

4 Aktivnosti

Scratch se često koristi u podučavanju kodiranja, informatike i računalnog razmišljanja. Učitelji ga također koriste kao alat u mnogim drugim predmetima, uključujući matematiku, znanost, povijest, geografiju i umjetnost. Pošto je mladima lako savladati Scratch zbog gotovo nepostojeće sintakse u pisanom obliku, s jednostavnim vizualnim sučeljem, mladi ga koriste za stvaranje digitalnih priča, igrica i animacija.

Python se obično koristi za razvoj web stranica i softvera, automatizaciju zadataka, analizu podataka i vizualizaciju podataka. Budući da ga je relativno lako naučiti, Python se koristi i za podučavanje vještine programiranja. Python su prihvatili i mnogi ne-programeri, poput računovođa i znanstvenika, za razne svakodnevne zadatke, poput organiziranja financija.

Code Club projekt

Na svjetskoj razini postoji mnogo organizacija i udruga koje se bave podučavanjem djece programiranju. Najpoznatiji je volonterski klub programiranja za djecu Code Club. U Republici Hrvatskoj djeluje njihov ogranak Code Club Croatia. Djeca, uz pomoć volontera i edukatora prolaze kroz kurikulum pripremljenih projekata iz programiranja i izrađuju animacije, igre i web stranice. Code Club je namijenjen djeci od 7 do 13 godina. Centralni Code Club na svjetskoj razini brine o stvaranju materijala i projekata za mlade, a volonterski ogranci brinu oko provedbe projekata. Code Clubovi provode projekte ne samo u Scratchu, već i u Pythonu, Unity-u, sa Raspberry Pi mikroračunalima, Micro:Bit pločicama i slično. Klubove po državama obično osnivaju učitelji i nastavnici informatike u školama, a kao pomoć provedbi mogu sudjelovati i volonteri kao pomoć osnivačima.

Natjecanje iz informatike u RH

Iako u Republici Hrvatskoj postoje organizirana natjecanja iz informatike za osnovne i srednje škole, ne postoje organizirana natjecanja u Scratchu. Natjecanje u informatici se sastoji od 3 kategorije: Algoritmi, Razvoj softvera i Osnove informatike i digitalne komunikacije. U kategorijama se učenici natječu raspoređeni u dobne skupine, za svaki viši razred osnovne škole postoji po jedna dobna skupina, a učenici srednjih škola imaju svoje 4 skupine. Natjecati se mogu isključivo redoviti učenici osnovnih i srednjih škola Republike Hrvatske. U svim kategorijama natjecanja, osim u kategorijama Razvoj softvera za osnovnu i srednju školu, učenici se natječu samostalno. Svaki natjecatelj mora imati jednog ili više mentora. Dodatno, uz navedene 3 kategorije, za učenike osnovne škole postoji i kategorija natjecanja u programskom jeziku LOGO. Težina zadataka varira ovisno o potkategorijama natjecanja i o razini natjecanja. Osnovnoškolci imaju jednostavnije zadatke prilagođene njihovom uzrastu, a srednjoškolci rješavaju kompleksnije probleme od kojih neki zahtijevaju podosta znanja i vještine.

U kategoriji Algoritmi, učenici se natječu u rješavanju složenih problema gdje je cilj napisati što bolji programski kod koji će se izvršavati što je brže moguće. Učenici na početku natjecanja dobiju nekoliko zadataka, a na raspolaganju imaju programski jezik uz pomoć kojeg moraju programirati svoja rješenja zadanih problema. Učenici mogu birati žele li kod pisati u programskom jeziku Python, Basic, Pascal, C ili C++. Po završetku natjecanja ispituje se ispravnost rješenja

zadataka po osmišljenim testnim primjerima koji nose određeni broj bodova, te se formira konačni poredak.

U kategoriji Razvoj softvera, učenici se natječu u ulozi pravih arhitekata računalnih sustava, jer moraju napraviti što bolje i što veće programe kako bi dokazali da su najbolji. U ovoj kategoriji, način održavanja natjecanja je ponešto drugačiji od ostalih kategorija. Na školskoj razini natjecanja ne radi se ljestvica poretka. Razina je samo prezentacijskog tipa. Na županijskoj razini povjerenstva moraju napraviti ljestvicu poretka. Učenici prezentiraju svoj rad pred komisijom. Prezentirani radovi ocjenjuju se prema kriterijima vrednovanja, a temeljem ocjena formira se rang lista. Natjecanje izgleda tako da natjecatelji razviju ideju koju onda implementiraju u bilo kojem programskom jeziku. Natjecatelji se mogu koristiti i gotovim programskim paketima, ali tada rad mora sadržavati barem dio vlastitog koda. Krajnji rezultat natjecatelji moraju prezentirati kroz audio/video uradak od 8 minuta.

Kategorija Osnove informatike i digitalne komunikacije je namijenjena za učenicima koji ne vole složene probleme ili velike programske sustave. Učenici se natječu u znanju osnova informatike i digitalne kompetencije. Težina pitanja i zadataka ovisi o uzrastu i razini na kojoj se natječe. Specifičnost ove kategorije je da se učenici u srednjoškolskoj skupini rangiraju u tri različite podskupine: prirodoslovno-matematičke gimnazije, strukovne škole i ostale gimnazije.

5 Anketni upitnik

Anketni upitnik pod nazivom „Prikaz metodičke obrade i prijelaza s programskog jezika Scratch u programski jezik Python“ proveden je s učenicima petih razreda iz četiri različite škole. Ispunjavanju upitnika pristupilo je ukupno 229 učenika. Ispitivanje je provedeno jednokratno na četiri različite skupine.

Cilj

Cilj ovog istraživanja bio je utvrditi lakoću tranzicije s programskog jezika Scratch u programski jezik Python kod učenika petih razreda koji su na satima Informatike u razrednoj nastavi obrađivali programski jezik Scratch, a sada, u predmetnoj nastavi, na satima Informatike obrađuju programski jezik Python.

Hipoteze

H1: Učenicima je programski jezik Scratch jednostavan i pristupačan.

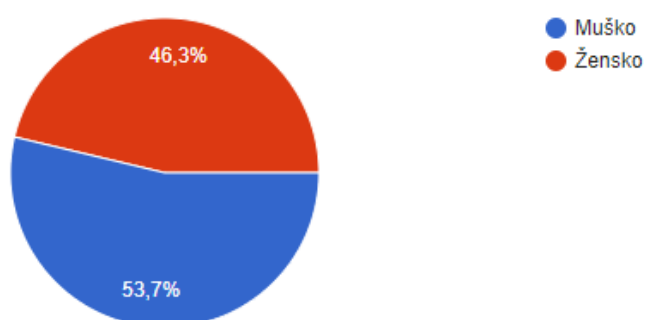
H2: Učenicima je učenje programskog jezika Scratch omogućilo lakši prijelaz u programski jezik Python.

5.1 Uzorak ispitanika

Uzorak ispitanika sastojao se od učenika petih razreda. Ukupni uzorak bio je 229 učenika, sudjelovalo je 106 djevojčica i 123 dječaka.

1. Spol

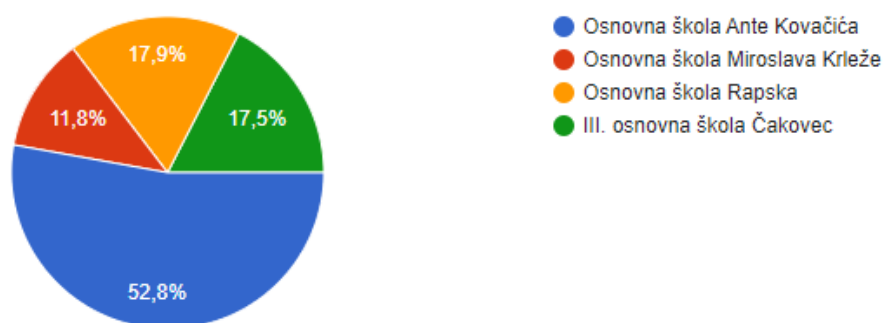
229 odgovora



Slika 11: Spol sudionika

Iz slike 11 vidljivo je da je od ukupnog broja ispitanika, 53,7% muških, dok je 46,3% ženskih ispitanika.

229 odgovora



Slika 12: Škole čiji su učenici sudjelovali u istraživanju

Iz slike 12 vidljivo je da je učenici koji su sudjelovali u anketnom upitniku pohađaju četiri različite škole. Anketni upitnik je proveden u više škola kako bi rezultati bili što vjerodostojniji. Ispunjavanju anketnog upitnika pristupili su učenici sljedećih škola:

- Osnovna škola Ante Kovačića, Grad Zagreb (njihovi učenici čine 52,8% ukupnih ispitanika, odnosno 121 ispitanik pohađa navedenu školu)
- Osnovna škola Rapska, Grad Zagreb (njihovi učenici čine 17,9% ukupnih ispitanika, odnosno 41 ispitanik pohađa navedenu školu)
- III. osnovna škola Čakovec, Čakovec (njihovi učenici čine 17,5% ukupnih ispitanika, odnosno 40 ispitanika pohađa navedenu školu)
- Osnovna škola Miroslava Krležu, Grad Zagreb (njihovi učenici čine 11,8% ukupnih ispitanika, odnosno 27 ispitanika pohađa navedenu školu)

5.2 Opis istraživanja

Istraživanje je provedeno putem online Google upitnika, te je sudjelovanje bilo potpuno dobrovoljno i anonimno. U provođenju istraživanja pomogao mi je jedan učitelj i tri učiteljice. Svi navedeni su diplomirani učitelji Informatike te predaju Informatiku u razrednoj nastavi u navedenim školama. Ispitanici su anketni upitnik rješavali unutar Google obrasca, a istraživanje se provodilo u svibnju i lipnju, dakle krajem školske godine. Istraživanje je s namjerom provedeno krajem školske godine jer su na taj način učenici imali dovoljno vremena da savladaju osnove programskog jezika Python koje se očekuju od njih kurikulumom.

Upitnik se sastoji od 16 pitanja, od kojih 5 nezavisnih i 11 zavisnih varijabli. Sedam pitanja odnose se na programski jezik Scratch, a tri pitanja odnose se na programski jezik Python. Posljednje pitanje u upitniku odnosi se na hipotezu izrade ovog anketnog upitnika, odnosno je li učenicima pomoglo učenje programskog jezika Scratch u savladavanju programskog jezika Python.

5.3 Metode obrade podataka

Za obradu dobivenih podataka upotrijebljen je programski paket IBM SPSS, a rezultati su prikazani u tabličnom obliku i pomoću grafova.

6 Rezultati

Rezultati istraživanja biti će prikazani kao samostalne vrijednosti i u ovisnosti o nezavisnim varijablama.

6.1 Rezultati koji se odnose na programski jezik Scratch

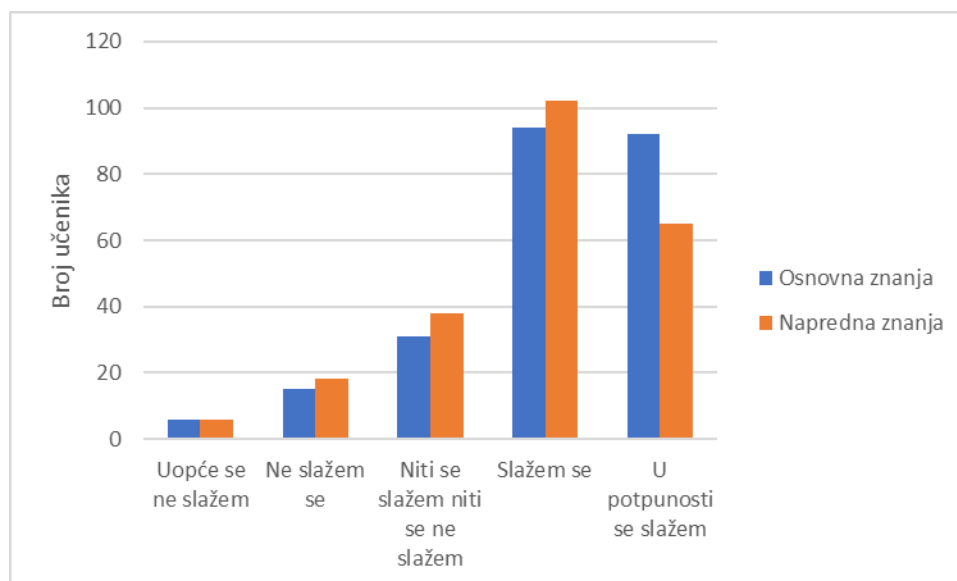
Pregledom rezultata (Tablica 1 i Tablica 2) vidljivo je da je većina učenika, po svom mišljenju, savladala osnovna i napredna znanja korištenja programskog jezika Scratch (ubacivanje likova i pozadina, kretanje likova, ponavljanje naredbi, ako onda, ako onda inače, itd.).

Tablica 1: Savladao/Savladala sam osnovna znanja unutar programskog jezika Scratch

	Frequency	Percent
Valid Ne Slazem se	18	7.9%
Niti se slazem niti se ne slazem	38	16.6%
Slazem se	102	44.5%
U potpunosti se slazem	65	28.4%
Uopce se ne slazem	6	2.6%
Total	229	100.0%

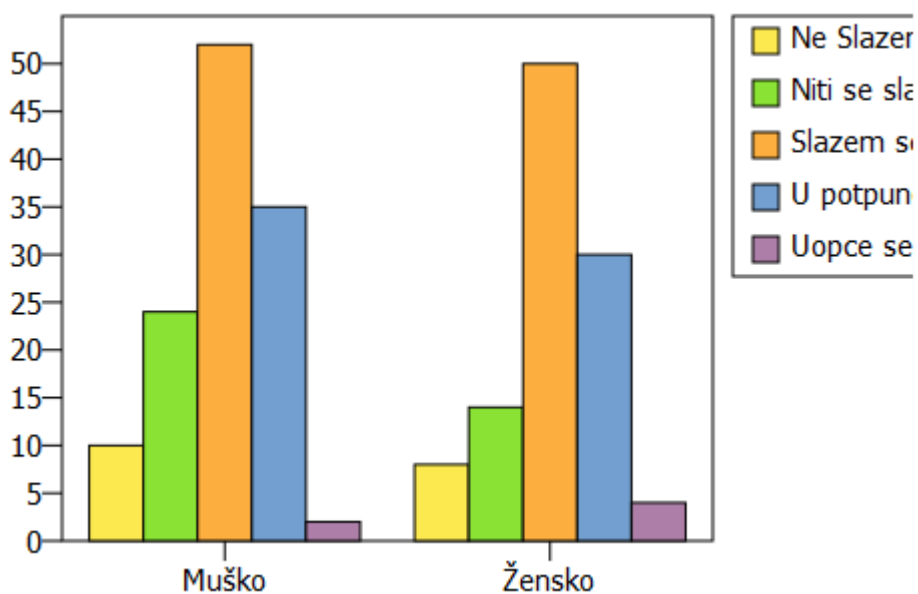
Tablica 2: Savladao/Savladala sam napredna znanja unutar programskog jezika Scratch

	Frequency	Percent
Valid Ne Slazem se	6	2.6%
Niti se slazem niti se ne slazem	31	13.5%
Slazem se	94	41.0%
U potpunosti se slazem	92	40.2%
Uopce se ne slazem	6	2.6%
Total	229	100.0%



Slika 13: Usporedba rezultat poznavanja osnovnih i naprednih funkcija programa stupčastim dijagramom

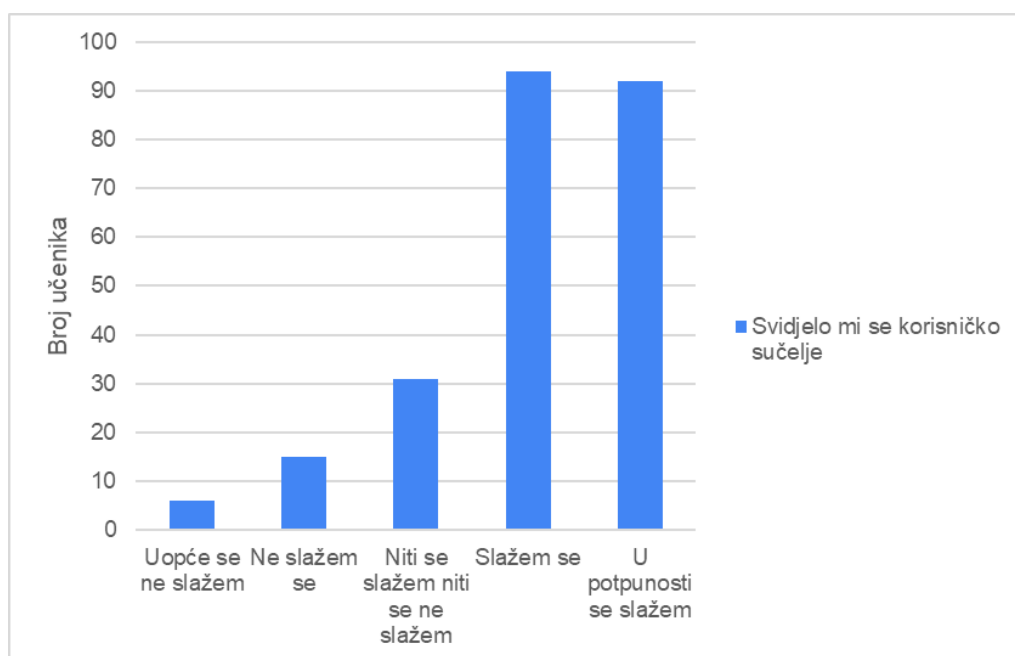
Iz slike 13 može se iščitati kako su učenici samouvjereni u svoje poznavanje osnovnih i naprednih značajki programskog jezika Scratch, iako su nešto manje sigurni u napredne funkcije.



Slika 14: Usporedba rezultata usvojenosti naprednih funkcija po spolu

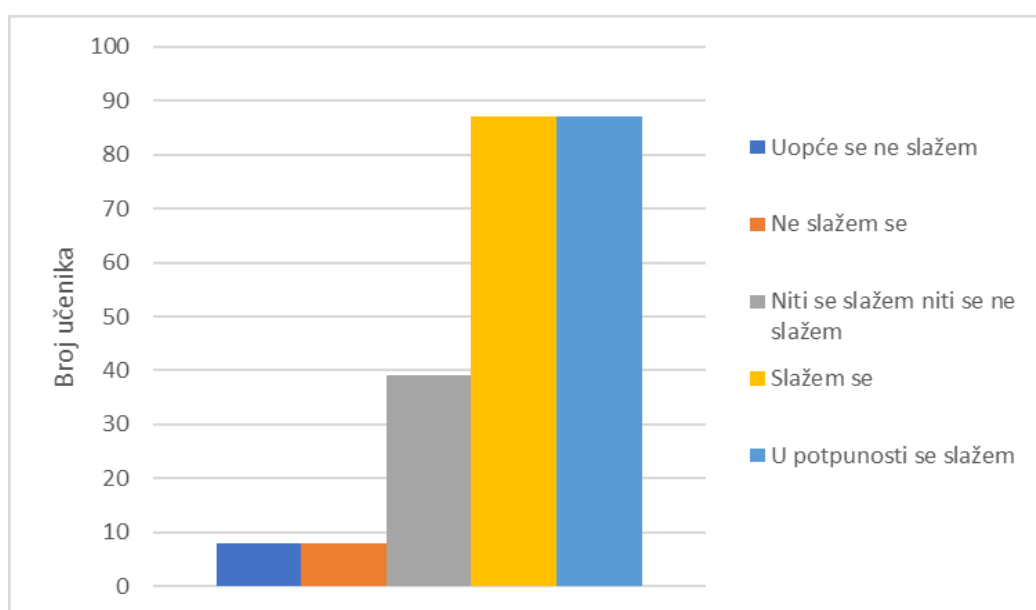
Iz slike 14 može se vidjeti kako su rezultati podjednaki kod oba spola i da je mali broj učenika koji nisu usvojili napredne funkcije.

Ono što programski jezik Scratch čini tako zabavnim je njegovo sučelje koje je jednostavno i pristupačno, po mišljenju učenika, kao što možemo vidjeti na slici 15. Ovi podaci nam ukazuju na to da učenicima u ranoj dobi više odgovara programiranje blokovima, građenje programa i vrlo intuitivno sučelje.



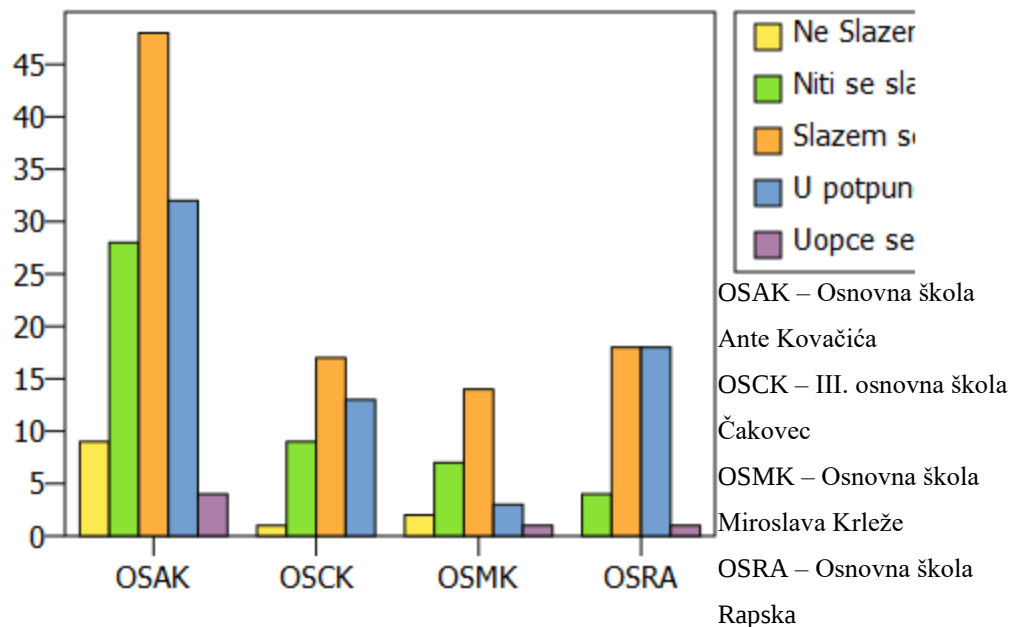
Slika 15: Svidjelo mi se korisničko sučelje u programskom jeziku Scratch (mogu s lakoćom pronaći naredbe)

Čak 186 učenika je reklo kako im se svidjelo korisničko sučelje u programskom jeziku Scratch, dok je svega 24 učenika reklo da im se nije svidjelo.



Slika 16: Zadovoljan/Zadovoljna sam znanjem koje sam stekao/stekla u programskom jeziku Scratch.

Učenici osnovnih škola koji su sudjelovali u anketi su s učenjem programskog jezika Scratch započeli tek prošle godine. No, kao što možemo vidjeti na slici 16, učenici su veoma zadovoljna stečenim znanjem. Čak 174 učenika je reklo da je zadovoljno stečenim znanjem, dok je njih svega 16 nezadovoljno.



Slika 17: Odnos zadovoljstva stečenim znanjem po školama

Podaci na slici 17 ukazuju da su učenici u sve četiri škole zadovoljni sa stečenim znanjem iz programskog jezika Scratch. Zanimljivo je kako su učenici Osnovne škole Rapska izrazito zadovoljni stečenim znanjem, tek su dva učenika rekla kako se uopće ne slažu s tim. U postotku zadovoljnih učenika najbolja je Osnovna škola Rapska, koja ima 87,8% učenika koji su zadovoljni ili iznimno zadovoljni, no učitelji svake od navedenih škola su očito učenicima pružili dovoljno znanja da budu zadovoljni.

6.2 Rezultati usporedbe programskog jezika Scratch i programskog jezika Python

Zadaci koje učenici obrađuju u programskom jeziku Scratch, u razrednoj nastavi, većinom su jednostavni i svrha im je da učenici pokušaj shvatiti sintaksu programa kroz igru. Programi se objašnjavaju korak po korak i u udžbenicima ima malo zadataka slobodnog tipa. S obzirom da je programiranje osmišljeno kao slaganje kockica, učenicima je približeno slaganje programa te isto može postati intuitivno. Programiranje u programskom jeziku Python donosi veliku promjenu u ueničko poimanje programiranja, postaje sve ono čega se učenici boje – riječi koje

ne razumiju na prvi pogled. Iako je sintaksa programskog jezika Python vrlo jednostavna u usporedbi s ostalim jezicima, ona dolazi kao šok i velika promjena od igre koju su igrali u programskom jeziku Scratch. Zadaci postaju kompleksniji, puno ih je više i nisu potpuno intuitivni učenicima. Iako u udžbenicima nedostaje zadataka otvorenog tipa i za programski jezik Python. Učenik odjednom nema više Mačka ispred sebe s kojim može upravljati kako želi, sada mora krenuti od nule i sve napraviti koristeći se „strašnim“ riječima.

Tablica 3: Zadatke u programskom jeziku Scratch rješavao/rješavala sam s lakoćom.

	Frequency	Percent
Valid Slazem se	87	38.0%
U potpunosti se slazem	87	38.0%
Niti se slazem niti se ne slazem	39	17.0%
Ne Slazem se	8	3.5%
Uopce se ne slazem	8	3.5%
Total	229	100.0%

Iz tablice 3 može se iščitati kako 174 učenika, što je ukupno 76% ispitanika, smatra da su zadaci u programskom jeziku Scratch bili jednostavni. Smisao programskog jezika Scratch je da se učenici igraju i tako uče, no postavlja se pitanje hoće li išta naučiti ukoliko su zadaci u toj mjeri jednostavni?

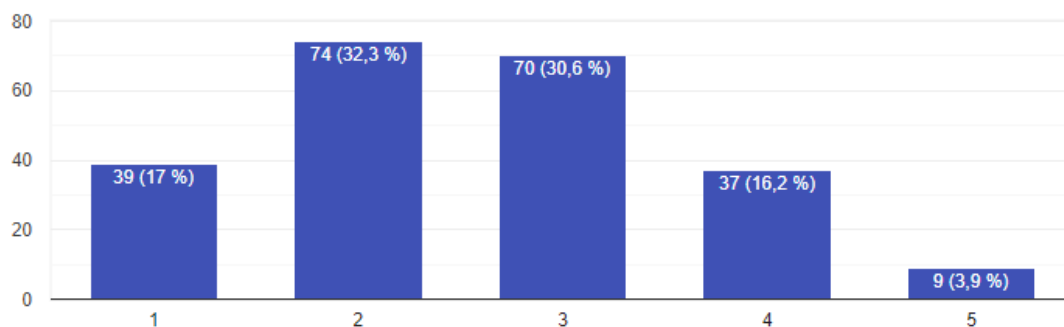
Tablica 4: Zadatke u programskom jeziku Python smatram jednakom teškim kao zadatke u programskom jeziku Scratch.

	Frequency	Percent
Valid Ne Slazem se	89	38.9%
Niti se slazem niti se ne slazem	54	23.6%
Slazem se	50	21.8%
Uopce se ne slazem	20	8.7%
U potpunosti se slazem	16	7.0%
Total	229	100.0%

Učenicima je postavljeno pitanje da usporede težinu zadataka u programskom jeziku Scratch sa zadacima u programskom jeziku Python. Iz tablice 4 vidljivo je kako 89 učenika, što čini 38,9% ispitanika, smatra zadatke u programskom jeziku Python težima nego u programskom jeziku Scratch. Rezultati su naime kod dosta učenika odskočili od očekivanog, pa je tako čak 50 učenika, što čini 21,8% ispitanika, reklo kako se slaže s time da su zadaci u programskom jeziku Python jednako zahtjevni kao i zadaci u programskom jeziku Scratch, dok se njih 54

(23,6%) nije moglo odlučiti. Ipak, kada se zbroje samo oni koji smatraju da su zadaci u programskom jeziku Python zahtjevniji dobijemo brojku od 109 učenika, dok ukupno njih 74 smatra da su zadaci podjednako teški.

Učenicima je zatim bilo postavljeno pitanje da ocjene težinu programskog jezika Scratch i programskom jeziku Python. Mogli su birati u rasponu od 1 do 5 – ocjena 1 predstavlja iznimno jednostavno, a ocjena 5 predstavlja iznimno zahtjevno.

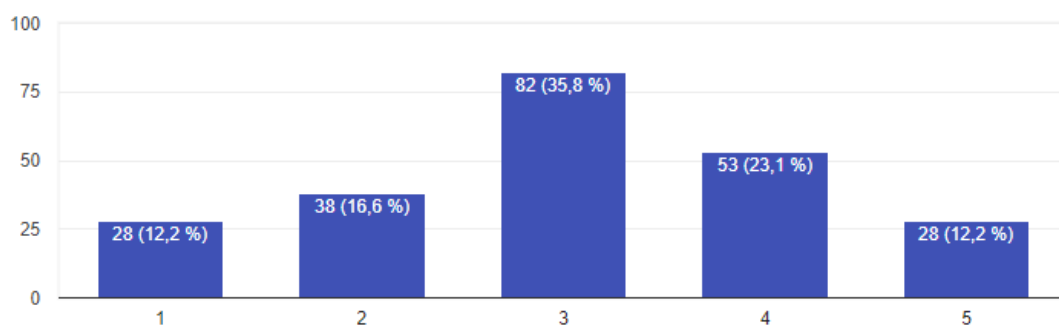


Slika 18: Ocjeni težinu programskog jezika Scratch ocjenom od 1 do 5.

Tablica 5: Srednja vrijednost i standardna devijacija

Ocjeni težinu programskog jezika Scratch ocjenom od 1 do 5.		
N	Valid	229
	Missing	0
Mean		2.58
Std Dev		1.07

Iz slike 18 vidljivo je kako 39 učenika, što čini 17% ispitanika, smatra kako je programski jezik Scratch iznimno jednostavan, dok njih 9, što čini 3,9% ispitanika, smatra kako je programski jezik Scratch iznimno zahtjevan. Učenici su u prosjeku težinu programskog jezika ocijenili s 2,58, dok je standardna devijacija vrlo mala, 1,07, što se može vidjeti u tablici 5.



Slika 19: Ocjeni težinu programskog jezika Python ocjenom od 1 do 5.

Tablica 6: Srednja vrijednost i standardna devijacija

Ocjeni težinu programskog jezika Python ocjenom od 1 do 5.	
N Valid	229
Missing	0
Mean	3.07
Std Dev	1.17

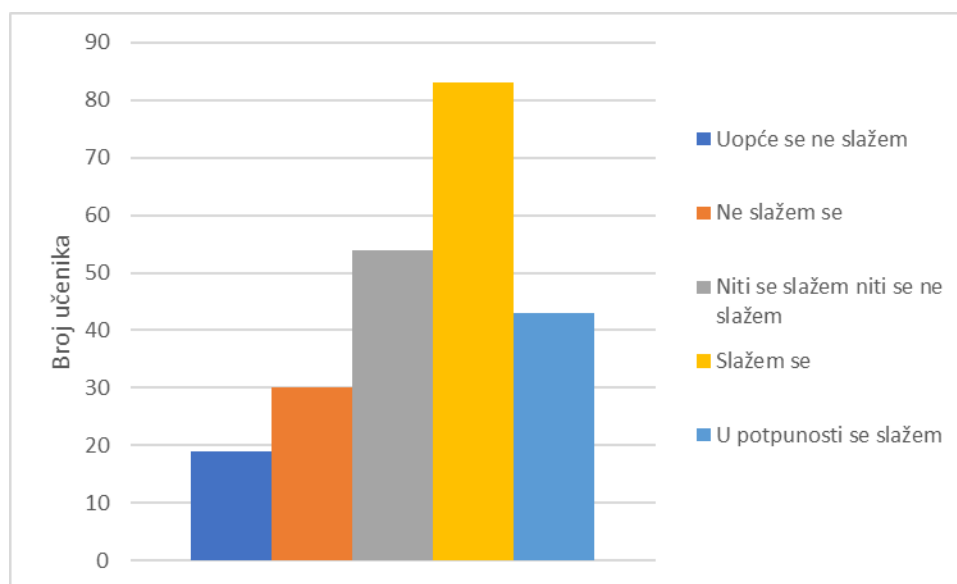
Iz slike 19 vidljivo je kako 28 učenika, što čini 12,2% ispitanika, smatra kako je programski jezik Python iznimno jednostavan, dok njih 28, što čini 12,2% ispitanika, smatra kako je programski jezik Python iznimno zahtjevan. Učenici su u prosjeku težinu programskog jezika ocjenili s 3,07, dok je standardna devijacija 1,17.

U ovom slučaju prosječne ocjene su vrlo slične, no to daje potpuno pogrešnu sliku stvarnih ocjena. Ako uzmemo samo krajnje ocjene i maknemo ocjenu 3 (prosječna težina), dobivamo potpuno drugačije rezultate. Pa je tako ocjenu 1 i 2 (iznimno jednostavno i jednostavno) za zahtjevnost programskog jezika Scratch dalo 113 učenika što, nakon što izbacimo učenike koji su dali ocjenu 3, čini 71,1% ispitanika. Ocjenom 4 i 5 (zahtjevan i izrazito zahtjevan) zahtjevnost programskog jezika Scratch ocjenilo je 36 učenika, što čini 28,9% ispitanika. Naspram toga, ocjene 1 i 2 za zahtjevnost programskog jezika Python dalo 66 učenika što, nakon što izbacimo učenike koji su dali ocjenu 3, čini 44,9% ispitanika. Ocjenom 4 i 5 zahtjevnost programskog jezika Python ocjenio je 81 učenik, što čini 55,1% ispitanika.

Jasno je kako je programski jezik Python zahtjevniji od programskog jezika Scratch, ali zadaci bi se trebali prilagođavati dobi učenika, što bi samim time trebalo značiti da bi iz učeničke perspektive programski jezici trebali biti jednake težine. Nesrazmjer težine dvaju jezika učenicima stvara problem te samim time gube volju za učenjem, ukoliko nisu temeljito i dobro motivirani.

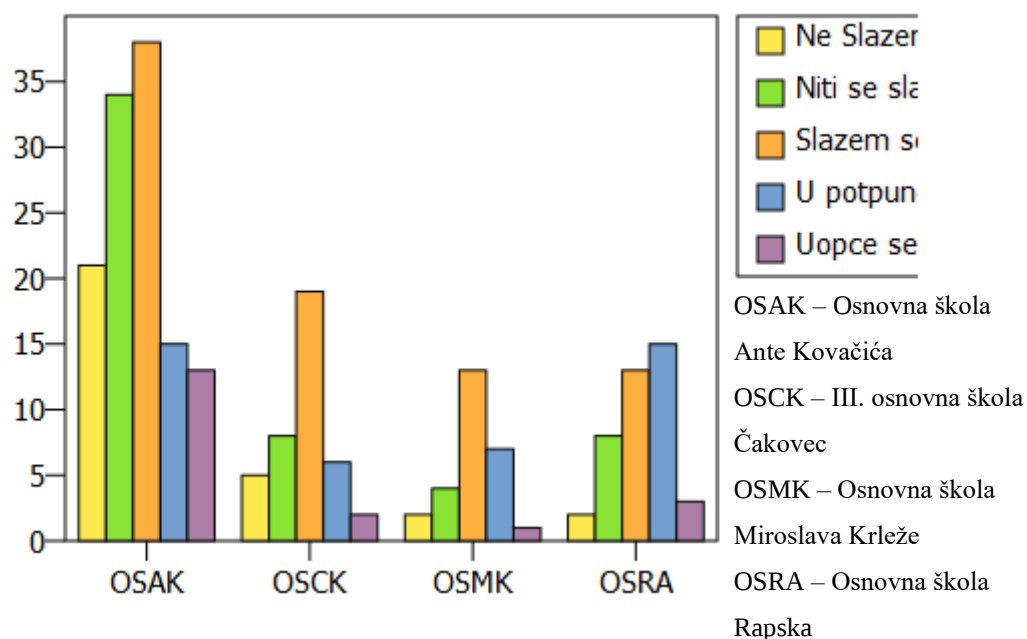
6.3 Rezultati utjecaja učenja programskog jezika Scratch na prilagodbu u programskom jeziku Python

Smisao uvođenja programskog jezika Scratch u osnovne škole je prilagodba učenika na logično, algoritamsko, razmišljanje. Učenici u programskom jeziku Python moraju raditi istu stvar kao i u programskom jeziku Scratch – graditi. Razlika je u tome što se u programskom jeziku Scratch gradi već otprije definiranim blokovima, dok se u programskom jeziku Python gradi riječima (naredbama). Kada učenik dođe do spoznaje da „gradi“ program, samo učenje postaje znatno lakše.



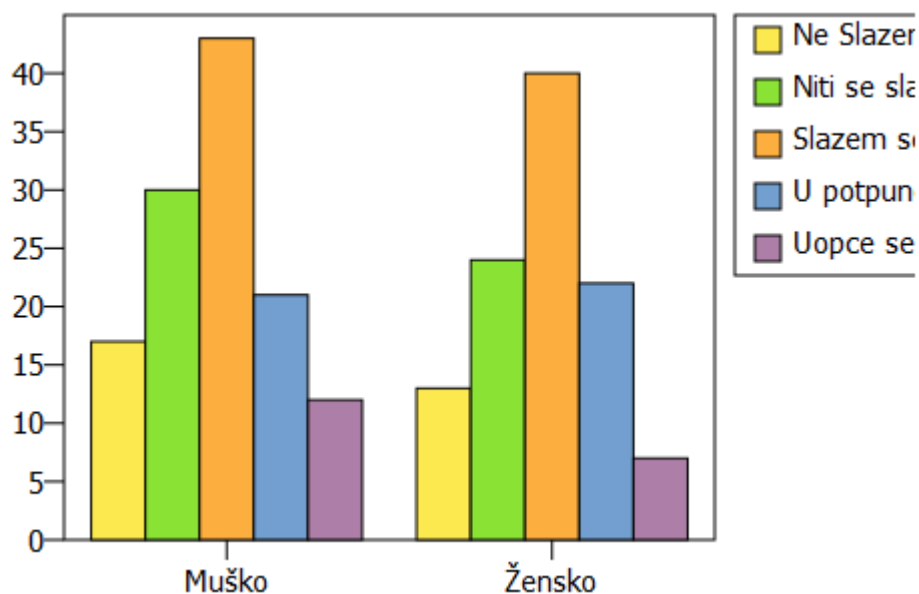
Slika 20: Učenje programskog jezika Scratch pomoglo mi je u počecima učenja programskog jezika Python.

Na slici 20 možemo vidjeti kako se 126 učenika, što čini 55% ispitanika, slaže ili u potpunosti slaže s time da im je učenje programskog jezika Scratch pomoglo u učenju programskog jezika Python. No, ovdje su rezultati nešto lošiji nego na prijašnjim grafovima. Ovdje se 49 učenika, što čini 21,4% ispitanika, ne slaže ili uopće ne slaže s time da im je učenje programskog jezika Scratch pomoglo u učenju programskog jezika Python. To je nezanemariv broj učenika. Također, veliki broj učenika, njih 54, što čini 23,6% ispitanika, se nije moglo odlučiti jesu li imali koristi od učenja programskog jezika Scratch.



Slika 21: Odnos koristi učenja programskog jezika Scratch u učenju programskog jezika Python po školama

Kada uspoređujemo (slika 21) odnos koristi učenja programskog jezika Scratch u učenju programskog jezika Python po školama vidimo kako 23 učenika, što čini 56,1% ispitanika, Osnovne škole Rapska smatra kako su imali koristi od učenja programskog jezika Scratch. Veliki broj učenika Osnovne škole Ante Kovačića, njih 34, što čini 28,1% ispitanika, prema rezultatima istraživanja, smatra kako nisu imali koristi od navedenog.



Slika 22: Odnos koristi učenja programskog jezika Scratch u učenju programskog jezika Python po spolu

Kada uspoređujemo (slika 22) odnos koristi učenja programskog jezika Scratch u učenju programskog jezika Python po spolu vidimo kako su rezultati vrlo slični, što znači da spol, kao nezavisna varijabla, ne utječe na smatranje koristi učenja programskog jezika Scratch u učenju programskog jezika Python po spolu.

7 Rasprava

Rezultati istraživanja pokazali su da su učenici u velikom dijelu samouvjereni u svoje poznavanje programskog jezika Scratch te da su imali koristi od njega u prijelazu na programski jezik Python. Učenicima se u velikom broju, njih čak 70,7%, svidjelo korisničko sučelje programskog jezika Scratch. Znanje koje su stekli smatra zadovoljavajućim 76% ispitanika, što znači da učitelji rade dobar posao u poučavanju programskog jezika Scratch. Sličan postotak vrijedi za svaku od četiri navedene škole. Zadatke u programskom jeziku Scratch njih 71,2% rješavalo je s lakoćom, što bi moglo značiti da su zadaci previše jednostavni. Uvođenje zadataka otvorenih tipa moglo bi produbiti razumijevanje programskih jezika i učiniti učenje zabavnijim. Čak 113 učenika, što čini 49,3% učenika smatra kako težina programskog jezika Scratch nije zahtjevnja, dok njih 48, što čini 20,1% ispitanika, smatra kako je težina programskog jezika Scratch zahtjevnja. Kod programskog jezika Python situacija je nešto drugačija, 66 učenika, što čini 28,8% ispitanika, smatra kako težina programskog jezika Python nije zahtjevnja, dok njih 81, što čini 35,3% ispitanika, smatra kako je težina programskog jezika Python zahtjevnja. Iz navedenog se može zaključiti kako učenici smatraju programski jezik Python znatno težim od programskog jezika Scratch. U konačnici, 126 učenika, što čini 55% ispitanika, smatra kako im je učenje programskog jezika Scratch pomoglo mi je u počecima učenja programskog jezika Python, dok njih 49, što čini 21,4% ispitanika, smatra kako im učenje programskog jezika Scratch nije pomoglo u počecima učenja programskog jezika Python. Dakle, više od polovine učenika smatra da su imali koristi od navedenog, no očito, globalno gledano, uvođenje programskog jezika Scratch nije u potpunosti postiglo željeni rezultat.

8 Zaključak

Programiranje je sve zastupljenije u današnjem obrazovanju pa tako i kurikulum nastavnog predmeta Informatike prati taj trend. Prema udžbenicima Školske knjige i Alfe programski jezici koji se uče po redovnom programu su Scratch i Python. Slijed učenja programiranja osmišljen je tako da se u razrednoj nastavi obrađuje programski jezik Scratch, dok je se u razrednoj nastavi obrađuje programski jezik Python. Udžbenici oba izdavača imaju ograničen, manji, broj zadataka vezanih za programski jezik Scratch i za svaki zadatak postoji predložak odgovara. Broj zadataka u oba udžbenika je znatno veći u predmetnoj nastavi i zadaci su nešto zahtjevniji te ne nude predložak odgovara za većinu zadataka.

Učenje programskih jezika moguće je i u izvannastavnim aktivnostima, koje omogućuju učenicima da detaljnije nauče pojedine stvari te ih bolje usvoje. Uz izvannastavne aktivnosti, tu su i natjecanja iz Informatike na školskoj, županijskoj i državnoj razini koja su veoma kompetitivna te pokazuju da je nastava Informatike na visokoj razini.

Istraživanje je provedeno s ciljem utvrđivanja koristi učenja programskog jezika Scratch u kasnijem učenju programskog jezika Python kod učenika petih razreda iz četiri različite osnovne škole. Rezultatima je utvrđeno kako je učenje programskog jezika Scratch uistinu korisno u učenju programskog jezika Python te bi se trebalo poticati. Iako rezultati nisu na najvišoj razini, dobri su pokazatelj napretka nastavnog predmeta Informatike kada je u pitanju programiranje.

Učenicima se treba pružiti mogućnost da pomoću nečeg jednostavnijeg, kao što je programski jezik Scratch, u ranoj dobi nauče sintaksu i algoritme, ne kao teorijske pojmove, već da razviju instinkt i razumijevanje istih. U odličnoj su prilici da igranjem postignu vrlo korisna znanja, što će nam svima služiti u budućnosti.

LITERATURA

1. Đurđević, I. (2014). Procjene studenata Učiteljskog studija o tri računalna programa namijenjena malim početnicima u programiranju. Radovi Zavoda za znanstveni i umjetnički rad u Požegi, 3, (str. 93-108).
2. Bubica, N., Mladenović, M., Boljat, I. (2013). Programiranje kao alat za razvoj apstraktnog mišljenja, (str. 1-11), URL: https://bib.irb.hr/datoteka/702093.Programiranje_kao_alat_za_razvoj_apstraktnog_miljenja-CUC-zbornik.pdf (pristupljeno 01.09.2022.)
3. Bubica, N. (2014). Strategije poučavanja i faktori koji utječu na unapređenje znanja programera početnika, URL: <http://www.pmfst.eu/wpcontent/uploads/2014/06/Istraziva--ki-seminar1-Bubica.pdf>
4. Ministarstvo znanosti i obrazovanja (MZO), Kurikulum nastavnog predmeta Informatike za osnovne i srednje škole, 2018.; URL: https://narodne-novine.nn.hr/clanci/sluzbeni/2018_03_22_436.html
5. Valić, B., Radovan, A., Pavlović, D. (2013)., Korištenje programskog jezika Scratch za podučavanje osnova programiranja od malih nogu.; URL: https://radovi2013.cuc.carnet.hr/modules/request.php?module=oc_proceedings&action=view.php&a=Accept&id=50&type=2 (pristupljeno 01.09.2022.)
6. Buklijaš, S. (2010). Scratch- Vizualni programski jezik za djecu, URL: https://cuc.carnet.hr/2010/images/b1_52a14.pdf?dm_document_id=182&dm_dnl=1 (pristupljeno 01.09.2022.)
7. Vukotić, D., Tanković, N. (2011). Alati za razvoj aplikacija bez kodiranja.; URL: https://bib.irb.hr/datoteka/523767.Vukoti_TankoviAlati_za_razvoj_aplikacija bez_kodiranja.pdf (pristupljeno 01.09.2022.)
8. Otvoreno društvo za razmjenu ideja (ODRAZI), SCRATCH - Vodič za korisnike i korisnice, Zagreb, URL: http://www.odrazise.org/scratch/Scratch_vodic.pdf (pristupljeno 01.09.2022.)

9. Scratch - Imagine, Program, Share (Scratch community), URL: <https://scratch.mit.edu/> (pristupljeno 01.09.2022.)
10. Scratch Wiki, URL: https://en.scratch-wiki.info/wiki/Scratch_3.0 (pristupljeno 01.09.2022.)
11. Code Club, URL: <https://international.codeclub.org/>, (pristupljeno 06.10.2022.)
12. Web stranice natjecanja iz informatike, Agencija za odgoj i obrazovanje, URL: <https://informatika.azoo.hr/>, (pristupljeno 01.09.2022.)

POPIS SLIKA

<i>Slika 1: Logo programskog jezika Scratch</i>	6
<i>Slika 2: Razvojno okruženje za Scratch 3.0</i>	9
<i>Slika 3: Naslovnica udžbenika iz informatike za 2. razred osnovne škole, e-Svijet 2</i>	12
<i>Slika 4: Naslovnica priručnika za učenje Scratcha nakladnika Školska knjiga, Istražite Scratch 3.0</i>	13
<i>Slika 5: Naslovnica udžbenika iz informatike za 5. razred nakladnika Alfa, LikeIT</i>	14
<i>Slika 6: Logo programskog jezika Python</i>	14
<i>Slika 7: Visual Studio Code, razvojno okruženje za programski jezik Python</i>	15
<i>Slika 8: Udžbenik iz informatike za 5. razred osnovne škole, nakladnika Alfa, #mojPortal5</i>	16
<i>Slika 9: Priručnik za učenje programskog jezika Python, nakladnik Školska knjiga, Algoritmi u Pythonu</i>	17
<i>Slika 10: Gradivo nastavne cjeline Programiranje, informatika za 5. razred osnovne škole</i>	18
<i>Slika 11: Spol sudionika</i>	21
<i>Slika 12: Škole čiji su učenici sudjelovali u istraživanju</i>	21
<i>Slika 13: Usporedba rezultat poznavanja osnovnih i naprednih funkcija programa stupčastim dijagramom</i>	24
<i>Slika 14: Usporedba rezultata usvojenosti naprednih funkcija po spolu</i>	24
<i>Slika 15: Svidjelo mi se korisničko sučelje u programskom jeziku Scratch (mogu s lakoćom pronaći naredbe)</i>	25

<i>Slika 16: Zadovoljan/Zadovoljna sam znanjem koje sam stekao/stekla u programskom jeziku Scratch.</i>	25
<i>Slika 17: Odnos zadovoljstva stečenim znanjem po školama</i>	26
<i>Slika 18: Ocjeni težinu programskog jezika Scratch ocjenom od 1 do 5.....</i>	28
<i>Slika 19: Ocjeni težinu programskog jezika Python ocjenom od 1 do 5.....</i>	28
<i>Slika 20: Učenje programskog jezika Scratch pomoglo mi je u počecima učenja programskog jezika Python.....</i>	30
<i>Slika 21: Odnos koristi učenja programskog jezika Scratch u učenju programskog jezika Python po školama.....</i>	31
<i>Slika 22: Odnos koristi učenja programskog jezika Scratch u učenju programskog jezika Python po spolu</i>	31

POPIS TABLICA

<i>Tablica 1: Savladao/Savladala sam osnovna znanja unutar programskog jezika Scratch.....</i>	23
<i>Tablica 2: Savladao/Savladala sam napredna znanja unutar programskog jezika Scratch.....</i>	23
<i>Tablica 3: Zadatke u programskom jeziku Scratch rješavao/rješavala sam s lakoćom.</i>	27
<i>Tablica 4: Zadatke u programskom jeziku Python smatram jednakom teškim kao zadatke u programskom jeziku Scratch.</i>	27
<i>Tablica 5: Srednja vrijednost i standardna devijacija.....</i>	28
<i>Tablica 6: Srednja vrijednost i standardna devijacija.....</i>	29

Izjava o samostalnoj izradi rada

Izjavljujem da je moj diplomski rad izvorni rezultat mog rada te da se u izradi istog nisam koristio drugim izvorima osim onih koji su u njemu navedeni.
