

# Motivacija učenika za računalno programiranje u nastavi informatike

---

**Tonković, Aleksandra**

**Master's thesis / Diplomski rad**

**2022**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:137:805364>

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

*Download date / Datum preuzimanja:* **2024-11-05**



*Repository / Repozitorij:*

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



SVEUČILIŠTE JURJA DOBRILE U PULI  
FAKULTET INFORMATIKE



Aleksandra TONKOVIĆ

**MOTIVACIJA UČENIKA ZA RAČUNALNO  
PROGRAMIRANJE U NASTAVI INFORMATIKE  
DIPLOMSKI RAD**

Pula, rujan 2022. godine

SVEUČILIŠTE JURJA DOBRILE U PULI  
FAKULTET INFORMATIKE

Aleksandra TONKOVIĆ

**MOTIVACIJA UČENIKA ZA RAČUNALNO  
PROGRAMIRANJE U NASTAVI INFORMATIKE**

DIPLOMSKI RAD

**JMBAG:** 0131015119

**Studijski smjer:** Sveučilišni diplomski studij informatika – nastavnički smjer

**Kolegij:** IT i edukacija

**Znanstveno područje:** Društvene znanosti

**Znanstveno polje:** Informacijske i komunikacijske znanosti

**Znanstvena grana:** Informacijski sustavi i informatologija

**Mentor:** izv. prof. dr. sc. Ivan Pogarčić

Pula, rujan 2022. godine



## IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Aleksandra Tonković, kandidat za magistra nastave informatike, mag. educ. inf., ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da nijedan dio Diplomskoga rada nije napisan na nedozvoljeni način, odnosno da je prepisan iz kojega necitiranog rada te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student



U Puli 9. rujna 2022.



## IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Ja, Aleksandra Tonković dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom *Motivacija učenika za računalno programiranje u nastavi informatike* koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama. Za korištenje autorskoga djela na gore navedeni način ne potražujem naknadu.

U Puli 9. rujna 2022.

Potpis

---

## ZAHVALA

Iskreno i od srca zahvaljujem mentoru izv. prof. dr. sc. Ivanu Pogarčiću, prije svega na ljudskosti i što me svojom stručnošću usmjeravao prema cilju, kvalitetnom i poučnom pisanju diplomskog rada.

Neizmjerne hvala mojem prijatelju pedijatru Edgaru Glavašu, dr. med. koji je odvajao svoje dragocjeno vrijeme za dobronamjerne sugestije tijekom pisanja rada, iako često ni sam nije imao dovoljno vremena za sebe i obitelj.

Zahvaljujem dr.sc. Ines Virč na korisnim komentarima i pokazanom interesu tijekom lektoriranja rada.

Hvala mojoj dragoj prijateljici i kolegici dr. sc. Martini Lončar što je uvijek uz mene.

Hvala mojim sinovima, Mihaelu i Tomi, što su imali napretek strpljenja i podrške. Moj ponos.

Svima veliko hvala za ohrabrenje da ustrajem i ne odustajem.

## SADRŽAJ

|  |    |
|--|----|
| 1. UVOD  | 1  |
| 2. RAČUNALNO PROGRAMIRANJE – POSTAVKE                                    | 4  |
| 2.1. POVIJESNI PREGLED RAČUNALNOG PROGRAMIRANJA                          | 6  |
| 2.2. MOTIVACIJA  | 9  |
| 2.2. OBRAZOVNI PRISTUPI UČENJU PROGRAMIRANJA                             | 11 |
| 2.3. PROGRAMIRANJE U VIRTUALNOM OKRUŽENJU                                | 13 |
| 2.4. PROGRAMSKI JEZICI U OSNOVNOJ ŠKOLI                                  | 17 |
| 2.5. MOTIVACIJA UČENIKA U ŠKOLI  | 22 |
| 3. DOSADAŠNJA ISTRAŽIVANJA MOTIVACIJE UČENIKA ZA RAČUNALNO PROGRAMIRANJE | 24 |
| 4. RASPRAVA  | 39 |
| 5. ZAKLJUČAK   | 41 |
| 6. LITERATURA  | 42 |
| 7. SAŽETAK   | 60 |
| 8. SUMMARY   | 61 |

## 1. UVOD

Globalizacijsko vrijeme iziskuje sve veći utjecaj računalne tehnologije. Samim time porasla je potreba za informatičkom edukacijom u osnovnoškolskom i srednjoškolskom obrazovanju za potrebe privatnoga i društvenoga sektora. Veća je potražnja za informatičkim obrazovanjem (Balanskat i Engelhardt, 2015). Eksponencijalni rast tehnologija utjecao je na računalno programiranje. Učenje računalnog programiranja uključuje stjecanje teorijskog razumijevanja i učenje razvijanja programa u praksi, u kontekstu škole.

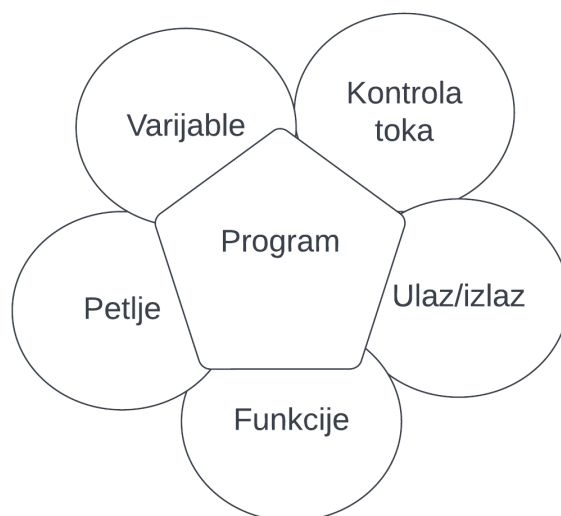
Kurikulumi nastave informatike u školama pojedinih država se izrazito razlikuju, što pokazuju i pojedina istraživanja u tom području. Stoga, nedosljednost kurikuluma ne daje konačne zaključke o najboljoj metod(ologij)i i motivaciji učenika za računalno programiranje u nastavi informatike. Izrazito je važno potaknuti djecu u što ranijoj dobi na računalno programiranje putem igre, kako bi se povećala motivacija i samopouzdanje kod učenika, ali i obrazovni rezultati. Motivacija učenika za pojedini predmet pozitivno utječe na njihova postignuća (Marsh i sur., 2005; Trautwein i Köller, 2005; Wilkins, 2004). Potvrđeno je da samopouzdanje učenika pozitivno djeluje na postignuća učenika (npr. Armstrong i Price, 1982; Lantz i Smith, 1981). Nedostaje rezultata istraživanja na tu tematiku, kao i mjernih instrumenata. Kako bi se bolje mogla odrediti najbolja metoda(ologij)a motivacija učenika za računalno programiranje u nastavi informatike, poželjna je procjena i evaluacija učenika i njihovih karakteristika te kurikulum u nastavi informatike.

Računalno programiranje u obrazovanju, samim time i u nastavi informatike, zauzima važno mjesto u istraživanju (Peko i sur., 2006; Reić Ercegovac, King i Behnke, 2005). Međusobna suradnja u obrazovnom sustavu, važan je element u proučavanju u prisutnosti nastavnika (Wilson i Stacey, 2004; Ko i Rossen, 2010). Važna je pedagoška uloga nastavnika (Berge, 1995, 2009) koja se odnosi na područje rada, stručnost, računalnu komunikaciju i olakšavanje međudjelovanja (Liu i sur., 2005). Suradnja nastavnika,



administrativnog osoblja, studenata i okoline pridonosi napretku nastave (Yang i Cornelious, 2005; Berge, 2009).

Aдекватni kurikulumi pridonose uspješnijem snalaženju s tehnologijom. Značajnu ulogu imaju aktivnosti, strategije i modeli u procesu računalnog programiranja (Johnson, 2002; Bezjak, 2009; Pierce i Jones, 1998; Putnam, 2001; Putnam i Borko, 2000) za vidljiv napredak učenika uzimajući u obzir tehnologije za današnje tehnološko društvo (Kelley i Kellam, 2009; Kovačević, 2012; Milat, 1996; Verbitsky, 2012). Današnje je društvo povećalo interes za programiranjem i programerima, što je dovelo do različitih informatičkih tečajeva u obrazovanju (Qian i Lehman, 2016), nastavi programiranja u školama (Saez-Lopez i sur., 2016, Moreno-Leon i sur., 2016). Računalno programiranje uključuje rješavanje problema (Calao i sur., 2015; Fessakis i sur., 2013), kreativno razmišljanje (Gupta i sur., Navarrete, 2013), algoritamsko razmišljanje (Hromkovic i sur., 2016), refleksivno razmišljanje (Kalelioglu, 2015). Povezuje različite aspekte matematike, znanosti i prirodnih i društvenih znanosti i jezika (Baki i Özpınar, 2007; Baytak i sur., 2011; Brown i sur., 2008; Burke, 2012; Ferrer-Mico i sur., 2012; Ke, 2014; Moreno-Leon i sur., 2016; Navarrete, 2013; Saez-Lopez i sur., 2016). Računalno programiranje se počinje učiti u osnovnoj školi (Han i sur., 2016; Heintz i sur., 2016). Računalno programiranje uključuje razmišljanje, način kodiranja, apstrakciju i analizu (Lye i Koh, 2014) i mentalne vještina (Benzer i Erümit, 2017).



**Slika 1.** Znanja koja je potrebno prenijeti na učenike kako bi mogli učinkovito programirati (autor)

Rad se sastoji od uvodnoga dijela, postavki računalnog programiranja, govori se o samim počecima i povijesti računalnog programiranja, učenju programskih jezika u osnovnoj školi, obrazovnoj dimenziji, virtualnim modelima učenja, motivaciji učenika u školi, rezultatima istraživanja na danu temu u svijetu te slijede rasprava, zaključak, literatura i sažetak.

U sadašnjem vremenu, digitalna pismenost važna je za svakog čovjeka. Potražnja za IT stručnjacima neprestano raste, a obrazovanje na fakultetskoj razini je sve popularnije. Postoje velike razlike među studentima, onih koji ne znaju programirati i/ili logički i algoritamski razmišljati. Stoga je važno informacijsku tehnologiju što više uključiti u kurikulum svake škole. Edukaciju računalnog programiranja potrebno je osigurati u obrazovnom sustavu, ali ne samo kao pasivno razumijevanje, već i interaktivno (su)djelovanje učenika. Tako će učenici jednog dana postati aktivni sudionici digitalnog doba, a samim time i razvijati kritički pristup. Usklađenost cjelokupnog obrazovnog sustava je važna, od opremljenosti, ujednačenog kurikuluma do edukacije učitelja.

## 2. RAČUNALNO PROGRAMIRANJE – POSTAVKE

Termin računalno razmišljanje uvodi Papert (1980) za opis misaonih radnji kod računalnih postupaka. Wing je (2006) objavila radove na temu računalno razmišljanje kao vještina za rješavanje problema koje dijete može svladati. Nardelli (2019) smatra da je računalno razmišljanje intrinzično povezano s programiranjem. Računalnim razmišljanjem moguće je riješiti probleme koji se ne mogu riješiti drukčije (Denning, 2017). Računalno programiranje je kognitivna sposobnost (npr. García Peñalvo i sur., 2016; Shute, Sun, i Asbell-Clarke, 2017).

Računalno razmišljanje je oblik rješavanja problema (npr. Lu i Fletcher, 2009; Wing, 2006; Yadav, Hong i Stephenson, 2016), pronalaženje algoritamskih rješenja (Barr i Stephenson, 2011.) koja su algoritamska i apstraktna (Lu i Fletcher, 2009.; Wing, 2008.) Računalno razmišljanje ima važnu ulogu u rješavanju problema.

Složeni problemi čije rješenje uključuje računalno razmišljanje su definirani s mnogo čimbenika. Rješenja takvih problema zahtijevaju procese generalizacije, apstrakcije i razvoj algoritamskih rješenja (Kazimoglu, 2013; Wing, 2006). Rješenje se smatra algoritamskim kada je nakon svakog koraka jasno koji korak treba izvršiti kao sljedeći ili koje uvjete treba uzeti u obzir da bi se odlučilo koji će se korak izvršiti kao sljedeći. U svrhu upoznavanja djece s programiranjem, smatra se da se takva algoritamska rješenja sastoje od osnovnog skupa koncepata računskog mišljenja u obliku blokova kontrolnog toka, kao što su petlje, uvjeti i drugo (Brennan i Resnick, 2012).

Računalno razmišljanje uključuje kognitivne procese koji su iznimno važni u rješavanju računalnih problema (García Peñalvo i sur., 2016.). Primjenjivost je iznimno široka i nadilazi razinu računalstva. Već 1970-ih godina predloženo je da učenje programiranja može pomoći djeci da izražavaju svoje vlastite kognitivne procese s ciljem promišljanja.

Papert (2005) smatra da je poučavanje programiranja u osnovnoj školi važno i korisno za razvoj dječjih sposobnosti rasuđivanja.

Danas je trend sve većeg informatičkog obrazovanja u obliku samoobrazovanja (Code.org; Code; Kodable), kao dio kurikuluma osnovnih škola propisan od strane MZO-a. U cijelom svijetu, popularizacija za programiranjem u sustavu obrazovanja je u porastu s ciljem logičkog razmišljanja i rješavanja problema. Sve veći broj država integrira i/ili planira integrirati računalno razmišljanje i programiranje u kurikulum nastave informatike na razini osnovne škole (npr. Balanskat i Engelhardt, 2015).

Britansko kraljevsko društvo dalo je inicijativu za pokretanje programa 2014. godine vezano za računalno programiranje u školi. Uvedeno je kao predmet za sve učenike svake godine osnovnog i srednjeg obrazovanja (Dredge, 2014).

Osnovana je Europska inicijativa za kodiranje 2014. godine s ciljem promicanja integracije računskog razmišljanja i programiranja u službene kurikulume (Balanskat i Engelhardt, 2015), kao i pridavanje važnosti računalnom programiranju. U SAD-u postoje brojni programi kao što su CE21, CPATH, BPC, CS 10K, CS4HS ili Computing in the Core (Wing, 2014). Smisao tih programa je pripremanje nastavnika za informatičko obrazovanje. Sličan razvoj je zamijećen u Kini, Koreji i Singapuru (Wing, 2011, 2014).

## 2.1. POVIJESNI PREGLED RAČUNALNOG PROGRAMIRANJA

Povijesni pregled računalnog programiranja je povezan s hardverskim razvojem. Sami počeci su bili u SAD-u i Australiji. Prvo programsko računalo je proizvedeno u Australiji i testirano je 1949. godine. To računalo je kasnije korišteno kao radni stroj na sveučilištu u Melbourneu, a sada se nalazi u muzeju u navedenom gradu. Zahtjevi digitalnih uređaja imali su potrebu za programerima. To su u početku bili matematičari inženjeri. Programiranje se vršilo u nekom od oblika strojnog koda. Godine 1960. bila su poznata 73 programska jezika, a jedanaest godina kasnije 164. (Sammet, 1972, 1981).

Programski odbor za Povijest programskih jezika (HOPL) je u Los Angelesu procijenio važnost jezika putem sljedećih kriterija:

- (1) jezik je u uporabi najmanje 10 godina
- (2) jezik ima značajne utjecaje
- (3) jezik je još uvijek u uporabi (Bergin i Gibson, 1996).

FORTRAN-a je započeo svoj razvoj 1954. godine (Parker i Davey, 2017).

ALGOL je objavljen 1958. godine, koristio se u tečajevima računalnog programiranja, bio je precizan i koristan za algoritme (Keet, 2004). COBOL je razvijen 1959. godine i naveliko je korišten u svijetu u smislu poslovne aplikacije.

U Velikoj Britaniji je BASIC bio izuzetno važan. Njegova primjena je postigla trend u srednjim školama. Uvedena su mikroračunala 1970-ih godina. Većina sveučilišta je prihvatila programski jezik Pascal kao uvodni programski jezik. Bio je široko dostupan i dizajnerski pristupačan.

Kasnije je akademskoj zajednici Pascal postupno zamijenjen s drugim programskim jezicima počevši s C-om i C++-om, da bi se na kraju prebacio na Javu i C#.

Već je 1980-ih godina porasla dostupnost programskih jezika. Tharp (1982) nabroja dostupne jezike za uvodno računalno programiranje FORTRAN, COBOL, Jovial, Ada, ALGOL, Pascal, Pl/I i Spitbol.

Luker (1989) je proučavao funkcionalno programiranje, proceduralno programiranje, objektno orijentirano programiranje i paralelno programiranje.

Howattovi (1995) kriteriji za evaluaciju programskih jezika su uključivali široki jezični dizajn i implementaciju, ljudske faktore, programsko inženjerstvo i domenu primjene.

Roberts (2004a) je uočio izazove s kojima se učenici suočavaju kod uvodnog računalnog programiranja:

- (1) porastao je broj programskih detalja koje učenici moraju savladati
- (2) mijenjaju se jezici, knjižnice i alati o kojima ovise uvodni tečajevi brže nego u prošlosti.

Gee, Wills i Cooke (2005) su uočili sve veću upotrebu skriptnih jezika za podučavanje računalnog programiranja.

Tijekom više godina izmišljeni su jezici za rješavanje problema i/ili olakšano učenje algoritama. Postavlja se često pitanje koje programske jezike bi trebalo učiti i primjenjivati, odabrati onaj koji se često koristi ili onaj koji najbolje podržava razvoj i motivaciju kod učenika?! Tako su nastala dva argumenta vezana za odabir programskog jezika. Jedan je pragmatični odabir, a drugi obrazovni. Pragmatični pristup preporučuje odabir jezika koji će učenicima pomoći da dobiju posao prema zahtjevima tržišta. King (1992) smatra da odabirom programskog jezika u trendu jezika ima praktične vrijednosti koje se odnose i na povećanu motivaciju učenika za učenje programskog jezika za koji znaju da je tražen u svijetu i široko primjenjiv.

Novi programski jezik treba zadovoljiti potrebe za nastavom s ciljem učinkovitosti. No, to nema jamstva da će poslodavac i dalje tražiti taj programski jezik kada se učenik jednoga dana nađe na tržištu rada. Trendovi u svijetu nisu uvijek razumljivi. Programeri koji su

krenuli po prvi put na tržište rada, 75 % njih je znalo C, dok je samo 25 % znalo Adu (Parker i Davey, 2017). Trošak kodiranja u Adi je otprilike upola manji od troška kodiranja u C-u i otkriveno je 90 % manje grešaka nakon isporuke proizvoda (Zeigler, 1995).

Postoji nekoliko jezika dizajniranih za nastavu (Pascal, LOGO). Najčešće se programski jezici koji se koriste u nastavi, ne primjenjuju u industriji. Percepcija učenika je da ti programski jezici imaju manju praktičnu vrijednost.

Pedagoški pristup inzistira da jezik koji se koristi u početnom računalnom programiranju, nastava treba biti dizajnirana za podučavanje koncepata programiranja i rješavanja problema s minimalnom vremenskom složenošću. Nije postignut konačan dogovor oko toga koji je pristup najoptimalniji. Postoji potreba da znanstvenici detaljnije procjene interese i potrebe učenika te pomno odabiru najadekvatniji programski jezik za kasnije računalno programiranje primjenjivo na tržištu rada.

**Tablica 1.** Pregled povijesti razvoja programskih jezika.

| Godina | Jezik    |  |
|--------|----------|--|
| 1957.  | FORTTRAN | prvi implementirani jezik visoke razine                |
| 1960.  | COBOL    | prvi poslovno orijentirani programski jezik            |
| 1960.  | LISP     | za istraživanje lista i umjetne inteligencije          |
| 1962.  | APL      | matematička notacija kao oblik programiranja           |
| 1964.  | BASIC    | prvi jezik namijenjen ne-programerima                  |
| 1965.  | Simula   | prvi objektno orijentiran jezik                        |
| 1967.  | Logo     | prvi jezik namijenjen učenju, uvodi koncept "kornjače" |
| 1970.  | Pascal   | namijenjen učenju strukturiranog programiranja         |
| 1972.  | C        | prototip najpoznatije obitelji programskih jezika      |
| 1978.  | SQL      | relacijske baze podataka                               |

## 2.2. MOTIVACIJA

Prema Piagetu (1972), apstrakcija kod djece se razvija između 7. i 11. godine života. Ona je iznimno bitna za računalno razmišljanje i programiranje. Postoji sugestija da je u predškolskoj dobi prerano učiti djecu kako računalno razmišljati (Armoni i Gal-Ezer, 2014). U početku su djeca spremna kodirati kraće programe (Teague i sur., 2012). Važna je motivacija djece kroz didaktički i metodički pristup računalnom razmišljanju kao što su karte, papir i olovka radi lakšeg razumijevanja samih početaka programiranja.

Važno je kod djece stvoriti pozitivnu sliku o sebi te sposobnost i vjerovanje da mogu odraditi određeni zadatak. Istraživanja su pokazala da djeca koja vjeruju u svoju sposobnost, imaju više uspjeha (Oliver i Simpson, 1988).

Motivacija djece te upornost prema računalnom programiranju značajna je za postizanje uspjeha. Postoje četiri aspekta uvjerenja o motivacijskoj vrijednosti: intrinzičnoj vrijednosti, vrijednost postignuća, korisnosti i uvjerenje o cijeni (Eccles i sur., 1983; Wigfield i Eccles, 1992).

Uvjerenje o intrinzičnoj vrijednosti smatra da je učenik intrinzično motiviran i zainteresiran je za određeno područje učenja. Obuhvaća stav učenika prema području učenja kao i aktivnost za to područje u slobodno vrijeme. Pozitivan stav prema predmetu obavezan je uvjet za postizanja postignuća, ali može povećati angažman aktivnost učenika (Oliver i Simpson, 1988). Prediktor je dobrovoljnog angažmana učenika s predmetom u slobodno vrijeme (Durik, Vida i Eccles, 2006; Nagengast i sur., 2011). Uvjerenje u vrijednost postignuća u vezi s predmetom predstavlja važnost učenika za određeni predmet (Gaspard, 2017). Uvjerenje o korisnosti vrijednosti predmeta se odnosi na očekivanja učenika o korisnosti predmeta u različitim aspektima (su)života, svakodnevni život, škola, buduća karijera ili društveni život (Gaspard, 2017). Vjerovanje o unutarnjoj vrijednosti i uvjerenje o vrijednosti postignuća obuhvaćaju unutarnje motivacijske čimbenike, a uvjerenje o korisnosti ekstrinzične motivacijske čimbenike (Trautwein i sur., 2013).

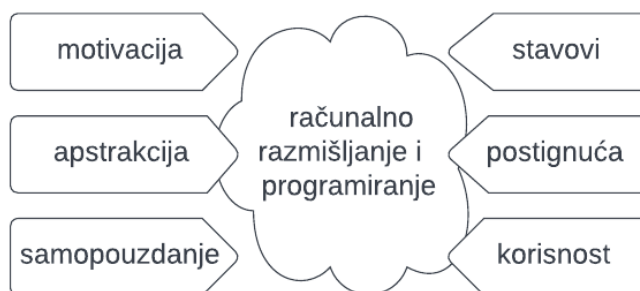


Motivacija je iznimno važna za postizanje obrazovnih postignuća učenika (Marsh i sur., 2005.; Trautwein i Köller, 2005.; Wigfield i Cambria, 2010.; Wilkins, 2004). Pozitivno samopoimanje sebe kao dio samopouzdanja pozitivno utječe na obrazovna postignuća učenika, posebno u prirodoslovlju i matematici, kao i daljni poticaj za nastavak obrazovanja u tim područjima (npr. Armstrong i Price, 1982; Lantz i Smith, 1981; Oliver i Simpson, 1988; Wilkins, 2004). Poželjno je procijeniti važnost samopoimanja za računalno programiranje, kao i motivacije za računalno programiranje, napraviti evaluaciju obrazovnih postignuća u (ranom) informatičkom obrazovanju.

Evaluirano je nekoliko upitnika za mjerenje stavova učenika prema računalnom programiranju. Jedna vrsta upitnika je više usmjerena na računalno programiranje, dok druga vrsta procjenjuje stavove prema korištenju računala.

Jedan od instrumenata je Computing Attitude Survey (CAS) za procjenu stavova studenata prema rješavanju problema i njihovih uvjerenja o prirodi znanja u računalnim znanostima (Dorn i Elliott Tew, 2015). Postoji i mjerni instrument za mjerenje stava učenika srednjih škola prema informatici i informacijskoj tehnologiji (Heersink i Moskal, 2010) te anketa o stavovima o računalnom programiranju za studente preddiplomskih znanosti i inženjerstva (Hoegh i Moskal, 2010).

Druga kategorija koristi mjerne instrumente Inventar uvjerenja o mikroračunalima (MBI) (Riggs i Enochs, 1993) za procjenu samoučinkovitosti učenika srednjih škola i uvjerenja o očekivanim ishodima vezanim uz korištenje računala (Riggs i Enochs, 1993), mjeri stavove srednjoškolaca prema interakciji s računalima (Selwyn, 1997). Ovi mjerni instrumenti ne uključuju povezanosti motivacije i samopoimanja učenika s računalnim programiranjem.



**Slika 2.** Međuodnos intrinzičnih i ekstrinzičnih karakteristika prema računalnom razmišljanju i programiranju (autor)

## 2.2. OBRAZOVNI PRISTUPI UČENJU PROGRAMIRANJA

Inovativni obrazovni modeli u učenju programiranja odgovaraju zahtjevima i očekivanju djece u digitalno doba. Ti modeli djeluju tako da djeca stječu određena znanja i razvijaju odgovarajuće programske vještine. Programiranje zahtijeva ne samo razumijevanje djece i poznavanje teorije, već i njihovu sposobnost primjene na rješavanje postavljenih stvarnih problema u životnim svakodnevnim situacijama.

Cilj novih teorija je da pomognu djeci da shvate koncept programiranja na eksperimentalan, motivacijski i zabavan način. Znanstvenici spominju različite pristupe podučavanja programiranja bazirano na različitim pedagoškim i didaktičkim okvirima, modelima i/ili obrazovnim pristupima.

Holistički pristup učenju se temelji na digitalnim igrama za osnovnu školu izvan škole. Novi trend koji prevladava u obrazovnim tehnologijama je nova paradigma podučavanja. Ona se temelji na upotrebi procesa gamifikacije. Budući da su djeca upoznata popularnim platformama za igranje, model učenja zasnovanog na igrama (GBL) ima za cilj korištenje igre za učenje (ne samo) programiranja. Neki znanstvenici tvrde da ova metoda djecu priprema da udovolje zahtjevima suvremenog digitalnog doba tako da ih se uči da budu inovativni, kreativni i prilagodljivi. Metoda uključuje nekoliko okvira predloženih za

dizajn edukativnih igara. Četiri-dimenzionalni okvir (Freitas i Jarvis, 2006) sastoji se od četiri osnovna principa: kontekst, reprezentacija, učenik i pedagogija.

Konceptualni okvir ima za cilj pružiti upute za dizajn edukativnih igara (Looi i sur., 2014). Okvir "Dizajn, igraj i doživi" (Tekinbas i Zimmerman, 2010) ima za cilj osmisliti igre za učenje, u edukativne svrhe. Okvir za modeliranje opisuje učenje kao kružni proces koji konstruira kognitivne dijelove kroz aktivnosti unutar same igre (Kiili, 2005). Višedimenzionalni okvir (Song i Zhang, 2008.) je predložen u pogledu pravilnog dizajna edukativne igre. Okvir modela dizajna obrazovnih igara (Ibrahim i Jaafar, 2009) uključuje dizajn, pedagogiju i čimbenike učenja. Okvir za masovnu multiplayer online igranje uloga (Malliarakis et al., 2012.) ima za cilj upoznati djecu srednje škole, koji su početnici u računalnom programiranju, s varijablama, petljama i sl. te ih uključiti u algoritamsku logiku. Za djecu upoznatu s kodiranjem, (Skalka i Drlík, 2018) određen je nastavni okvir za mikroučenje za poboljšanje postignutih vještina programiranja.

Na taj način djeca koriste ICT kao dio digitalne tehnologije. Ona im pomaže da kroz digitalne igre napreduju u svojem učenju programiranja kroz zabavan način. Digitalne tehnologije temeljene na igricama su jedan od najučinkovitijih načina za osiguravanje kreativnosti u učenju na digitalnim igrama (DGBL). Istraživanje (Ott i Pozzi, 2012) je pokazalo da digitalne igre pomažu u digitalnom učenju kreativnosti. Učenje kroz kreiranje računalnih igra daje novu dimenziju u inovativnom pedagoškom podučavanju s ciljem razvoja računalnog razmišljanja.

### 2.3. PROGRAMIRANJE U VIRTUALNOM OKRUŽENJU

Programiranje u virtualnom okruženju pomaže učenicima steći vještine razmišljanja i programiranja, socijalne i komunikacijske vještine, međusobna (su)pomoć, asertivnost i uvažavanje. Platforme za takvo računalno programiranje su Claroline i Moodle. Claroline je nastao 2001. u Belgiji. Pristupačnost je izrazito jednostavna bez potrebe posebnog prethodnog tehničkog predznanja.

Moodle platforma daje mogućnost izvora znanja u obliku zadataka, vježbi, multimedije, videa, prezentacije, foruma, lekcije, kvizovi, zadaće. Komunikacija između nastavnika i učenika je interaktivna.

Uz Moodle postoji veliki broj platforma s obrazovnim ciljem kao što su Blackboard, Claroline, WebCT, Alpha LMS, Link2school, CentraOne, Consensus, Web-guru, Lmswizdom, Wiziq.

Jedna od digitalnih načina računalnog programiranja s ciljem motivacije učenika je gemifikacija. Učenicima se iznova nudi rješavanje zadataka bez nekakvih sankcija i negativnih bodova, slobodno mogu birati zadatke prema interesima i sposobnostima. Učenici mogu raditi u timu, povećan je angažman učenika, ali i natjecateljski duh.

Gemifikacija potiče i nagrađuje učenike koja podržavaju računalno programiranje u virtualnom okruženju s ciljem produktivnosti i razvoja socijalnih vještina. Takav način učenja zauzima važno mjesto u obrazovanju s ciljem bolje motivacije, pohađanja nastave i postizanja boljih rezultata (Hung, 2017). Barata i suradnici (2013) su proveli gemificirani tečaj s ciljem bolje motivacije učenika za računalno programiranje. Tečaj je uključivao skupljanje bodova, ploče s rezultatima, izazove i značke. Na kraju tečaja vidljivo je da je porasla motivacija učenika za programiranje, ali se ocjene nisu značajno poboljšale. Berkling i Thomas (2013) su pokrenuli mrežnu platformu za računalno učenje kroz igre.

Brewer i suradnici su istraživali problem nedostatka motivacije kod učenika za računalno programiranje. Motivacija djece u izvršavanju zadataka porasla je za 24 %. De Freitas i de Freitas (2013) su nazvali gemifikaciju na satovima informatike pod nazivom "Classroom Live". Važan im je bio dizajn igre, uveli su bodove, razine i nagrade. Rezultati su pokazali da je studentima u takvom ozračju ugodnije bilo učiti računalno programiranje, ali je vidljiv napredak i u angažmanu. Kapp (2012) je naznačio da gamifikacija može povećati učenikov angažman u procesu učenja. Nicholson (2015) smatra da igranje temeljeno na nagrađivanju ima kratku učinkovitost. Motivacija ovisi o kontekstu (Hartnett, 2016).

Primjer aplikacije za povećanje motivacije učenika je DevHub/Roblox, Stack Overflow, gdje je moguća virtualna međusobna komunikacija. Platforma nudi niz postavljanja pitanja i dobivanja odgovora, za razmjenu znanja i suradnju jezicima C++, Javascript, Python, Java, C#, CSS, HTML, Mysql, PHP, RUBY. Pruža jedinstvenu bazu kodova, radi na svim platformama koje Roblox podržava. Jedna od platformi za mlađe od 18 godina (Comscore), igrači mogu biti uključeni na Robloxu. Microsoft Ribbon Hero nudi mogućnost gemifikacije u učenju alata Microsoft Office. To su Microsoft Word, Excel, PowerPoint i OneNote u sustavima Microsoft Office 2007 i 2010.

Platforma Khan Academy pruža različite mogućnosti računalnog programiranja kroz igru. Postoje gamificirani sustavi koji pomažu povećati angažman učenika prilikom učenja novih tehnika (Nah i sur., 2014). Platforme pružaju niz novih mogućnosti u virtualnom okruženju.

U internetskim tečajevima je to fleksibilno podučavanje uzimajući u obzir dob učenika, njegovi interesi bez unaprijed strogo definiranih kriterija i pravila (Nicholson, 2015). KhanAcademy je platforma koja se ne plaća. Učenici samo biraju koje zadatke će rješavati te kojim tempom. Usmjerena je na učenje iz područja matematike, informatike, biologije, umjetnosti, ekonomije, čitanja, jezika, životnih vještina. Nastavnik može sam formirati razred, grupe, dodati učenike te odabrati tečajeve programiranja različitih programskih jezika.

Ukoliko ima nejasnoća, postoje videolekcije i primjer kodova u različitim programskim jezicima. Učenici rade vlastitim tempom kako im najbolje odgovara prema vlastitim mogućnostima i sposobnostima.

Code game je besplatna platforma koja razvija logičko razmišljanje, omogućuje interdisciplinarnost i timski rad. Pruža edukacije putem gemifikacije otvorenog i zatvorenog tipa. Suraduju s više od 6500 škola i organizacija iz 21 države diljem svijeta s ciljem povezanosti, socijalizacije i stvaranje igara.

Codeacademy je platforma za učenje koja je specijalizirana za računalno programiranje s gamifikacijom (Ryzac, 2012). Može i online tečaj. Učenici su u mogućnosti izraditi interaktivne vježbe kodiranja s bodovanjem i nagrađivanja putem značaka s ciljem što bolje motivacije. Učenici imaju pristup nastavnim materijalima, nastavnici imaju mogućnost praćenja trenutne situacije svakog pojedinog učenika i njihovog napretka. Omogućene su bonus nagrade s ciljem još veće motivacije. Codeacademy je prilagođena za početnike, ima i vremensko ograničenje.

Codeschool nudi računalno programiranje kroz igru. Uključuje videosadržaje, razne zadatke, imaju priliku izrađivati projekt. Dobivaju bedževe i bodove s ciljem bolje motivacije. Pruža mogućnost upoznavanja različitih programskih jezika HTML&CSS, Python, JavaScript, Java, SQL, Ruby, C++, R, C#, PHP, Go, Bash/Shell, Swift, Kotlin. Ponuđen je račun preko kojeg se učenici mogu prijaviti i gdje se vidi pojedinačni napredak i postignuća. Sadrži i mogućnost sudjelovanja u pojedinim tečajevima, kao i pristup nastavnim materijalima. Platforme zasnovane na gemifikaciji imaju poseban dizajn, sadrže prilagođene zadatke, rangiranje prema razinama, nagrađivanje, pregled nastavnih sadržaja, kvizove, forum, diskusije s ciljem što veće motivacije učenika. Potiče natjecanje među učenicima i/ili timovima. Bodovi se skupljaju rješavanjem kvizova i zadataka vezanih uz odrađene lekcije, izvršavanjem dodatnih vježbi, poticanjem na izazov kao oblik natjecanja, ispravljanjem prikazanih pogreški (forum, diskusija), pomaganje drugima podučavanjem ili prikazivanjem neke vježbe s ciljem razvoja empatije, samopoštovanja i socijalizacije. Mjeri se vrijeme izvršavanja koda i provjerava se točnost koda. Može biti kao web-stranica

napravljena u različitim programskim jezicima HTML-u, JavaScript-u i CSS-u. JavaScript daje mogućnost rada bilo kojeg računalnog programiranja. Teoretski, platforma se može koristiti za podučavanje bilo kojeg programskog jezika (Python, Javascript, SQL, C ++, C#, Ruby, PHP, HTML).

**Tablica 2.** Primjeri online okruženja i njihove značajke (autor)

|                               |   |
|-------------------------------|---|
| Claroline (2001.)             |   |
| Moodle (2002.)                | jedan od najraširenijih sustava za online učenje        |
| DevHub/Roblox (2018.)         | programiranje, virtualna komunikacija i međusobna pomoć |
| Microsoft Ribbon Hero (2010.) | učenje alata Microsoft Office u obliku mini igara       |
| Khan academy (2008.)          | uz računalno programiranje i drugi predmeti             |
| Code game (2018.)             | gamificirano poučavanje programiranja                   |
| Codeacademy (2011.)           |   |
| Codeschool (2012.)            |   |



## 2.4. PROGRAMSKI JEZICI U OSNOVNOJ ŠKOLI

Današnje digitalno doba se sve više oslanjanja na područje informatike i digitalne kompetencije. Traženo je sve više znanja i vještina povezanih s računalnim programiranjem. Učenike treba što ranije upoznati s programiranjem.

Postavlja se pitanje koji bi bio programski jezik najprikladniji za tu dob. Klasični pristup podučavanju programiranja uključuje proceduralni programski jezik s rješavanjem matematičkih zadataka. Istraživanja su pokazala da vizualni programski jezici poput Scratcha i programskih igara umjesto rješavanja matematičkih problema pozitivno utječu na motivaciju učenika za programiranje. Motivacija i sama vještina rješavanja problema ključna je za programiranje.

Programiranje je teško i programski jezici su prilično komplicirani za početnike. Programski jezici za početnike trebali bi biti jednostavni kako učenik ne bi bio preopterećen složenošću alata. Programski jezici koji se koriste u obrazovne svrhe, trebali bi se temeljiti na ideji osmišljavanja malog i jednostavnog jezika za djecu koji će podržati prve korake u učenju programiranja.

Python ispunjava sve zahtjeve za taj jezik i smatra se jednim od najprikladnijih početnih tekstualnih programskih jezika.

Vizualna programska okruženja temelje se na vizualnim uputama, omogućavajući djeci da se usredotoče na sam problem. Najpoznatiji vizualni programski jezici su Scratch, Alice i Greenfoot. Ti jezici pružaju novi kontekst programiranja i usmjereni su na kontekst igara, pripovijedanja, animacija,

Programiranje je izazov za nastavnike i učenike. Smatra se teškim za učenje i razumijevanje, bez obzira na godine. Većina učenika osnovnih škola imaju negativan stav prema programiranju. Neka istraživanja su pokazala da čimbenici kao što su stav i

motivacija za programiranje utječe na uspjeh u učenje programiranja. Programiranje je način razmišljanja.

Neka istraživanja su ukazala da čimbenici kao stav i motivacija imaju značajan utjecaj na faktor uspjeha pri učenju programiranja. Jedan od prvih programskih jezika koji se koristi kao alat za učenje geometrije svladavanjem vještina rješavanja problema je Papert's Logo. Izvođenje programa napisano u Logou vizualno je atraktivno, korištenje drag and drop blokova predstavljenih u obrascu.

U Republici Hrvatskoj informatika je obavezni predmet samo u 5. i 6. razredu osnovne škole. U svakom razredu programiranje je jedno od ponuđenih tema u kurikulumu gdje učitelji slobodno biraju na temelju hrvatskog nacionalnog kurikuluma jezike za programiranje.

Programiranje nije popularno među djecom, te izbor odgovarajućeg programskog jezika može biti iznimno važan u motivaciji i odnosu prema programiranju. Djeca bi trebala najprije svladati osnovne koncepte programiranja.

Znanstvenici su proveli istraživanje na Harvardovoj ljetnoj školi gdje je cilj istraživanja bio odrediti stav učenika prema programiranju. Rezultati su pokazali da djeca smatraju Scratch zabavnim, a imali su i pozitivan stav prema programiranju nakon korištenja Scratcha, mali broj učenika koji nije volio Scratch.

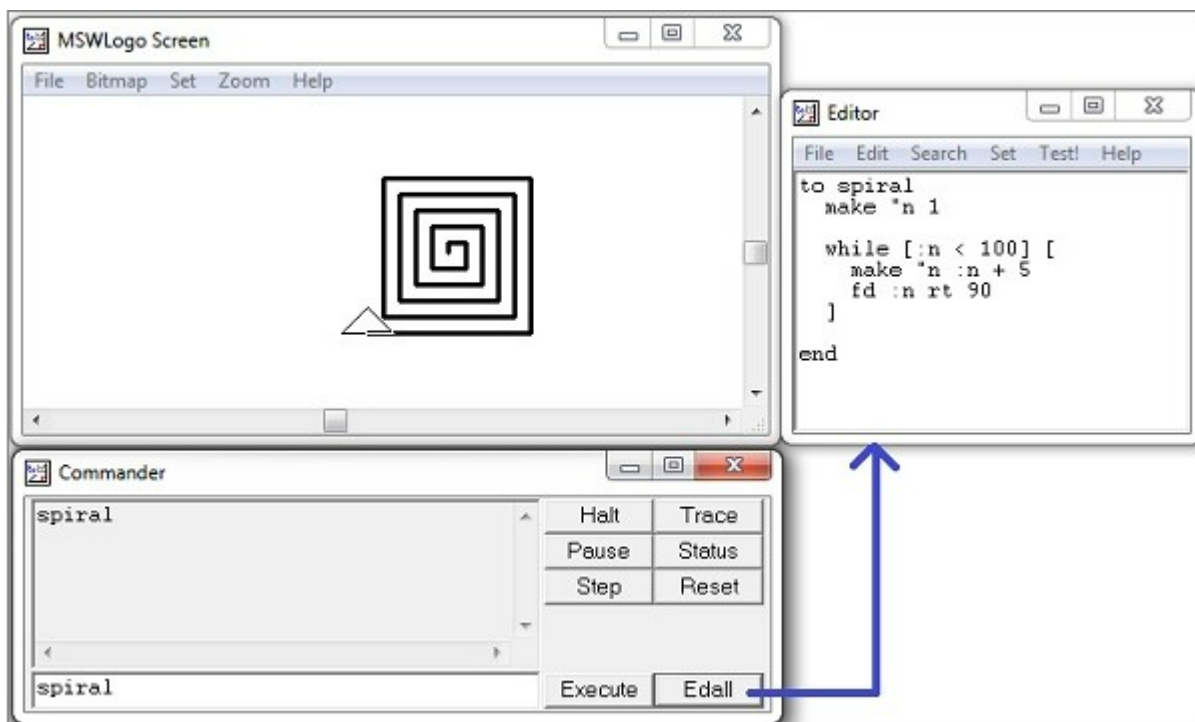
Drugo slično istraživanje u blizini Los Angelesa pokazuje da su djeca uspjela svladati osnove programskih koncepata u Scratchu kroz zabavan način. Sljedeće istraživanje uspoređivalo je stavove i ishode učenja učenika šestih razreda kroz programiranje u Logu ili Scratchu u ljetnoj školi. Logo okruženje je pomoglo u razvoju samopouzdanja učenika, zanimanje za računalno programiranje i razumijevanje konstrukt petlje. Scratch pokazuje poboljšanje ishoda učenja za učenike. Logo poboljšava kreativnost učenika i vještina rješavanja problema među učenicima 5. razreda Katoličke osnovne škole u blizini Jakarte.

**Tablica 3.** Neki programski jezici namijenjeni početnicima i godina uvođenja u uporabu (autor)

|                   |   |
|-------------------|---|
| BASIC (1964.)     | Prvi jezik namijenjen ne-programerima.                            |
| Logo (1967.)      | Implementacija kornjače. Dijalekt Lisp-a.                         |
| Python (1991.)    | Jezik visoke razine s naglaskom na čitljivost koda                |
| Scratch (2003.)   | Vizualni jezik. Dijalekt JavaScripta.                             |
| Alice (1994.)     | Vizualni jezik za učenje kroz stvaranje 3d svijeta. Dijalekt Jave |
| Greenfoot (2003.) | Vizualni jezik. Dijalekt Jave.                                    |

```
10 MAX = 5000
20 X = 1 : Y = 1
30 IF (X > MAX) GOTO 100
40 PRINT X
50 X = X + Y
60 IF (Y > MAX) GOTO 100
70 PRINT Y
80 Y = X + Y
90 GOTO 30
100 END
```

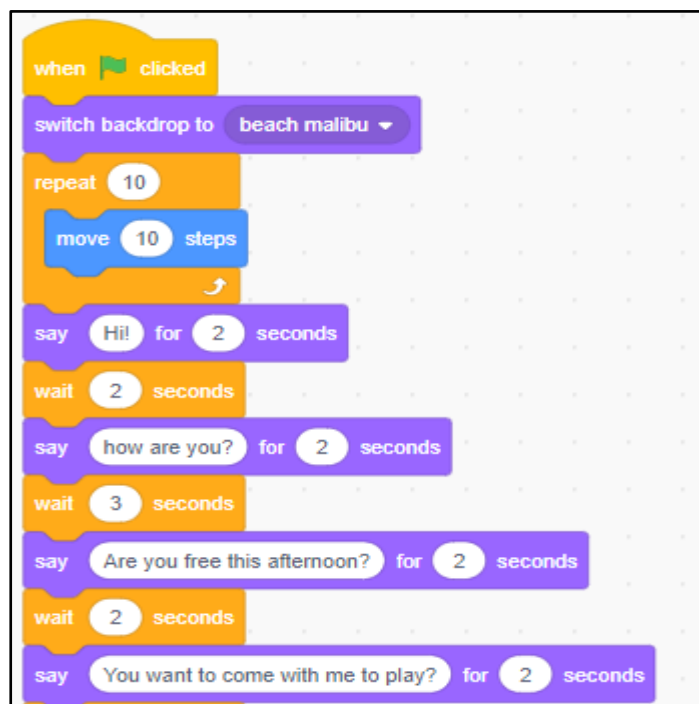
**Slika 3.** Primjer programa u BASIC-u (screenshot programskog jezika)



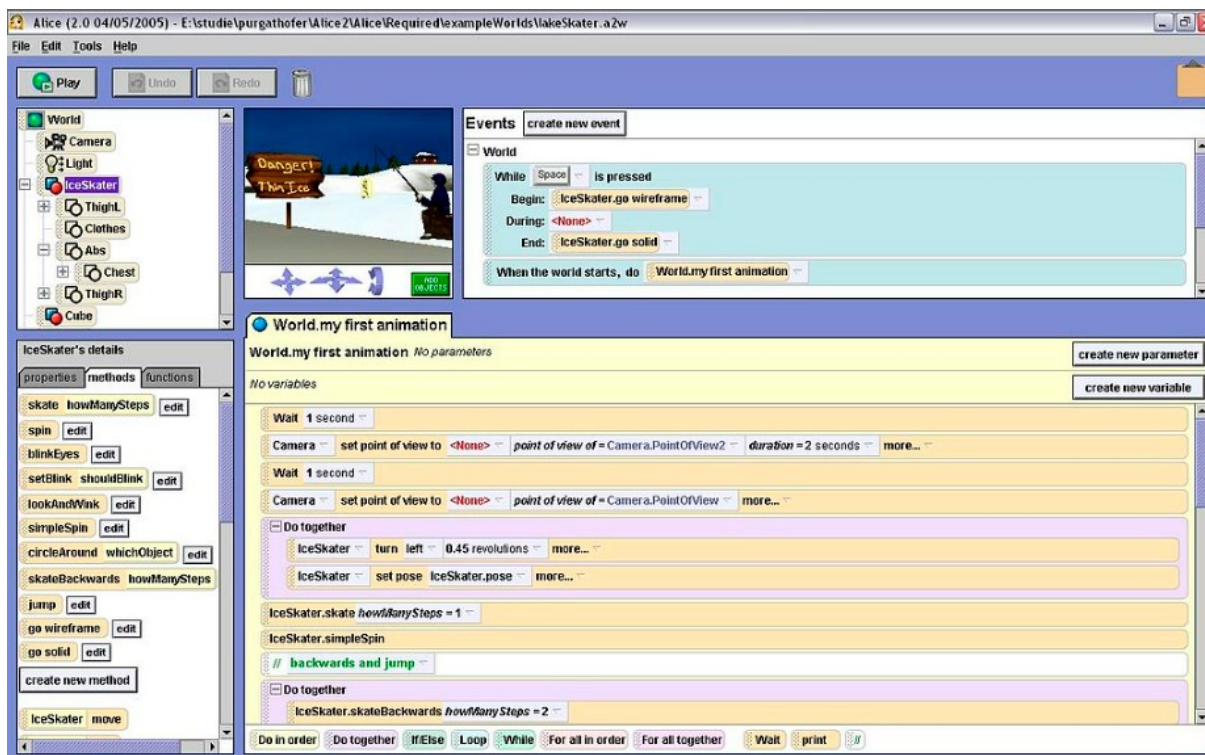
Slika 4. Primjer programa u Logou (screenshot danog programskog jezika)

```
num = 11
# If given number is greater than 1
if num > 1:
    # Iterate from 2 to n / 2
    for i in range(2, int(num/2)+1):
        # If num is divisible by any number between
        # 2 and n / 2, it is not prime
        if (num % i) == 0:
            print(num, "is not a prime number")
            break
        else:
            print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```

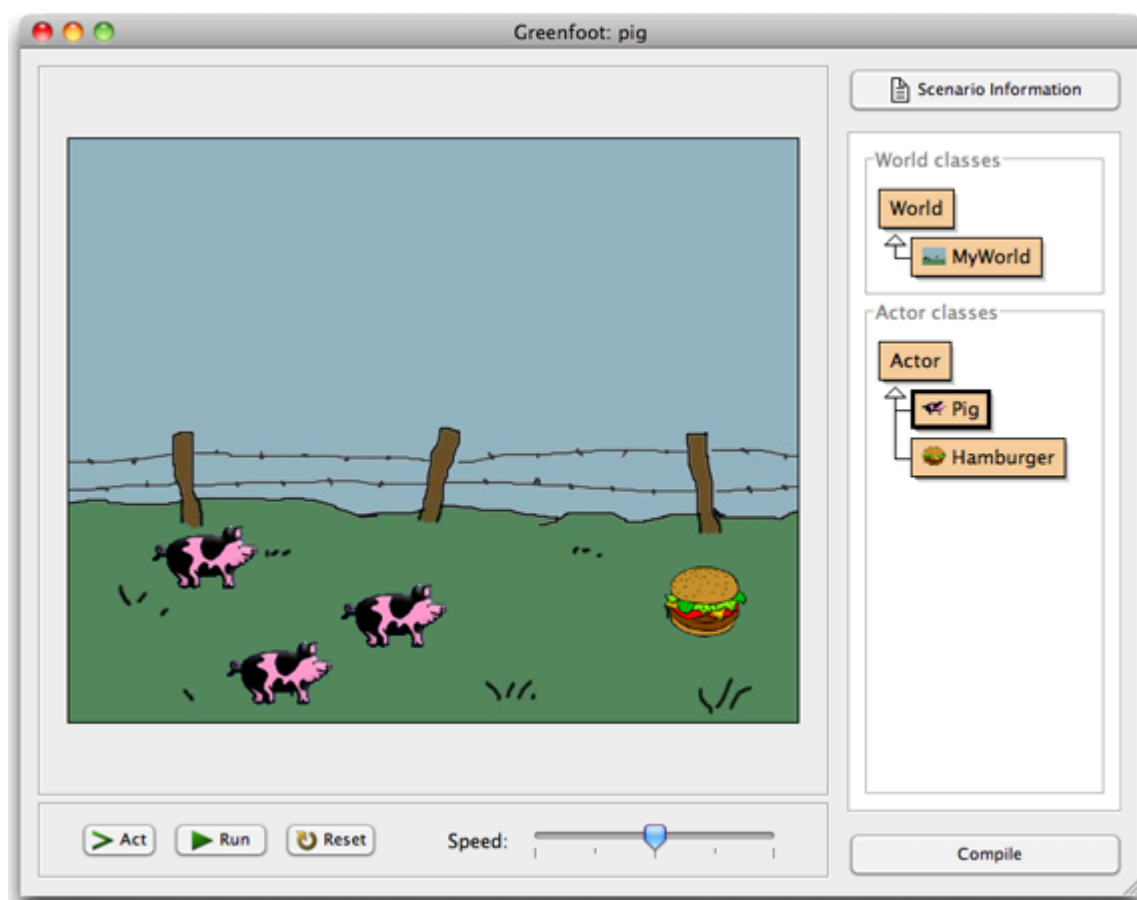
Slika 5. Primjer programa u Pythonu (screenshot danog programskog jezika)



Slika 6. Primjer programa u Scratch okruženju (screenshot danog programskog jezika)



Slika 7. Primjer programa u Alice okruženju (screenshot danog programskog jezika)



Slika 8. Primjer programa u Greenfoot okruženju (screenshot danog programskog jezika)

## 2.5. MOTIVACIJA UČENIKA U ŠKOLI

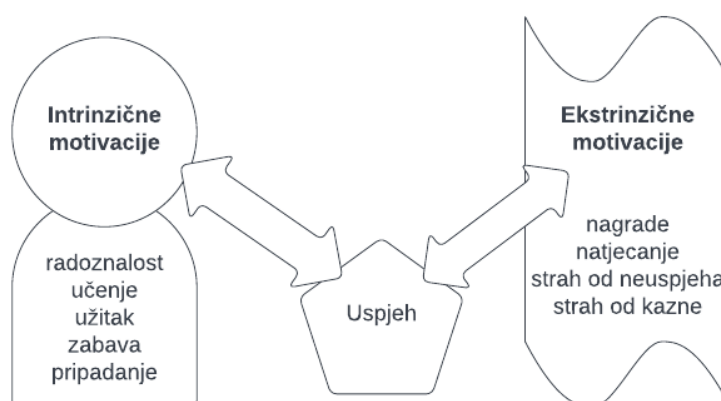
Rezultati istraživanja su pokazali da je školska motivacija iznimno značajna za školski uspjeh i upornost kod djece (Pintrich, 2003). Osmišljeni su intervencijski programi s ciljem poboljšanja motivacije učenika u školi (Wigfield i Wentzel, 2007). Tako razlikujemo motivaciju učenika općenito prema školi i motivaciju učenika prema pojedinim nastavnim predmetima (Bandura, 1997; Elliot, 2005; Green, Martin i Marsh, 2007; Pintrich, 2003).

Ispitivanje motivacije učenika prema školskim predmetima uključuje teorijska stajališta:

- Teorija ciljeva
- Teorija samoučinkovitosti
- Samopoimanje
- Model očekivane vrijednosti, i to prvenstveno kroz čitanje, pisanje i matematiku.

Drugi pristup ispitivanja zahtjeva pristup teoriji samoodređenja (SDT; Ryan i Deci, 2002). Pristupi su doveli do boljeg razumijevanja motivacijske dinamike i međusobnih ishoda kod učenika mlađe i starije školske dobi.

Motivacija je definirana prema SDT (Ryan i Deci, 2002) kao razlozi koji leže u ponašanju, uključenost u školske aktivnosti. Samoodređenje uključuje intrinzičnu motivaciju koja potječe od same sebe, aktivnosti koje se rade iz užitka i zabave. S druge strane, ekstrinzična motivacija se odnosi na aktivnost izazvanu vanjskim podražajem, nagradom, izbjegavanjem kazne i varira u smislu samoodređenja.



Slika 8. Međuodnos motivacije i uspjeha (autor)

Intrizična motivacija proizlazi iz učenikove radoznalosti, učenja radi užitka, zabave, ali i osjećaja pripadnosti. Ekstrizična motivacija je poticana od nagrada, natjecateljskog angažmana, mogućeg niskog samopouzdanja, a samim time i bojazni od neuspjeha i/ili kazne kod kuće i/ili u školi. Jedna i druga motivacija vezane su za uspjeh koji je krajnji cilj (slika 8).



### **3. DOSADAŠNJA ISTRAŽIVANJA MOTIVACIJE UČENIKA ZA RAČUNALNO PROGRAMIRANJE**

- **Socijalno-kognitivne interakcije, motivacija i kognitivni rast u programiranju logotipa i CAI okruženja za rješavanje problema (Nastasi i Clements, 2021)**

Istraživanje se odnosilo na dva okruženja učenika: računalno programiranje uz pomoć Loga i računalno rješavanje problema. Usmjerenost je bila i na kooperativnost, interakciju, sukob, rješavanje, motivaciju i samoevaluaciju učenika.

Dobnu skupinu učenika činili su četvrti i šesti razredi osnovne škole. Djeca koja su sudjelovala u Logu uspješnije su pokušavala i rješavala probleme, stvarala pravila i bila su zadovoljnija. Logo je potaknuo više motivaciju kod učenika.

- **Spirale održive akademske motivacije, kreativnosti i povjerenje osoblja visokog obrazovanja (Blaškova i sur., 2021)**

U radu se ističe međusobna korelacija kreativnosti i motivacije. Postavljena je hipoteza odnosa motivacije, kreativnosti i povjerenja u visokom obrazovanju u Slovačkoj i Poljskoj. U istraživanju su sudjelovali svi djelatnici visokoobrazovnih ustanova (N = 181). Postoji međusobna povezanost na sveučilištu.

- **Intrinzične, identificirane i kontrolirane vrste motivacije za školske predmete kod mlađe osnovnoškolske djece (Guay i sur, 2021)**

U radu se istražuje motivacija učenika prema pojedinim nastavnim predmetima i razlika među njima. Drugi dio istraživanja ispituje intenzitet i kvalitetu motivacije u nastavnim predmetima. Nastavni predmeti su se odnosili na pisanje, čitanje i matematiku. Sudjelovalo je 425 francusko-kanadskih učenika osnovnoškolske dobi, prvi i treći razred. Rezultati su pokazali da kod učenika postoji intrinzična i kontrolirana motivacija za nastavne predmete. Veće razlike u motivaciji su prisutne kod starijih učenika. Istraživanje ukazuje na

postojanje više vrste motivacija kod učenika osnovnoškolske dobi prema nastavnim predmetima.

- **Usporedba programiranja osnovnoškolaca – uspjeh temeljen na programskom okruženju** (Mladenović i sur., 2021)

Spominju se programsko okruženje kao podloga za poučavanje programiranja koje se bazira na igricama. Navedeno je Scratch i Logo. Ispitani su stavovi i ishodi učenika za računalno programiranje u navedenim programskim jezicima. Uzorak je obuhvatio učenike 7. razreda. Rezultati su pokazali da Scratch bolje pomaže u razumijevanju računalnog programiranja.

- **Holistički pristup učenju temeljen na digitalnim igrama s obzirom na izvanškolsko osnovno obrazovanje o programiranju** (Panskyi i Rowinska, 2020)

Osnovno obrazovanje u računalnom programiranju sastojalo se od triju dijelova: vizualno programiranje, programiranje i robotika, ming programiranje i elektronika. Učenici koji su sudjelovali u holističkom pristupu su bili od četvrtog do desetog razreda, i to u Poljskoj. Proveden je upitnik koji je pokazao da taj pristup pomaže učenicima u računalnom programiranju, osnažuje ih za srodne znanosti u budućnosti, inženjerstvu, matematici i tehnologiji. Isto tako, holistički pristup olakšava učenje i poučavanje u integraciji s izvanškolskim obrazovanjem.

- **Upoznavanje osnovnoškolaca s programiranjem korištenjem razvoja igrica u Pythonu** (Mladenović i sur., 2021)

Postavlja se retoričko pitanje koji je najpodobniji programski jezik za što ranije učenje programiranja i koji je to najprihvatljiviji pristup u nastavi informatike?! Istaknuto je da vizualni programski jezici kao što je Scratch pozitivno djeluju na motivaciju učenika za računalno programiranje. U radu je ispitan utjecaj uspjeha i motivacije učenika za računalno programiranje. Učenici četvrtih i petih razreda su sudjelovali u istraživanju dviju

različitih škola. Procedularno programiranje je bilo u Pythonu, a programiranje igara u Scratchu. Nije bilo statističke značajnosti među školama, a ni između spolova. Učenici koji imaju bolji uspjeh bolje se snalaze u rješavanju problema u Pythonu, dok se slabiji učenici bolje snalaze u Scratchu nego u Pythonu.

- **Poticanje razvoja vještina 21. stoljeća uključivanjem učenika u autentičnu igru, dizajn projekata u srednjoj školi, sat računalnog programiranja** (Thomas i sur., 2011)

Učenici koji su pohađali srednju školu, kreirali su igre za učenike osnovnoškolske dobi. Ovakav pristup videoigricama, dizajnu, programiranju i suradnji njeguje vještine za suvremeno društvo stvarajući moderno građanstvo.

- **Programiranje u paru dvaju računala: Istraživanje povratne veze – intervencija za poboljšanje suradničkog razgovora u osnovnoj školi** (Zakarija i sur., 2021)

Učenici mlađe školske dobi trebaju pravodobne smjernice za suradnju u okruženju koje njeguje interakciju i suradnju. Istražuje se suradnja učenika u programiranju kada su uparena sva računala. Prikupljeni su videopodatci. Istaknuta je međusobna podrška vršnjaka u ovakvom kontekstu programiranja.

- **Motivacija za učenje i razumijevanje programiranja učenika osnovnih škola u Japanu** (Cheung i sur., 2018)

Istraživanje je usmjereno na motivaciju učenika za računalno programiranje vizualnim putem u osnovnoj školi. Sve više su prihvatljivi vlastiti roboti koje konstruiraju i dizajniraju učenici, a zatim im daju razne funkcije kretanja, govorenja putem programskog jezika. Zanimljivo je saznati kolika je motivacija učenika u nastavi informatike. Proveden je upitnik u tu svrhu. Upitnik je pokazao da su motivacija i stupanj razumijevanja učenika iznimno visoki.

- **Popunjavanje praznine u uputama za programiranje: Tekstualno poboljšano grafičko programsko okruženje za učenike nižih razreda** (Cheung i sur., 2009)

Istraživanje je provedeno u Japanu. Proveden je uvodni tečaj kako bi se učenicima pružilo motivirajuće programsko okruženje za učenje bogatom grafikom i tekstualnim okruženjem. Učenici do 8. razreda pozitivno reagiraju na grafička i tekstualna okruženja, dok stariji učenici ne pridaju značajnu važnost tome. Više vole konvencionalno okruženje. U radu je predloženo da učenici nižih razreda osnovne škole koji su neuki u programiranju, na inovativan i interaktivan način budu okruženi poboljšanim programskim okruženjem. Na taj način učenici smišljaju vlastite priče i/ili programe uz pomoć blokova. Ovakvo okruženje se pokazalo kao iznimno motivirajuće i pozitivno iskustvo za učenje programiranja.

- **Iskustvo u programiranju potiče veću STEM motivaciju među djevojčicama prvog razreda** (Master i sur., 2017)

Pokazano je da su djevojčice manje motivirane za STEM područje od dječaka prvih razreda. Dječaci pokazuju veći interes i bolji su po rezultatima u robotici i programiranju. Iskustvo programiranja robota povećava motivaciju kod djevojčica u STEM području.

- **Učenje programiranja robotike za učenike osnovnih i srednjih škola** (Ohnishi i sur., 2017)

Obrazovna politika u Japanu daje iznimnu važnost učenja programiranja učenika. Učenici su dobili set robota za sastavljanje s ciljem stvaranja proaktivnog stava prema učenju. Cilj je bio izučiti učinkovitost robota za učenje programiranja kod učenika koja se pokazala iznimno dobrom.

- **Akadska motivacija osnovnoškolske djece u dvama obrazovnim pristupima – inovativni i tradicionalni** (Gordeeva i sur., 2018)

Ispitano je koliko intrinzična motivacija učenika utječe na uspjeh učenika i inovativne mogućnosti. Ispitanici su bili učenici trećih i četvrtih razreda. Učenici koji su imali tradicionalan način poučavanja pokazali su iznimno manju ekstrinzičnu motivaciju (roditelji, učitelji, općenito). Ekstrinzična motivacija igra ulogu u postignućima učenika kod načina rada (tradicionalno naspram inovativno). Autonomna motivacija pozitivno utječe na percepciju školske klime prema uspjehu učenika. Nema razlike kod učenika vezano za intrinzičnu motivaciju.

- **Promjene u samopoimanju kompetencije i intrinzična motivacija među učenicima osnovnoškolske dobi (Bouffard i sur., 2003)**

Provedeno je longitudinalno istraživanje koje je trajalo tri godine u Kanadi. Cilj je bio ispitati ulogu dječje kompetencije i intrinzične motivacije za čitanje, matematiku, uspjeh učenika. Na početku je napravljena inicijalna procjena učenika prvih razreda osnovne škole (N = 115). Ispitivanja su nastavljena i sljedeće dvije godine upitnikom o dječjoj kompetenciji i intrinzičnoj motivaciji kod učenika u čitanju i matematici. Ocjene na kraju školske godine su predstavljale mjeru uspješnosti. Nema značajnog utjecaja intrinzične motivacije na uspjeh učenika u nijednom razredu. Kompetencija je povezana s uspjehom učenika u svakom razredu u čitanju i matematici. Nema značajne razlike prema spolu učenika.

- **Korištenje upravljanja računalom – instruktivni softver za povećanje motivacije i postignuća u djece osnovne škole (Terrell i Rendulić, 1996)**

Korištenje instruktivnim softverom daje pozitivan učinak na učeničku motivaciju s ciljem što boljeg uspjeha. Bolji su rezultati u intrinzičnoj motivaciji, kao i postignućima učenika, kognitivnim vještinama. Što je viša motivacija učenika, to on postiže bolji uspjeh.

Smatra se da je za znanstveno djelovanje najvažnija intrinzična motivacija jer samo tako dolazi do pravog i istinskog znanja i učenja (Biehler, 1974; Csikszentmihalyi i Nakamura, 1989; Deci i Ryan, 1985.; Higgins i Sorrentino, 1990).

- **Program razvoja životnih vještina za osnovnu školu (Gim, 2021)**

Provedeno je istraživanje u osnovnoj školi u Koreji. Stjecanje životnih vještina uključuje i kompetencije koje se stječu kroz računalno kodiranje. Svrha istraživanja je razviti program Entry koje daju mogućnost pružiti učenicima poboljšanje online životnih vještina kroz aktivnosti računalnog kodiranja. Program se pokazao prikladnim za vrijeme korone u smislu online učenja kroz priručnik i platformu.

- **Evaluacija metoda podučavanja temeljenih na igrama - Računalno razmišljanje u osnovnoj školi (Leifheit i sur., 2018)**

Istraživanje je obuhvatilo razne pristupe temeljene na igrama s ciljem što bolje motivacije kod učenika primarnog obrazovanja. Provedena je procjena prikladnosti pristupa koji se zasnivaju na igricama s ciljem računalnog programiranja.

Napravljeno je terensko istraživanje računalnog razmišljanja kod učenika. Uključivalo je razumijevanje, formuliranje, sustavno rješavanje složenijih problema koji su zahtijevali apstrakciju, generalizaciju, parametrizaciju, algoritmizaciju i particioniranje, programiranje.

Održan je tečaj s ciljem boljeg razumijevanja računalnog razmišljanja koje je temeljeno na materijalima s code.org. Tečaj je uključivao algoritme, nizove, petlje, grananja i događanja. Učenje i aktivnosti su bile bazirane na igricama. Uzorak ispitanika su bili učenici 3. i 4. razreda osnovne škole, N = 33. Razumijevanje učenika se provjeravalo testovima, a zainteresiranost i motivacija samoprocjenom upitnika prije i nakon tečaja. Učenici su u prosjeku postigli 82 % ciljeva učenja, pozitivno su ocijenili svoja iskustva u tečaju kao i iznimnu motivaciju. Dio kurikuluma je bio iznimno složen za učenike, ugniježdeni uvjeti.

- **Izrada upitnika o samopoimanju, motivacijska uvjerenja i stav prema programiranju** (Leifheit i sur., 2019)

Istraživanje je pokazalo da su samopoimanje, motivacijska uvjerenja i stavovi prema školskom predmetu relevantni za učenje i obrazovna postignuća.

Napravljen je upitnik kojim se procjenjuje samopoimanje, motivacijska uvjerenja i stavovi prema programiranju. Taj upitnik je korišten u pilot-istraživanju kod učenika koji su pohađali tečaj računalnog programiranja u obliku izvannastavnih aktivnosti u školi. U istraživanju su sudjelovali učenici u dobi od sedam do deset godina, N = 31. Postoji povezanost aspekata procjene upitnikom. Nema značajne povezanosti rezultata upitnika i vještina programiranja.

- **Razvoj upitnika za procjenu – samopoimanje i stavovi prema programiranju** (Leifheit i sur., 2020)

Ovaj upitnik za procjenu prema programiranju temelji se na postojećim instrumentima i uključeni su aspekti samopoimanje i stavovi učenika. Upitnik sadrži:

- prethodno programiranje prema samoprocjeni/iskustvo i razumijevanje
- samopoimanje
- intrinzičnu vrijednost/uvjerenje,
- uvjerenje o vrijednosti/postignuća
- uvjerenje o korisnoj vrijednosti
- uvjerenje o trošku
- usklađenost i ustrajnost.

Istraživanje je provedeno na uzorku od 197 učenika osnovnih škola od sedam do deset godina s ciljem evaluacije računalnog razmišljanja. Podatci su analizirani radi pouzdanosti i valjanosti konstrukta. Rezultati su pokazali dobru pouzdanost. Rezultati pokazuju dobru prikladnost mjernog instrumenta za procjenu samopoimanja i stava učenika prema programiranju.

- **Učinci izvannastavnog tečaja računalnog razmišljanja za osnovnu školu – školska djeca: randomizirano kontrolirano terensko ispitivanje** (Leifheit i sur., 2020)

Proveden je tečaj računalnog programiranja u sklopu izvannastavnih aktivnosti s ciljem samopoimanja i motivacijom učenika. Tečaj je uključivao upoznavanje o sekvenci, petljama, uvjetima i događajima i njihovim primjenama. Učenje je temeljeno na igrama te mjestovito učenje (unplugged/plugged-in). Uzorak je obuhvatio 197 učenika 3. i 4. razreda koji su sudjelovali na tečaju. Rezultati su pokazali da je tečaj bio učinkovit za promicanje pozitivnog samopoimanja i veću motivaciju za programiranje kod učenika osnovne škole.

- **Shvatite kako računala razmišljaju. Program tečaja za napredovanje računalnog razmišljanja i vještine sustavnog rješavanja problema darovite djece osnovnoškolske dobi** (Leifheit i sur., 2020)

U Njemačkoj su osnovane brojne Hector dječje akademije. Djeluju dobrovoljno i podrška su darovitoj djeci. Hector institut za empirijska obrazovna istraživanja zajedno s Njemačkim institutom za međunarodna obrazovna istraživanja daju znanstvenu podršku i vrednuju Hector dječje akademije. Akademija pruža brojne tečajeve u području matematike, informatike, prirodnih znanosti za darovite učenike. Tečajevi počivaju na teorijskom znanju iz područja didaktike, psihologije i brojnih istraživanja nastave.

Neki od temeljnih Hector tečajeve su:

- Razumjeti kako računala razmišljaju
- Razgovor o prirodnim znanostima: Mali stručnjaci – Iznosimo svoje znanje
- Mali istraživači – Radimo kao znanstvenici
- Spreman za matematičku olimpijadu



- Eksperimentirajte sigurno u kemijskom laboratoriju
- Matematika za slušanje – djeca sastavljaju s LEGO-om
- Kako djeluju biljke
- Pneumatika
- Ribolovna tehnika i električna energija
- Tajni spisi.

Cilj prvog tečaja je poticati računalno razmišljanje i pobuditi interes za informatiku kod darovitih učenika osnovnoškolske dobi. Termini koja daroviti učenici kod algoritama usvajaju su nizovi, petlje, uvjetna grananja, događaji, operatori i podatci i varijable (Brennan i Resnick, 2012). Učenici pritom usvajaju misaone procese, sudjeluju u aktivnostima od jednostavnog ka složenom u računalnom programiranju.

Unplugged aktivnosti su društvene igre i kartaške igre u prirodnoj veličini s obrazovnom svrhom.

Računalno razmišljanje je veoma važno za računalno programiranje. Programiranje se odnosi algoritamskog koda za izvršavanje instrukcija. Računalno razmišljanje pomaže učenicima da razviju sposobnost kako razumjeti složene probleme i pronaći strategije za njihovo sustavno rješavanje (Wing, 2014).

Taj proces se u Hectorovom tečaju sastoji od sljedećih dijelova:

- Razumjeti problem: srž problema i kontekst, ono što ga definira, razumjeti parametre njegovu višerazinsku strukturu
- Precizno formulirati problem: izraziti problem na način da je algoritamski rješenje moguće
- Riješiti problem: razviti algoritam rješenja za formulirani problem koji se sastoji od uređenog skupa izjava
- Implementacija i evaluacija rješenja – ono razvijeno u prethodnom koraku

- Poboljšati rješenje: poboljšati pogrešne ili nedostatne dijelove algoritma, za postizanje željenih rezultata.

Tečaj ima ciljeve usvojiti temeljne informatičke pojmove, a to su:

- slijed
- petlja
- uzorak
- događaj
- uvjet
- varijabla
- postojan
- operatori
- algoritam.

Cilj tečaja je:

- promicanje kompetencije u razvoju strategija algoritamskih rješenja za IT probleme
- pobuđivanje interesa za informatičke sadržaje i probleme

- **Uloga samopoimanja i motivacije u okviru pristupa računalnim razmišljanjem do ranog informatičkog obrazovanja (Leifheit, 2020) – doktorski rad**

U ovom dijelu rada se nalaze istraživačka pitanja i studije vezano za područje rada.

Prvi dio istraživanja se odnosi na sljedeće tvrdnje:

- povezanost motivacijskih predispozicija s programiranjem u mlađoj školskoj dobi
- najbolji način računalnog razmišljanja kroz motivaciju učenika školske dobi koje je ujedno korisno za njihovo samopoimanje.

S tim ciljem, napravljen je novi mjerni instrument koji je mjerio samopoimanje učenika te njihovu motivaciju vezano za računalno programiranje prema spolu.

Vršena je procjena prikladnosti unplugged igrice za pomoć osnovnoškolcima u stvaranju početnog razumijevanja koncepta računalnog mišljenja. Proveden je tečaj u školama.

Evaluirane su lekcije kao dio tečaja računalnog razmišljanja koji se sastojao od 18 lekcija po 45 minuta. Lekcije su bile po uzoru na tečaj code.org.

Cilj tečaja je bilo promovirati razumijevanje računalnog programiranja kod učenika i potaknuti njihov interes i motivaciju. Tečaj je kroz lekcije uključivao jednostavne algoritme, učenici su učili kako ukloniti greške kod programiranja.

Učenje je uvedeno kroz unplugged aktivnosti temeljene na igricama koje koriste svakodnevne predmete (papir, olovke) umjesto digitalnog okruženja i koda.

Deset lekcija je uključivalo unplugged aktivnosti, dok je u preostalih osam lekcija bila primjena vježbi programiranja uz pomoć tableta i stolnih računala.

Smatralo se da unplugged aktivnosti učenja stavljaju naglasak na računalne koncepte, a ne na korištenje tehnologije (Prottsman, 2014).

Glavni dio sata na tečaju se sastajao od aktivnosti učenika koje su bile bazirane na igricama. Prije toga bi se učenici prisjetili aktivnosti s prethodnog sata.

Zadatak nastavnika je bio potaknuti razgovor s učenicima što su naučili kroz aktivnosti i kako se to može povezati s algoritmima u računalnom programiranju.

Uzorak istraživanja je obuhvatio učenike njemačke osnovne škole,  $N = 33$ . Učenici su polazili 3. i 4. razred. Bila je potrebna pisana roditeljska suglasnost vezano za prijavu tečaja njihove djece.

Mjerni instrument: na kraju svakog sata, učenici su dobivali nastavne listiće za rješavanje s ciljem analiziranjem ciljeva učenja koje su postigli.

Mjerni upitnici za samoprocjenu (Likertova skala od 1 do 5) odnosili su se na ispitivanje zainteresiranosti i motivacije učenika za računalnim programiranjem.

Metodološka ograničenja dovela su do razvoja novog instrumenta za procjenu motivacijskih mogućnosti kod učenika osnovne škole, kao i mogućnost novog studija o podučavanju računalnog programiranja.

Drugi dio istraživanja je obuhvatio upitnik o samopoimanju te motivacijska uvjerenja i stav prema programiranju.

Dorađen je instrument na temelju postojećeg instrumenta za mjerenje samopoimanja i motivacijskih uvjerenja učenika o matematici (Brisson i sur., 2017; Gaspard, 2017; Leifheit i sur., 2019), koji je već bio prilagođen i potvrđen za druge školske predmete, uključujući biologiju, fiziku i jezike (Gaspard, Häfner, Parrisius, Trautwein i Nagengast, 2017).

Mjerni instrument se odnosi na samopoimanje učenika i motivaciju vezano za računalno programiranje na pedeset zadataka na skali procjene od 1 do 7:

1. samoprijavljeno iskustvo i razumijevanje programiranja
2. samopoimanje
3. uvjerenje o intrinzičnoj vrijednosti
4. uvjerenje o vrijednosti postignuća
5. uvjerenje o korisnosti
6. trošak, uvjerenje
7. usklađenost i upornost. (Leifheit i sur., 2019).

Ustanovljeno je da su samopoimanje i motivacija kao oblici intrinzične motivacije povezani s računalnim programiranjem. U matematici se pokazalo da je upornost i usklađenost učenika povezana sa sposobnošću računalnog razmišljanja.

Sljedeće istraživanje je uključivalo 197 učenika osnovne škole u dobi od sedam do deset godina koji su polazili treći i četvrti razred u Njemačkoj. Svi učenici su pohađali tečaj računalnog razmišljanja u sklopu izvannastavnih aktivnosti na temelju interesa i školskog uspjeha prema preporuci nastavnika.

### Mjerni instrument

Provedena je skala procjena o samopoimanju učenika i stavu prema programiranju na skali procjene od jedan četiri (1 = potpuno se slažem; 2 = prilično se slažem; 3 = radije ne pristati; 4 = uopće se ne slažem) koje je obuhvatilo:

- (1) samoprijavljeno iskustvo i razumijevanje programiranja
- (2) samopoimanje
- (3) uvjerenje o intrinzičnoj vrijednosti
- (4) uvjerenje o vrijednosti postignuća
- (5) vrijednost korisnosti uvjerenje
- (6) uvjerenje o troškovima
- (7) usklađenost i upornost.

Tečaj računalnog razmišljanja je bio poticajan u samopoimanju vezano uz programiranje i motivacijska uvjerenja o programiranju učenika osnovnih škola.

Nema značajne razlike prema spolu učenika što se tiče tečaja na motivaciju poimanja o programiranju. Nema značajne razlike prema spolu koje se odnosi na:

- (P1) iskustvo i razumijevanje
- (P3) uvjerenje unutarnje vrijednosti
- (P4) postignuće/uvjerenje o vrijednosti
- (P5) uvjerenje o korisnoj vrijednosti
- (P6) uvjerenje o trošku
- (P7) usklađenost i upornost u programiranju.

#### 4. RASPRAVA

Samopojmanje učenika nakon tečaja računalnog programiranja je značajno poboljšano u usporedbi s učenicama. Ovo istraživanje ukazuje i na ograničenja u smislu mogućnosti da se kod budućih istraživanja obuhvati veći uzorak ispitanika. Isto tako, istraživanjem se nije moglo točno odrediti koje specifične komponente tečaja su bile posebice učinkovite. Ograničenje je i to što nema velik broj istraživanja na tu temu, tako da nije moguće raditi usporedbe. Istraživanje ne predviđa rezultate dugoročnih učinaka tečaja. To iziskuje dodatna mjerenja kako bi se dobio širi uvid u učinke računalnog programiranja kod učenika.

Tečaj je imao pozitivne učinke na razumijevanje računalnog programiranja, samopojmanje učenika o računalnom programiranju i izrazitu intrinzičnu motivaciju. Tečaj je bio koristan u domeni računarstva, učenici su pokazali sposobnost računalnog razmišljanja.

Ovo istraživanje je dalo prve dokaze o učinkovitosti tečaja računalnog programiranja s ciljem poboljšanja računalnog razmišljanja, pozitivnog samopojmanja i povećanje motivacije učenika. Učenici i učenice imaju podjednaku korist od računalnog programiranja.

Unplugged metode poučavanja temeljene na igrama su bile motivirajuće za poučavanje računalnog razmišljanja u osnovnoj školi. Učenici su s vremenom osjećali sve veću motivaciju tijekom tečaja. Učinkovito poučavanje kroz računalno programiranje je izrazito motivirajući učenicima u osnovnoj školi.

Tečaj računalnog razmišljanja je bio poticajan u online programiranju u kojem je sudjelovalo više od 1000 učenika, dok su škole bile zatvorene u Njemačkoj tijekom pandemije COVID-19 2020, ali i šire.

Nadam se u budućnosti većem broju instrumenata za procjenu dane teme koji će biti dostupni znanstvenicima, ali i nastavnicima s ciljem šire primjene instrumenata ocjenjivanja, ali i veći broj istraživanja na zadanu tematiku. Istraživanje može pomoći

nastavnicima i kreatorima obrazovnih politika kako ukomponirati nove metodičke pristupe i metode poučavanja računalnog programiranja u postojeće kurikulume u školama koji su nadasve motivirajući za učenike. Učinkovitost pristupa i metoda mogu poslužiti kao orijentacija za buduća istraživanja.

Uočljivo je da je u međusobnom odnosu intrinzičnih i ekstrinzičnih karakteristika prema računalnom razmišljanju i programiranju povezano više pojmova koji jedan na drugog utječu, motivacija, apstrakcija, samopouzdanje učenika, postignuća koja se postignu uz pomoć računalnog razmišljanja i programiranja, ali i dugotrajna korisnost.

Kako bi to sve bilo učinkovito potrebno je usvojiti i neke pojmove i poveznice kod računalnog razmišljanja i programiranja (ulaz/izlaz, varijable, kontrola toka, petlje, funkcije).

Primjeri online okruženja prikazuju raširenost i raznovrsnost sustava za računalno programiranje za učenike. Najrašireniji sustav za online učenje je Moodle. DevHub/Roblox služi za programiranje, virtualnu komunikaciju i međusobnu pomoć. Code game, Codeacademy i Codeschool nude gamificirano poučavanje programiranja.

## 5. ZAKLJUČAK

Računalno programiranje postaje imperativ i najvažnija vještina u današnje doba. Zauzima važno mjesto u obrazovanju i potrebno ga je implementirati u druge discipline kako bi ono postalo interdisciplinarno. Ne postoje mnoga istraživanja o najboljem učinku načina učenja i podučavanja računalnog programiranja za učenike osnovne škole. No, u jednom se slažu znanstvenici, tj. da je motivacija učenika ta koja im daje poticaj za bolji angažman, potiče ih za razne aktivnosti s ciljem bolje suradnje, (su)pomoći, razvijanje natjecateljskog duha i učinka boljih rezultata. Samim time se razvijaju i socijalne vještine i povećava samopouzdanje što dovodi do kvalitetnijeg i efikasnijeg učenja i ugodnog ozračja u radnom okruženju.

Takvim načinom doprinosi se boljitku i povećanju profesionalne programske zajednice te zadovoljavanju potrebama tržišta rada. Ovo područje rada iziskuje iznova proširena istraživanja zbog aktualnosti tematike, ali i boljeg razumijevanja i odabira računalnog programiranja u obrazovanju.



## 6. LITERATURA

Armoni, M., & Gal-Ezer, J. (2014). Early computing education: Why? What? When? Who? *ACM Inroads*, 5(4), 54–59.

Armstrong, J. M., & Price, R. A. (1982). Correlates and predictors of women's mathematics participation. *Journal for Research in Mathematics Education*, 13(2), 99.

Baki, A., & Özpınar, İ. (2007). Effects of LOGO assisted geometry teaching material on students' academic achievement and students' views on practice. *Cukurova University Faculty of Education Journal*, 34(3), 153 - 163.

Balanskat, A., & Engelhardt, K. (2015). Computing our future. Computer programming and coding. Priorities, school curricula and initiatives across Europe. European Schoolnet, Brussels.

Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York: W. H. Freeman.

Barata, G., Gama, S., Jorge, J., Goncalves, D.(2013). Engaging Engineering Students with Gamification. In: 5th International Conference on Games and Virtual Worlds for Serious Applications, 1–8.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54.

Baytak, A., Land, S. M., & Smith, B. K. (2011). Children as educational computer game designers: An exploratory study. *Turkish Online Journal of Educational Technology*, 10(4), 84-92.

Benzer, A. İ., & Erümit, A. K. (2017). The analysis of the graduate theses related to programming instruction. *Journal of Instructional Technologies & Teacher Education*, 6(3), 99-110.

Berge, Z. (1995). Facilitating computer conferencing: Recommendations from the field. *Educational Technology*, 15(1), 22–30.

Berge, Z. (2009). Changing instructor's roles in virtual worlds. *Quarterly Review of Distance Education*, 9(4), 407–415 [www.infoagepub.com/quarterly-review-of-distance-education.html](http://www.infoagepub.com/quarterly-review-of-distance-education.html)

Bergin, T.J. and Gibson, R.G. (1996) *History of Programming Languages-II*. USA: ACM Press.

Berkling, K., Thomas, C. (2013). Gamification of a Software Engineering Course. In: *International Conference on Interactive Collaborative Learning*, 525–530.

Betts, B.W., Bal, J., Betts, A.W.(2013). Gamification as a Tool for Increasing the Depth of Student Understanding using a Collaborative E-learning Environment. *International Journal of Continuing Engineering Education and Life-Long Learning* 23(3-4), 213–228.

Bezjak, J. (2009). Project learning of model PUD – BJ- from idea to the product. Klagenfurt: LVM.

Blašková, M.; Tumová, D.; Blaško, R.; Majchrzak-Lepczyk, J. (2021). Spirals of Sustainable Academic Motivation, Creativity, and Trust of Higher Education Staff. *Sustainability* 13, 7057. <https://doi.org/10.3390/su13137057>

Bolliger, D. U. & Wasilik, O. (2009). Factors influencing faculty satisfaction with online teaching and learning in higher education. *Distance Education*, 30(1), 103–116. <http://dx.doi.org/10.1080/01587910902845949>

Bonnie K. Nastasi, B.K.&Clements D.H. (1990). *Social-Cognitive Interactions, Motivation, and Cognitive Growth in Logo Programming and CAI Problem-Solving Environments*, American Psychological Association.

Boulden , Collin F. Lynch , Kristy Elizabeth Boyer & Eric N. Wiebe (2021): Two-Computer Pair Programming: Exploring a Feedback Intervention to improve Collaborative Talk in Elementary Students., *Computer Science Education*, DOI: 10.1080/08993408.2021.1877987

Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2008). Computer aided instruction as a vehicle for problem solving: Scratch programming environment in the middle years classroom. Paper presented at 2008 Annual Conference & Exposition, Pittsburgh, Pennsylvania. Retrieved from <https://peer.asee.org/3826>.

Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education*, 4(2), 121-135.

C. Y. Cheung, Ngai, G., Stephen C. F. Chan and Lau, W.Y. (2021). *Filling the Gap in Programming Instruction: A Text-enhanced Graphical Programming Environment for Junior High Students*, The Hong Kong Polytechnic University.

Calao, L. A., Moreno-Leon, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with Scratch. In G. Conole, T. Klobučar, C. Rensing, J. Konert, & E. Lavoué (Eds.), *Design for Teaching and Learning in a Networked World*. Toledo: Springer International Publishing, 17-27. [https://doi.org/10.1007/978-3-319-24258-3\\_2](https://doi.org/10.1007/978-3-319-24258-3_2)

Covington, D., Petherbridge, D. & Warren, S. (2005). Best practices: A triangulated support approach in transitioning faculty to online teaching, *Online Journal of Distance Learning Administration*, 8(1). <https://www.westga.edu/~distance/ojdla/spring81/covington81.htm>

De Freitas, A.A., de Freitas, M.M.(2013). Classroom Live: A Software-assisted Gamification Tool. *Computer Science Education* 23(2), 186–206.

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33–39. *Development*, 15(1), 1–12.

Diana V. Pshenichnuka, D.& Sidnevaa, A.N. (2003). Changes in self-perceptions of competence and intrinsic motivation among elementary schoolchildren, *British Journal of Educational Psychology*.

Dorn, B., & Elliott Tew, A. (2015). Empirical validation and application of the computing attitudes survey. *Computer Science Education*, 25(1), 1–36.

Durik, A. M., Vida, M., & Eccles, J. S. (2006). Task values and ability beliefs.

Džaferagić-Franca, A., Omerović, M. (2012). Aktivno učenje u osnovnoj školi. *Metodički obzori*, 7(1), 167-181. Glasser, W. (1994). *Kvalitetna škola*. Zagreb: Educa.

Eccles, J. S., Alder, T. F., Futterman, R., Goff, S. B., Kaczala, C. M., Meece, J. L., *Educational Psychologist*, 42, 191–196.

Elliot, A. J. (2005). A conceptual history of the achievement goal construct. In A. J. Elliot.

Ferrer-Mico, T., Prats-Fernández, M. À., & Redo-Sanchez, A. (2012). Impact of Scratch programming on students' understanding of their own learning process. *Procedia-Social and Behavioral Sciences*, 46, 1219-1223. <https://doi.org/10.1016/j.sbspro.2012.05.278> for Elementary and Junior High School Students, Hiroshima University.

Freitas, S.D., Jarvis, S. (2006). A Framework for Developing Serious Games to meet Learner Needs. *Proceedings of the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*. 4–7

García Peñalvo, F. J., Reimann, D., Tuul, M., Rees, A., Jormanainen, I., et al. (2016). An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers.

Gaspard, H. (2017). Promoting value beliefs in mathematics : A multidimensional perspective and the role of gender.

Gee, Q. H., Wills, G. and Cooke, E. (2005). “A first programming language for IT students.” *Proceedings of the 6th Annual Conference of the Learning and Teaching Support Network Centre for Information and Computer Sciences*, York, UK.

Gordeevaa, T. & Sychevb, A. (2018). Academic Motivation of Elementary School Children in Two Educational Approaches — Innovative and Traditional. *Russian Psychological Society*, doi: 10.11621/pir.2018.0402

Green, J., Martin, A., & Marsh, H. W. (2007). Motivation and engagement in English, Mathematics and Science high school subjects: Towards an understanding of multidimensional domain specificity. *Learning and Individual Differences*, 17, 269–279. doi:10.1016/j.lindif.2006.12.003

Guay, F., Ratelle, C. F., & Chanal, J. (2021). Intrinsic, identified, and controlled types of motivation for school subjects in young elementary school children  
*Canadian Psychology*, 49, 233–240.

Guo, P. J. (2013). Online Python Tutor: Embeddable Web-Based Program Visuali.

Gupta, N., Tejovanth, N., & Murthy, P. (2012). Learning by creating: Interactive programming for Indian high schools. In 2012 IEEE International Conference on Technology Enhanced Education. <https://doi.org/10.1109/ICTEE.2012.6208643>

Han, B., Bae, Y., & Park, J. (2016). The effect of mathematics achievement variables on Scratch programming activities of elementary school students. *International Journal of Software Engineering and Its Applications*, 10(12), 21-30.  
<https://doi.org/10.14257/ijseia.2016.10.12.03>

Heersink, D., & Moskal, B. M. (2010). Measuring high school students' attitudes toward computing. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 446–450). New York, NY, USA: ACM.

Heintz, F., Mannila, L., & Farnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. In *Frontiers in Education Conference*. <https://doi.org/10.1109/FIE.2016.7757410>

Hoegh, A., & Moskal, B. M. (2009). Examining science and engineering students'

Howatt, J. W. (1995). "A project-based approach to programming language evaluation." *ACM SIGPLAN Notices*, 30 (7), 37-40. <http://academic.luther.edu/~howaja01/v/lang.pdf>

Hromkovic, J., Kohn, T., Komm, D., & Serafini, G. (2016). Examples of algorithmic thinking in programming education. *Olympiads in Informatics*, 10, 111-124.  
<https://doi.org/10.15388/oi.2016.08>

Hung, A.C.Y. (2017). A Critique and Defense of Gamification. *Journal of Interactive Online Learning* Vol 15, Number 1. <https://www.ncolr.org/jiol/issues/pdf/15.1.4.pdf>

Ibrahim, R., Jaafar, A. (2009). Educational games (EG) design framework: Combination of game design, peda-gogy and content modeling. 2009 International Conference On Electrical Engineering and Informatics.

Ihantola, P. (2011) Automated Assessment of Programming Assignments: Visual Feedback, Assignment Mobility, and Assessment of Students' Testing Skills. Espoo, Finland: Aalto University. *Journal of Educational Psychology*, 95, 667–686. doi:10.1037/0022-0663.95.4.667

Johnson, E. B. (2002). Contextual teaching and learning: what it is and why it's here to stay. Thousand Oaks, CA: Corwin Press, INC. 31.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200-210. <https://doi.org/10.1016/j.chb.2015.05.047>

Kapp, K.M. (2012). Games, Gamification, and the Quest for Learner Engagement. *Training and Development* 66(6), 64–68.

Kazimoglu, C. (2013). Empirical evidence that proves a serious game is an educationally effective tool for learning computer programming constructs at the computational thinking level (Unpublished doctoral dissertation). University of Greenwich.

Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, 73, 26-39. <https://doi.org/10.1016/j.compedu.2013.12.010>

Keet, E. E. (2004). "A personal recollection of software's early days (1960-1979): Part 1." IEEE Annals of the History of Computing (October-December).

Kelley, T., Kellam, N. (2009). A Theoretical Framework to Guide the ReEngineering of Technology Education. Journal of Technology Education 20 (2). 32.

Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. The Internet and Higher Education, 8(1), 13–24.

King, P. E., Behnke, R. R. (2005). Problems associated with evaluating student performance in groups. College Teaching, 53(2), 57-61.

Ko, S. & Rossen, S. (2010). Teaching Online: A Practical Guide. New York: Routledge.

Kosowski, A., Małafiejski, M., & Noiński T. (2008). Application of an online judge & contester in academic tuition. Lecture Notes in Computer Science 4823, 343- 354.

Kovačević, S. (2012). Kurikulumska matrica tehničkih kompetencija u odgoju i općem obrazovanju (Doktorska disertacija, Sveučilište u Zagrebu, Filozofski fakultet). Zagreb: Filozofski fakultet.

Lamprou, A., & Repenning, A. (2018). Teaching how to teach computational thinking: 69–74. <https://doi.org/10.1145/3197091.3197120>

Leifheit, L. (2020) The Role of Self-Concept and Motivation Within the "Computational Thinking" Approach to Early Computer Science Education. Dissertation. Tübingen.

Leifheit, L., Golle, J., Trautwein, U., Ostermann, K., Ninaus, M., & Moeller, K. (unsubmitted manuscript). Motivational Effects of an Extracurricular Computational Thinking Training for Elementary School Children: A Randomized Controlled Field Trial.



Leifheit, L., Jabs, J., Ninaus, M., Moeller, K., & Ostermann, K. (2018). Programming unplugged: An evaluation of game-based methods for teaching computational thinking in primary school. In Proceedings of the 12th European Conference on Game-Based Learning (ECGBL) (pp. 344–353).

Leifheit, L., Tsarava, K., Moeller, K., Ostermann, K., Golle, J., Trautwein, U., & Ninaus, M. (2019). Development of a Questionnaire on Self-Concept, Motivational Beliefs, and Attitude Towards Programming. In Proceedings of the 14th Workshop in Primary and Secondary Computing Education (WiPSCE). ACM. (pp. 1–9).

Leifheit, L., Tsarava, K., Ninaus, M., & Moeller, K. (2018). “Verstehen wie Computer denken”. Ein Trainingsprogramm zur Förderung von informatischem Denken und systematischen Problemlösefähigkeiten besonders begabter Kinder im Grundschulalter.

Leifheit, L., Tsarava, K., Ninaus, M., Ostermann, K., Golle, J., Trautwein, U., & Moeller, K. (2020, June). SCAPA: Development of a Questionnaire Assessing Self-Concept and Attitudes Toward Programming. In Proceedings of the 25th Conference on Innovation and Technology in Computer Science Education (ITiCSE). ACM. (pp. 138–144).

Liu, X., Bonk, C.J., Magjuka, R.J., Lee, S.H. & Su, B. (2005). Exploring four dimensions of online instructor roles: a program level case study. *Journal of Asynchronous Learning*

Looi, H.C., Seyal, A.H., Darussalam, B. (2014). Problem-based Learning: An Analysis of its Application to the Teaching of Programming. Proceedings of the International Proceedings of Economics Development and Research (IPEDR). Orlando, Florida.

Lu, J. J., & Fletcher, G. H. (2009). Thinking about computational thinking. In Proceedings of the 40th ACM Technical Symposium on Computer Science Education (pp. 260–264).

Luker, P. (1989). "Academic staff development in universities with special reference to small group teaching." (Unpublished PhD Thesis), University of Nottingham.

Malliarakis, C., Satratzemi, M., Xinogalos, S. (2012). Towards the Constructive Incorporation of Serious Games Within Object Oriented Programming. Proceedings of the 6th European Conference on Games Based Learning (ECGBL 2012). 4–5 October, Cork, Ireland, 301–308.

Marsh, H. W., Köller, O., Trautwein, U., Lüdtke, O., & Baumert, J. (2005). Academic self-concept, interest, grades, and standardized test scores: Reciprocal effects models of causal ordering. *Child Development*, 76(2), 397–416. Mathematics and Science high school subjects: Towards an understanding of mathematics participation. *Journal for Research in Mathematics Education*,

Matijević, M., Radovanović, D. (2011). *Nastava usmjerena na učenika*. Zagreb: Školske novine.

Milat, J. (1996). Tehnička kultura bitna je odrednica sustava obrazovanja. *Društvena istraživanja*, 5.

Mladenović, M., Rosić, M. i Mladenović, S. (2021). Comparing Elementary Students' Programming Success based on Programming Environment, *Modern Education and Computer Science*, 2021, 8, 1-10.

Moreno-Leon, J., Robles, G., & Roman-Gonzalez, M. (2016). Code to learn: Where does it belong in the K-12 curriculum? *Journal of Information Technology Education: Research*, 15, 283-303. <https://doi.org/10.28945/3521>

Nagengast, B., Trautwein, U., Scalas, L. F., Hau, K.-T., Marsh, H. W., & Xu, M. K.

Nah. H.F., Zeng, Q., & Rajasekhar, V.T., & Padmanabhuni, A.A., & Eschenbrenner, B. (2014). Gamification of Education: A Review of Literature. Department of Business and Information Technology, Missouri University of Science and Technology. United States.  
[https://ca.jei.com/official/data/Gamification\\_Of\\_Education.pdf](https://ca.jei.com/official/data/Gamification_Of_Education.pdf)

Nardelli, E. (2019). Do we really need computational thinking? Communications of the ACM, 62(2), 32–35.

Navarrete, C. C. (2013). Creative thinking in digital game design and development: A case study. Computers & Education, 69, 320-331.

Nevin, A. I., Thousand, J. S. & Villa, R. A. (2009). Collaborative teaching for teacher educators—What does the research say? Teaching and Teacher Education, 25(4), 569-574.  
<http://dx.doi.org/10.1016/j.tate.2009.02.009>

Nicholson, S. (2015). A recipe for meaningful gamification. In T. Reiners & L. A. Wood (Eds.), Gamification in education and business, 1–20, New York, NY: Springer.

Nikčević-Milković, A. (2004). Aktivno učenje na visokoškolskoj razini. Život i škola, 50(2), 47-54.

Ohnishi, Y., Kimitoshi Honda, K, Nishioka, R. (2017). Robotics Programming Learning for Elementary and Junior High School Students, Hiroshima University.

Oliver, J. S., & Simpson, R. D. (1988). Influences of attitude toward science, achievement motivation, and science self concept on achievement in science: A longitudinal study. Science Education, 72(2), 143–55.

Ott, M., Pozzi, F. (2012). Digital games as creativity enablers for children. Behaviour & Information Technology, 31(10), 1011–1019.

Panskyi, T., Rowinska, Z., Biedron, S. (2021). *Informatics in Education*, 2021, Vol. 20, No. 2, 255–276, Vilnius University, ETH Zürich, DOI: 10.15388/infedu.2021.12.

Papert, S. (1980). *Mindstorms: Computers, children, and powerful ideas*. NY: Basic.

Papert, S. (2005). You can't think about thinking without thinking about thinking about something. *Contemporary Issues in Technology and Teacher Education*, 5(3), 366–367.

Parker, K.R.&Davey, B. (2012).The History of Computer Language Selection. Arthur Tatnall. *Reflections on the History of Computing : Preserving Memories and Sharing Stories*, AICT-387, Springer, pp.166-179, IFIP Advances in Information and Communication Technology (SURVEY), ff10.1007/978-3-642-33899-1\_12ff. fihal-01526795f

Parker, K.R. i Davey, B. (2012). The History of Computer Language Selection. Arthur Tatnall. *Reflections on the History of Computing : Preserving Memories and Sharing Stories*, AICT-387, Springer, pp.166-179, IFIP Advances in Information and Communication Technology (SURVEY), ff10.1007/978-3-642-33899-1\_12ff. fihal-01526795f pedagogical approaches to embedding 21st century problem solving in k-12

Peko, A., Sablić, M., Livazović, G. (2006). Suradničko učenje u mlađoj školskoj dobi. *Život i škola*, 15-16(1-2), 17-28.

Piaget, J. (1972). Intellectual evolution from adolescence to adulthood. *Human Development*, 15(1), 1–12.

Pierce, J., W., Jones, B. (1998). *Problem-Based Learning: Learning and Teaching in Context of Problems*. U: *Contextual Teaching and Learning: Preparing Teachers to*

Enhance Student Success in and Beyond School (str. 75-106). Washington DC: ERIC Clearinghouse on Teaching and Teacher Education.

Pintrich, P. R. (2003). A motivational science perspective on the role of student motivation predictors of high school literacy choices: A developmental analysis. *Journal of Computer Attitude Scale for 16–19 Education*, 28(1), 35–41.

Putnam, R., T., Borko, H. (2000). What Do New Views of Knowledge and Thinking Have to Say about Research on Teacher Learning?. *Educational Researcher* 29 (1): 4-15.

Qian, Y., & Lehman, J. D. (2016). Correlates of success in introductory programming: A study with middle school students. *Journal of Education and Learning*, 5(2), 73-83. <https://doi.org/10.5539/jel.v5n2p73>

Reić Ercegovac, I., Jukić, T. (2008). Suradničko učenje u razrednoj nastavi. *Život i škola*, 20(2), 69-80.

Reynolds, D., Treharne, D. & Tripp, H. (2003). ICT—the hopes and the reality *British Journal of Educational Technology*, 34(2), 151–167. [http:// dx.doi.org/10.1111/1467-8535.00317](http://dx.doi.org/10.1111/1467-8535.00317)

Riggs, I. M., & Enochs, L. G. (1993). A microcomputer beliefs inventory for middle rope. European Schoolnet, Brussels.

Ryan, R. M., & Deci, E. L. (2002). Overview of self-determination theory: An organismic dialectical perspective. In E. L. Deci & R. M. Ryan (Eds.), *Handbook of self-determination research* (pp. 3–33). Rochester, NY: The University of Rochester Press.

Ryzac (2012). Codecademy. <http://www.codecademy.com/>

Saez-Lopez, J. M., Roman-Gonzalez, M., & Vazquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129-141. <https://doi.org/10.1016/j.compedu.2016.03.003>

Sammet, J. E. (1972). "Programming languages: History and future." *Communications of the ACM* 15(7): 601.

Sammet, J. E. (1981). The Early History of COBOL. In R. L. Wexelblat (Ed.), *History of programming languages I: ACM.school students: Scale development and validation*. *Journal of Research on Computing in Education*, 25(3), 383–390.

Selwyn, N. (1997). Students' attitudes toward computers: Validation of a computer attitude scale for 16–19 education. *Computers & Education*, 28(1), 35–41.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158.

Skalka J., Drlík, M. (2018). Conceptual Framework of Microlearning-Based Training Mobile Application for Improving Programming Skills. In: Auer M., Tsiatsos T. (eds) *Interactive Mobile Communication Technologies and Learning. IMCL 2017. Advances in Intelligent Systems and Computing*, 725. Springer, Cham.

Song, M., Zhang, S. (2008). EFM: A Model for Educational Game Design. In: Pan Z., Zhang X., El Rhalibi A., Woo W., Li Y. (eds) *Technologies for E-Learning and Digital Entertainment. Edutainment 2008. Lecture Notes in Computer Science*, 5093. Springer, Berlin, Heidelberg.

Swacha, J., & Baszuro, P. (2014). *Gamification-based e-learning Platform for Computer Programming Education*. Poland.

Teague, D. M., Corney, M. W., Fidge, C. J., Roggenkamp, M. G., Ahadi, A., & Lister, R. (2012). Using neo-Piagetian theory, formative in-class tests and think alouds to better understand student thinking: A preliminary report on computer programming. In Proceedings of 2012 Australasian Association for Engineering Education (AAEE) Annual Conference.

Tekinbas, K., Zimmerman, E. (2010). Rules of play: Game Design Fundamentals. Cambridge, The MIT Press. that work [Special Issue: Promoting motivation at school: Interventions that work]. The Guilford Press.

Thérèse Bouffard, T. (2016). Self-evaluation bias of school competence: Risks or opportunities for future academic success, Université du Québec à Montréal. thinking. Educational Research Review, 22, 142–158.

Thomas, M.K., Ge Xun, Greene, B. A. (2011). Fostering 21st Century Skills Development by Engaging Students in Authentic Game Design Projects in a High School Computer Programming Class, Educational Computing Research, Vol. 44(4) 391-408.

Threlkeld, R. (2006). Online education in community colleges: Conversations with the field. League for Innovation. <http://www.league.org/publication/whitepapers/1006.pdf>

Trautwein, U., Nagengast, B., Marsh, H. W., Gaspard, H., Dicke, A.-L., Lüdtke, O., & Jonkmann, K. (2013). Expectancy-value theory revisited: From expectancy-value theory to expectancy-valueS theory? Theory driving re-search: New wave perspectives on self-processes and human development, 233–249.

Verbitsky A., A., Kalashnikov V., G. (2012). Category of »Context« and Contextual Approach in Psychology. Psychology in Russia, State of the Art 5: 117-130.

Wigfield, A., & Cambria, J. (2010). Students' achievement values, goal orientations,

and interest: Definitions, development, and relations to achievement outcomes, Vol. 30, Academic Press.

Wigfield, A., & Wentzel, K. (2007). Introduction to motivation at school: Interventions that work [Special Issue: Promoting motivation at school: Interventions that work]. *Educational Psychologist*, 42, 191–196.

Wilkins, J. L. (2004). Mathematics and science self-concept: An international investigation. *The Journal of Experimental Education*, 72(4), 331–346.

Wilson, G. & Stacey, E. (2004). Online interaction impacts on learning: Teaching the teachers to teach online. *Australasian Journal of Educational Technology*, 20(1),33-48.  
<http://dro.deakin.edu.au/eserv/DU:30013096/staceyonlineinteractionimpacts-2004.pdf>

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725.

Wing, J. M. (2011). Research notebook: Computational thinking — what and why. *The Link Magazine*, 6.

Wing, J. M. (2014). Computational thinking benefits society. Retrieved from <http://socialissues.cs.toronto.edu/index.html?p=279.html>

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in k-12 classrooms. *TechTrends*, 60(6), 565–568.



Yang, Y. & Cornelious, L. F. (2005). Preparing Instructors for Quality Online Instruction. *Online Journal of Distance Learning Administration*, 8(1).  
<http://www.westga.edu/~distance/ojdla/spring81/yang81.htm>

Zakaria, Z., Vandenberg, J., Tsan, J., Cadieux, D. (2021). Two-Computer Pair Programming: Exploring a Feedback Intervention to improve Collaborative Talk in Elementary Students., *Computer Science Education*, DOI: 10.1080/08993408.2021.1877987

Zeigler, S.F. (1995). "Comparing development costs of C and Ada." Rational Software Corporation, Santa Clara, Calif, March 30.

Zhao Y. & Frank K. (2003). Factors affecting technology use in schools: An ecological perspective. *American Educational Research Journal*, 40, 807–840.  
<http://dx.doi.org/10.3102/00028312040004807>

## 7. SAŽETAK

Eksponencijalni rast tehnologija je utjecao na važnu vještinu u obrazovanju, a to je računalno programiranje. Ta vještina danas postaje imperativ na svim razinama u obrazovanju. U ovom radu se nalaze teorijski okviri i rezultati znanstvenih istraživanja vezanih za motivaciju učenika za računalno programiranje i za njeno poboljšanje, provedbu i učinkovitost. Učenje računalnog programiranja uključuje stjecanje teorijskog razumijevanja i učenje razvijanja programa u praksi, u virtualnom okruženju i okruženju u učionici. Kurikulumi nastave informatike u školama pojedinih država se izrazito razlikuju, što pokazuju i pojedina istraživanja u tom području. Nedosljednost kurikuluma ne daje konačne zaključke o najboljoj metod(ologij)i i motivaciji učenika za računalno programiranje u nastavi informatike. Izrazito je važno potaknuti djecu od što ranije dobi na računalno programiranje putem igre kako bi se povećala motivacija i samopouzdanje kod učenika, ali i obrazovni rezultati.

Ovo istraživanje će poslužiti za detaljnije razumijevanje motivacije učenika za računalno programiranje.

**KLJUČNE RIJEČI:** motivacija učenika, računalno programiranje, nastava

## 8. SUMMARY

The exponential growth of technology has affected an important skill in education, which is computer programming. This skill is becoming imperative today at all levels of education. This paper contains theoretical frameworks and results of scientific research related to student motivation for computer programming and its improvement, implementation and effectiveness. Learning computer programming involves gaining a theoretical understanding and learning to develop programs in practice, in a virtual environment and in a classroom environment. Curricula for teaching informatics in the schools of individual countries are extremely different, as shown by individual studies in the field. The inconsistency of the curriculum does not give final conclusions about the best methodology(ology) and student motivation for computer programming in computer science classes. It is extremely important to encourage children from an early age to computer programming through play, in order to increase students' motivation and self-confidence, as well as educational results.

This research will serve for a more detailed understanding of students' motivation for computer programming.

**KEYWORDS:** student motivation, computer programming, teaching