

# Računalno razmišljanje i programiranje u osnovnoškolskom obrazovanju

---

Škvorc, Dunja

Master's thesis / Diplomski rad

2021

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:147:340886>

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

*Download date / Datum preuzimanja:* **2024-07-06**

*Repository / Repozitorij:*

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



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

**Dunja Škvorc**

**RAČUNALNO RAZMIŠLJANJE I PROGRAMIRANJE U**  
**OSNOVNOŠKOLSKOM OBRAZOVANJU**

**Diplomski rad**

**Čakovec, lipanj 2021.**

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

**Dunja Škvorc**

**RAČUNALNO RAZMIŠLJANJE I PROGRAMIRANJE U**  
**OSNOVNOŠKOLSKOM OBRAZOVANJU**

**Diplomski rad**

**Mentor rada:**

**Izv.prof.dr.sc. Predrag Oreški**

**Čakovec, lipanj 2021.**

## SAŽETAK:

Kurikularnom reformom i uvođenjem obaveznog provođenja izborne nastave Informatike od 1. do 4. razreda u svim školama Republike Hrvatske sve je veća potreba za kreiranjem kvalitetnih nastavnih sredstava. Hrvatski izdavači u udžbenicima nude različite sadržaje za ostvarenje odgojno-obrazovnih ishoda, no nisu svi sadržaji jednako kvalitetni i primjereni svim učenicima. Glavnu ulogu u odabiru sadržaja za ostvarenje odgojno-obrazovnih ishoda ima učitelj koji kritički analizira sadržaje udžbenika. Sadržaji u udžbeniku često u potpunosti ne odgovaraju individualnim potrebama učenika. Iz svega navedenog proizlazi da je važna sastavnica kvalitetnog poučavanja i učenja učiteljeva kompetencija za odabir tih sadržaja, a prema potrebi i izmjeni te kreiranju vlastitih zadataka i primjera. Razvoj računalnog razmišljanja i programiranja ima višestruke koristi, no kod učenika predstavlja apstraktan proces, stoga je vrlo važna motivacija i poticaj, na što utječu brojni faktori: odabir sadržaja primjerenog razvojnog stadiju učenika, odabir primjera iz svakodnevnog života te mogućnost primjene izrađenih programa, odabir odgovarajućeg programskog jezika. Ovaj diplomski rad teorijski objašnjava računalno razmišljanje i programiranje, sadrži pregled provođenja nastavnog predmeta Informatika u Republici Hrvatskoj i *Kurikuluma nastavnog predmeta Informatika za osnovnu školu i gimnazije*, analizu sadržaja udžbenika koji ostvaruju ishode u domeni *Računalno razmišljanje i programiranje*, nudi prijedloge za ostvarenje odgojno-obrazovnih ishoda i prijedloge za povezivanje sadržaja *Računalnog razmišljanja i programiranja* s ostalim predmetima u sklopu projektne nastave. Polazište u kreiranju prijedloga za ostvarenje odgojno-obrazovnih ishoda bili su svakodnevni problemi s kojima se učenici susreću, primjerenost zadataka s obzirom na kognitivni razvoj učenika te primjenjivost programa na ostalim nastavnim predmetima.

Ključne riječi: *računalno razmišljanje; programiranje; kurikulum; udžbenici; odgojno-obrazovni ishodi*

## **SUMMARY:**

With the curricular reform and the introduction of compulsory elective classes in informatics from 1st to 4th grade in all schools of the Republic of Croatia, there is a growing need to create quality teaching aids. Croatian publisher's in textbooks offer different content for achieving educational outcomes, but not all content is of equal quality and appropriate for all students. The main role in the selection of content for the achievement of educational outcomes is played by the teacher who critically analyzes the contents of the textbook. The content of the textbook often does not fully correspond to the individual needs of the students. It follows from all the above that an important component of quality teaching and learning is the teacher's competence to select those contents, and if necessary, to change and create their own tasks and examples. The development of computational thinking and programming has multiple benefits, but it is an abstract process for students, so motivation and encouragement is very important, which is influenced by many factors - choosing content appropriate for the developmental stage of students, choosing examples from everyday life and the possibility of applying for appropriate programs, programming language. This diploma thesis theoretically explains computational thinking and programming, contains an overview of the implementation of the subject Informatics in the Republic of Croatia and the *Curriculum of the subject Informatics for primary school and gymnasium*, analysis of textbooks that achieve outcomes in the field of computational thinking and programming, offers proposals for educational outcomes and suggestions for connecting the content of computational thinking and programming with other subjects within the project teaching. The starting point in creating proposals for the achievement of educational outcomes were everyday problems that students encounter, the adequacy of tasks to the cognitive development of students, and the applicability of the program in other subjects.

Keywords: *computational thinking; programming; curriculum; textbooks; educational outcomes*

## SADRŽAJ:

SAŽETAK .....	I
1. Uvod .....	1
2. Računalno razmišljanje.....	2
3. Programiranje .....	4
4. Nastavni predmet Informatika u osnovnoj školi.....	4
5. Kurikulum nastavnog predmeta Informatika.....	5
5.1. Domena <i>Računalno razmišljanje i programiranje</i> .....	6
6. Prijedlozi ostvarenja ishoda iz domene <i>Računalno razmišljanje i programiranje</i> u hrvatskim udžbenicima i radnim bilježnicama s obzirom na preporuke iz <i>Kurikuluma nastavnog predmeta Informatika</i> .....	8
7. Pristup poučavanju .....	19
8. Programska podrška .....	21
8.1. Aplikacije i igre za razvijanje logike programiranja.....	23
8.2. Programski jezik <i>Scratch</i> .....	23
8.3. Džepno računalo <i>Micro:bit</i> .....	23
8.4. Robot <i>mBot</i> .....	24
8.5. Programski jezik <i>Logo</i> .....	24
8.6. Programski jezik <i>Python</i> .....	24
9. Prijedlozi za ostvarenje odgojno-obrazovnih ishoda iz domene <i>Računalno razmišljanje i programiranje</i> po razredima .....	25
9.1. 1. razred.....	25
9.2. 2. razred.....	29
9.3. 3. razred.....	32
9.4. 4. razred.....	35
9.5. 5. razred.....	38
9.6. 6. razred.....	41
9.7. 7. razred.....	45
9.8. 8. razred.....	50
10. Projektna nastava- povezanost Programiranja i računalnog razmišljanja s ostalim nastavnim predmetima .....	54
11. Zaključak.....	58
12. Literatura.....	59
POPIS TABLICA: .....	63
PRILOZI .....	64

## 1. Uvod

Kurikularnom reformom *Škola za život* koja je, nakon eksperimentalnog razdoblja, 2019. godine uvedena u sve škole fokus s nastavnog sadržaja prelazi na ishode, a s učitelja na učenike. Kreiranjem *Kurikuluma za nastavni predmet Informatika za osnovne škole i gimnazije* javila se potreba kreiranja novih udžbenika, ali i promjena nastavnog procesa. Odlukom Ministarstva znanosti i obrazovanja od školske godine 2020./2021. izborna nastava Informatike od 1. do 4. razreda postaje obavezna u svim osnovnim školama Republike Hrvatske. Tom odlukom postaje sve veća potreba za udžbenicima, radnim bilježnicama i materijalima za učenike, materijalima i edukacijom za učitelje te opremom za škole.

*Kurikulum za nastavni predmet Informatika za osnovne škole i gimnazije* podijeljen je na 4 domene: *Informacije i digitalna tehnologija, Računalno razmišljanje i programiranje, Digitalna pismenost i komunikacija* te *e-Društvo*. S obzirom na zastupljenost, *Računalno razmišljanje i programiranje* najzastupljenija je domena u svim godinama učenja. Važnost računalnog razmišljanja i programiranja može se promatrati kroz više aspekata, no u osnovnoj školi cilj ove domene nije stvaranje budućih programera već razvoj logičkog i algoritamskog razmišljanja te razvoj sposobnosti rješavanja problema. Računalnim razmišljanjem i programiranjem razvija se inovativnost, stvaralaštvo, poduzetnost, upornost, samopouzdanje, a potiče se suradnja i timski rad.

Suvremena nastava teži projektnoj nastavi, problemskoj nastavi, međupredmetnoj i predmetnoj povezanosti. Programiranje nudi široke mogućnosti upotrebe u drugim nastavnim predmetima. *Kurikulum za nastavni predmet Informatika* preporučuje korištenje programiranja za izradu programa koji su usko povezani s nastavnim sadržajima drugih predmeta ili s usvojenim sadržajem nižih razreda. Suvremenom nastavom učenici se potiču na razmišljanje, samostalno rješavanje problema/zadataka koji u konačnici dovode do novih spoznaja.

Rezultat svega navedenog potreba je za analizom *Kurikuluma*, aktualnih udžbenika i materijala te promišljanje i kreiranje materijala za kvalitetno usvajanje ishoda propisanih *Kurikulumom*.

## 2. Računalno razmišljanje

Računalno razmišljanje je misaoni proces koji sudjeluje u formuliranju problema i njihovih rješenja tako da se rješenje predstavlja u obliku koji agent za obradu informacija može učinkovito provesti. (Wing, n.d.) Ono se može promatrati kao formuliranje i rješavanje problema na način kao što to odvija računalo, (Tomljenović, 2018) uz pomoć računala, čovjeka ili u njihovoj kombinaciji. Osim formuliranja i rješavanja problema važno je i samo uočavanje problema. *The International Society for Technology in Education (ISTE)* i *The Computer Science Teachers Association (CSTA)* razvili su operativnu definiciju računalnog razmišljanja. Računalno razmišljanje definirali su kao postupak rješavanja problema koji uključuje određene, ali ne ograničene, karakteristike: formuliranje problema na način koji omogućava upotrebu računala ili drugih alata za rješavanje tog programa, logičko organiziranje i analizu podataka, predstavljanje podataka kroz apstrakciju kao što je model i simulacija, automatizaciju rješenja algoritamskim razmišljanjem, identifikaciju, analizu i primjenu mogućih rješenja s ciljem najučinkovitije kombinacije koraka i resursa te generalizaciju postupka rješavanja problema na širok spektar problema. (ISTE & CSTA, 2011)

Shute i suradnici ističu kako većina autora navodi sljedeće komponente računalnog razmišljanja: apstrakcija, dekompozicija, algoritmi i otklanjanje pogrešaka. (Shute et al., 2017) Na temelju prethodnih istraživanja Shute i suradnici računalno razmišljanje kategoriziraju u šest glavnih aspekata: dekompozicija, apstrakcija, algoritmi, otklanjanje pogrešaka, iteracija i generalizacija. (Shute et al., 2017) Tomljenović daje tablični prikaz aspekata prema Shute i suradnicima.

*Tablica 1. Aspekti računalnog razmišljanja*

Izvor: (Tomljenović, 2018)

ASPEKT	DEFINICIJA
Dekompozicija	Rastaviti kompleksnije probleme/sustave na manje, lakše rješive dijelove-funkcionalne podcjeline
Apstrakcija	Uočiti bit problema. Sastoji se od podcjelina: a) Prikupljanje podataka i analiza: sakupiti najrelevantnije podatke iz više izvora i shvatiti vezu među njima b) Prepoznavanje uzorka: identificirati uzorak/pravilnosti u strukturi podataka



	c) Modeliranje: izgradnja modela ili simulacija funkcioniranja sustava i/ili kako će sustav funkcionirati u budućnosti
Algoritmi	Dizajniranje logičkih i uređenih instrukcija za postizanje rješenja problema. Instrukcije mogu iznijeti ljudi ili računala. Potkategorije su: a) Dizajn: kreiranje niza određenih koraka ka rješenju problema b) Usporednost: izvršavanje određenog broja koraka u isto vrijeme c) Efikasnost: dizajniranje minimalnog broja koraka za rješavanje problema, uklanjanje nepotrebnih koraka d) Automatizacija: automatiziranje izvođenje pri rješavanju sličnih problema
Otklanjanje pogrešaka	Detektirati, identificirati i popraviti pogreške u slučaju da rješenje problema ne funkcionira.
Iteracija	Ponavljanje procesa dok se ne postigne idealno rješenje.
Generalizacija	Prenošenje CT vještina na široki krug situacija u cilju efikasnog rješavanja problema

Računalno razmišljanje nije važno samo za programere te se ono ne razvija samo programiranjem, no Tomljenović napominje da ishodi dobro planiranih aktivnosti programiranja mogu biti razvoj vještina računalnog razmišljanja. (Tomljenović, 2018) Računalno razmišljanje može se razvijati i vježbanjem određenih aspekata računalnog razmišljanja koji najčešće nisu vezani uz programiranje. Mnogi primjeri zadataka za razvoj računalnog razmišljanja nude *CS Unplugged project* i *Dabar*. *CS Unplugged project* osmišljen je sa svrhom razvoja aspekata računalnog razmišljanja kroz igre i aktivnosti bez upotrebe računala. (Tomljenović, 2018) *Bebras*, u Hrvatskoj pod nazivom *Dabar*, međunarodno je natjecanje iz informatike i računalnog razmišljanja. Natjecanje je izraslo u projekt razvoja računalnog razmišljanja, (Tomljenović, 2018) a glavni cilj je širenje i popularizacija računalstva i informatike, razvoj računalnog razmišljanja kod učenika te usporedba znanja učenika u Hrvatskoj i Svijetu na području računalnih znanosti i informacijskih tehnologija. (Suradnici u učenju, n.d.) Računalnim razmišljanjem u obrazovanju nije cilj osposobljavanje za programiranje i obrazovanje programera već rješavanje problema u digitalnom svijetu, (Tomljenović, 2018) no i izvan njega. Iako računalno razmišljanje nije važno samo za programere, ono je temelj za svako programiranje.

### **3. Programiranje**

Programiranje je proces računalno potpomognutog rješavanja problema i otklanjanja pogrešaka (Mladenović, 2019) pri čemu se razvija logičko, apstraktno i računalno razmišljanje. Mladenović ističe da je glavna svrha razvoj strategija za rješavanje problema koja mogu biti iz programerskog, ali i ne-programerskog područja, (Mladenović, 2019) dok Bubuca i suradnici programiranje promatraju kao alat pomoću kojeg se razvijaju metakognitivne sposobnosti i apstraktno razmišljanje. (Bubica et al., 2013) Metakognitivne sposobnosti utječu na kontrolu učenja pri čemu osoba određuje ciljeve i strategije učenja kojima ostvaruje ciljeve učenja. Olabe i suradnici u programiranje uključuju izgradnju sustava, otklanjanje pogrešaka, pronalaženje prepreka i problema te njihovo rješavanje pri čemu se koristi metarazmišljanje. (Olabe et al., 2011) Iz navedenog proizlazi da je svrha programiranja, naročito učenja u osnovnoj školi, razvoj rješavanja problema i metakognitivnih sposobnosti što je usko povezano s računalnim razmišljanjem.

### **4. Nastavni predmet Informatika u osnovnoj školi**

Informatika je „djelatnost koja se osniva na računalstvu, računalnoj i informacijskoj znanosti i koja primjenom informacijske i komunikacijske tehnologije rješava probleme i zadatke u različitim područjima ljudskog djelovanja (poslovna, medicinska, pravna informatika).“ (Hrvatska enciklopedija, 2020) Ministarstvo znanosti i obrazovanja (u daljnjem tekstu: MZO) u *Kurikulumu nastavnog predmeta Informatika za osnovnu školu i gimnazije* (u daljnjem tekstu: *Kurikulum nastavnog predmeta Informatika*) pod pojmom Informatika u obrazovnom sustavu podrazumijeva stjecanje vještina kod uporabe informacijske i komunikacijske tehnologije, njihovu upotrebu u obrazovnom procesu te rješavanje problema uporabom nekog programskog jezika. (Ministarstvo znanosti i obrazovanja, 2019)

Nastava informatike u svim razredima osnovne škole provodi se dva sata tjedno, odnosno 70 sati godišnje. Informatika je obavezni nastavni predmet samo u 5. i 6. razredu osnovnoškolskog obrazovanja. Od 2020./21. školske godine sve osnovne škole Informatiku obavezno trebaju nuditi kao izborni predmet za učenike od 1. do 4. razreda osnovne škole. (Ministarstvo znanosti i obrazovanja, 2020)

U današnje vrijeme sve je veća potreba za digitalnim vještinama i kompetencijama. U mnogim zemljama, između ostalih Estoniji, Izraelu, Finskoj i Ujedinjenom Kraljevstvu, sve je veća potpora vlade u integraciji kodiranja, kao nove pismenosti, u nastavu te podržavanje kreativnih zadataka u rješavanju problema. (Papavlasopoulou et al., 2020) Uvođenje obavezne

izborne nastave Informatike u primarnom obrazovanju podršku dobivaju i učenici u Hrvatskoj. Kako bi se omogućilo održavanje nastave Informatike bilo je predviđeno opremanje 1137 područnih škola hibridnim računalima. (Ministarstvo znanosti i obrazovanja, 2020)

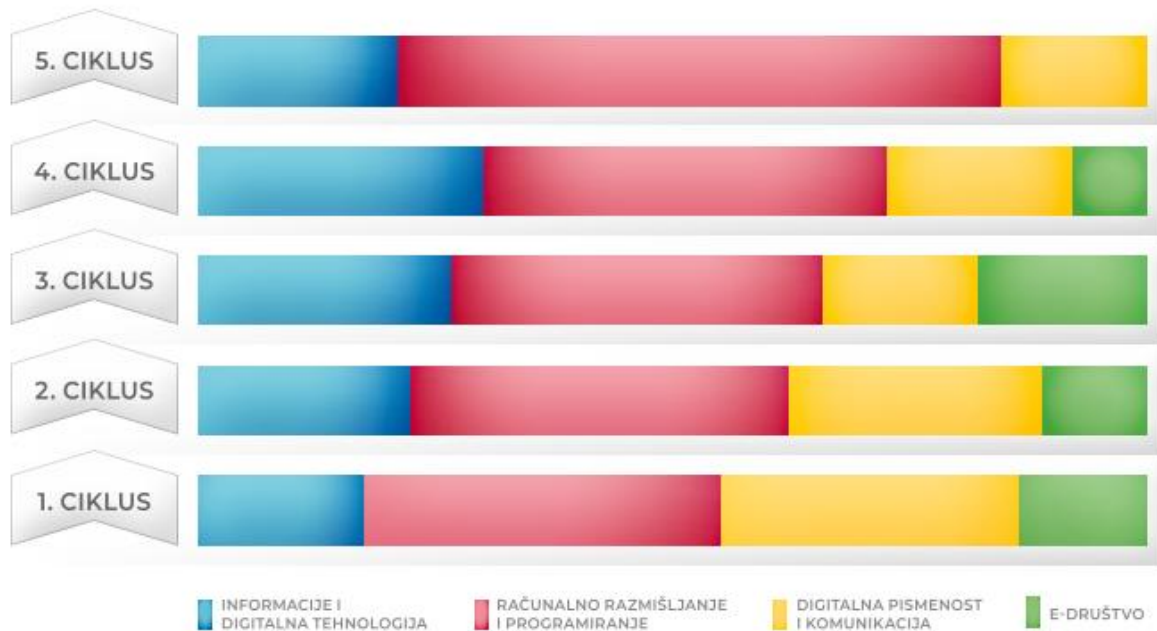
## **5. Kurikulum nastavnog predmeta Informatika**

*Kurikulum za nastavni predmet Informatika* primjenjuje se od školske godine 2018./19. za učenike od 5. do 8. razreda osnovne škole te za učenike od 1. do 4. razreda srednje škole, dok se za učenike od 1. do 4. razreda osnovne škole primjenjuje od školske godine 2020./21.

*Kurikulum nastavnog predmeta Informatika* se sastoji od sljedećih poglavlja: opis nastavnog predmeta, odgojno-obrazovni ciljevi učenja i poučavanja, domene u organizaciji predmetnog kurikuluma, odgojno-obrazovni ishodi prema razredima i domenama, učenje i poučavanje nastavnog predmeta te vrednovanje odgojno-obrazovnih ishoda. (Ministarstvo znanosti i obrazovanja, 2019) *Kurikulum nastavnog predmeta Informatika* je mješovitog tipa koji zadaje jezgru, a učitelju daje slobodu u odabiru nastavnih materijala, organizaciji i metodama rada. U svom opisu nastavnog predmeta Informatika prepoznaje i naglašava važnost poznavanje temeljnih informatičkih koncepata i nužnosti informatičkih kompetencija u svakodnevnom životu, a težište obrazovnog procesa usmjerava se rješavanju problema i programiranju u svrhu razvoja računalnog razmišljanja koje omogućava razumijevanje, analizu i rješavanje problema odabirom odgovarajućih strategija, algoritama i programskih rješenja. (Ministarstvo znanosti i obrazovanja, 2019) Razvoj računalnog načina razmišljanja uključuje mnoge tehnike rješavanja problema koje se razvijaju kod učenika, a uključuju prikazivanje „informacija apstrakcijama, logičko povezivanje i analizu podataka, automatizaciju rješenja uporabom algoritamskoga razmišljanja, prepoznavanje, analizu i primjenu mogućih rješenja s ciljem postizanja učinkovitoga rezultata vodeći računa o dostupnim resursima, formuliranje problema načinom primjerenim uporabi računala i računalnih alata, generalizaciju procesa rješavanja problema primjenjivoga na čitav niz sličnih problema.“ (Ministarstvo znanosti i obrazovanja, 2019, p. 6)

Odgojno-obrazovni ciljevi učenja i poučavanja odnose se na informatičku pismenost, digitalnu mudrost, razvoj kritičkog mišljenja, kreativnost i inovativnost, računalno mišljenje, rješavanje problema i programiranje, komunikaciju i suradnju u digitalnom okruženju, sigurnosne preporuke i pravne odrednice. Domene kojima se realiziraju ciljevi predmeta Informatike su *Informacije i digitalna tehnologija, Računalno razmišljanje i programiranje, Digitalna pismenost i komunikacija* te *e-Društvo*. (Ministarstvo znanosti i obrazovanja, 2019)

Domene se međusobno isprepliću i nadopunjuju te jedan nastavni sat može objediniti više domena, odnosno ishoda. Ishodi nisu poredani kronološki već učitelj ima slobodu kreativno planirati proces i redoslijed poučavanja te prilagođavati sadržaj ostalim nastavnim predmetima. (Ministarstvo znanosti i obrazovanja, 2019)



Slika 1. Prikaz udjela domene po ciklusima  
Izvor: (Ministarstvo znanosti i obrazovanja, 2019)

Na slici 1. prikazan je udio pojedine domene po ciklusima učenja. Vidljivo je da najveći udio u svim ciklusima zauzima domena *Računalno razmišljanje i programiranje* što potvrđuje prethodno navedeno, težište je na procesu usvajanja rješavanja problema i programiranju. Udio pojedine domene je ugrubo određen budući da ne sadrži postotak. U prvom (1. i 2. razred), drugom (3., 4. i 5. razred) te trećem ciklusu (6., 7. i 8. razred) udio sadržaja iz domene *Računalno razmišljanje i programiranje* je prilično podjednak, otprilike 1/3.

### 5.1. Domena *Računalno razmišljanje i programiranje*

Domena *Računalno razmišljanje i programiranje* označena je oznakom *B*. U opisu domene istaknuta je važnost razvoja računalnog razmišljanja i programiranja kod učenika. Ističe se važnost razvoja logičkog i algoritamskog razmišljanja, naročito korištenjem računala, čime učenici postaju stvaratelji. Kod razvoja računalnog razmišljanja naglasak se stavlja na cjelokupan proces stvaranja programa od početne ideje pa do konačnog proizvoda. Pritom se fokus manje stavlja na sintaksu i semantiku programskog jezika. (Ministarstvo znanosti i obrazovanja, 2019) Budući da kurikulum ne postavlja ishode koji se odnose na određeni

programski jezik, učiteljima se daje mogućnost odabira programskog jezika kojim će ostvariti ishode. Aktivnosti i sadržaji koji proizlaze iz ishoda trebaju biti osmišljeni tako da razvijaju inovativnost, stvaralaštvo, poduzetnost te razvijaju znanja za daljnji život. (Ministarstvo znanosti i obrazovanja, 2019) Kod rješavanja problema potrebno je polaziti od aspekata računalnog razmišljanja te omogućiti razdvajanje složenih problema na jednostavne. Tijekom rješavanja problema potrebno je uključiti standardne postupke razvoja programa, a učenike treba poticati na inovativnost, poduzetnost te preuzimanje inicijative kod izrade dizajna i razvoja novih modela i proizvoda. (Ministarstvo znanosti i obrazovanja, 2019) Kvalitetnim informatičkim obrazovanjem razvijaju se vještine logičkog zaključivanja, modeliranja, apstrahiranja, rješavanja problema. Računalnim razmišljanjem potiče se preciznost i sustavnost, a može se primjenjivati u različitim područjima, dok programiranje razvija samopouzdanje, preciznost, upornost, sposobnost komunikacije i timski rad. (Ministarstvo znanosti i obrazovanja, 2019)

*Tablica 2. Odgojno-obrazovni ishodi u domeni Računalno razmišljanje i programiranje po razredima*

Izvor: (Ministarstvo znanosti i obrazovanja, 2019)

<b>1. RAZRED</b>	B. 1. 1. Učenik rješava jednostavan logički zadatak.
	B. 1. 2. Učenik prati i prikazuje slijed koraka potrebnih za rješavanje nekoga jednostavnog zadatka.
<b>2. RAZRED</b>	B. 2. 1. Učenik analizira niz uputa koje izvode jednostavan zadatak, ako je potrebno ispravlja pogrešan redoslijed.
	B. 2. 2. Učenik stvara niz uputa u kojemu upotrebljava ponavljanje.
<b>3. RAZRED</b>	B. 3. 1. Učenik stvara program korištenjem vizualnoga okruženja u kojem se koristi slijedom koraka, ponavljanjem i odlukom te uz pomoć učitelja vrednuje svoje rješenje.
	B. 3. 2. Učenik slaže podatke na koristan način.
<b>4. RAZRED</b>	B. 4. 1. Učenik stvara program korištenjem vizualnog okruženja u kojem koristi slijed, ponavljanje, odluku i ulazne vrijednosti.
	B. 4. 2. Učenik rješava složenije logičke zadatke s uporabom računala ili bez uporabe računala.
<b>5. RAZRED</b>	B. 5. 1. Učenik se koristi programskim alatom za stvaranje programa u kojemu se koristi ulaznim i izlaznim vrijednostima te ponavljanjem.
	B. 5. 2. Učenik stvara algoritam za rješavanje jednostavnoga zadatka, provjerava ispravnost algoritma, otkriva i popravljiva pogreške.
<b>6. RAZRED</b>	B. 6. 1. Učenik stvara, prati i preuređuje programe koji sadrže strukture grananja i uvjetnoga ponavljanja te predviđa ponašanje jednostavnih algoritama koji mogu biti prikazani dijagramom, riječima govornoga jezika ili programskim jezikom.
	B. 6. 2. Učenik razmatra i rješava složeniji problem rastavljajući ga na niz potproblema.

<b>7. RAZRED</b>	B. 7. 1. Učenik razvija algoritme za rješavanje različitih problema koristeći se nekim programskim jezikom pri čemu se koristi prikladnim strukturama i tipovima podataka.
	B. 7. 2. Učenik primjenjuje algoritam (sekvencijalnog) pretraživanja pri rješavanju problema.
	B. 7. 3. Učenik dizajnira i izrađuje modularne programe koji sadrže potprograme u programskom jeziku.
	B. 7. 4. Učenik se koristi simulacijom pri rješavanju nekoga, ne nužno računalnoga, problema.
<b>8. RAZRED</b>	B. 8. 1. Učenik identificira neki problem iz stvarnoga svijeta, stvara program za njegovo rješavanje, dokumentira rad programa i predstavlja djelovanje programa drugima.
	B. 8. 2. Učenik prepoznaje i opisuje algoritam sortiranja, primjenjuje jedan algoritam sortiranja za rješavanje zadanoga problema u programskom jeziku.
	B. 8. 3. Učenik prepoznaje i opisuje mogućnost primjene rekurzivnih postupaka pri rješavanju odabranih problema te istražuje daljnje mogućnosti primjene rekurzije.

## **6. Prijedlozi ostvarenja ishoda iz domene *Računalno razmišljanje i programiranje* u hrvatskim udžbenicima i radnim bilježnicama s obzirom na preporuke iz *Kurikuluma nastavnog predmeta Informatika***

U ovom poglavlju slijedi prikaz prijedloga različitih izdavača konkretiziranih udžbenicima, radnim bilježnicama i ostalim materijalima za učitelje i učenike iz domene *Računalno razmišljanje i programiranje*. Koristi se tablični prikaz s ciljem usporedbe materijala različitih izdavača s preporukama iz *Kurikuluma za nastavni predmet Informatika*. Odabrani izdavači su: *Školska knjiga*, *SysPrint*, *Alfa* i *Profil Klett*.

Sadržaji se usvajaju tijekom cijelog školovanja, a ishodi su podijeljeni po razredima. Ishodi se načelom spiralnog modela proširuju i produbljuju kroz razrede. (Ministarstvo znanosti i obrazovanja, 2019) Usvojeni ishodi na kraju određenog razreda smatraju se usvojenima te se koriste u ostalim razredima. Iznimku predstavlja peti razred, u kojem je nastava Informatike obavezna, nakon četiri godine izborne nastave. Ishodi u petom razredu izazovni su učenicima kojima je to prva godina učenja, dok je većina razrade ishoda kod ostalih učenika već usvojena. Preporuke za ostvarivanje odgojno-obrazovnog ishoda sadrže sadržaje, situacije i aktivnosti za ostvarivanje određenog odgojno-obrazovnog ishoda. U preporukama se velika važnost pridaje analizi problema i osmišljavanju rješenja, uspoređivanju različitih pristupa, argumentiranju svojih postupaka i uvažavanju tuđih, vizualnom predstavljanju rješenja i sagledavanju cijelog postupka, naročito u prva četiri razreda. (Ministarstvo znanosti i obrazovanja, 2019) U nastavku

učenja to prelazi na višu razinu, a sadržaj programiranja se sve više povezuje s drugim nastavnim predmetima.

Tablica 3. 1. razred

<p><i>Kurikulum nastavnog predmeta Informatika</i> (Ministarstvo znanosti i obrazovanja, 2019)</p>	<p><i>Školska knjiga (e-SVIJET 1)</i> (Blagus, Ljubić Klemše, Flisar Odorčić, Bubica, et al., 2020)</p>	<p><i>SysPrint (Mišolovka 1)</i> (Sokol, Mandić, Purgar, et al., 2020)</p>
<p>-1/3 sadržaja iz domene <i>Računalno razmišljanje i programiranje</i></p> <p>- korištenje slagalica, igre <i>Memory</i>, labirinta, zagonetka, razvojnih logičkih i kreativnih zadataka, mozgalica, zadataka s natjecanja <i>Dabar</i></p> <p>- poticanje logičkog mišljenja, kreativnosti i intelektualnog zadovoljstva uz pomoć računala ili bez njega</p> <p>- primjeri iz svakodnevnog života, poput spremanja za školu, slaganja sendviča, prelaska ceste...</p> <p>- korištenje grafičkih programa i aplikacije s naznakom programiranja, poput <i>Code.org</i>, <i>Run Marco</i> ili <i>Hour of Code</i>, a prema mogućnostima i hardverska rješenja za vizualizaciju programiranja</p>	<p>POZITIVNO</p>	
	<p>-zadatci prepoznavanja, spajanja, sastavljanja oblika, labirinta, premetaljki, izbacivanja uljeza, sudokua, zagonetki, povezivanja, slijeđenja koraka/uputa, zadatci iz područja matematike, slagalica, prepoznavanje i nadopunjavanje niza</p> <p>-<i>Scratch</i>: upoznavanje osnova rada, govor likova, kretanje i skretanje likova, korištenje olovke (crtanje)</p> <p>-primjeri iz svakodnevnog života: prelaženje ceste</p> <p>-priručnik za korištenje <i>mBota</i>, <i>Scratcha</i> i <i>Minecrafta</i></p>	<p>-zadatci povezivanja i zaključivanja, poput izbacivanja uljeza, sastavljanja riječi, slagalica, prepoznavanja i nadopunjavanja niza, rebusa, labirinta, slijeđenja koraka/uputa, sudokua, zadataka iz područja matematike</p> <p>- crtanje u grafičkom programu <i>Korni 1</i>- osnove naredbe u programu, crtanje kvadrat i pravokutnik, kretanje bez crte te kretanje ukoso</p>
	<p>NEGATIVNO</p>	
<p>-samo osam nastavnih tema (premali omjer s obzirom na ostale domene)</p>	<p>-samo šest nastavnih tema</p> <p>-samo dvije teme s logičkim zadacima bez računala</p> <p>-nema primjera iz svakodnevnog života</p>	

	-nema korištenja <i>Code.org</i> , <i>Run Marco</i> ili <i>Hour of Code</i> (samo <i>Scratch</i> )  -pre malo prijedloga iz svakodnevnog života	-nema korištenja <i>Code.org</i> , <i>Run Marco</i> ili <i>Hour of Code</i> (samo <i>Korni 1</i> )
--	--	--

Tablica 4. 2. razred

<i>Kurikulum nastavnog predmeta Informatika</i> (Ministarstvo znanosti i obrazovanja, 2019)	<i>Školska knjiga (e-SVIJET 2)</i> (Blagus, Ljubić Klemše, Flisar Odorčić, Ružić, et al., 2020a)	<i>SysPrint (Mišolovka 2)</i> (S. Horvat & Prpić, 2020)
-1/3 sadržaja iz domene <i>Računalno razmišljanje i programiranje</i>  - analiza uputa na poznatom zadatku, otkrivanje pogrešaka i ispravljanje kroz različite igre, poput igre sakrivanja, davanja uputa za kretanje, vođenje robota labirintom...  - grafički programi i aplikacije s naznakom programiranja, zadatci s natjecanja <i>Klokan bez granica</i> i <i>Dabar</i> te hardverska rješenja za vizualizaciju programiranja  - pomicanja likova sa ili bez računala te davanje uputa za kretanje, koraci se mogu prikazivati i crtežima	<b>POZITIVNO</b>	
	-19 nastavnih tema -baza 1. razred, proširivanje: slijed koraka, prepoznavanje i nadopunjavanje niza, rješavanje problema, ispravljanje pogrešaka, računanje s objektima, prepoznavanje oblika, slagalice, labirinti, osmosmjerke, razvrstavanje prema svrsi/ karakteristikama, sudoku, labirint- navođenje  - <i>Scratch</i> : skupine blokova kretanja, izgleda, upravljanja i događaja; uvođenje blokova ponavljanja, izrada likova i pozadina, spremanje programa  -priručnici za <i>mBot</i> , <i>Scratch</i> i <i>Minecraft</i>	-12 nastavnih tema -zadatci navođenje robota, mnogo različitih zadataka iz svakodnevnog života koji zahtijevaju slijeđenje koraka, prepoznavanje i nadopunjavanje niza  - <i>Korni 1</i> : pokretanje (upoznaje veličine kuta i ponavljanja)  -preporuka za rješavanje zadataka <i>Code.org</i> i <i>RunMarco</i>



- ponavljanje sa zadanim brojem- crtanje geometrijskih likova	NEGATIVNO	
	-od grafičkih programa koristi se samo <i>Scratch</i> -nema crtanja geometrijskih likova -pre malo primjera iz svakodnevnog života	-apstraktno: veličina kutova

Tablica 5. 3. razred

<i>Kurikulum nastavnog predmeta Informatika</i> (Ministarstvo znanosti i obrazovanja, 2019)	<i>Školska knjiga (e-SVIJET 3)</i> (Blagus, Ljubić Klemše, Flisar Odorčić, Ružić, et al., 2020b)	<i>SysPrint (Mišolovka 3)</i> (Sokol, Mandić, Lohajner, et al., 2020a)
-1/3 sadržaja iz domene <i>Računalno razmišljanje i programiranje</i> - primjeri događaja koji utječu na odluke, primjerice utjecaj vremenske prognoze - grafički programi i aplikacije s naznakom programiranja, a prvi puta je preporuka za grafički programski jezik <i>Scratch</i> -zadatci s natjecanja te hardverska rješenja za vizualizaciju, prema mogućnosti koristi se robot kojim učitelj demonstrira unošenje različitih naredbi -razvoj preciznosti i jednostavnosti u objašnjavanju rješenja problema	POZITIVNO	
	-veliki broj raznolikih zadataka prilagođenih dobi i potrebama učenika -sadržaji povezani s drugim domenama i nastavnim predmetima -zadatci sastavljanja rebusa, križaljka, nizova... -svrstavanje i grupiranje podataka prema pravilima u tablice i mape, prikaz dijagramima - <i>Scratch</i> : blokovi ponavljanja i grananja, pokretanje likova (tipkama), dodavanje zvukova- stvaranje priča i animacija likova -priručnik za <i>mBot</i> , <i>Scratch</i> i <i>Minecraft</i>	- raznoliki zadatci s natjecanja Dabar i donošenja odluka -mnogi zadatci i primjeri iz svakodnevnog života povezani s donošenjem odluka - <i>Scratch</i> : upoznavanje s osnovama rada, dodavanje olovke, korištenje blokova ponavljanja i grananja; crtanje i izrada igara

- svrstavanje podataka, primjerice raspored učenika s obzirom na datum rođenja, boju/vrstu odjeće... - prikaz podataka tablično i dijagramima	NEGATIVNO	
	-23 nastavne teme (više od 1/3 s obzirom na ostale teme) -premalo primjera iz svakodnevnog života	- samo 7 nastavnih tema -nema svrstavanja podataka, malo tabličnih prikaza, nema korištenja dijagrama

Tablica 6. 4. razred

<i>Kurikulum nastavnog predmeta Informatika (Ministarstvo znanosti i obrazovanja, 2019)</i>	<i>Školska knjiga (e-SVIJET 4) (Blagus, Ljubić Klemše, Ružić, &amp; Stančić, 2020)</i>	<i>SysPrint (Mišolovka 4) (Sokol, Mandić, Lohajner, et al., 2020b)</i>
-1/3 sadržaja iz domene <i>Računalno razmišljanje i programiranje</i> -upoznavanje osnovnih segmenata izrade programa- ulaz, obrada podataka i izlaz, npr. računanje s ulaznim vrijednostima ili razgovor likova u grafičkim programskim jezicima i aplikacijama s naznakom programiranja -korištenje demonstracije robotima, zadataka s natjecanja, te hardverskih rješenja za vizualizaciju (prema mogućnosti) -razvoj strategija analiziranja i rješavanja problema upotrebom različitih igara, zagonetki i	POZITIVNO	
	-vrlo raznoliki zadatci bez korištenja računala na principu natjecanja <i>Dabar</i> - <i>Scratch</i> : korištenje naučenih blokova, razgovor likova, izrada različitih igara, blokovi grananja, jednostavno korištenje varijabli -upoznavanje varijabli i segmenata izrade programa -povezivanje sadržaja s drugim domenama i nastavnim predmetima -priručnik za korištenje <i>mBota</i> , <i>Scratcha</i> i <i>Minecrafta</i>	-rad u <i>Micro:bitu</i> -raznoliki zadatci s natjecanja <i>Dabar</i> , matematičke mozgalice -upoznavanje varijabla i segmenata izrade programa - <i>Scratch</i> : upoznavanje i korištenje varijabla, računanje s ulaznim vrijednostima, korištenje blokova grananja te izrada igara -projektni zadatci

mozgalica, primjerice s natjecanja	NEGATIVNO	
	-22 nastavne teme (više od 1/3 s obzirom na ostale teme)	-samo 7 nastavnih tema -zadatci u <i>Scratchu</i> su složeni s obzirom na uzrast učenika i dosadašnje iskustvo sa <i>Scratchom</i>

Tablica 7. 5. razred

<i>Kurikulum nastavnog predmeta Informatika</i> (Ministarstvo znanosti i obrazovanja, 2019)	<i>Školska knjiga (Moj portal 5)</i> (Ružić et al., 2020a)	<i>Alfa (Like IT 5)</i> (Rihter et al., 2019)
-instalacija ili poveznica za programski alat -upoznavanje sučelja i načina korištenja alata -korištenje videosadržaja i alata za vizualizaciju -učenici se potiču na samostalno uređivanje i mijenjanje pokaznih primjera te izradu niza uputa kao rješenja problema -svakodnevne aktivnosti - javno predstavljanje, opisivanje postupka analize i rješavanja problema - osmišljavanje i zadavanje jednostavnih problema u paru - rješenje problema od analize do testiranja, prepravljavanja i vrednovanja rješenja	POZITIVNO	
	-primjeri iz svakodnevnih aktivnosti -zadatci za samostalno osmišljavanje algoritama, dijagrama toka... - <i>Python</i> : instaliranje, osnove rada, varijable, ulazne i izlazne vrijednosti (na primjerima računanja opsega i površine), crtanje, petlja, - usvajanje načina rada programa te analize i ispravljanje pogrešaka -dodatni priručnici za <i>mBot</i> , <i>Scratch</i> , <i>Logo</i> , <i>Python</i> , <i>Raspberry Pi</i> , <i>Minecraft</i> , i korištenje <i>Pythona</i> u <i>Micro:bitu</i>	-primjeri iz svakodnevnih aktivnosti -zadatci za samostalno osmišljavanje algoritama i dijagrama toka, ispravljanja pogrešaka, upoznavanje različitih vrste algoritama te upotrebljavanje na zadatcima -zadatci koji zahtijevaju rješenje problema od analize do testiranja, prepravljavanja i vrednovanje rješenja - <i>Scratch</i> : upoznavanje s programskim jezikom, upoznavanje svih kategorija blokova + olovka, korištenja petlja, grananja, varijabla, ulaznih i izlaznih vrijednosti -dodatni zadatci za rad u paru i skupini, mogućnosti

<p>-zadatci s natjecanja i hardverska rješenja za vizualizaciju</p>		<p>za dodatno učenje i proučavanje, zanimljivosti -materijal i zadatci za rad s <i>Micro:bitom</i> -materijal i zadatci za rad u <i>Pythonu</i>: osnove rada, ulazne i izlazne vrijednosti, varijable i ponavljanje</p>
<p>NEGATIVNO</p>		
	<p>-loš redoslijed ponuđenih tema -crtanje geometrijskih likova više odgovara ishodima i preporukama za 6. razred -premalo primjera iz svakodnevnog života -nema zadataka koji zahtijevaju rješenje problema od analize do testiranja, prepravljanja i vrednovanje rješenja</p>	<p>-</p>

Tablica 8. 6 razred

<p><i>Kurikulum nastavnog predmeta Informatika</i> (Ministarstvo znanosti i obrazovanja, 2019)</p>	<p><i>Školska knjiga (Moj portal 6)</i> (Ružić et al., 2020b)</p>	<p><i>Profil Klett (Informatika 6)</i> (Deljac et al., 2020a)</p>
<p>-istraživanje ponašanja algoritama uređivanjem i mijenjanjem pokaznih primjera -kreiranje niza uputa za rješenje problema -korištenje ulaznih vrijednosti i algoritamskih struktura slijeda, grananja i ponavljanja -primjeri: ispisivanje određenog niza brojeva s određenim svojstvima, traženje najvećih/najmanjih vrijednosti kod najviše tri vrijednosti, stvaranje dijaloga među likovima, te ponavljanje izvršavanja aktivnosti (kretanje likova, mijenjanje obilježja, računanje opsega i površine geometrijskih likova)... -sadržaj s mrežnih stranica <i>Code week, Hour of code, App studio, Code Academy</i> te hardverska rješenja za vizualizaciju</p>	<p>POZITIVNO</p>	
	<p>-primjeri iz svakodnevnog života, jednostavno i složeno grananje, ispravljanje pogrešaka -<i>Python</i>: grananje, ponavljanje (crtanje geometrijskih likova, ispis parnih brojeva, pogodi broj...) uvjetno ponavljanje -projektni zadatci: program za izračunavanje prosječne vrijednosti, program za provjeru točnosti izračuna -dodatni priručnici za <i>mBot, Scratch, Logo, Python, Raspberry Pi, Minecraft</i>, korištenja <i>Pythona</i> u <i>Micro:bitu</i></p>	<p>-<i>Logo</i>: zbrajanje brojeva, izračun površina i opsega, izračun prosječne visine, petlje -<i>Python</i>: petlje (ispis parnih brojeva, raspored od najvećeg do najmanjeg...), crtanje (olimpijski krugovi...), grananje (crtanje uz uvjet) -raznoliki zadatci poredani po težini, prilagođeni djeci -analiza i ispravljanje pogrešaka -rješavanje problema po principu razumijevanja problema, izrade plana, provedbe i pregled rezultata</p>

-analiziranje, planiranje i rješavanje problema u skupinama - rastavljanje problema na manje, uočavanje dijelova koda koji su rješenje nekog poznatog problema, mijenjanje dijelova koda i uklapanje u veće rješenje - logički tipovi zadataka, logičke zagonetke, zadatci kombinacija...	NEGATIVNO	
	-nema sadržaja s mrežnih stranica <i>Code week, Hour of code, App studio, Code Academy</i> -pre malo analiza, ispravljanja i rješavanja problema -pre malo rastavljanja problema na manje i prepoznavanja manjih dijelova rješenja	-nema sadržaja s mrežnih stranica <i>Code week, Hour of code, App studio, Code Academy</i>

Tablica 9. 7. razred

<i>Kurikulum nastavnog predmeta Informatika</i> (Ministarstvo znanosti i obrazovanja, 2019)	Školska knjiga (Moj portal 7) (Ružić et al., 2020c)	Profil Klett (Informatika 7) (Deljac et al., 2020b)
-analiza problema, prepoznaje ulazne vrijednosti za rješenje problema te moguće izlazne vrijednosti -na primjerima različitih problema pokazuje se potreba za složenim tipom podataka, niza ili stringova -korištenje jednostavnih funkcija -na primjerima se analiziraju, predviđaju i	POZITIVNO	
	- <i>Python</i> : brojevi, znakovni i logički podatci, jednostavne funkcije, sekvencijalno traženje (traženje slova u riječi, imena u nizu imena...), crtanje (liste i funkcije) -primjeri iz svakodnevnih situacija: lista s visinama učenika, ocjenama iz nekog predmeta (prikupljanjem podataka za analizu, oblikovanje, grafički prikaz i	- <i>Logo</i> : ponavljanje naučenog, tipovi podataka, potprogrami (crtanje niza geometrijskih likova...), sekvencijalno traženje (u dvije liste) - <i>Python</i> : ponavljanje naučenog, složeni tipovi podataka, jednostavne funkcije, grananje, potprogrami (skupovi lektira) -analiza problema

<p>izmjenjuju algoritmi za rješavanje sličnih problema</p> <p>-sekvencijalno traženje: korištenje situacije iz svakodnevnog života s potrebom pretraživanja, uočavanje obrazaca i razlika, primjerice nađi pandu na slici, uoči riječ, pronađu osobu u gužvi...</p> <p>-prepoznavanje i opisivanje kriterija za pretraživanje</p> <p>-primjerima pojasniti programski kod</p>	<p>analiza pomoću programa), zbrajanje brojeva</p> <p>-analiza i ispravljanje pogrešaka</p> <p>-projektni zadatci povezani s drugim nastavnim predmetima: prikaz dijagramom ovisnost sile o duljini</p> <p>-dodatni priručnici za <i>mBot</i>, <i>Scratch</i>, <i>Logo</i>, <i>Python</i>, <i>Raspberry Pi</i>, <i>Minecraft</i>, korištenja <i>Pythona</i> u <i>Micro:bitu</i></p>	<p>-zadatci iz svakodnevnog života: liste najmnogoljudnijih gradova, ispis članova obitelji prema uvjetu</p> <p>-prepoznavanje i korištenje potprograma u drugim programima</p>
<p>- predložiti i analizirati rješenje potproblema te povezati s problemom</p> <p>-korištenje alata za vizualizaciju</p> <p>-prikupljanje podataka, analiza, grafički prikaz i analiza pomoću programa (izostanci u razredu, kretanje temperature zraka...)</p>	<p>NEGATIVNO</p>	<p>-</p> <p>-premalo primjera iz svakodnevnog života povezanih s sekvencijalnim traženjem</p> <p>-apstraktni primjeri</p> <p>-premalo teorijskog dijela koji objašnjava primjenu</p>

Tablica 10. 8 razred

<p><i>Kurikulum nastavnog predmeta Informatika</i> (Ministarstvo znanosti i obrazovanja, 2019)</p>	<p><i>Školska knjiga (Moj portal 8)</i> (Ružić et al., 2021)</p>	<p><i>Alfa (Like IT 8)</i> (Rihter et al., 2021)</p>
<p>-razgovor o primjerima zanimljivih problema za rješavanje, samostalno predlaganje ideja ili biranje projektnog zadatka -priprema i analiza problema u skupinama, parovima ili pojedinačno -teme iz nižih razreda i drugih predmeta (primjena u tim razredima i predmetima za ponavljanje) - izrada matematičkih/ fizikalnih operacija i postupaka; izrada mobilne aplikacije ili Internet of things te korištenje različitih primjera iz učeničkih i studentskih aplikacija s raznih smotri, natjecanja i kampanja -gotove funkcije sortiranja -slaganje/organiziranje objekata prema kriteriju, primjerice od najmanjem do najvećeg</p>	<p style="text-align: center;"><b>POZITIVNO</b></p> <p>-primjeri i svrha sortiranja podataka (sortiranje brojeve od najmanjeg do najvećeg) -<i>Python</i>: funkcija sortiranja, rekurzija (računanje umnoška prvih n brojeva, crtanje stabla) -prepoznavanje potprograma u programu -rekurzija (Matrojske, školjka morskog puža...) -grafičko sučelje- <i>Tkinter</i>: dodavanje prozora, gumba, povezivanje gumba s akcijom, polja za ispis, polja za unos -izrada programa povezana s ostalim predmetima u <i>Tkinteru</i>- molekule, računanje opsega i površine -projektni zadatci: program s izbornikom za vježbanje fizike -izrada dokumentacije za projekt -timska izrada web stranice</p>	<p>-izrada aplikacija za pametne uređaje, pretvaranje <i>Scratch</i> projekta u <i>JavaScript</i> te pokretanje na mobitelu -<i>Scratch</i>: rekurzija (ispis brojeva do određenog, Sierpinskog trokut...), sortiranja podataka, izrada gumba -rekurzija na primjerima: Hanoa, Sierpinskog trokuta, Kochove pahulje -primjeri sortiranja podataka -primjena <i>Scratch</i> programa u matematici i fizici -<i>Python</i>: rekurzija (ispis brojeva do određenog, Kochove pahulja...), sortiranje -<i>Tkinter</i>: dodavanje prozora, gumba, povezivanje gumba s akcijom, polja za ispis, polja za unos -videosadržaji</p>



<p>-prikaz i diskusija o grafičkim primjerima, trokut Sierpinskog, Kochova pahulja</p> <p>-prikaz primjera rekurzivnih fenomena iz svakodnevnog života te rasprava o mogućnostima i korištenje konkretnih modela, Matrojske, tornjevi Hanoa</p> <p>-opisivanje i pokazivanje osnovnih koraka rekurzivnoga postupka</p>	<p>-priručnici za programiranje <i>mBota, Minecraft, Python, Raspberry Pi</i></p>	
	<p>NEGATIVNO</p>	
	<p>-potrebni zanimljivi primjeri: Hanoa, Sierpinskog trokuta, Kochove pahulje</p> <p>- nema primjera Internet of things</p>	<p>-potrebno je više primjera iz drugih nastavnih predmeta i razreda te projektni zadatci</p> <p>-nema primjera Internet of things</p> <p>- primjereniji tekstualni programski jezici (umjesto <i>Scratcha</i>)</p>

## 7. Pristup poučavanju

Kurikularnom reformom fokus se premješta sa sadržaja i učitelja na ishode, a učenje i poučavanje usmjerava na svakog učenika. Kurikulum daje ishode koji se trebaju ostvariti u pojedinoj godini učenja i smjernice, no učitelj ima slobodu u izboru načina, redosljeda i vremena, nastavnih metoda i sredstava kojima će ostvariti propisano te nastavu učiniti što kvalitetnijom i interaktivnijom, a aktivnosti i sadržaje prilagođava potrebama i interesima učenika, razreda i škole. (Ministarstvo znanosti i obrazovanja, 2019) Oblici rada trebaju biti raznoliki, prilagođeni aktivnostima i mogućnostima učenika. Grupnim radom i radom u paru potiče se suradničko učenje i poticajno okruženje. Odabir sadržaja, metoda, sredstava i dužina aktivnosti ovisi o kognitivnom stadiju u kojem se učenici nalaze.

Prema Piagetu djeca do 6. do 11. godine nalaze se u stadiju konkretnih operacija, dok nakon toga prelaze u stadij formalnih operacija. Učenici u stadiju konkretnih operacija apsorbiraju informacije i prilagođavaju ih sebi te spoznaju konkretnu realnost i perspektivu ponašanja drugih osoba. (T. Horvat, 2019) Upravo s razlogom spoznaje konkretne realnosti učenicima u stadiju formalnih operacija potrebno je pristupiti s konkretnim svakodnevnim aktivnostima.

Učenici nisu spremni za prihvaćanje apstraktnih pojmova s kojima se često susreću kroz programiranje. Najbolji način spoznaje je kroz opipljive i vizualizirane primjere i aktivnosti, a tijekom učenja potrebno je pravilno odabrati aplikacije i programske jezike. Učenicima u tom stadiju potrebna je interakcija, aktivno učenje i rješavanje problema. Učenike treba staviti u poziciju u kojoj se susreću s nekim svakodnevnim problemom, a rješenje takvog problema potrebno je riješiti računalnim razmišljanjem, a u konačnici i programiranjem. Olabe i suradnici navode da konstrukcionizam smatra programiranje, kao čin izgradnje sustava i otklanjanja pogrešaka, pronalaženja prepreka i problema te njegovo rješavanje, najučinkovitijim načinom učenja djeteta. (Olabe et al., 2011) Iskustva učenja snažnija su kada su učenici aktivno uključeni u proces stvaranje vlastitih projekata i to tako da djeca primjenjuju prethodna znanja te otkrivaju nova. (Papavlasopoulou et al., 2020) Ključan objekt u konstrukcionizmu je „objekt za razmišljanje“, to je ono što će učenicima pružiti priliku za interakciju i aktivno sudjelovanje. (Papavlasopoulou et al., 2020) Obrazovna strategija konstrukcionističkog učenja je programiranje računalnih igara. Kurikulum nastavnog predmeta Informatika navodi „Igra je najprirodnija aktivnost djece i još uvijek vodeća pedagoška metoda u mlađemu uzrastu zbog čega je u najnižim razredima naglasak upravo na učenju igrama i zabavom uporabom digitalne tehnologije.“ (Ministarstvo znanosti i obrazovanja, 2019, p. 34) Mladenović pak navodi da pedagoška primjena igre može zamijeniti udžbenike ili laboratorije te kako ona nije samo nastavna metoda, već se treba koristiti i kao nastavno sredstvo. (Mladenović, 2019)

Učenici u stadiju konkretnih operacija zadržavaju pažnju na određenom predmetu ili situaciji 15 minuta, a provedba aktivnosti može trajati najduže 60 minuta uz pauze. (T. Horvat, 2019) Takve spoznaje dovode do zaključka da se nastavna treba realizirati u dva nastavna sata u tjednu. Do 11. godine djeca sve uspješnije zadržavaju pažnju i rješavaju duže zadatke, (T. Horvat, 2019) stoga se mogu organizirati i dulje aktivnosti te organizirati blok satovi u kojima je lakše organizirati projektnu i istraživačku nastavu.

Piaget je postavio razvojne faze te ljudsku populaciju s obzirom na godine svrstao u određeni stadij. Bruner je smatrao da se takvim pristupom djeca ograničavaju te je konceptom spiralnog kurikuluma objasnio kako se do složenih ideja dolazi poučavanjem na pojednostavljenoj razini te kasnije nadogradnjom na sve složenije razine. (McLeod, 2019) Bruner smatra da se proces učenja odvija kroz tri faze: akcijsku, ikoničku i simboličku. Akcijska faza podrazumijeva manipulaciju konkretnim objektima, ikonička slikama objekta, dok simbolička podrazumijeva reprezentaciju objekta. (Mladenović, 2019) Iz tog možemo zaključiti da je pristup učenju od konkretnog ka apstraktnom. Kroz takav princip učenici su sposobni

samostalno otkrivati i rješavati probleme. Bruner je smatrao da je najučinkovitije učenje programiranja otkrivanjem, a ne izlaganjem nastavnika. (McLeod, 2019) Stoga, uloga učitelja nije poučavanje informacija, već olakšavanje procesa učenja. Iz toga polazi da je svrha potpore omogućiti učeniku postizanje višeg stupnja razvoja pojednostavljuvanjem zadataka i ideja, motiviranjem i ohrabrivanjem, isticanjem važnih elemenata zadataka ili pogrešaka te davanjem modela koji se mogu oponašati. (McLeod, 2019) Učenici se mogu uključiti u izbor sadržaja, alata i programa, a preporuka je da učenici predstavljaju svoj rad. (Ministarstvo znanosti i obrazovanja, 2019) Aktivnim uključivanjem učenika u nastavni sat učenicima se daje na važnosti, potiče ih se i motivira. Motivacija se postiže odgovarajućim odabirom aktivnosti, metoda i sredstava, a veliku ulogu u motivaciji ima odabir programskih jezika.

*Kurikulum* naglašava potrebu povezivanja s ostalim domenama, nastavim predmetima i međupredmetnim temama. Kada učenik nauči određene segmente izrade programa, tada se može u problemskoj ili projektnoj nastavi iskoristiti naučeno i izraditi program koji se može koristiti na drugim nastavnim predmetima. Problemska nastava omogućava svladavanje vještina rješavanja problema te suradničko učenje. (CARNet, n.d.-a) Kao takva potiče računalno razmišljanje, a zadatak može zahtijevati i korištenje programiranja. Kod projektne nastave potreban je interdisciplinarni pristup. Takvim oblikom nastave učenici stječu dugotrajnije znanje, vještine i kompetencije, rješavaju problem, razvijaju pozitivan odnos prema radu, potiče radoznalost, inovativnost i osjećaj odgovornosti. (CARNet, n.d.-b) Projektna i problemska nastava višestruko je korisna za razvoj računalnog razmišljanja i programiranja, ali i korelacije s ostalim nastavnim predmetima.

## **8. Programska podrška**

Kod učenja programiranja učenici uče koncept programiranja, sintaksu i snalaženje u okruženju za programiranje. (Krpan, 2018) prema (Robins et al., 2003) Složena sintaksa učenicima predstavlja najveći problem te je često izvor frustracija, demotivacije i odustajanja. (Bubica et al., 2013) Osim problema sintakse, učenici se susreću s problemom vizualizacije algoritma. „Mini jezici“ razvili su se u svrhu jednostavnijeg programiranja s ograničenim skupom naredbi. (Krpan, 2018) Osim uklanjanja problema sintakse, u vizualnim programskim jezicima učenici apstraktne pojmove, poput petlji, grananja, varijabli mogu doživjeti u konkretnom vizualnom okruženju. (Bubica et al., 2013) S obzirom na sintaksu, programske jezike za početnike dijelimo na vizualno-blokovske, vizualno-tekstualne i proceduralno-tekstualne programske jezike. (Mladenović, 2019) Vizualno-blokovski jezici kao ulazne

podatke imaju tekstualne podatke, no u obliku blokova dok su izlazni podatci grafički prikazi, (Mladenović, 2019) kao primjerice u *Scratchu*. Takvi jezici odgovarali bi Brunerovoj ikoničkoj fazi. Vizualno-tekstualni jezici su jezici u kojima su ulazni podatci tekstualni, a izlazni grafički. (Mladenović, 2019) Jedan od primjera vizualno-tekstualnog jezika je *Logo*. Proceduralno-tekstualni jezici su jezici u kojima su ulazni i izlazni podatci tekstualni, (Mladenović, 2019) a jedan od njih je *Python*. Vizualno-tekstualni i proceduralno-tekstualni jezici odgovarali bi Brunerovoj simboličkoj fazi. S obzirom na razvojnu dob, Mladenović daje prikaz optimalnih programskih jezika. Programski jezici trebaju biti poticajni i motivirajući za učenike. Ako odabran programski jezik prelazi kognitivnu razinu učenika, može doći do kognitivnog preopterećenja koji dovodi do smanjenja samopouzdanja, motivacije te na posljetku do neuspjeha. (Mladenović, 2019) Do dosade i nedovoljne razine izazova dolazi ako se odabere programski jezik koji kognitivno nije izazovan, odnosno ukoliko su učenici „prerasli“ taj jezik. (Mladenović, 2019)

Tablica 11. Vrsta programskog jezika s obzirom na kognitivni razvoj

Izvor: (Mladenović, 2019)

Programski jezici/okruženje	Razine kognitivnog razvoja prema Piagetu		
	Konkretna	Pred-formalna	Formalna
Vizualni-blokovski (Primjer: Scratch, Alice, Hour of code, Microsoft MakeCode micro:bit editor)	optimalno	nije izazovno	nije izazovno
Vizualni-tekstualni (Primjer: Logo, Python-uz korištenje turtle biblioteke, Greenfoot)	preopterećenje	optimalno	nije izazovno
Proceduralni-tekstualni (Primjer: Python, BASIC)	preopterećenje	preopterećenje	optimalno

Iako Tablica 11. prikazuje primjerene programske jezike prema kognitivnim razvojnim fazama Piageta, ona odgovara i Brunerovim fazama učenja ako uključimo akcijsku fazu prije učenja samih programskih jezika. Razilaženje nastaje u 5. razredu kada nastavni predmet Informatike u hrvatskim školama postaje obavezan. Učenici koji se po prvi puta susreću s programiranjem, prema Bruneru bi trebali krenuti od akcijske faze, dok se prema Piagetu nalaze u konkretnom, odnosno pred-formalnom stadiju. Ipak, Bruner smatra da se kod starijih učenika

proces odvija brže te bi znatno kraće vrijeme trebali provesti na prvoj, odnosno drugoj fazi.(McLeod, 2019)

Poticajnu i motivirajuću atmosferu donosi vizualizacija programskih jezika. U vizualnim programskim jezicima vizualizacija je uglavnom dvodimenzionalna, no učenicima su još zanimljivija hardverska rješenja u kojima sami ili uz pomoć učitelja vizualiziraju i čine opipljivim svoj kod. Hardverska rješenja predstavljaju različite vrste robota i mikroračunala na koje se prenosi kod.

### **8.1. Aplikacije i igre za razvijanje logike programiranja**

*Code.org* nudi tečajeve za učenike i resurse za učitelje. Zadatci su prilagođeni i raspoređeni prema dobi i kognitivnom razvoju učenika, a temelje se na blokovima koji se spajaju u naredbe prema kojima likovi izvršavaju zadani zadatak. Blokovi se mogu pretvoriti u tekstualne naredbe u *JavaScriptu*.

*SpiriteBox* je igra temeljena na programiranju pomoću ikona, a kasnije i tekstualnim naredbama (temeljenim na *JavaScriptu*). Igra je primjerena početnicima, a zanimljiva je i naprednijima.

### **8.2. Programski jezik Scratch**

*Scratch* je grafički programski jezik koji pripada u skupinu vizualno-blokovskih jezika. S obzirom na kognitivni razvoj učenika primjeren je za rad u stadiju konkretnih operacija što odgovara učenicima do 6. razreda, no on se već u 5. razredu postupno može zamijeniti zahtjevnijim programskim jezikom. Pogodan je za početnike koji će naučiti osnove rada i koncepte, no nudi dovoljnu dubinu i za naprednije. Namijenjen je izradi interaktivnih priča, igara i animacija, a omogućuju i dijeljenje projekata s drugim članovima mrežne zajednice. (*Službena Web Stranica- Scratch*, n.d.) Bazira se na blokovima koji su, s obzirom na funkciju, podijeljeni u kategorije te obojeni određenom bojom. Blokovi se poput kockica u skriptama slažu u cjeline te se dobivaju naredbe za određeni lik ili pozadinu. Likovi se nalaze na pozornicama, odnosno pozadinama, gdje se pokretanjem programa izvršava sastavljeni kod, a rezultat su različite radnje u kojima sudjeluju likovi. *Scratch* je dostupan putem web browsera pa nije potrebna instalacija. Trenutno je dostupna verzija *Scratch 3.0*.

### **8.3. Džepno računalo Micro:bit**

Džepno računalo *Micro:bit* razvijeno je od BBC-a u svrhu upoznavanja softvera i hardvera te razumijevanja načina rada računala. *Micro:bit* se sastoji od LED zaslona, gumba, senzora te ulaznih i izlaznih značajki, dok nova verzija ima dodatan mikrofon i zvučnik te gumbe za ulaz

i napajanje. (*Službena Stranica Micro:Bit*, n.d.) *Micro:bit* se može programirati putem web browsera u *MakeCode editoru*, predstavnik vizualno-blokovskog jezika, *JavaScriptu* ili *Python editoru*, predstavniku proceduralno-tekstualnog jezika. Osim putem web browsera, *Micro:bit* je moguće spojiti na mobilne uređaje preuzimanjem besplatne aplikacije *Micro:bit*. Zbog raznolikih programskih jezika koje *Micro:bit* podržava te mnogih mogućnosti koje on nudi, on je primjeren svim dobnim i kognitivnim razinama. *Micro:bitom* se mogu ostvariti mnogi projekti koji se mogu koristiti u drugim nastavnim predmetima, poput kompasa i mjerenja temperature u nastavi Prirode i društva; kalkulator u nastavi Matematike, igrača kockica za društvene igre, različite igre s *Micro:bitom*...

#### **8.4. Robot *mBot***

*mBot* je STEM programirajući robot za početnike. Namijenjen je učenju programiranja robota na jednostavan i zabavan način. *mBot* se programira koristeći aplikaciju *mBlock*, koji pripada u vizualno-blokovske programske jezike. Spajanjem blokova kreira se program koji se prenosi na *mBot*. *mBot* prati naredbe i izvršava program. Tako je kod vizualiziran, a učenici motivirani. Može se koristiti i bez programiranja korištenjem funkcije daljinskog upravljanja što omogućava upoznavanje robota već od 1. razreda. Primjeren je i starijim uzrastima zbog širokih mogućnosti, korištenja senzora, komunikacije s drugim *mBotovima*, a može se koristiti i Arduino C programiranje. (*mBot*, n.d.)

#### **8.5. Programski jezik *Logo***

*Logo* pripada u vizualno-tekstualne programske jezike kod kojeg se podatci unose tekstualno, dok su izlazni podatci vizualni. *Logo* je namijenjen učenju geometrije putem programiranja. Junak *Loga* je kornjača koja ostavlja trag prema naredbama. Postoje mnoge varijacije *Logo* jezika, prema Slavinu njih 197. (Slavin, n.d.) Neke od njih su besplatne poput *FMS Logo*, a dostupne su i verzije u web browseru.

#### **8.6. Programski jezik *Python***

*Python* je skriptni jezik visoke razine. Pripada u proceduralno-tekstualne programske jezike koji su primjereni učenicima u stadiju formalnih operacija. Primjeren je osnovnoškolskoj naobrazbi u kojima nema većih programa, budući da se programi brzo izvršavaju bez kompajliranja. Najnovija verzija *Pythona* je 3.9.4. Tekstualno sučelje koje dolazi uz *Python* je IDLE, integrirano okruženje za razvoj i učenje, dok je prozor za izvršavanje koda *Python Shell*. (*Python*, n.d.) *Turtle* biblioteka dostupna uz preuzimanje *Pythona* služi za grafički prikaz.

## 9. Prijedlozi za ostvarenje odgojno-obrazovnih ishoda iz domene *Računalno razmišljanje i programiranje po razredima*

Ovo poglavlje sadrži prijedloge za ostvarenje odgojno-obrazovnih ishoda iz domene *Računalno razmišljanje i programiranje po razredima*. Svaka nastavna tema sadrži broj sati potrebnih za provedbu, kratki opis i ishod/ishode koji se ostvaruju.

### 9.1.1. razred

Ishodi na kraju 1. razreda u domeni *Računalno razmišljanje i programiranje* (Ministarstvo znanosti i obrazovanja, 2019):

B. 1. 1. Učenik rješava jednostavan logički zadatak.

B. 1. 2. Učenik prati i prikazuje slijed koraka potrebnih za rješavanje nekoga jednostavnog zadatka.

Tablica 12. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 1. razred

Broj sati (23)	Nastavne teme	Opis	Ishod
2	Logički zadatci	-rješavanje jednostavnih i raznolikih logičkih zadataka (slagalice, memory, labirint, zagonetke, izbacivanje uljeza) u skupinama i individualno	B. 1. 1.
2	Matematički zadatci	-rješavanje jednostavnih matematičkih zadataka (sudoku, nizovi, korištenje novaca, magični četverokut) u skupini i individualno	B. 1. 1.
2	Tko sam ja?	-učenici se upoznaju s četiri plišane igrače te prema zagonetkama i logičkim zadatcima trebaju otkriti njihova imena, koje ime kome pripada, koliko imaju godina, kome pripadaju te koliko godina imaju njihovi vlasnici i gdje žive	B. 1. 1.
2	Tangram i origami	-svaki učenik ima izrezane dijelove za tangram te prema predlošku sastavlja različite oblike; upoznaje se s origami tehnikom i izrađuje origami uz pomoć učitelja; sam osmišljava oblike tangram i origami tehnikom	B. 1. 1.
2	Slijed događaja	-učenici opisuju različite svakodnevne aktivnosti- jutarnju/večernju rutinu, slaganje doručka, redoslijed oblačenja -učenici stvaraju redoslijed događaja koje treba patiti učenik robot kako bi došao s jednog mjesta na drugo -učenici strelicama prikazuju slijed događaja kojima lika dovode s jednog mjesta na drugo	B. 1. 1. B. 1. 2.
2	Kockice i Code.org	-učenici pomoću kockica (kocke i kvadri) grade različite oblike prema predlošku i bez predloška; učenici na web stranici <i>Code.org</i> rješavaju <i>Tečaj 1</i> (1. sat: 1., 2., 3. nivo; 2. sat: 4., 5., 6., 7. nivo)	B. 1. 1.
1	Upoznavanje sa <i>Scratchom</i>	-učitelj učenike upoznaje sa <i>Scratchom</i>	B. 1. 2.

		-pokazuje dodavanje/crtanje likova i pozadina, objašnjava sastavljanje blokova, upoznaje učenike s blokovima <i>Kada je pritisnuta zastavica i reci</i> -prikazuje kratak program	
2	Prvi program u <i>Scratchu</i>	-učenici izrađuju prvi program prema slijedu događaja -učenici se upoznaju s blokom <i>idi 10 koraka</i> , osmišljavaju kratak slijed događaja i izrađuju prvi samostalni rad u <i>Scratchu</i> ; učenici prikazuju svoje radove	B. 1. 2.
2	Riješi problem	-učenici se susreću s problemom- žele igrati društvenu igru, no nemaju igraću kockicu; ispred učenika se postavljaju različiti materijali i pribor te se od njih očekuje da riješe problem; učenici osmišljavaju rješenje problema -učenici realiziraju i predstavljaju rješenje problema; učitelj pokazuje rješenje u <i>Scratchu</i> i na <i>Micro:bitu</i> ; učenici igraju igru koristeći svoju igraću kockicu	B. 1. 1. B. 1. 2.
2	Programiram u <i>Scratchu</i>	-učenici se upoznaju s blokom <i>promijeni pozadinu</i> te dodaju više likova -učenici kreiraju program s 3 lika u kojem koriste naučene blokove	B. 1. 2.
2	Upoznajem <i>mBot</i>	-učitelj učenicima predstavlja <i>mBot</i> - <i>mBot</i> pokreću daljinskim upravljačem -učenici s učiteljem kreiraju jednostavan program za kretanje robota unaprijed	B. 1. 2.
2	Upoznajem <i>Micro:bit</i>	-učitelj učenicima predstavlja <i>Micro:bit</i> -učitelj prikazuje jednostavan program na <i>Micro:bitu</i> (prikaz ikona) -učenici prema uputama kreiraju jednostavan program, uz pomoć učitelja spremaju ga na <i>Micro:bit</i>	B. 1. 2.

Tablica 13. Prijedlog uvođenja nastavnog sadržaja Logički zadatci- Igraj se i uči

<b>Nastavni sadržaj:</b>	Logički zadatci- Igraj se i uči (1. Prilog a))
<b>Ishod:</b>	B. 1. 1. Učenik rješava jednostavan logički zadatak.
<b>Razrada ishoda:</b>	Učenik prepoznaje problem i smišlja mogućnosti rješavanja jednostavnoga logičkog zadatka. Uz učiteljevu pomoć rješava jednostavne logičke zadatke. Potiče sebe i druge na ustrajnost u pronalaženju rješenja.
<b>Struktura i trajanje etapa sata:</b>	
UVODNI DIO (5 min) <b>Koji predmet čemu služi?</b>	
Učitelj prikazuje predmete (glačalo, pametni telefon, jabuku) i ispituje učenike o funkcijama tih predmeta. Potiče ih da razmisle koji predmeti imaju sličnu funkciju, npr. jabuka je hrana- učenici nabrajaju različitu hranu...	
GLAVNI DIO (33 min)	



### **Natjecanje u skupinama- Igre Memory i Slagalice**

Učenici se dijele u tri skupine. Tijekom igara u glavnom dijelu sata učenici skupljaju bodove te se na kraju proglašava pobjednička skupina.

#### Igra Memory

Učitelj na ploču postavlja okrenute kartice s ilustracijama predmeta. Učenici se u skupinama dogovaraju koja polja otvaraju, a za svaki pronađeni par skupina dobiva jedan bod. Igra završava kada učenici pronađu sve parove.

#### Igra Slagalice

Svaka skupina učenika dobiva 12 dijelova slagalice, prikazani predmet je računalo. Cilj je najbrže složiti slagalicu u označeno mjesto na ploči tako da samo jedan učenik sastavlja slagalicu, a ostali ga navode. Igra se prekida kada jedna skupina sastavi slagalicu, a broj bodova jednak je broju točno postavljenih dijelova slagalice. Nakon dodijele bodova, preostale dvije skupine nastavljaju s igrom dok ne dovrše slagalicu.

#### **Izbacivanje uljeza**

Učenici razgovaraju o prikazanom predmetu te njegovoj svrsi. Učenici dobivaju listić s tri zadatka. U prvom zadatku trebaju pobjeći predmete koji pripadaju u određenu skupinu (oboje dijelove računala, oboje voće i povrće), u drugom zadatku trebaju izbaciti uljez (1. ilustracije: tanjur, pribor za jelo, olovka i čaša; 2. ilustracije: četkica za zube, majica, ručnik i sapun), a u trećem spojiti predmete sa sličnom funkcijom.

#### **ZAVRŠNI DIO (7 min) Pogodi predmet**

Učitelj opisuje jedan predmet koji se nalazi u učionici, a učenici pogađaju. Učenik koji pogodi predmet sljedeći opisuje odabrani predmet.

Uputa za sljedeći sat: Logički zadatci- Možeš i sam

Učenici u uvodom dijelu sata igraju igru pogađanja zamišljenog predmeta. U glavnom dijelu sata samostalno rješavaju logičke zadatke: slagalice, labirint, zagonetke, osmišljavaju svoje zagonetke. U završnom dijelu sata učenici čitaju i pogađaju osmišljene zagonetke.

Tablica 14. Prijedlog uvođenja nastavnog sadržaja Moj prvi program u Scratchu

<b>Nastavni sadržaj:</b>	Moj prvi program u <i>Scratchu</i> - Slijedi upute (1. Prilog b))
<b>Ishod:</b>	B. 1. 2. Učenik prati i prikazuje slijed koraka potrebnih za rješavanje nekoga jednostavnog zadatka.
<b>Razrada ishoda:</b>	Učenik prepoznaje jednostavni zadatak (problem) iz svakodnevnoga života. Analizira zadatak (različiti načini rješavanja), opisuje korake

	njegova rješavanja. Prikazuje korake rješavanja zadatka (slikom, riječima).
<b>Struktura i trajanje etapa sata:</b>	
<b>UVODNI DIO (5 min)- Crtanje prema uputama</b>	
Učitelj na prezentaciji prikazuje slijed radnji koje učenici trebaju pratiti te prema slijedu nacrtati u bilježnicu (na sredini papira trokut, lijevo od njega krug, u krugu ime). Provjeravaju jesu li svi točno nacrtali te ističu važnost slijeđenja naredbi.	
<b>GLAVNI DIO (35 min)- Izrada programa u Scratchu</b>	
Učitelj upućuje učenike na web stranicu <i>scratch.mit.edu</i> . S učenicima ponavlja što su naučili na prethodnom satu- dodavanje i crtanje likova i pozadina, prostor u kojem se sklapaju blokovi te skupine blokova. Učitelj pokazuje blok <i>reci</i> te s učenicima razgovara o tom bloku. Učitelj prikazuje slijed radnji koje učenici trebaju izvršiti u <i>Scratchu</i> (nacrtati dvorište s kućom kao pozadinu, nacrtati lik djevojčice ili dječaka, dodati blok kada je zastavica pritisnuta, dodati blok <i>reci</i> te se predstaviti u polju za tekst.)	
<b>ZAVRŠNI DIO (5 min)- Provjera programa</b>	
Učitelj prikazuje program na svom računalu. Učenici provjeravaju točnost, te razgovaraju o problemima na koje su naišli.	
Uputa za sljedeći sat: Prvi program u <i>Scratchu</i> - Osmisli sam	
Učitelj u uvodnom dijelu sata prikazuje program u <i>Scratchu</i> u kojem je korišten blok <i>idi 10 koraka</i> i <i>blok reci</i> . Učenika upoznaje s novim blokom. U glavnom dijelu sata učenici osmišljavaju i na papir simbolima prikazuju redoslijed događaja te osmišljavaju svoj prvi samostalni rad u <i>Scratchu</i> u kojem koriste blokove <i>idi 10 koraka</i> i <i>reci</i> te crtaju/mijenjaju pozadine i likove. U završnom dijelu sata učenici predstavljaju svoj rad i objašnjavaju što su izradili.	

### 9.2.2. razred

Ishodi na kraju 2. razreda u domeni *Računalno razmišljanje i programiranje* (Ministarstvo znanosti i obrazovanja, 2019):

B. 2. 1. Učenik analizira niz uputa koje izvode jednostavan zadatak, ako je potrebno ispravlja pogrešan redosljed.

B. 2. 2. Učenik stvara niz uputa u kojemu upotrebljava ponavljanje.

Tablica 15. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 2. razred

Broj sati (24)	Nastavne teme	Opis	Ishod
1	Logički zadatci	-individualno rješavanje složenijih zadataka od zadataka u 1. razredu (zadatci s natjecanja <i>Dabar</i> ): labirint, nizovi, rebusi, matematički zadatci, zadatci riječima, izbacivanje uljeza	B. 2. 1.
1	Slijed događaja	-rješavanje zadataka slijeda: zadatci sa slijedom, prepoznavanje rezultata slijeda događaja, sastavljanje sendviča prema uputama, crtanje prema uputama	B. 2. 1.
2	Slijedi i pronađi me	-korištenjem kartica koje pokazuju smjer, učitelj sastavlja slijed događaja kojeg jedan učenik slijedi u naznačenim poljima učionice, učenici sastavljaju slijed događaja, učenici ispravljaju slijed događaja koji kreira učitelj -učenici u grupama izrađuju plan pronalaska blaga slijedenjem uputa i rješavanjem logičkih zadataka, skupine mijenjaju planove te pronalaze blago	B. 2. 1.
1	Igraj i uči 1	-učenici na tabletima igraju prva dva nivoa igre <i>SpiriteBox</i>	B. 2. 1.
1	Pričaj priču	-učitelj prikazuje događaje iz učenicima poznatih djela; učenici raspoređuju ponuđene događaje u poznata djela (lektire) te određuju redosljed događaja, osmišljavaju priču i redosljed događaja, događaje izmiješaju te međusobno mijenjaju i rješavaju	B. 2. 1.
2	Što znaš u <i>Scratchu</i> ?	-učenici se motiviraju da izrade program u <i>Scratchu</i> koristeći blokove kojih se sjećaju; ponavljanje naučenog u 1. razredu; učenje novih blokova- <i>skreni, sljedeći kostim, sviraj zvuk, čekaj 1 sekundu, okreni se u smjeru i govori</i> -kreiranje programa prema slijedu događaja	B. 2. 1.
2	Ponovi ovo sve	-pjevanje pjesmice <i>Kad si sretan</i> , razgovor o ponavljanju u pjesmici, pisanje na ploču skraćenicama što se ponavlja (3*pljeskanje...), razgovor o izmjeni godišnjih doma- kako bismo prikazali to ponavljanje kroz 4 godine, ponavljanje vježba na tjelesnoj i zdravstvenoj kulturi -kreiranje ponavljanja karticama te praćenje koraka: hodanje cik-cak, u obliku kvadrata; učenici kreiraju ponavljanje karticama te izvršavaju zadano	B. 2. 2.
2	Ponavljanje u <i>Scratchu</i>	-kreiranje slijeda događaja, promatranje što se događa ako stavimo unutar bloka <i>ponovi (10)</i> ; učenici objašnjavaju što	B. 2. 1. B. 2. 2.

		se događa, učitelj prikazuje program, a učenici ga pokušavaju rekreirati -dramska igra ogledalo (ponavljanje za drugim učenikom); kreiranje programa prema uputama korištenjem blokova ponavljanja	
2	Animacija likova	-prikaz kretanja- što se događa kada se mi krećemo; kreiranje programa u <i>Scratchu</i> za kretanje lika; samostalno kreiranje programa za kretanje lika -animacija prvog slova imena i prezimena uz pozadinsku glazbu	B. 2. 2.
2	Crtaj u <i>Scratchu</i>	-dodavanje proširenja olovka u <i>Scratchu</i> i crtanje prema želji; crtanje prikazanog (isprekidane crte, cik-cak crte, stepenice), zajedničko rješavanje -crtanje kvadrata i pravokutnika na ploči, razgovor o ponavljanju u crtanju; crtanje u <i>Scratchu</i>	B. 2. 2.
1	Riješi problem: Čestitka za Dan očeva	-postavlja se problem- izrada čestitke pomoću <i>Scratcha</i> , izrada čestitke u <i>Scratchu</i> pomoću naučenih blokova, preuzimanje i slanje programa, pokazivanje i objašnjavanje čestitke	B. 2. 1. B. 2. 2.
1	Logički zadatci	-individualno rješavanje složenijih zadataka od zadataka u 1. razredu (zadatci s natjecanja <i>Dabar</i> ):labirint, nizovi, rebusi, matematički zadatci, zadatci riječima, izbacivanje uljeza, slijed događaja	B. 2. 1.
2	Ponavljanje u <i>Micro:bitu</i>	-ponavljanje naučenog u <i>Micro:bitu</i> , individualna izrada jednostavnog programa s prikazom ikona prema izboru -izrada jednostavnog programa korištenjem ponavljanja (prikaz ikona i teksta određeni broj puta pritiskom na određeni gumb)	B. 2. 2.
2	Ponavljanje u <i>mBotu</i>	-ponavljanje naučenog za <i>mBot</i> , pokretanje <i>mBota</i> kroz labirint daljinskim upravljanjem -kreiranje programa za kretanje u obliku kvadrata	B. 2. 2.
1	Čestitka za Majčin dan	-motivacija pitanjem: <i>Što za tebe znači mama?</i> , izrada čestitke u <i>Scratchu</i> pomoću naučenih blokova, preuzimanje i slanje programa, pokazivanje i objašnjavanje čestitke	B. 2. 1. B. 2. 2.
1	Igraj i uči 2	-učenici na tabletima igraju treći nivo igre <i>SpiriteBox</i>	B. 2. 1. B. 2. 2.

Tablica 16. Prijedlog uvođenja nastavnog sadržaja Ponovi ovo sve

<b>Nastavni sadržaj:</b>	Ponovi ovo sve (2. Prilog a))
<b>Ishod:</b>	B. 2. 2. Učenik stvara niz uputa u kojemu upotrebljava ponavljanje.
<b>Razrada ishoda:</b>	Učenik prati niz uputa predloženih slikom ili riječima koje izvode jednostavan zadatak. Analizira niz uputa i otkriva pogrešan redoslijed. Ispravlja pogrešan redoslijed u uputama za rješavanje jednostavnoga zadatka.
<b>Struktura i trajanje etapa sata:</b>	

<b>UVODNI DIO (5 min)- Kad si sretan</b>
Slušanje pjesmice <i>Kad si sretan</i> . Razgovor o ponavljanju u pjesmici.
<b>GLAVNI DIO (35 min)- Što se ponavlja?</b>
Prikazivanje karticama slijed događaja u pjesmici. Uočavanje elemenata koji se ponavljaju i prikaz karticama. Učitelj miješa kartice i broj ponavljanja te učenici pjevaju i ponavljaju prema prikazanom. Razgovor o ponavljanjima u prirodi i svakodnevnim aktivnostima. Razgovor o ponavljanju dana u tjednu ( <i>Kako bismo prikazali 3 tjedna?</i> ) te izmjeni godišnjih doba- učenici prikazuju karticama ( <i>Kako bismo to prikazali za 4 godine?</i> ).
<b>ZAVRŠNI DIO (5 min)- Prikaži tablicama</b>
Učitelj karticama prikazuje vježbe te broj ponavljanja. Učenici izvršavaju prikazano.

Tablica 17. Prijedlog uvođenja nastavnog sadržaja *Crtaj u Scratchu*

<b>Nastavni sadržaj:</b>	Crtaj u <i>Scratchu</i> (2. Prilog b))
<b>Ishod:</b>	B. 2. 2. Učenik stvara niz uputa u kojemu upotrebljava ponavljanje.
<b>Razrada ishoda:</b>	Učenik prati niz uputa predloženih slikom ili riječima koje izvode jednostavan zadatak. Analizira niz uputa i otkriva pogrešan redoslijed. Ispravlja pogrešan redoslijed u uputama za rješavanje jednostavnoga zadatka.
<b>Struktura i trajanje etapa sata:</b>	
<b>UVODNI DIO (5 min)- Crtajmo</b>	
Učitelj prikazuje program s nacrtanim isprekidanim crtama, u obliku stepenica i cik-cak crte. Učenici iznose zapažanja te pretpostavke kako nacrtati zadano.	
<b>GLAVNI DIO (32 min)- Pokušaj</b>	
Učitelj prikazuje proširenje olovku te blokove koji će biti potrebni. Učitelj učenike potiče na razmišljanje i davanje ideje za izradu. Učenici samostalno pokušavaju nacrtati prikazano. Učenici prikazuju program. Učitelj s učenicima izrađuje i objašnjava program.	
<b>ZAVRŠNI DIO (8 min)- Izradi sam</b>	
Učenici crtaju oblike prema želji. Prikazuju i objašnjavaju nacrtano.	

### 9.3. 3. razred

Ishodi na kraju 3. razreda u domeni *Računalno razmišljanje i programiranje* (Ministarstvo znanosti i obrazovanja, 2019):

B. 3. 1. Učenik stvara program korištenjem vizualnoga okruženja u kojem se koristi slijedom koraka, ponavljanjem i odlukom te uz pomoć učitelja vrednuje svoje rješenje.

B. 3. 2. Učenik slaže podatke na koristan način.

Tablica 18. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 3. razred

Broj sati (24)	Nastavne teme	Opis	Ishod
1	Logički zadatci	-individualno rješavanje slijeda događaja, logičkih zadataka: izbroji trokute, kvadrate, računanje objektima; izrada križaljka, osmosmjerki i rebusa u paru te zamjena; raspored zadataka prema težini u tablice	B. 3. 1. B. 3. 2.
1	Prikaži karticama	-na podu se postavljaju krugovi, kvadrati i trokuti koji predstavljaju različiti oblik kretanja; učenici karticama prikazuju kretanje (kartice za ravno, lijevo, desno i ponavljanje) te prolaze kroz poligon	B. 3. 2.
1	Razvrstavanje podataka	-igra slobodnog kretanja i razvrstavanja prema zadanom kriteriju (boji kose, boji odjeće...); razvrstavanje predmeta prema kriterijima koje učenik odabere, objašnjavanje logike razvrstavanja	B. 3. 2.
1	Prikaži u tablicama	-razvrstavanje lektira po abecedi, životinja po visini, učenika po starosti/visini, članova obitelji po visini/godinama u tablice; razvrstavanje blokova iz <i>Scratcha</i> u tablice prema različitim kriterijima	B. 3. 2.
2	Moje slobodno vrijeme	-razgovor o slobodnom vremenu, sastavljanje tablice- <i>Što radim u slobodno vrijeme?</i> od onog što najviše rade do onog što najrjeđe rade, upoznavanje dijagrama i primjera -sastavljanje dijagrama prema tablici, uspoređivanje učeničkih dijagrama, sastavljanje dijagrama s nastavnim predmetima, učitavanje dijagrama (Tko najviše voli Hrvatski jezik/Matematiku...)	B. 3. 2.
2	Što znam u <i>Scratchu</i> ?	-ponavljanje naučenih blokova u <i>Scratchu</i> ; izrada zajedničkog programa crtanja kvadrata i pravokutnika -prikaz programa s crtanjem nizova, individualna izrada prikazanog programa	B. 3. 1.
1	Animacije i pokretanje likova	-ponavljanje animacije likova; ponavljanje kretanja likova s izmjenom kostima; dodavanje najjednostavnije odluke (ako si na rubu okreni se); izrada programa kretanja lika s okretanjem na rubu	B. 3. 1.
1	Odluči se!	-razgovor o odlasku u školu; kreiranje slijeda događaja, razgovor o utjecaju semafora na kretanje, utjecaj vremenske prognoze na naše ponašanje; igra pljesni-skoči (ako učenik pljesne- skoči)	B. 3. 1.

2	Odluke u <i>Scratchu</i>	-upoznavanje bloka <i>ako-onda</i> u <i>Scratchu</i> , prikaz programa s razgovorom (ako liku odgovorimo da pada kiše, on kaže da će uzeti kišobran), upoznavanje <i>Ako-onda-inače</i> bloka u <i>Scratchu</i> (ako liku odgovorimo da pada kiše- lik kaže da će uzeti kišobran, ako ne pada- lik kaže da neće uzeti kišobran) -izrada programa s jednostavnom odlukom (ako dodiruje-zaustavi se/promijeni položaj), učenici objašnjavaju program, samostalna izrada programa	B. 3. 1.
1	Logički zadatci	-slijed događaja, prikaz podataka u tablicama (izračunaj pa poredaj po redu, prebroji pa poredaj od najmanjeg do najvećeg u tablici, razvrstaj prema zadanom/slobodnom kriteriju)	B. 3. 1. B. 3. 2.
1	Tablice i dijagrami	-razvrstavanje podataka prema različitim kriterijima (izmiješani pojmovi koji se svrstavaju u tablici prema definiranim i slobodnim kriterijima) i sastavljanje stupčastog dijagrama (prema tablicama koje prikazuju broj)	B. 3. 2.
2	Igre u <i>Scratchu</i> 1	-igra- učenici hodaju kada učitelj pljesne oni skoče, izrada igre skakanja preko prepreke -igranje igre labirinta s navođenjem učenika; izrada labirinta u <i>Scratchu</i> ; pokretanje likova strelicama (ako dodirne prepreku- vraća se na početak; ako dodirne predmet- predmet nestane)	B. 3. 1.
2	<i>Micro:bit</i>	-upoznavanje senzora za temperaturu i svjetlost; kreiranje programa za mjerenje temperature s uvjetnim prikazom simbola; kreiranje programa za praćenje količine svjetlosti s uvjetnim prikazom simbola	B. 3. 1.
2	<i>mBot</i>	-razgovor o mogućnostima <i>mBota</i> ; ponavljanje naučenog; kreiranje programa za praćenje linije te uvjetni prikaz na zaslonu (prikaz jednog simbola kada se ne kreće, drugog kada se kreće)	B. 3. 1.
1	Organiziram mape	-učenici igraju igru u kojoj raspoređuju mape i dokumente u tablice, učenici svoje radove iz informatike, koje su skupljali u jednoj mapi, pregledavaju te prema potrebi mijenjaju imena, smisljeno raspoređuju u mape i dodjeljuju imena mapama; objašnjavaju logiku raspodijele dokumenata u mape	B. 3. 2.
1	Igra u <i>Scratchu</i> 2	-prikaz izrađene igre Labirint, samostalna izrada igre: Labirint u kojem se lik pomiče 20 koraka unatrag kada dodirne prepreku, a kada dodirne predmet, predmet mijenja poziciju	B. 3. 1.
2	Riješi problem: organiziraj Dan škole	-razgovor o Danu škole i programu za taj dan, promatranje plana za ovogodišnji Dan škole (neorganizirano)- kako bismo mogli bolje organizirati/ što bismo mogli dodati/ kako bismo program mogli poboljšati/što bismo mogli dodati? -izrada programa za Dan škole u skupinama- raspored u tabličnom prikazu (priredba- izvođači i teme- raspored; izložbe- raspored prema prostorijama škole; sportsko natjecanje- raspored grupa u igre; vrijeme trajanja...)	B. 3. 2.

Tablica 19. Prijedlog uvođenja nastavnog sadržaja *Moje slobodno vrijeme*

<b>Nastavni sadržaj:</b>	Moje slobodno vrijeme (3. Prilog a)
<b>Ishod:</b>	B. 3. 2. Učenik slaže podatke na koristan način.
<b>Razrada ishoda:</b>	Određuje zajedničke karakteristike grupe podataka. Razvrstava podatke u grupe. Reda podatke prema kriteriju koji omogućuje njihovo učinkovito korištenje.
<b>Struktura i trajanje etapa sata:</b>	
UVODNI DIO (5 min)- <b>Što je slobodno vrijeme?</b>	
Učitelj učiteljima dijeli listiće na kojima se nalazi aktivnosti tijekom dana. Učenici aktivnosti dijele na obaveze i aktivnosti u slobodno vrijeme. Razgovor o ostalim aktivnostima u slobodno vrijeme.	
GLAVNI DIO (35 min)- <b>Tablice i dijagrami</b>	
Učenici sastavljaju tablicu od najčešće do najrjeđe aktivnosti koje rade u slobodno vrijeme. Učitelj prikazuje dijagrame koji pokazuju kako Ana i Ivan provode vrijeme tablicom i dijagramom. Uspoređivanje i komentiranje dijagrama. Zajedničko kreiranje tablice u Infogramu, prikaz dijagrama s obzirom na broj sati pojedinih predmetima u nastavi.	
ZAVRŠNI DIO (5 min)- <b>Protumači</b>	
Učenici tumače izrađeni dijagram- veličina kružnog isječka s obzirom na broj sati...	
Uputa za sljedeći sat: <i>Moje slobodno vrijeme- dijagrami</i> Učenici prema tablici koju su sastavili uz pomoć učitelja sastavljaju dijagram- <i>Što radim u slobodno vrijeme</i> . Uspoređivanje dijagrama. Sastavljanje dijagrama s nastavnim predmetima prema tome koji predmet najviše vole. Uspoređivanje te utvrđivanje tko najviše voli pojedini predmet...	

Tablica 20. Prijedlog uvođenja nastavnog sadržaja *Odluke u Scratchu 2*

<b>Nastavni sadržaj:</b>	Odluke u <i>Scratchu 2</i> (3. Prilog b)
<b>Ishod:</b>	B. 3. 1. Učenik stvara program korištenjem vizualnoga okruženja u kojem se koristi slijedom koraka, ponavljanjem i odlukom te uz pomoć učitelja vrednuje svoje rješenje.
<b>Razrada ishoda:</b>	Prepoznavanje problema u jednostavnom zadatku s odlukom, osmišljavanje niza koraka koji vode k rješenju. Uz pomoć učitelja ili samostalno otkriva i prikazuje korake rješavanja jednostavnoga zadatka koji sadrži odluku. Stvaranje programa korištenjem vizualnim



	okruženjem u kojemu se koristi slijedom koraka, ponavljanjem i odlukom.
<b>Struktura i trajanje etapa sata:</b>	
UVODNI DIO (5 min)- <b>Prikaz programa u Scratchu</b>	
Učenici hodaju prostorom, kada učitelj pljesne oni skoče. Osmišljavanje niza koraka i sastavljanje kartica koje to prikazuju. Učitelj prikazuje program u kojem se lik kreće, a pritiskom na razmaknicu skoči. Učenike potiče na razmišljanje kako napraviti program.	
GLAVNI DIO (35 min)- <b>Blokovi odluke</b>	
Učitelj prikazuje i objašnjava kako se mijenja vrijednost y. Učenici zaključuju kako se mijenja y kod skakanja. Učenici individualno izrađuju program. Učenici prikazuju svoj rad. Učitelj prikazuje kako koristiti blok <i>ako-onda</i> u programu.	
ZAVRŠNI DIO (5 min)- <b>Izmjeni</b>	
Učenici izmjenjuju svoj program te provjeravaju i testiraju.	

#### 9.4. 4. razred

Ishodi na kraju 4. razreda u domeni *Računalno razmišljanje i programiranje* (Ministarstvo znanosti i obrazovanja, 2019):

B. 4. 1. Učenik stvara program korištenjem vizualnog okruženja u kojem koristi slijed, ponavljanje, odluku i ulazne vrijednosti.

B. 4. 2. Učenik rješava složenije logičke zadatke s uporabom računala ili bez uporabe računala.

Tablica 21. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 4. razred

Broj sati (26)	Nastavne teme	Opis	Ishod
2	Logički zadatci	-individualno rješavanje složenih zadataka: matematički labirinti, sudoku, slagalice na ploči, osmišljavanje palindroma -palindromi, abecedna simetrija, šifriranje (osmišljavanje i rješavanje u paru)	B. 4. 2.
1	<i>Code.org</i>	-individualno rješavanje <i>Code.org</i> , 2. tečaj- ponavljanje slijeda, petlja i odluka	B. 4. 1. B. 4. 2.
1	Program	-učenicima se prikazuje program u koji se unosi godina rođenja, a kao izlaznu vrijednost prikazuje se broj godina; učenicima se objašnjava što je program te segmenti izrade programa	B. 4. 1.
2	Ponavljanje i odluke	-ponavljanje slijeda, ponavljanja i odluka u <i>Scratchu</i> -individualna izrada igre <i>Labirint</i>	B. 4. 1.

2	Varijable	-raspoređivanje namirnica u košare i spremanje napitaka u čašu; izrada programa s korištenjem odgovora, razgovor o spremanju vrijednosti u varijable -razgovor o razlici između vrijednosti i varijabli, izrada <i>Scratch</i> programa u kojem se spremaju vrijednosti u varijable (spremanje imena, prezimena i godina te korištenje u govoru)	B. 4. 1.
2	Dijalog s likovima	-kreiranje razgovora s likom u <i>Scratchu</i> - kreiranje razgovora s odlukama (odlazak u svemir, vraćanje natrag- prema odluci ), kreiranje programa sa spremanjem odgovora u varijable (informacije o omiljenoj hrani, predmetu) i korištenje tih vrijednosti -individualno kreiranje razgovora prema uputama	B. 4. 1.
2	Program za uspoređivanje i zbrajanje	-razgovor o programu koji bi nam olakšao računanje i uspoređivanje, kreiranje programa za usporedbu -kreiranje programa za zbrajanje (i oduzimanje), ponavljanje o segmentima izrade programa	B. 4. 1.
2	Igre u <i>Scratchu</i> 1	-razgovor o promijeni vrijednosti u varijabli, prikaz programa u kojem jedan lik lovi drugog i pritom sakuplja bodove -individualna izrada igre, prikaz igre i ispravljanje pogrešaka	B. 4. 1.
2	Logički zadatci	-individualno rješavanje složenih zadataka riječima koji zahtijevaju strategije rješavanje prema Polyai (učenici se vode po koracima)	B. 4. 2.
2	<i>Micro:bit</i>	-upoznavanje varijabli u <i>Micro:bitu</i> , izrada igre <i>Kamen, škare, papir</i> -izrada programa za brojanje koraka pomoću <i>Micro:bita</i> , testiranje programa i razgovor o radu programa	B. 4. 1.
2	<i>Micro:bit</i> 2	-priča- Ivan je krenuo u pustolovinu, no razbio mu se kompas, upoznavanje mogućnosti izrade kompasa u <i>Micro:bitu</i> -objašnjavanje načina rada kompasa, izrada kompasa, zapisivanje prijevoda naučenih blokova, korištenje kompasa i razgovor o ograničenju	B. 4. 1.
2	Logički zadatci	-individualno rješavanje složenih zadataka bez vođenja po koracima	B. 4. 2.
2	Igre u <i>Scratchu</i> 2	-igranje igre <i>Pogodi broj</i> , poticanje učenika na razmišljanje kako bi mogli sami igrati igru, prikaz igre u <i>Scratchu</i> , individualno kreiranje programa i igranje igre -razgovor o igri <i>Pong</i> , razgovor o izradi, grupno kreiranje igre <i>Pong</i>	B. 4. 1.
2	Projektna nastava: Programi za množenje	-učenici u tri skupine dobivaju zadatke za izradu programa za množenje u <i>Scratchu</i> ili <i>Micro:bitu</i> , učenici izrađuju i predstavljaju svoje radove	B. 4. 1. B. 4. 2.

Tablica 22. Prijedlog uvođenja nastavnog sadržaja *Micro:bit*

<b>Nastavni sadržaj:</b>	<i>Micro:bit</i> (4. Prilog a))
<b>Ishod:</b>	B. 4. 1. Učenik stvara program korištenjem vizualnog okruženja u kojem koristi slijed, ponavljanje, odluku i ulazne vrijednosti.
<b>Razrada ishoda:</b>	Opisuje situacije u svojem programu u kojemu bi se trebao koristiti odlukom i ulaznim vrijednostima. Stvara program u kojem se koristi ulaznim vrijednostima, slijedom, ponavljanjem i odlukom. Analizom odnosa ulaznih i izlaznih vrijednosti vrednuje uspješnost rješenja.
<b>Struktura i trajanje etapa sata:</b>	
UVODNI DIO (5 min)- <b>Igram igru</b>	
Igranje igre <i>Kamen, škare, papir</i> u paru. Razgovor o varijablama i njihovoj svrsi. Poticanje učenika na razmišljanje- kako bi sami mogli igrati igru <i>Kamen, škare, papir</i> .	
GLAVNI DIO (38 min)- <b>Varijable u <i>Micro:bitu</i></b>	
Upoznavanje kreiranja varijabla i blokova- zapisivanje novih blokova u bilježnicu. Razgovor i kreiranje zajedničkog programa za igru <i>Kamen, škare, papir</i> (Razgovor- Kamo ćemo spremiti broj koji će <i>Micro:bit</i> odabrati? Koji blok ćemo koristiti ako želimo da <i>Micro:bit</i> prikaže papir ako je odabran broj 1? Kojim blokom ćemo prikazati papir/škare/kamen?). Individualna izrada programa.	
ZAVRŠNI DIO (7 min)- <b>Provjera programa</b>	
Provjera programa i ispravljanje eventualnih pogrešaka, opisivanje rada programa. Igranje igre.	

Tablica 23. Prijedlog uvođenja nastavnog sadržaja *Igre u Scratchu 2*

<b>Nastavni sadržaj:</b>	<i>Igre u Scratchu 2</i> (4. Prilog b))
<b>Ishod:</b>	B. 4. 1. Učenik stvara program korištenjem vizualnog okruženja u kojem koristi slijed, ponavljanje, odluku i ulazne vrijednosti.
<b>Razrada ishoda:</b>	Opisuje situacije u svojem programu u kojemu bi se trebao koristiti odlukom i ulaznim vrijednostima. Stvara program u kojem se koristi ulaznim vrijednostima, slijedom, ponavljanjem i odlukom. Analizom odnosa ulaznih i izlaznih vrijednosti vrednuje uspješnost rješenja.
<b>Struktura i trajanje etapa sata:</b>	
UVODNI DIO (7 min)- <b>Igram igru</b>	

Igranje igre <i>Pogodi broj</i> . Jedan učenik zamišlja broj, ostali učenici pogađaju zamišljen broj. Poticanje na razmišljanje pitanjem: <i>Kako bismo mogli sami igrati ovu igru?</i>
<b>GLAVNI DIO (30 min)- Izrada programa u Scratchu</b>
Pokazivanje igre <i>Pogodi broj</i> u <i>Scratchu</i> . Zajedničko igranje igre. Razgovor o tome kako je program izrađen i na koji princip radi. Individualno kreiranje programa.
<b>ZAVRŠNI DIO (8 min)- Provjera programa</b>
Provjeravanje programa i ispravljanje eventualnih pogrešaka, objašnjavanje na koji način program radi. Igranje igre.

### 9.5. 5. razred

Ishodi na kraju 5. razreda u domeni *Računalno razmišljanje i programiranje* (Ministarstvo znanosti i obrazovanja, 2019):

B. 5. 1. Učenik se koristi programskim alatom za stvaranje programa u kojemu se koristi ulaznim i izlaznim vrijednostima te ponavljanjem.

B. 5. 2. Učenik stvara algoritam za rješavanje jednostavnoga zadatka, provjerava ispravnost algoritma, otkriva i popravlja pogreške.

Tablica 24. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 5. razred

Broj sati (26)	Nastavne teme	Opis	Ishod
2	Slijedi upute	-učitelj daje upute, a učenici izvršavaju zadano; uspoređivanje i ispravljanje pogrešaka; individualno rješavanje složenih logičkih zadataka -individualno rješavanje složenih logičkih zadataka; definiranje računalnog razmišljanja i upotrebe	B. 5. 2.
2	Riješi problem	-rješavanje složenih logičkih zadataka s natjecanja Dabar -zadavanje problema: predškolska djeca- crvene majice, učenici osnovne škole- zelene majice, učenici srednjih škola- plave majice- učenici iznose rješenje kako bi oni riješili problem na skupini izmiješane djece; razgovor o tome kako bi računalno riješili problem	B. 5. 2.
2	Algoritmi	-tekstualni prikaz rješenja problema s prošlog sata; upoznavanje s pojmom algoritmi; prikaz jednog primjera za algoritam slijeda, grananja i ponavljanja, prikaz više algoritama, svrstavanje u skupine te objašnjavanje (prelazak preko ceste bez semafora, prelazak preko ceste sa semaforom, kuhanje čaja, pečenje pizze, jedenja juhe, crtanja pravokutnika), individualno kreiranje algoritma punjenja boca vodom u tvornici	B. 5. 2.

2	Kreiraj algoritam	-učenicima se daju problemi za koje je potrebno napisati algoritam- spremanje kreveta, dodjeljivanje ocjene (ovisnost o bodovnoj ljestvici), izrada muffina; učenici objašnjavaju algoritam, uspoređuju ga, ispravljaju i svrstavaju u skupinu- algoritmi slijeda/grananja/ ponavljanja	B. 5. 2.
2	Ispravi pogreške	-učenicima se daju neispravno napisani algoritmi- oni ispravljaju pogreške i objašnjavaju zašto -učenicima se prikazuje jedan neispravan <i>Scratch</i> program te traže pogrešku; upoznaju se s pojmom <i>bug</i> , vrstama pogrešaka te načinom testiranja programa	B. 5. 2.
2	<i>Scratch</i>	-učenici s predznanjem prezentiraju osnovne značajke <i>Scratcha</i> i pokazuju primjere programa izrađenih u <i>Scratchu</i> te ih objašnjavaju; upoznavanje s pojmovima programski jezik, program, programer -učenici s predznanjem (nakon 4 godine učenja) rade u paru s učenicima bez predznanja te zajedno kreiraju slijed događaja prema zadanim koracima	B. 5. 1.
2	Petlja i grananje, varijable	-crtanje kvadrata, trokuta, pravokutnika, paralelograma- upoznavanje bloka ponavljanja -učenici kreiraju program s odlukom (prelazak preko ceste- zasebno sa semaforom, bez semafora; jedan program za oba); razgovor kamo se spremaju vrijednosti i na koji način; upoznavanje varijabla	B. 5. 1.
2	Riješi problem: <i>Scratch</i>	-kreiranje algoritma za unos broja artikala i cijene te izračuna ukupne cijene; izrada programa u <i>Scratchu</i> -kreiranje algoritma za unos broja i duljine stranica- crtanje mnogokuta prema unesenim podacima	B. 5. 1.
2	<i>Micro:bit</i>	-učenici s predznanjem prezentiraju osnovne značajke <i>Micro:bita</i> , učenici prema želji izrađuju odabrani program (program za brojanje kod pritiska gumba, program za brojanje koraka kod hodanja), učenici testiraju program i ispravljaju pogreške; učenici izrađuju igraču kockicu	B. 5. 1.
2	<i>Logo</i>	-upoznavanje s <i>Logo</i> programskim jezikom, upoznavanje s instalacijom <i>FMSLogo</i> ; -prikaz osnovnih naredbi i njihovog korištenja; učenici samostalno crtaju ravnu crtu; prikaz nekoliko programa; spremanje i otvaranje programa	B. 5. 1.
2	Crtaj	-prikaz programa crtanja pravca, isprekidanih crta, stepenica; učenici samostalno crtaju -zajedničko crtanje paralelnih crta i kvadrata bez petlje	B. 5. 1.
2	Crtaj geometrijske likove	-prikaz kvadrata, pravokutnika i jednakostraničnog trokuta bez petlje; samostalno crtanje i uočavanje ponavljanja, ponavljanje naredbe ponavljanja -stvaranje programa s petljom te crtanje geometrijskih likova pomoću petlje; testiranje programa te objašnjavanje provjere ispravnosti i svrhe	B. 5. 1.
2	Projektna nastava: Crtaj mnogokute	-zadavanje problema: izrada programa za crtanje mnogokuta; kreiranje algoritma za crtanje, izrada programa za crtanje mnogokuta; testiranje; provjera i usporedba	B. 5. 1.

	programa, učenici objašnjavaju gdje se koriste ulazne vrijednosti, što se obrađuje te što su izlazne vrijednosti	
--	--	--

Tablica 25. Prijedlog uvođenja nastavnog sadržaja Kreiraj algoritam

<b>Nastavni sadržaj:</b>	Kreiraj algoritam (5. Prilog a))
<b>Ishod:</b>	B. 5. 2. Učenik stvara algoritam za rješavanje jednostavnoga zadatka, provjerava ispravnost algoritma, otkriva i popravlja pogreške.
<b>Razrada ishoda:</b>	Analizira jednostavan problem, predviđa korake za rješavanje toga problema i prikazuje ih (grafički, usmeno ili tekstom) predviđajući redoslijed njihova izvršavanja. Učenik kritički provjerava ispravnost svojega algoritma tako da uspoređuje očekivano rješenje problema s dobivenim rješenjem. Ovisno o ispravnosti rješenja preuređuje i ponovno testira svoje rješenje.
<b>Struktura i trajanje etapa blok sata:</b>	
UVODNI DIO (10 min)- <b>Ponavljanje naučenog</b>	
Učenici ponavljaju naučeno o algoritmima- što su algoritmi, čemu služe, njihovu podjelu.	
GLAVNI DIO (65 min)- <b>Izrada algoritama</b>	
Učenicima se predstavljaju problemi: spremanje kreveta, izrada muffina, dodjeljivanje ocjene (ovisnost o bodovnoj ljestvici). Zajednička analiza problema te individualna izrada algoritama. Prikaz i uspoređivanje algoritma.	
ZAVRŠNI DIO (15 min)- <b>Ispravljanje pogrešaka i svrstavanje u vrste algoritama</b>	
Ovisno o usporedbi, ispravljanje pogrešaka prema potrebi. Svrstavanje u vrste algoritama te objašnjavanje razloga.	

Tablica 26. Prijedlog uvođenja nastavnog sadržaja Crtaj geometrijske likove

<b>Nastavni sadržaj:</b>	Crtaj geometrijske likove (5. Prilog b))
<b>Ishod:</b>	B. 5. 1. Učenik se koristi programskim alatom za stvaranje programa u kojemu se koristi ulaznim i izlaznim vrijednostima te ponavljanjem.
<b>Razrada ishoda:</b>	Slaže jednostavan niz uputa koristeći se blokovima/naredbama. Povezujući nekoliko blokova ili naredbi u cjelinu, istražuje njihovo djelovanje. Analizira zadani problem te odabire ili predlaže niz blokova/naredbi kao moguće rješenje problema. Učenik samostalno

	razvija rješenje nekoga problema koristeći se ulaznim i izlaznim podacima, pridruživanjem vrijednosti te ponavljanjem.
<b>Struktura i trajanje etapa blok sata:</b>	
UVODNI DIO (10 min)- <b>Ponavljanjem naučenog</b>	
Učenici ponavljaju naučene naredbe u <i>Logo</i> programskom jeziku i crtaju kvadrat bez petlje.	
GLAVNI DIO (65 min)- <b>Geometrijski likovi</b>	
Učitelj prikazuje programska rješenja za crtanje pravokutnika i jednakostraničnog trokuta. Učenici analiziraju rješenje i individualno izrađuju programe. Provjeravaju ispravnost rješenja te uočavaju ponavljanja u programu. Učenici se upoznaju s naredbom ponavljanja i kreiranjem programa za crtanje kvadrata s ponavljanjem. Samostalno izrađuju program za crtanje pravokutnika i jednakostraničnog trokuta s ponavljanjem. Učitelj prikazuje program u kojem se unosi ulazna vrijednost duljine stranice, učenici uređuju program koji odgovara crtanju trokuta.	
ZAVRŠNI DIO (15 min)- <b>Provjera programa</b>	
Učenici objašnjavaju način provjere ispravnosti programa, provjeravaju ispravnost i objašnjavaju način rada programa.	

### 9.6. 6. razred

Ishodi na kraju 6. razreda u domeni *Računalno razmišljanje i programiranje* (Ministarstvo znanosti i obrazovanja, 2019):

B. 6. 1. Učenik stvara, prati i preuređuje programe koji sadrže strukture grananja i uvjetnoga ponavljanja te predviđa ponašanje jednostavnih algoritama koji mogu biti prikazani dijagramom, riječima govornoga jezika ili programskim jezikom.

B. 6. 2. Učenik razmatra i rješava složeniji problem rastavljajući ga na niz potproblema.

Tablica 27. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 6. razred

Broj sati (26)	Nastavne teme	Opis	Ishod
2	Logički slijed	-učenici rješavaju zadatak te objašnjavaju kako su došli do rješenja; učitelj predstavlja strategiju rješavanja problema prema Polyi; učenici rješavaju složene zadatke prema planu	B. 6.2.
2	Algoritmi	-učitelj prikazuje jedan algoritam slijeda, grananja i ponavljanja- učenici ih objašnjavaju; učenici kreiraju algoritam za zbrajanje brojeva; prikaz dijagrama toka i objašnjavanje simbola; učenici kreiraju dijagram toka za množenje brojeva te objašnjavaju	B. 6. 2.

2	Riješi problem	-učenici rješavaju zadatak (cvijeće s 4 i 6 latica, ukupno 30 latica), postoji li samo jedno rješenje; individualno rješavanje odabranih zadataka iz natjecanja Dabar (prema strategiji Polya) -prikaz algoritma dijagramima toka- učenici predviđaju ponašanje algoritma	B. 6. 1. B. 6. 2.
2	Python	-upoznavanje sa sučeljem, instalacijom i osnovnim operatorima <i>Pythona</i> ; prikaz nekoliko programa; zajedničko kreiranje programa za ispis riječi, programa za zbrajanje (bez ulaznih vrijednosti) prema kreiranom dijagramu toka; samostalna prenamjena programa u program za množenje	B. 6. 1.
2	Ulaz- obrada-izlaz	-diskusija o opsegu i površini; kreiranje dijagrama toka za računanje površine i opsega pravokutnika bez ulaznih vrijednosti; prikaz programa za izračun površine s unosom ulaznih vrijednosti; razgovor o segmentima i varijablama u kreiranom programu; individualno kreiranje dijagrama toka i programa za izračun opsega; individualno kreiranje dijagram toka i programsko rješenje za kupovinu kino ulaznica; prikaz i diskusija, testiranje programa i ispravljanje prema potrebi	B. 6. 1.
2	Algoritam grananja 1	-prikaz programskog rješenja s jednostavnim grananjem- učenici predviđaju ponašanje, upoznaju s grananjem <i>if</i> ; zajedničko kreiranje programa za ispis broja ako je paran i upoznavanje s pravilima kod pisanja koda; individualno preuređenje programa u kojem program ispisuje broj ako je pozivan -prikaz programskog rješenja s <i>if...else</i> grananjem- učenici predviđaju ponašanje; zajedničko kreiranje programa za ispis djeljiv sa 7/nije djeljiv sa 7 s obzirom na unesenu vrijednost	B. 6. 1.
2	Algoritam grananja 2	-individualno kreiranje programa za ispis najmanjeg od 3 unesena broja -prikaz programskog rješenja s <i>if...elif...else</i> grananjem- učenici predviđaju ponašanje; zajedničko kreiranje programa za ispis pozitivan/negativan/0 s obzirom na ulaznu vrijednost; individualna izrada programa za uspoređivanje brojeva (<,>=)	B. 6. 1.
2	Pokušaj sam/sama 1!	-učenicima se predstavljaju problemi (ispis ako je broj višekratnik broja 9; paran/neparan s obzirom na ulaznu vrijednost; uspoređivanje dva broja)- učenici kreiraju dijagram toka na zadani problem te kreiraju programsko rješenje, testiraju te prikazuju svoje rješenje	B. 6. 1.
2	Petlja s odlukom	-prikaz programskog rješenja s petljom (ispis parnih brojeva do određenog broja), predviđanje ponašanja; upoznavanje s petljom i pravilima; nabrojanje situacija s uvjetnim ponavljanjem; kreiranje dijagrama toka za ispis višekratnika broja 3 do određenog broja- objašnjavanje programa -prikaz dijagrama toka s uvjetnim ponavljanjem ( <i>while</i> )- upis brojeva do 10, program se prekida upisom 0	B. 6. 1.



2	Pokušaj sam/sama 2!	-učenicima se predstavlja problem (upis i određivanje dvoznamenkastih brojeva, s upisom 0 program se prekida)-učenici kreiraju dijagram toka na zadani problem te kreiraju programsko rješenje, testiraju te prikazuju svoje rješenje -učenici osmišljavaju situacije s uvjetnim ponavljanjem te kreiraju dijagram toka	B. 6. 1.
2	Potproblem u problemu	-učenici se prisjećaju crtanja u <i>Logo</i> programskom jeziku (crtaju kvadrat, jednakostranični trokut); učitelj prikazuje programe s nizom kvadrata/trokuta, dva pravokutna trokuta koji čine kvadrat; učenici planiraju crtanje te provode, provjeravaju ispravnost i objašnjavaju	B. 6. 2.
2	Crtaj uz uvjete	-crtanje paralelograma i trapeza; crtanje niza s određenim brojem ponavljanja; prikaz uvjetnog crtanja i objašnjavanje naredba (ako se unose broj 1- crtanje kvadrata; ako se unese broj 2- crtanje pravokutnika); individualna izrada programa za uvjetno crtanje niza paralelograma/trapeza	B. 6. 1. B. 6. 2.
2	Projektna nastava: Trokut	-učenici dobivaju plan za kreiranje lekcije iz matematike u 6. razredu na temu <i>Vrste trokuta</i> (crtanje jednakostraničnog, jednakokračnog i raznostraničnog trokuta te računanje opsega i površine); učenici se dijele u skupine, kreiraju dijagrame toka i kreiraju programska rješenja- crtanje u <i>Logo</i> , a izračun u <i>Pythonu</i>	B. 6. 1. B. 6. 2.

Tablica 28. Prijedlog uvođenja nastavnog sadržaja *Ulaz-obrađa-izlaz*

<b>Nastavni sadržaj:</b>	Ulaz-obrađa-izlaz (6. Prilog a)
<b>Ishod:</b>	B. 6. 1. Učenik stvara, prati i preuređuje programe koji sadrže strukture grananja i uvjetnoga ponavljanja te predviđa ponašanje jednostavnih algoritama koji mogu biti prikazani dijagramom, riječima govornoga jezika ili programskim jezikom.
<b>Razrada ishoda:</b>	Učenik interpretira problem te prepoznaje ulazne vrijednosti i algoritamske strukture koje se upotrebljavaju za rješavanje problema, samostalno planira i slaže niz uputa (naredbi) kao rješenje problema primjenom algoritamskih struktura slijeda, grananja i ponavljanja. Učenik samostalno ili uz pomoć učitelja analizira zadani problem te predlaže neko algoritamsko rješenje, rješenje problema prikazuje dijagramom, riječima govornoga jezika ili naredbama programskoga jezika. Predviđa ponašanje algoritma te provjerava ispravnost algoritma prateći njegovo ponašanje (olovkom) ili testiranjem programa (algoritma) nekim ulaznim vrijednostima (na računalu). Učenik predviđa odgovarajuće ulazne (testne) primjere te kritički

	provjerava ispravnost rješenja i prema potrebi preuređuje svoje rješenje.
<b>Struktura i trajanje etapa blok sata:</b>	
<b>UVODNI DIO (15 min)- Opseg i površina</b>	
Učenici diskutiraju o opsegu i površini. Kreiraju dijagram toka za izračun opsega i dijagram toka za izračun površine pravokutnika. Učenici kreiraju program za izračun opsega i površine za stranice duljine 5 i 7 cm.	
<b>GLAVNI DIO (65 min)- Segmenti programa</b>	
Učenicima se postavlja pitanje što bi trebali učiniti ako žele izračunati opseg i površinu pravokutnika stranica duljine 10 i 12 cm. Učitelj prikazuje dijagram toka i program s unosom ulaznih vrijednosti za izračun površine pravokutnika. Učenici objašnjavaju program. Ponavljanje o segmentima programa i varijablama, određivanje ulaznih i izlaznih vrijednosti te razgovor o događajima tijekom obrade. Učenici samostalno izrađuju dijagram toka i program s unosom ulaznih vrijednosti za izračun opsega pravokutnika. Učenicima se zadaje novi zadatak: program za kupnju ulaznica za kino (s unosom cijene i broja ulaznica)- učenici izrađuju dijagram toka i programsko rješenje.	
<b>ZAVRŠNI DIO (10 min)- Provjera programa</b>	
Učenici prikazuju program, testiraju i prema potrebi ispravljaju.	

Tablica 29. Prijedlog uvođenja nastavnog sadržaja Potproblem u problemu

<b>Nastavni sadržaj:</b>	Potproblem u problemu (6. Prilog b))
<b>Ishod:</b>	<p>B. 6. 1. Učenik stvara, prati i preuređuje programe koji sadrže strukture grananja i uvjetnoga ponavljanja te predviđa ponašanje jednostavnih algoritama koji mogu biti prikazani dijagramom, riječima govornoga jezika ili programskim jezikom.</p> <p>B. 6. 2. Učenik razmatra i rješava složeniji problem rastavljajući ga na niz potproblema.</p>
<b>Razrada ishoda:</b>	<p>B. 6.1. Učenik interpretira problem te prepoznaje ulazne vrijednosti i algoritamske strukture koje se upotrebljavaju za rješavanje problema, samostalno planira i slaže niz uputa (naredbi) kao rješenje problema primjenom algoritamskih struktura slijeda, grananja i ponavljanja.</p> <p>B.6.2. Učenik opisuje složeniji problem te prepoznaje neke korake/dijelove u rješenju problema. Uz pomoć učitelja razvija plan</p>

	rješavanja problema te u njemu prepoznaje potprobleme, manje probleme s kojima se već susretao, odnosno probleme koje zna riješiti. Analizira mogućnost implementiranja rješenja potproblema u rješenje složenoga problema primjenjujući moguće izmjene/prilagodbe ako je potrebno.
<b>Struktura i trajanje etapa blok sata:</b>	
<b>UVODNI DIO (15 min)- Ponavljanje</b>	
Učenici se prisjećaju crtanja u Logo i osnovnih naredbi te crtaju kvadrat i jednakostranični trokut pomoću petlje.	
<b>GLAVNI DIO (67 min)- Crtaj nizove</b>	
Učitelj prikazuje program s nizom kvadrata. Učenici izrađuju programe s nizom kvadrata bez razmaka između njih, jednakostraničnih trokuta s razmakom između njih i četiri kvadrata koji čine jedan veći kvadrat. Učenici planiraju i provode crtanje.	
<b>ZAVRŠNI DIO (8 min)- Provjera programa</b>	
Učenici prikazuju nacrtano, provjeravaju i prema potrebi ispravljaju.	

### 9.7. 7. razred

Ishodi na kraju 7. razreda u domeni *Računalno razmišljanje i programiranje* (Ministarstvo znanosti i obrazovanja, 2019):

- B. 7. 1. Učenik razvija algoritme za rješavanje različitih problema koristeći se nekim programskim jezikom pri čemu se koristi prikladnim strukturama i tipovima podataka.
- B. 7. 2. Učenik primjenjuje algoritam (sekvencijalnog) pretraživanja pri rješavanju problema.
- B. 7. 3. Učenik dizajnira i izrađuje modularne programe koji sadrže potprograme u programskom jeziku.
- B. 7. 4. Učenik se koristi simulacijom pri rješavanju nekoga, ne nužno računalnoga, problema.

Tablica 30. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 7. razred

Broj sati (26)	Nastavne teme	Opis	Ishod
2	Tipovi podataka	-razgovor o vrstama podataka s kojima se svakodnevno susrećemo; prikaz programa s različitim ulaznim podatcima te razgovor (brojčani, znakovni nizovi i logički tipovi podataka); učenici nabrajaju programe koje su radili u 6. razredu i koristili različite vrste podataka; prikaz indeksiranja znakova i dohvaćanja znakova; korištenje operatora kod različitih tipova podataka	B. 7. 1.
2	Funkcije i metode	-razgovor o funkcijama i prikaz programa za prikaz cijelih i decimalnih brojeva, ispis broja znakova u znakovnom nizu; razgovor o funkcijama (int(), float(), str(), len()); prikaz pravila za korištenje funkcija u <i>Pythonu</i> -razgovor o metodama i prikaz programa za ispis podataka velikim slovima, malim slovima, prikaz pozicije znaka i izmjena znakovnog niza; razgovor o metodama (lower(), upper(), index(), replace())	B. 7. 1.
2	Riješi problem 1	-izrada programa za prebrojavanje nekog znaka u riječi, razmaka/riječi u rečenici -učenici individualno ili u parovima sastavljaju programe-program koji spaja dva znakovna niza, program koji vraća nalazi li se znak u nizu, program koji vraća podatke napisane velikim/malim slovima (pritom određuju tip podataka te funkcije i metode koje će koristiti)	B. 7. 1.
2	Složeni tipovi podataka	-upoznavanje s listama, prikaz liste s imenima učenika; upoznavanje s pravilima i indeksiranjem elemenata liste, nadovezivanje, uvišestručenje i provjera elemenata lista (+, *, in); prikaz programa za ispis lista, provjera elemenata u listama;	B. 7. 1.
2	Funkcije i metode u listama	-upoznavanje funkcija za rad s listama- min(), max(), sum(), len(), del() -upoznavanje metoda za rad s listama- insert(), remove(), index(), count(), append(); stvaranje liste od ulaznih vrijednosti-metoda split(); stvaranje liste s visinama učenika, određivanje najveće, najmanje, dodavanje visine učitelja, obriši visinu učitelja, provjeri postojanje visine, određivanje koji indeks ima neka vrijednost	B. 7. 1.
2	Riješi problem 2	-učenici individualno izrađuju program za stvaranje lista s ulaznim vrijednostima; u listu unose ocijene iz pismenog ispita- određuju najveću, najmanju, srednju vrijednost; određuju koja ocjena je najčešća -učenici individualno stvaraju listu s popisom država Europske unije prema vremenu ulaska u EU- ispis država članica, ispis broja članica, ispis koja država je ušla 5., 8., 15.	B. 7. 1.
2	Sekvencijalno pretraživanje	-izrada liste s godišnjim dobima, pretraživanje liste; kreiranje programa za unos rečenice- pretraživanje slova u	B. 7. 1. B. 7. 2.

		rečenici, pozicije slova i broj ponavljanja slova; izrada liste parnih brojeva od unesenih brojeva u listi; testiranje i ispravljanje pogrešaka	
2	Riješi problem 3	-izrada liste s trkačima- program koji daje poziciju (+1=startni broj) ili poruku da nema trkača na listi -izrada novih lista s pozitivnim/negativnim brojevima iz početne liste	B. 7. 1. B. 7. 2.
2	Projektna nastava: Kretanje temperature zraka	-učenicima se predstavlja problem- potrebno je pratiti temperaturu zraka sljedećih mjesec dana, grafički prikazati i predvidjeti temperaturu zraka za sljedećih mjesec dana; učenici u skupinama po 4 određuju podatke koji su im potrebni, sastavljaju tablice u Infogramu (infogram.com) i grafičkog prikaza koji predviđa temperaturu zraka sljedećih mjesec dana; učenici u sljedećih mjesec dana svakodnevno mjere temperaturu	B. 7. 4.
2	Potprogrami- <i>Python</i>	-razgovor o naučenim ugrađenim funkcijama i njihovim svrhama; razlika između potprograma i glavnog programa; objašnjavanje definiranja funkcije i definiranje funkcije za oduzimanje prirodnih brojeva (u kombinaciji s ulaznim i bez ulaznih vrijednosti te s vraćanjem i bez vraćanja vrijednosti)	B. 7. 1. B. 7. 3.
2	Kalkulator	-kreiranje funkcija za preračunavanje mjernih jedinica; kreiranje programa na principu kalkulatora	B. 7. 1. B. 7. 3.
2	Potprogrami- <i>Logo</i>	-učenici kreiraju program za crtanje niza kvadrata i pravokutnika -promatranje prikaza kocke- učenici uočavaju od kojih geometrijskih likova možemo prikazati kocku- učenici individualno crtaju prikaz kocke	B. 7. 1. B. 7. 3.
2	Projektna nastava: Kretanje temperature zraka	-učenici unose izmjerene podatke u Infogram (infogram.com); linijama prikazuju predviđene i izmjerene podatke te ih uspoređuju; učenici prema izmjerenim podacima predviđaju daljnje kretanje temperature; učenici izrađuju prezentaciju/plakat s podacima i predviđanjima te analizom i vrednovanjem	B. 7. 4.

Tablica 31. Prijedlog uvođenja nastavnog sadržaja Sekvencijalno pretraživanje

<b>Nastavni sadržaj:</b>	Sekvencijalno pretraživanje (7. Prilog a))
<b>Ishod:</b>	B. 7. 1. Učenik razvija algoritme za rješavanje različitih problema koristeći se nekim programskim jezikom pri čemu se koristi prikladnim strukturama i tipovima podataka. B. 7. 2. Učenik primjenjuje algoritam (sekvencijalnog) pretraživanja pri rješavanju problema.

<b>Razrada ishoda:</b>	<p>B. 7. 1. Analizira problem, odabire strategiju rješavanja, rješenje realizira u obliku programa s odgovarajućim tipovima podataka. Provjerava ispravnost rješenja te ga preuređuje po potrebi.</p> <p>B. 7. 2. Učenik prepoznaje i objašnjava ideju nekoga algoritma (sekvencijalnoga) pretraživanja koju je potrebno primijeniti za rješavanje zadanoga problema. Razmišlja o kriteriju za pretraživanje, identificira različite algoritamske strukture kojima se provodi algoritam (sekvencijalnoga) pretraživanja.</p>
<b>Struktura i trajanje etapa blok sata:</b>	
UVODNI DIO (7 min)- <b>Izrada liste</b>	
Izrada liste s godišnjim dobima. Korištenjem metode <i>index()</i> pretražuje se <i>ljet</i> , <i>jesen</i> , <i>ponedjeljak</i> unutar liste. Učenici uočavaju razliku u izlaznoj vrijednosti.	
GLAVNI DIO (73 min)- <b>Sekvencijalno pretraživanje</b>	
<p>Ponavljjanje indeksiranja. Zajedničko kreiranje programa za unos rečenice. Pretraživanje slova u rečenici i ispis pozicije. Uočavanje ispisivanja prve pozicije na kojem se pojavljuje slovo u rečenici. Učitelj s učenicima nadograđuje program tako da daje ispis je li slovo u rečenici ili nije. Ako je rezultat pretrage istinit tada se ispisuje i pozicija traženog slova.</p> <p>Testiranje programa. Nadogradnja programa tako da se uz istinitost pretrage ispisuje i broj pojavljivanja slova. Testiranje programa. Upoznavanje s izradom prazne liste. Izrada liste u koju se unose brojevi te kreiranje nove liste s parnim brojevima iz te liste.</p>	
ZAVRŠNI DIO (10 min)- <b>Provjera programa</b>	
Testiranje programa, objašnjavanje i ispravljanje eventualnih pogrešaka.	

Tablica 32. Prijedlog uvođenja nastavnog sadržaja Potprogrami- Python

<b>Nastavni sadržaj:</b>	Potprogrami- Python (7. Prilog b))
<b>Ishod:</b>	<p>B. 7. 1. Učenik razvija algoritme za rješavanje različitih problema koristeći se nekim programskim jezikom pri čemu se koristi prikladnim strukturama i tipovima podataka.</p> <p>B. 7. 3. Učenik dizajnira i izrađuje modularne programe koji sadrže potprograme u programskom jeziku.</p>
<b>Razrada ishoda:</b>	B. 7. 1. Opisuje način rješavanja problema naredbama nekoga programskog jezika koristeći se različitim osnovnim tipovima podataka. Analizira problem, odabire strategiju rješavanja, rješenje

	<p>realizira u obliku programa s odgovarajućim tipovima podataka. Provjerava ispravnost rješenja te ga preuređuje po potrebi.</p> <p>B. 7. 3. Učenik opisuje problem, prepoznaje u njemu potprobleme. Učenik potproblem opisuje nizom uputa u programskom jeziku. Analizira problem te povezuje module programa odgovarajućim parametrima. Učenik odabire strategiju rješavanja problema rastavljajući ga na manje potprobleme, algoritamsko rješenje problema realizira u obliku programa koji može sadržavati više od jednoga modula, provjerava ispravnost rješenja ta ga preuređuje prema potrebi</p>
<b>Struktura i trajanje etapa blok sata:</b>	
<b>UVODNI DIO (13 min)- Naučene funkcije</b>	
Izrada programa za spajanje 2 stringa i zbrajanje 2 cijela broja. Razgovor o nužnosti korištenja ugrađenih funkcija i njihovoj svrsi.	
<b>GLAVNI DIO (67 min)- Definiraj funkciju</b>	
Prikaz programa s funkcijom koja određuje djeljivost s 5- prikaz glavnog programa i potprograma- uz objašnjavanje i definiranje. Razgovor o definiranju funkcije i pravilima koja se pritom treba pridržavati, pozivanju u programu. Prikaz pozivanja funkcije. Razgovor i objašnjavanje problema oduzimanja prirodnih brojeva. Prikaz i objašnjavanje programa s funkcijom za oduzimanje prirodnih brojeva (u kombinaciji s ulaznim i bez ulaznih vrijednosti te s vraćanjem i bez vraćanja vrijednosti). Učenici odabiru jedan od četiri prikazana programa te ga individualno izrađuju.	
<b>ZAVRŠNI DIO (10 min)- Provjera programa</b>	
Testiranje programa, objašnjavanje i ispravljanje eventualnih pogrešaka.	

### 9.8.8. razred

Ishodi na kraju 8. razreda u domeni *Računalno razmišljanje i programiranje* (Ministarstvo znanosti i obrazovanja, 2019):

B. 8. 1. Učenik identificira neki problem iz stvarnoga svijeta, stvara program za njegovo rješavanje, dokumentira rad programa i predstavlja djelovanje programa drugima.

B. 8. 2. Učenik prepoznaje i opisuje algoritam sortiranja, primjenjuje jedan algoritam sortiranja za rješavanje zadanoga problema u programskom jeziku.

B. 8. 3. Učenik prepoznaje i opisuje mogućnost primjene rekurzivnih postupaka pri rješavanju odabranih problema te istražuje daljnje mogućnosti primjene rekurzije.

Tablica 33. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 8. razred

Broj sati (26)	Nastavne teme	Opis	Ishod
2	Web stranica	-učitelj prikazuje <i>WordPress</i> za izradu web stranice-izrada računa, način kreiranja web stranice; prikaz nekoliko stranica napravljenih putem <i>WordPresa</i> -učitelj zadaje zadatak za cijelu godinu- izrada web stranice razredne zajednice; planiranje- cilj, datum objavljivanja, popis članova i raspodjela posla; planiranje izrade i početak izrade	B. 8. 1.
2	Rekurzija	-prikaz primjera rekurzije; učenici nabrajaju više primjera rekurzije; prikaz primjera rekurzije u <i>Pythonu</i> i <i>Logu</i> ; razgovor o potrebi za rekurzijom i njezinoj primjeni; prikaz dijagrama toka i objašnjavanje načina rada programa s rekurzijom	B. 8. 3.
2	Primjena rekurzije u <i>Pythonu</i>	-izrada dijagrama toka i programa za zbroj prvih n brojeva; prenamjena programa za zbrajanje brojeva između unesenih brojeva u <i>Pythonu</i>	B. 8. 3.
2	Primjena rekurzije u <i>Logu</i>	- izrada dijagrama toka i programa za crtanje niza sve većih kvadrata sa zajedničkim vrhom; niz koncentričnih kružnica; učenici osmišljavaju crtanje rekurzivnim postupcima	B. 8. 3.
2	Sortiranje	-razgovor o nužnosti sortiranja- što sve i zašto sortiramo; učenici dobivaju 3 rezultata s natjecanja bacanja kugle te sortiraju rezultate od najmanjeg do najvećeg, a zatim od najvećeg do najmanjeg; objašnjavaju postupak; iznose ideje o sortiranju u <i>Pythonu</i> ; izrada programa za rješenje problema- sortiranje 3 broja počevši s najmanjim; učenici uređuju program tako da ispisuje počevši s najvećim	B. 8. 2.
2	Funkcije za sortiranje	-učenici dobivaju 10 rezultata s natjecanja skoka u vis te sortiraju rezultate od najmanjeg do najvećeg te obratno; učenici iznose ideje o rješenju problema u <i>Pythonu</i> ; razgovor o nedostatku usporedbe kao što je primijenjena	B. 8. 2.



		za usporedbu tri broja na ovom primjeru; prikaz funkcija i razlika u funkcijama sort(), sorted(); izrada programa za sortiranje unesenih vrijednosti počevši od najmanjeg; izmjena- počevši od najvećeg	
2	Riješi problem sortiranjem	-predstavljanje problema- natjecanje iz atletike: potrebno je poredati rezultate skoka u vis, skoka u dalj, bacanja kladiva i bacanja kugle po redu počevši od najvećeg -sortiranje liste prema duljini znakovnog stringa	B. 8. 1. B. 8. 2.
2	Predstavljanje mobilnih aplikacija i IoT primjera	-učenici istražuju i pronalaze različite primjere IoT primjene, kreiraju prezentacije te predstavljaju; razgovor o primjeni i osmišljavanje primjera za izradu mobilnih aplikacija i primjenu IoT-a	B. 8. 1.
2	Projektna nastava: Matematika	-zadatci za samostalan ili rad u paru (učenik opisuje odabrani problem, analizira problem, predviđa ulazne vrijednosti problema te razvija dijagram toka, provjerava ispravnost rješenja te ga prema potrebi preuređuje, stvara dokumentaciju prema planu te argumentirano predstavlja svoj program) -zadatci za odabir: 1. Program za izračun kvadrata zbroja i razlike; 2. Računanje dijagonale pravokutnika/kvadrata; 3. Izračun površine baze i pobočja te volumena i oplošja pravilne četverostrane piramide	B. 8. 1.
2	Projektna nastava: Fizika 7. razred	-zadatci za samostalan ili rad u paru (učenik opisuje odabrani problem, analizira problem, predviđa ulazne vrijednosti problema te razvija dijagram toka, provjerava ispravnost rješenja te ga prema potrebi preuređuje, stvara dokumentaciju prema planu te argumentirano predstavlja svoj program); predstavljanje programa u 7. razredu -zadatci za odabir: 1. Dinamometar; 2. Izračun tlaka s pretvaranjem u $m^2$ ; 3. Izračun snage s unosom sile, puta i vremena	B. 8. 1.
2	Projektna nastava: Fizika	-zadatci za samostalan ili rad u paru (učenik opisuje odabrani problem, analizira problem, predviđa ulazne vrijednosti problema te razvija dijagram toka, provjerava ispravnost rješenja te ga prema potrebi preuređuje, stvara dokumentaciju prema planu te argumentirano predstavlja svoj program) -zadatci za odabir: 1. Pretvorba km/h u m/s i obrnuto sa sortiranjem do upisa 0; 2. Izračun količine vode koja stane u posudu oblika pravokutnika te sortiranje do upisa 0; 3. Usporedba gustoće s unosom $kg/m^3$ i $g/cm^3$ te sortiranje do upisa 0	B. 8. 1. B. 8. 2.
2	Samostalan odabir programa za izradu	-zadatci za samostalan ili rad u paru (učenik opisuje odabrani problem, analizira problem, predviđa ulazne vrijednosti problema te razvija dijagram toka, provjerava ispravnost rješenja te ga prema potrebi preuređuje, stvara dokumentaciju prema planu te argumentirano predstavlja svoj program)	B. 8. 1.
2	Predstavljanje web stranica	-učenici predstavljaju web stranice razreda; zajedničko ocjenjivanje	B. 8. 1.

Tablica 34. Prijedlog uvođenja nastavnog sadržaja Sortiranje

<b>Nastavni sadržaj:</b>	Sortiranje (Prilog 8. a))
<b>Ishod:</b>	B. 8. 2. Učenik prepoznaje i opisuje algoritam sortiranja, primjenjuje jedan algoritam sortiranja za rješavanje zadanoga problema u programskom jeziku.
<b>Razrada ishoda:</b>	Učenik prepoznaje da se problem učinkovitije rješava s podacima koji su sortirani. Uočava potproblem sortiranja u zadanome problemu. Opisuje postupak sortiranja riječima ili grafički te uz pomoć učitelja ili samostalno primjenjuje jedan algoritam sortiranja za rješavanje zadanog problema u kojemu programskom jeziku.
<b>Struktura i trajanje etapa blok sata:</b>	
UVODNI DIO (10 min)- <b>Što sve sortiramo?</b>	
Razgovor o sortiranju- što je sortiranje, kada koristimo sortiranje, što sve sortiramo, u kojem programu su već sortirali podatke i na koji način.	
GLAVNI DIO (65 min)- <b>Sortiranje rezultata</b>	
Učenicima se zadaju tri rezultata bacanja kugle- 21.35, 22.12, 21.40. Učenici trebaju sortirati rezultate od najmanjeg do najvećeg. Učenici sortiraju brojeve. Učenici objašnjavaju način na koji su sortirali rezultate. Učenici sortiraju rezultate od najmanjeg do najvećeg, objašnjavaju postupak, iznose ideje o rješenju zadatka pomoću <i>Pythona</i> . Zajedničko kreiranje programa za sortiranje 3 broja te prikazom rješenja počevši od najmanjeg. Testiranje programa.	
ZAVRŠNI DIO (15 min)- <b>Izmijeni</b>	
Učenici izmjenjuju program tako da ispisuje sortirane podatke počevši od najvećeg. Testiraju i prema potrebi ispravljaju program.	

Tablica 35. Prijedlog uvođenja nastavnog sadržaja Primjena rekurzije u Pythonu

<b>Nastavni sadržaj:</b>	Primjena rekurzije u <i>Pythonu</i> (Prilog 8. b))
<b>Ishod:</b>	B. 8. 3. Učenik prepoznaje i opisuje mogućnost primjene rekurzivnih postupaka pri rješavanju odabranih problema te istražuje daljnje mogućnosti primjene rekurzije.
<b>Razrada ishoda:</b>	Analizira odabrani problem te u njemu identificira osnovi slučaj rekurzije te način rekurzivnoga pozivanja. Pronalazi i predlaže rješenje

	(grafički, riječima/ uputama) odabranoga problema primjenom rekurzivnoga postupka.
<b>Struktura i trajanje etapa blok sata:</b>	
<b>UVODNI DIO (15 min)- Rješenje bez rekurzije</b>	
Učenicima se predstavlja zadatak zbroj prvih n brojeva. Učenici analiziraju problem i predviđaju ulazne vrijednosti. Prikaz rješenja bez rekurzije. Komentiranje rješenja i plan za korištenje rekurzije.	
<b>GLAVNI DIO (65 min)- Primjeni rekurziju</b>	
Zajednička izrada dijagrama toka za rješenje problema rekurzijom. Izrada programa. Testiranje. Preuređenje programa za zbrajanje brojeva između dva unesena broja.	
<b>ZAVRŠNI DIO (10 min)- Provjera programa</b>	
Testiranje i preuređivanje prema potrebi te predstavljanje programa.	

## 10. Projektna nastava- povezanost Programiranja i računalnog razmišljanja s ostalim nastavnim predmetima

Tablica 36. Projektna nastava u 4. razredu- Programi za množenje

<b>Nastavni sadržaj:</b>	<b>Programi za množenje (9. Prilog)</b>
<b>Ishodi:</b>	<p><b>B. 4. 1.</b> Učenik stvara program korištenjem vizualnog okruženja u kojem koristi slijed, ponavljanje, odluku i ulazne vrijednosti.</p> <p><b>B. 4. 2.</b> Učenik rješava složenije logičke zadatke s uporabom računala ili bez uporabe računala.</p>
<b>Korelacija:</b>	<b>MAT OŠ A.4.3.</b> Pisano množi i dijeli dvoznamenkastim brojevima u skupu prirodnih brojeva do milijun.
<b>Struktura i trajanje etapa blok sata:</b>	
UVODNI DIO (5 min)	
<p>Učitelj učenike prema sposobnostima i vještinama dijeli u tri skupine. Učenicima se dijele listići sa zadacima i listić za plan.</p> <p><b>ZADATCI:</b></p> <p>1. skupina</p> <p>Izradite program za vježbanje množenja pri čemu će program zadati 2 broja jedan četveroznamenkast, a drugi dvoznamenkasti. Program treba nuditi mogućnost prikaza umnoška (kada želimo provjeriti jesmo li točno riješili zadatak). Koristite <i>Scratch</i>.</p> <p>2. skupina</p> <p>Izradite program za vježbanje množenja pri čemu mi unosimo faktore. Program treba nuditi mogućnost prikaza umnoška (kada želimo provjeriti jesmo li točno riješili zadatak). Koristite <i>Scratch</i>.</p> <p>3. skupina</p> <p>Izradite program za vježbanje množenja pri čemu će program zadati 2 broja jedan četveroznamenkast, a drugi dvoznamenkasti. Program treba nuditi mogućnost prikaza umnoška (kada želimo provjeriti jesmo li točno riješili zadatak). Koristite <i>Micro:bit</i>.</p>	
GLAVNI DIO (65 min)	
<p>Učenici u skupinama analiziraju zadatak, osmišljavaju plan za izradu programa, provode plan, odnosno izrađuju program te ga testiraju.</p> <p>Učenici po potrebi traže pomoć učitelja koji ih usmjerava.</p>	
ZAVRŠNI DIO (20 min)	

Svaka skupina prikazuje i objašnjava svoj program, argumentira i objašnjava svoje postupke i probleme s kojima su se susreli tijekom izrade programa. Učenici predlažu načine korištenja programa i daljnji razvoj.

Tablica 37. Projektna nastava u 5. razredu- Program- crtaj mnogokute

<b>Nastavni sadržaj:</b>	<b>Program- crtaj mnogokute (10. prilog)</b>
<b>Ishodi:</b>	B. 5. 1. Učenik se koristi programskim alatom za stvaranje programa u kojemu se koristi ulaznim i izlaznim vrijednostima te ponavljanjem.
<b>Korelacija:</b>	MAT OŠ C.5.2. Opisuje i crta / konstruira geometrijske likove te stvara motive koristeći se njima MAT OŠ D.5.1. Mjeri i crta kutove, određuje mjere susjednih i vršnih kutova.
<b>Struktura i trajanje etapa blok sata:</b>	
UVODNI DIO (15 min)	
Učenici se dijele u parove. Učitelj dijeli listiće s prikazima pravilnih geometrijskih likova i tablicom. Učitelj učenicima predstavlja problem- izrada programa u kojem se zadaje veličina stranice i broj kutova, a rezultat je crtanje pravilnog mnogokuta.	
GLAVNI DIO (65 min)	
Učenici mjere veličine kutova te zaključuju o veličini kutova s obzirom na broj stranica. Učenici u paru kreiraju algoritam i program za crtanje mnogokuta. Učitelj obilazi učenike i nadgleda rad, po potrebi navodi.	
ZAVRŠNI DIO (10 min)	
Učenici provjeravaju ispravnost, uspoređuju i predstavljaju svoj program.	

Tablica 38. Projektna nastava u 6. razredu- Trokut

<b>Nastavni sadržaj:</b>	<b>Program za mnogokute (11. prilog)</b>
<b>Ishodi:</b>	B. 6. 1. Učenik stvara, prati i preuređuje programe koji sadrže strukture grananja i uvjetnoga ponavljanja te predviđa ponašanje jednostavnih algoritama koji mogu biti prikazani dijagramom, riječima govornoga jezika ili programskim jezikom. B. 6. 2. Učenik razmatra i rješava složeniji problem rastavljajući ga na niz potproblema.
<b>Korelacija:</b>	MAT OŠ C.6.2 Konstruira trokute, analizira njihova svojstva i odnose. MAT OŠ D.6.2. Računa i primjenjuje opseg i površinu trokuta i četverokuta te mjeru kuta.
<b>Struktura i trajanje etapa blok sata:</b>	
UVODNI DIO (10 min)	
Učenici se dijele u 4 skupine. Svaka skupina dobiva zadatak- pomoću programskih jezika <i>Logo</i> i <i>Python</i> osmisli program za ponavljanje nastavnog sadržaja trokut. Pritom je potrebno izraditi program za crtanje vrsta trokuta s obzirom na duljinu stranica te program za računanje opsega i površine trokuta.	
GLAVNI DIO (60 min)	
Učenici dijele problem u potprobleme te raspoređuju zadatke. Učenici izrađuju dijagrame toka te kreiraju programe. Učitelj obilazi učenike i nadgleda rad, po potrebi navodi.	
ZAVRŠNI DIO (20 min)	
Učenici provjeravaju ispravnost, uspoređuju i predstavljaju svoj program.	

Tablica 39. Projektna nastava u 7. razredu- Kalkulator

<b>Nastavni sadržaj:</b>	<b>Kalkulator (12. prilog)</b>
<b>Ishodi:</b>	B. 7. 1. Učenik razvija algoritme za rješavanje različitih problema koristeći se nekim programskim jezikom pri čemu se koristi prikladnim strukturama i tipovima podataka. B. 7. 3. Učenik dizajnira i izrađuje modularne programe koji sadrže potprograme u programskom jeziku.
<b>Korelacija:</b>	MAT OŠ D.7.5. Odabire i preračunava odgovarajuće mjerne jedinice.
<b>Struktura i trajanje etapa blok sata:</b>	
UVODNI DIO (10 min)	

Učitelj predstavlja problem- kreiranje kalkulatora za preračunavanje mjernih jedinica za duljinu- odabiru 3 mjerne jedinice za pretvorbu iz jedne u drugu. U programu je potrebno kreirati funkcije koje se pozivaju u glavnom programu. Učenici se dijele u parove.
GLAVNI DIO (60 min)
Učenici izrađuju plan i tablice za preračunavanje mjernih jedinica, kreiraju dijagram toka i program. Učitelj obilazi učenike i nadgleda rad, po potrebi navodi.
ZAVRŠNI DIO (20 min)
Učenici provjeravaju ispravnost, uspoređuju i predstavljaju svoj program.

Tablica 40. Projektna nastava u 8. razredu- Matematika

<b>Nastavni sadržaj:</b>	<b>Matematika (13. prilog)</b>
<b>Ishodi:</b>	B. 8. 1. Učenik identificira neki problem iz stvarnoga svijeta, stvara program za njegovo rješavanje, dokumentira rad programa i predstavlja djelovanje programa drugima.
<b>Korelacija:</b>	MAT OŠ D.8.1. Primjenjuje Pitagorin poučak. MAT OŠ D.8.2. Primjenjuje oplošje i volumen geometrijskih tijela. MAT OŠ B.8.1. Računa s algebarskim izrazima u $R$ .
<b>Struktura i trajanje etapa blok sata:</b>	
UVODNI DIO (10 min)	
Učenicima se prikazuje matematičko proširenje te način korištenja i naredbe koje će im trebati u kreiranju programa. Učenicima se zadaju problemi: 1. Program za izračun kvadrata zbroja i razlike; 2. Računanje dijagonale pravokutnika/kvadrata; 3. Izračun površine baze i pobočja te oplošja i volumena pravilne četverostrane piramide.	
GLAVNI DIO (60 min)	
Učenici rade u paru ili individualno- učenik opisuje odabrani problem, analizira problem, predviđa ulazne vrijednosti problema i kreira algoritam te izrađuje programsko rješenje.	
ZAVRŠNI DIO (20 min)	
Učenik provjerava ispravnost rješenja te ga prema potrebi preuređuje, prema planu stvara dokumentaciju te argumentirano predstavlja svoj program.	

## 11. Zaključak

*Računalno razmišljanje i programiranje* jedna je od četiri domene u nastavnom predmetu Informatika određen *Kurikulumom nastavnog predmeta Informatika za osnovne škole i gimnazije*. Važnost razvoja računalnog razmišljanja i programiranja istaknuto je u *Kurikulumu*, od razvoja logičkog i algoritamskog razmišljanja, sagledavanja cjelokupnog procesa stvaranja, modeliranja, apstrahiranja, razvoj sposobnosti rješavanja problema, sustavnost, samopouzdanje, upornost i timski rad. Kako bi se razvijalo razmišljanje i vještine potreban je odabir odgovarajućeg sadržaja i programskog jezika. Programski jezik ima veliku ulogu u motiviranju i razvoju interesa kod učenika. Ako je odabran pogrešan programski jezik dolazi do frustracije, demotivacije i odustajanja.

Udžbenici i materijali koje nude hrvatski izdavači nude mnoge zanimljive prijedloge za usvajanje ishoda, no učitelj je kreator nastavnog procesa te udžbeničku građu treba birati i prilagođavati potrebama učenika i aktualnosti nastave. Generalno, udžbenici nude premalo zadataka za projektnu nastavu i za odabir na kojima učenici vježbaju analizu problema, osmišljavaju rješenja, argumentiraju svoje postupke. U udžbenicima se uglavnom nudi samo jedna mogućnost rješenja zadatka, iako programiranje nudi mnoge mogućnosti rješenja jednog problema. Budući da se igra više ne smatra samo nastavnom metodom, već i nastavnim sredstvom, postavlja se pitanje potrebe za udžbenicima, naročito kod učenja programiranja.

Prijedlozima za ostvarenje odgojno-obrazovnih ishoda ponuđene su nastavne teme s opisom prema razradi ishoda. Polazište u kreiranju prijedloga kognitivni je razvoj učenika, ali i primjena zadataka u drugim nastavnim predmetima. Suvremena nastava teži individualiziranom pristupu učenja, stoga su prijedlozi primjenjivi, no podložni promjenama.



## 12.Literatura

1. Blagus, J., Ljubić Klemše, N., Flisar Odorčić, A., Bubica, N., Ružić, I., & Mihočka, N. (2020). *e-SVIJET 1*. Školska knjiga. Pristup: 5.4.2021. <https://www.e-sfera.hr/udzbenicke-serije/podraska/3d6ff748-1f05-4767-aad6-73b3aa8981ca>
2. Blagus, J., Ljubić Klemše, N., Flisar Odorčić, A., Ružić, I., & Mihočka, N. (2020a). *e-SVIJET 2*. Školska knjiga. Pristup: 5.4.2021. <https://www.e-sfera.hr/udzbenicke-serije/podraska/bfe9b919-708d-42c9-8240-8e7a85b8df88>
3. Blagus, J., Ljubić Klemše, N., Flisar Odorčić, A., Ružić, I., & Mihočka, N. (2020b). *e-SVIJET 3*. Školska knjiga. Pristup: 5.4.2021. <https://www.e-sfera.hr/udzbenicke-serije/podraska/e6e81764-edaf-45bb-aa9f-72fe641b4565>
4. Blagus, J., Ljubić Klemše, N., Ružić, I., & Stančić, M. (2020). *e-SVIJET 4*. Školska knjiga. Pristup: 5.4.2021. <https://www.e-sfera.hr/udzbenicke-serije/podraska/30eba414-5430-430c-84ff-9fa3dd32d10a>
5. Bubica, N., Mladenović, M., & Boljat, I. (2013). *Programiranje kao alat za razvoj apstraktnog mišljenja*. Pristup: 8.3.2021. [https://radovi.cuc.carnet.hr/modules/request.php?module=oc\\_program&action=summary.php&id=44](https://radovi.cuc.carnet.hr/modules/request.php?module=oc_program&action=summary.php&id=44)
6. CARNet. (n.d.-a). *Obrazovni trendovi za suvremenu nastavu: Problemska nastava*. Pristup: 18.4.2021. <https://mooc.carnet.hr/mod/book/view.php?id=25646&chapterid=6783>
7. CARNet. (n.d.-b). *Obrazovni trendovi za suvremenu nastavu: Projektna nastava*. Pristup: 18.4.2021. <https://mooc.carnet.hr/mod/book/view.php?id=25646&chapterid=6782>
8. Deljac, S., Ercegovac, M., Gregurić, V., Hajdinjak, N., Kijuk, A., Počuča, B., Rakić, D., & Svetličić, S. (2020a). *Informatika 6*. Profil Klett. Pristup: 6.4.2021. <https://hr.izzi.digital/DOS/16220/16221.html>
9. Deljac, S., Ercegovac, M., Gregurić, V., Hajdinjak, N., Kijuk, A., Počuča, B., Rakić, D., & Svetličić, S. (2020b). *Informatika 7*. Profil Klett. Pristup: 6.4.2021. <https://hr.izzi.digital/DOS/16254/16255.html>
10. Horvat, S., & Prpić, M. (2020). *MIŠolovka 2*. Pristup: 5.4.2021. Udžbenik.hr. <https://e.udzbenik.hr/2r/>

11. Horvat, T. (2019). *Kognitivni razvoj i učenje djeteta školske dobi*. Pristup: 12.4.2021. <https://repositorij.unipu.hr/islandora/object/unipu%3A4135/datastream/PDF/view>
12. Hrvatska enciklopedija. (2020). *Informatika*. Leksikografski Zavod Miroslav Krleža. Pristup: 7.1.2021. <https://www.enciklopedija.hr/Natuknica.aspx?ID=27412>
13. ISTE, & CSTA. (2011). Operational Definition of Computational Thinking. Pristup: 5.3.2021. <https://cdn.iste.org/www-root/ct-documents/computational-thinking-operational-definition-flyer.pdf>
14. Krpan, D. (2018). Posredovani prijenos u poučavanju programiranja s vizualnim programskim jezicima. *Politehnika*, 2(1), 71–79. Pristup: 17.4.2021. <https://hrcak.srce.hr/208140>
15. *mBot*. (n.d.). Makeblock. Pristup: 17.4.2021. <https://www.makeblock.com/mbot>
16. McLeod, S. A. (2019). Bruner - Learning Theory in Education. *Simply Psychology*. Pristup: 17.4.2021. <https://www.simplypsychology.org/bruner.html>
17. Ministarstvo znanosti i obrazovanja. (2019). *Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije*. Ministarstvo znanosti i obrazovanja. Pristup: 15.11.2020. [https://skolazazivot.hr/wp-content/uploads/2020/06/INF\\_kurikulum.pdf](https://skolazazivot.hr/wp-content/uploads/2020/06/INF_kurikulum.pdf)
18. Ministarstvo znanosti i obrazovanja. (2020). *Izborna informatika od iduće školske godine za učenike razredne nastave*. Ministarstvo znanosti i obrazovanja. Pristup: 18.11.2020. <https://mzo.gov.hr/print.aspx?id=3726&url=print>
19. Mladenović, M. (2019). *Poučavanje početnog programiranja oblikovanjem računalnih igara*. Pristup: 29.3.2021. <https://dr.nsk.hr/islandora/object/pmfst%3A760>
20. Olabe, J. C., Olabe, M. A., Basogain, X., Maiz, I., & Castaño, C. (2011). Programming and Robotics with Scratch in Primary Education. *Education in a Technological World: Communicating Current and Emerging Research and Technological Efforts, July 2016*, 356–363. Pristup: 29.3.2021. [https://www.researchgate.net/profile/Carlos-Castano-2/publication/268255644\\_Programming\\_and\\_Robotics\\_with\\_Scratch\\_in\\_Primary\\_Education/links/5630de6908ae0530378ce280/Programming-and-Robotics-with-Scratch-in-Primary-Education.pdf](https://www.researchgate.net/profile/Carlos-Castano-2/publication/268255644_Programming_and_Robotics_with_Scratch_in_Primary_Education/links/5630de6908ae0530378ce280/Programming-and-Robotics-with-Scratch-in-Primary-Education.pdf)
21. Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2020). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences.

- Computers in Human Behavior*, 105. Pristup: 23.2.2021.  
<https://doi.org/10.1016/j.chb.2019.03.003>
22. Python. (n.d.). Pristup: 14.4.2021. <https://www.python.org/>
23. Rihter, B., Rade, D., Dlačić, K. T., Topić, S., Novaković, L., Bujadinović, D., Pandurić, T., & Orlović, D. (2021). *Like IT 8*. Alfa. Pristup: 6.4.2021.  
[https://hr.mozaweb.com/course.php?cmd=single\\_book&bid=HR-ALFA-INF8-6267&from=subject&from\\_value=INF&subject=INF&grade=&t=mozaweb](https://hr.mozaweb.com/course.php?cmd=single_book&bid=HR-ALFA-INF8-6267&from=subject&from_value=INF&subject=INF&grade=&t=mozaweb)
24. Rihter, B., Rade, D., Toić Dlačić, K., Topić, S., Novaković, L., Bujadinović, D., & Pandurić, T. (2019). *Like IT 5*. Alfa. Pristup: 6.4.2021.  
[https://hr.mozaweb.com/course.php?cmd=single\\_book&bid=HR-ALFA-INF5-0495&from=subject&from\\_value=INF&subject=INF&grade=&t=mozaweb](https://hr.mozaweb.com/course.php?cmd=single_book&bid=HR-ALFA-INF5-0495&from=subject&from_value=INF&subject=INF&grade=&t=mozaweb)
25. Robins, A., Rountree, J., & Rountree, N. (2003). *Learning and Teaching Programming: A Review and Discussion*. Pristup: 29.3.2021.  
<http://www.science.smith.edu/classwiki/images/1/14/RobinsRev.pdf>
26. Ružić, I., Stančić, M., Bubica, N., Dimovski, Z., Leko, S., Mihočka, N., Vejnović, B., & Babić, M. (2020a). *Moj portal 5*. Školska knjiga. Pristup: 6.4.2021. <https://www.e-sfera.hr/udzbenicke-serije/podrska/1934ee5c-3880-4192-9287-790abf5c6ab0>
27. Ružić, I., Stančić, M., Bubica, N., Dimovski, Z., Leko, S., Mihočka, N., Vejnović, B., & Babić, M. (2020b). *Moj portal 6*. Školska knjiga. Pristup: 6.4.2021. <https://www.e-sfera.hr/udzbenicke-serije/podrska/8f16c4f9-fcc7-49ed-b9d4-822e9f69d6ec>
28. Ružić, I., Stančić, M., Bubica, N., Dimovski, Z., Leko, S., Mihočka, N., Vejnović, B., & Babić, M. (2020c). *Moj portal 7*. Školska knjiga. Pristup: 6.4.2021. <https://www.e-sfera.hr/udzbenicke-serije/podrska/6bba8935-8fc6-434a-a07b-8d3f10840e75>
29. Ružić, I., Stančić, M., Bubica, N., Dimovski, Z., Leko, S., Mihočka, N., Vejnović, B., & Babić, M. (2021). *Moj portal 8*. Školska knjiga. Pristup: 6.4.2021. <https://www.e-sfera.hr/udzbenicke-serije/podrska/c0273c34-fb37-4eb1-acba-0a25d493f20f>
30. Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22(9). Pristup: 8.3.2021.  
<https://doi.org/10.1016/j.edurev.2017.09.003>
31. Slavin, T. (n.d.). *Logo*. Kidscodex.Com. Pristup: 14.4.2021.

- <https://www.kidscodecs.com/logo-programming-language/>
32. Službena stranica *Micro:bita*. (n.d.). Pristup: 14.4.2021. <https://microbit.org/hr/get-started/first-steps/introduction/>
  33. Službena web stranica- *Scratch*. (n.d.). Pristup: 14.4.2021. <https://scratch.mit.edu/>
  34. Sokol, G., Mandić, M., Lohajner, G., & Purgar, J. (2020a). *MIŠolovka 3*. Školska knjiga. Pristup: 5.4.2021. <https://e.udzbenik.hr/3r/>
  35. Sokol, G., Mandić, M., Lohajner, G., & Purgar, J. (2020b). *MIŠolovka 4*. Školska knjiga. Pristup: 5.4.2021. <https://e.udzbenik.hr/4r/define.asp?r=4>
  36. Sokol, G., Mandić, M., Purgar, J., & Lohajner, G. (2020). *MIŠolovka 1*. Udžbenici.hr. Pristup: 5.4.2021. <https://e.udzbenik.hr/1r/>
  37. Suradnici u učenju. (n.d.). *Pravilnik međunarodnog natjecanja iz informatike i računalnog razmišljanja "Dabar."* Pristup: 29.3.2021. <https://ucitelji.hr/dabar/>
  38. Tomljenović, K. (2018). *Računalno razmišljanje i uloga učenja pomoću igre na njegov razvoj*. Učiteljski fakultet. Pristup: 23.2.2021. [https://www.inf.uniri.hr/images/studiji/poslijediplomski/kvalifikacijski/Kvalifikacijski\\_rad\\_Kreso\\_Tomljenovic.pdf](https://www.inf.uniri.hr/images/studiji/poslijediplomski/kvalifikacijski/Kvalifikacijski_rad_Kreso_Tomljenovic.pdf)
  39. Wing, J. M. (n.d.). *Research Notebook: Computational Thinking-What and Why?* The LINK. Pristup: 23.2.2021. <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>

## POPIS TABLICA:

Tablica 1. Aspekti računalnog razmišljanja .....	2
Tablica 2. Odgojno-obrazovni ishodi u domeni Računalno razmišljanje i programiranje po razredima.....	7
Tablica 3. 1. razred.....	9
Tablica 4. 2. razred.....	10
Tablica 5. 3. razred.....	11
Tablica 6. 4. razred.....	12
Tablica 7. 5. razred.....	13
Tablica 8. 6 razred.....	15
Tablica 9. 7. razred.....	16
Tablica 10. 8 razred.....	18
Tablica 11. Vrsta programskog jezika s obzirom na kognitivni razvoj .....	22
Tablica 12. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 1. razred.....	25
Tablica 13. Prijedlog uvođenja nastavnog sadržaja Logički zadatci- Igraj se i uči .....	26
Tablica 14. Prijedlog uvođenja nastavnog sadržaja Moj prvi program u Scratchu.....	27
Tablica 15. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 2. razred.....	29
Tablica 16. Prijedlog uvođenja nastavnog sadržaja Ponovi ovo sve.....	30
Tablica 17. Prijedlog uvođenja nastavnog sadržaja Crtaj u Scratchu .....	31
Tablica 18. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 3. razred.....	32
Tablica 19. Prijedlog uvođenja nastavnog sadržaja Moje slobodno vrijeme.....	34
Tablica 20. Prijedlog uvođenja nastavnog sadržaja Odluke u Scratchu 2 .....	34
Tablica 21. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 4. razred.....	35
Tablica 22. Prijedlog uvođenja nastavnog sadržaja Micro:bit .....	37
Tablica 23. Prijedlog uvođenja nastavnog sadržaja Igre u Scratchu 2.....	37
Tablica 24. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 5. razred.....	38
Tablica 25. Prijedlog uvođenja nastavnog sadržaja Kreiraj algoritam.....	40
Tablica 26. Prijedlog uvođenja nastavnog sadržaja Crtaj geometrijske likove.....	40
Tablica 27. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 6. razred.....	41
Tablica 28. Prijedlog uvođenja nastavnog sadržaja Ulaz-obrada-izlaz.....	43
Tablica 29. Prijedlog uvođenja nastavnog sadržaja Potproblem u problemu .....	44
Tablica 30. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 7. razred.....	46
Tablica 31. Prijedlog uvođenja nastavnog sadržaja Sekvencijalno pretraživanje.....	47
Tablica 32. Prijedlog uvođenja nastavnog sadržaja Potprogrami- Python.....	48
Tablica 33. Prijedlozi nastavnih tema s brojem sati, opisom i ishodima za 8. razred.....	50
Tablica 34. Prijedlog uvođenja nastavnog sadržaja Sortiranje .....	52
Tablica 35. Prijedlog uvođenja nastavnog sadržaja Primjena rekurzije u Pythonu .....	52
Tablica 36. Projektna nastava u 4. razredu- Programi za množenje .....	54
Tablica 37. Projektna nastava u 5. razredu- Program- crtaj mnogokute .....	55
Tablica 38. Projektna nastava u 6. razredu- Trokut .....	56
Tablica 39. Projektna nastava u 7. razredu- Kalkulator .....	56
Tablica 40. Projektna nastava u 8. razredu- Matematika .....	57

## PRILOZI

1. prilog- Materijal za 1. razred

a) *Logički zadatci- igray se i uči*

-ilustracije za igru Memory



-slagalica



-listić za učenike

 **IGRAJ SE I UČI** 

1. OBOJI:  
DIJELOVE RAČUNALA

VOĆE I POVRĆE

2. PREKRIŽI ULJEZ.

3. POVEŽI ILUSTRACIJE PREDMETA SA SLIČNOM SVRHOM.



b) *Moj prvi program u Scratchu- Slijedi upute*

-PowerPoint prezentacija

 **SLIJEDI UPUTE**  
MOJ PRVI PROGRAM U SCRATCHU

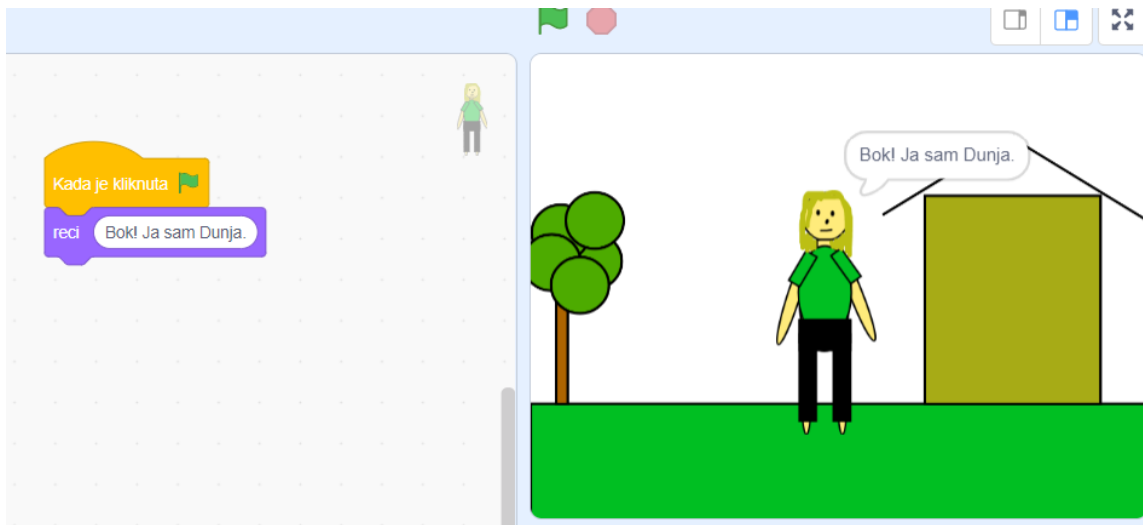
1. NA SREDINU PAPIRA NACRTAJ TROKUT.
2. DESNO OD TROKUTA NACRTAJ KRUG.
3. U KRUG UPIŠI SVOJE IME.

NACRTAJ U BILJEŽNICU

SLIJEDI KORAKE I IZNAĐI SVUJU PRVI PROGRAM.

1. NACRTAJ DVORIŠTE S KUĆOM.
2. NACRTAJ DJEVOJČICU/DJEČAKA.
3. DODAJ BLOK- KADA JE KLIKNUTA ZASTAVICA.
4. DODAJ BLOK- RECI.
5. PREDSTAVI SE U PROSTORU ZA TEKST.

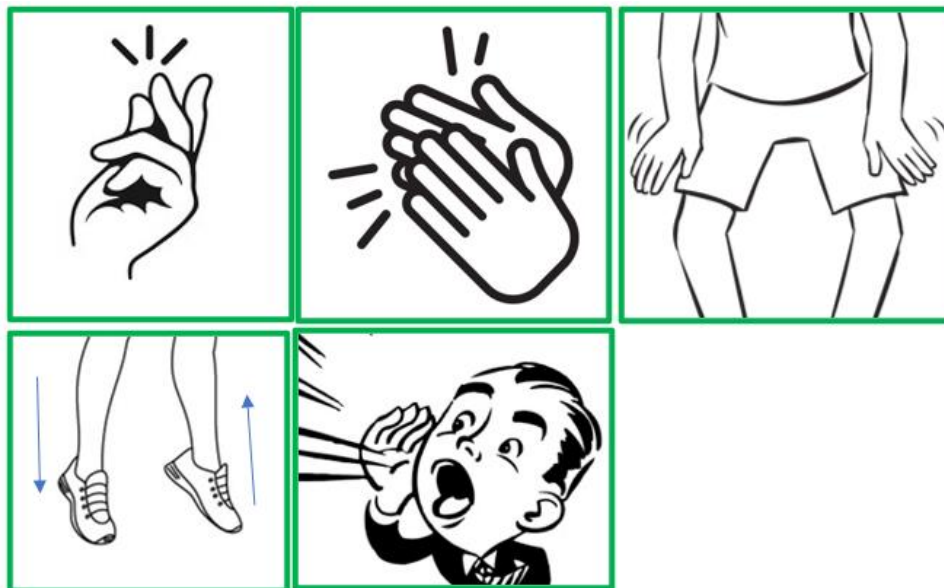
-rješenje zadatka u Scratch-u



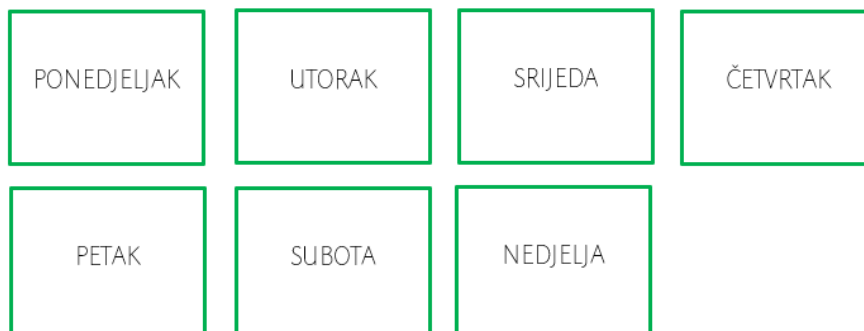
2. prilog- Materijal za 2. razred

a) *Ponovi ovo sve*

-kartice za prikaz pjesmice *Kad si sretan*



-kartice za prikaz dana u tjednu

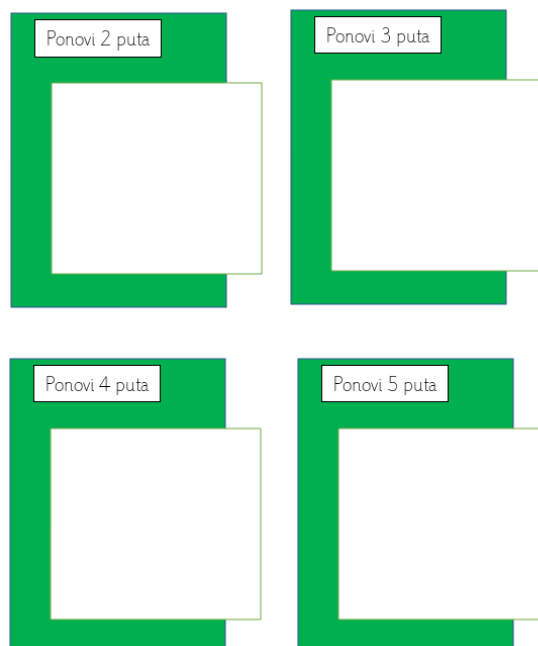




-kartice za prikaz godišnjih doba

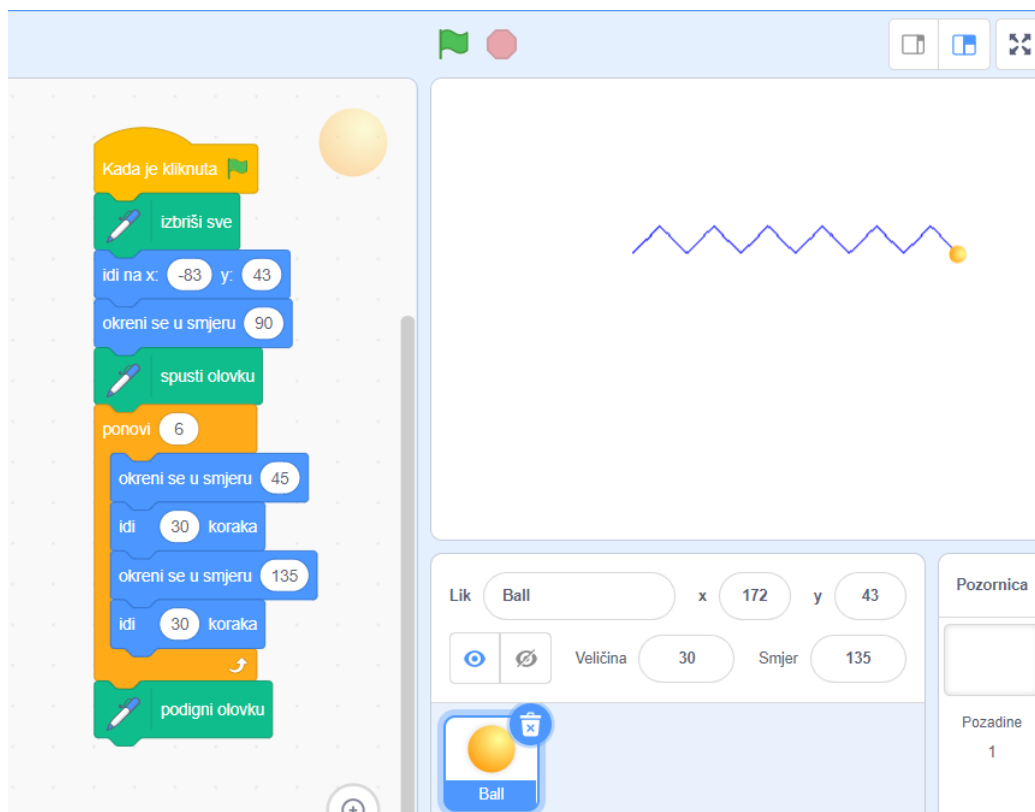
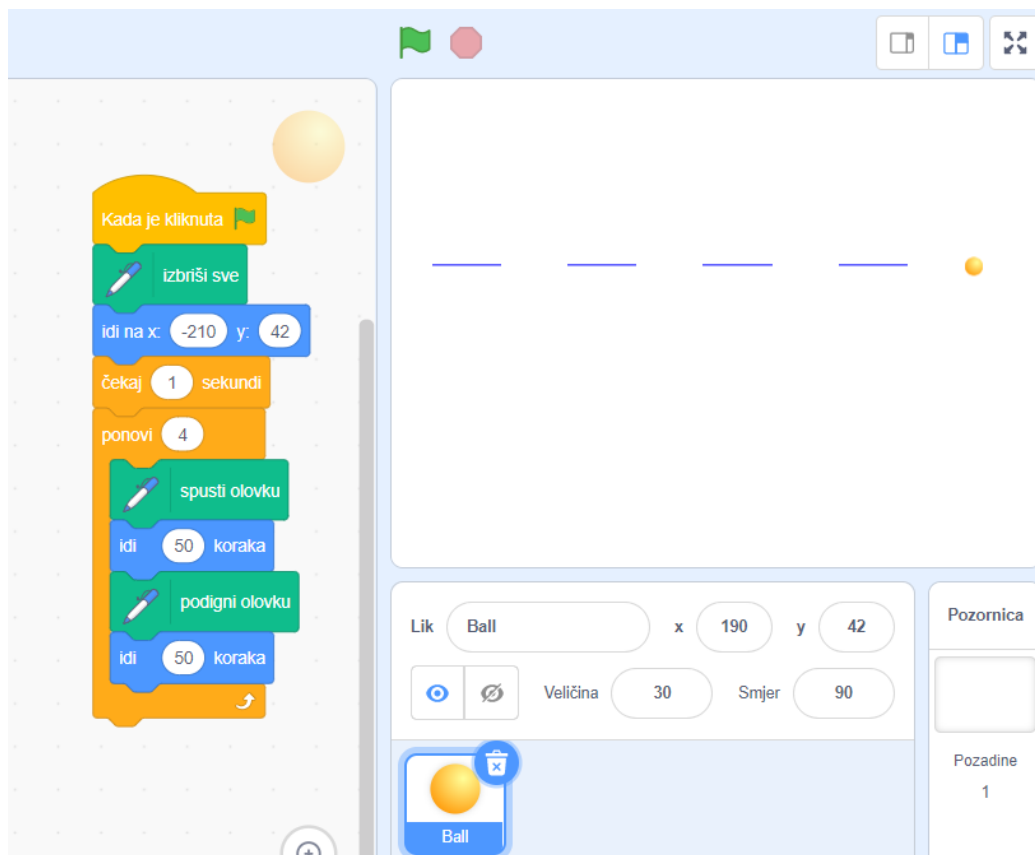


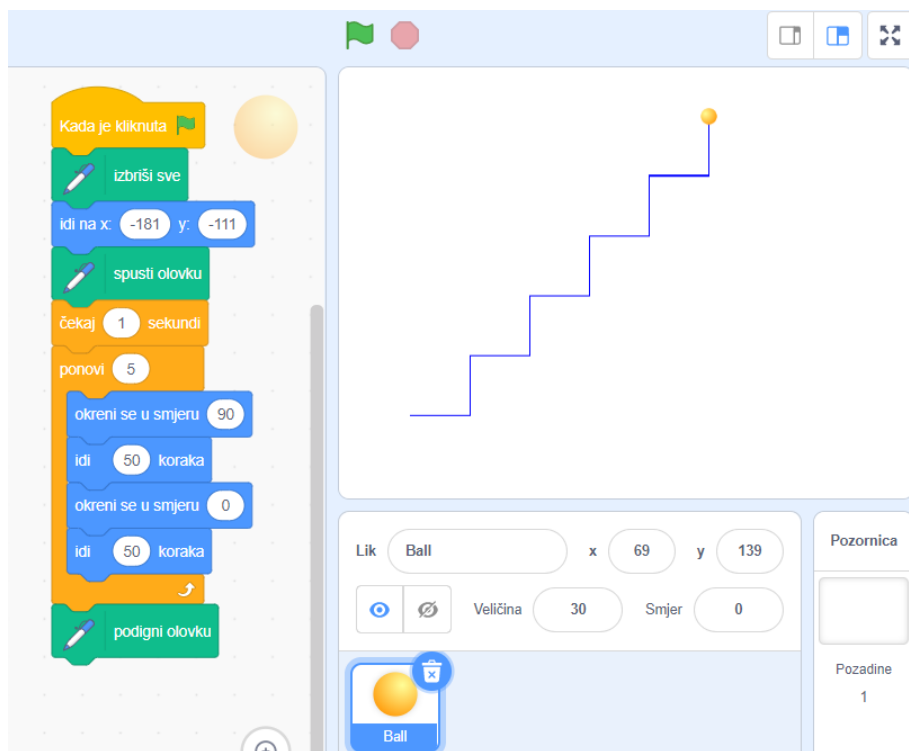
-kartice s blokovima za ponavljanje



b) *Crtaj u Sertachu*

-rješenja za crtanje isprekidane crte, cik-cak crte i u obliku stepenica





### 3. prilog- Materijal za 3. razred

#### a) *Moje slobodno vrijeme*

-tablica za razvrstavanje aktivnosti

#### Moje slobodno vrijeme

pripremanje stola, druženje s obitelji, gledanje televizora, postavljanje stola, briga o kućnim ljubimcima, odlazak u školu, igranje s prijateljima, pisanje domaće zadaće, pomaganje roditeljima u kućanskim poslovima

OBAVEZE	SLOBODNO VRIJEME

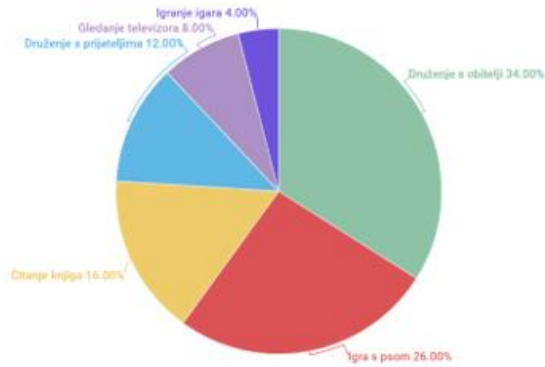
*najčešće radim
*najrjeđe radim

-slobodno vrijeme Ana i Ivan

Slobodno vrijeme- Ana

### Aktivnosti u slobodno vrijeme

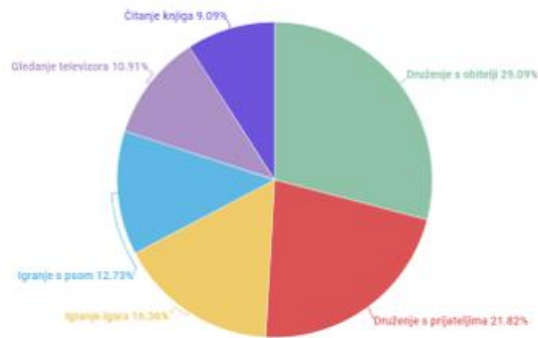
	A	B
1		postotak
2	Druženje s obitelji	34
3	Igra s psom	26
4	Čitanje knjiga	16
5	Druženje s prijateljima	12
6	Gledanje televizora	8
7	Igranje igara	4



Slobodno vrijeme- Ivan

### Aktivnosti u slobodno vrijeme

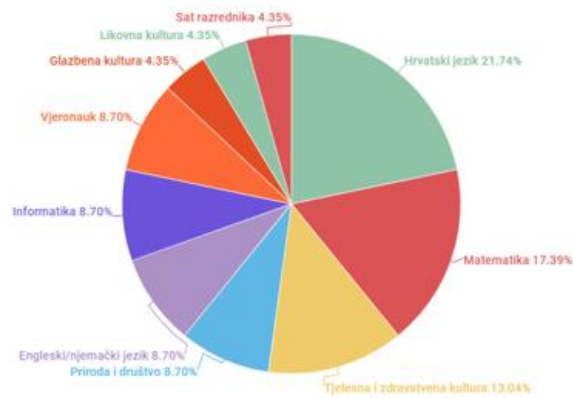
1		postotak
2	Druženje s obitelji	32
3	Druženje s prijateljima	24
4	Igranje igara	18
5	Igranje s psom	14
6	Gledanje televizora	12
7	Čitanje knjiga	10



-broj sati

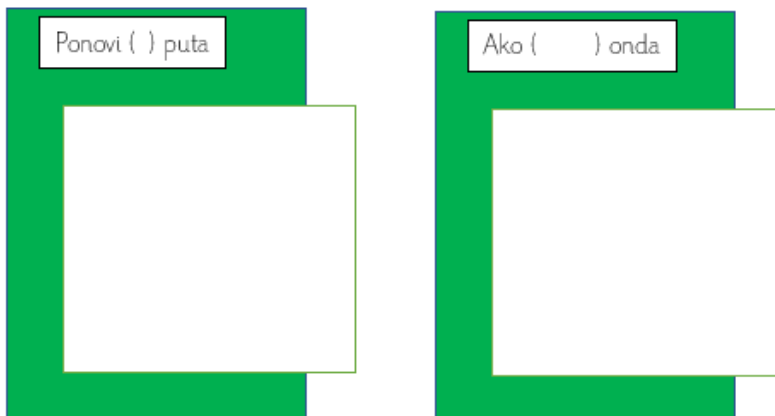
### Broj sati u tjednu-predmeti

1		Broj sati tjedno
2	Hrvatski jezik	5
3	Matematika	4
4	Tjelesna i zdravstvena kultura	3
5	Priroda i društvo	2
6	Engleski/njemački jezik	2
7	Informatika	2
8	Vjeronauk	2
9	Glazbena kultura	1
10	Likovna kultura	1
11	Sat razrednika	1

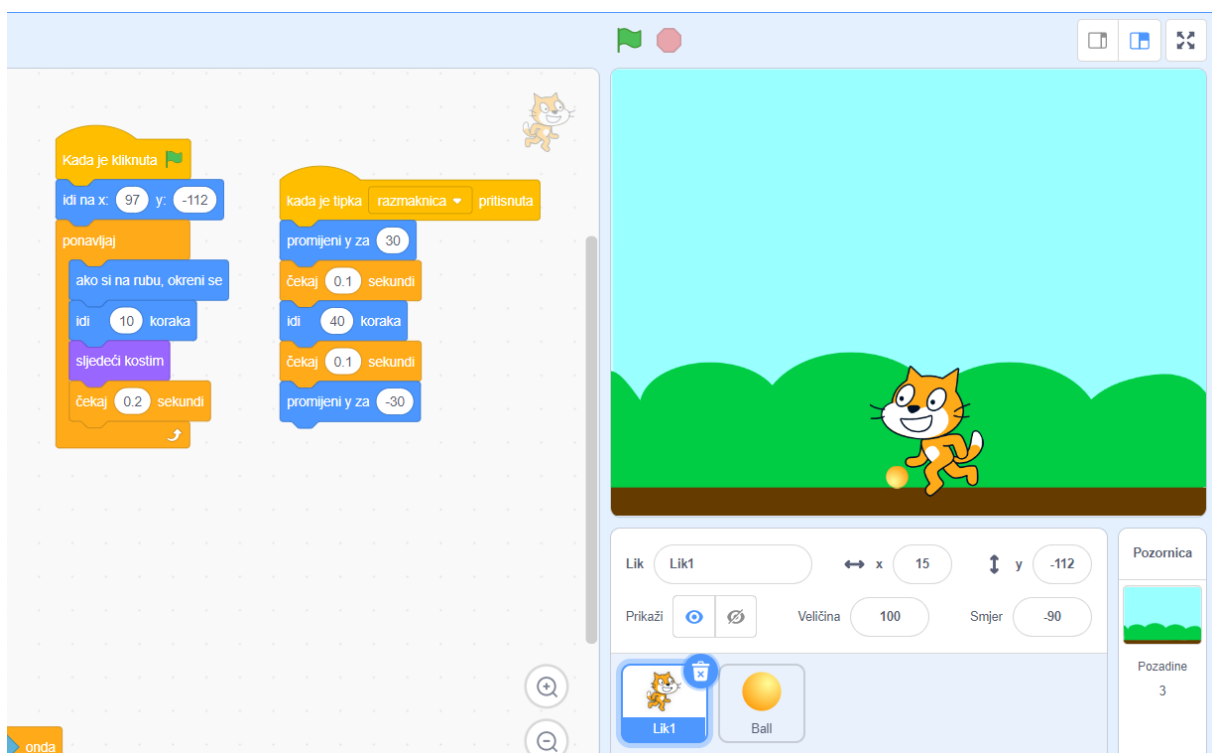


## b) Odluke u Scratchu

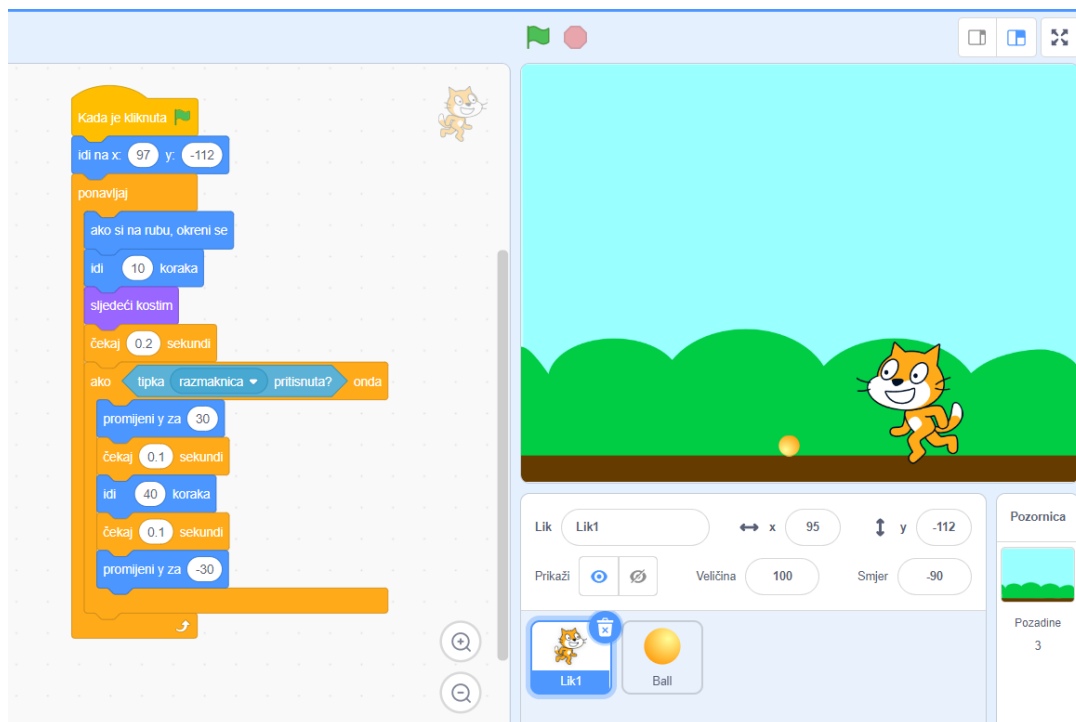
-kartice za kretanje



-Scratch program (prije ispravka)



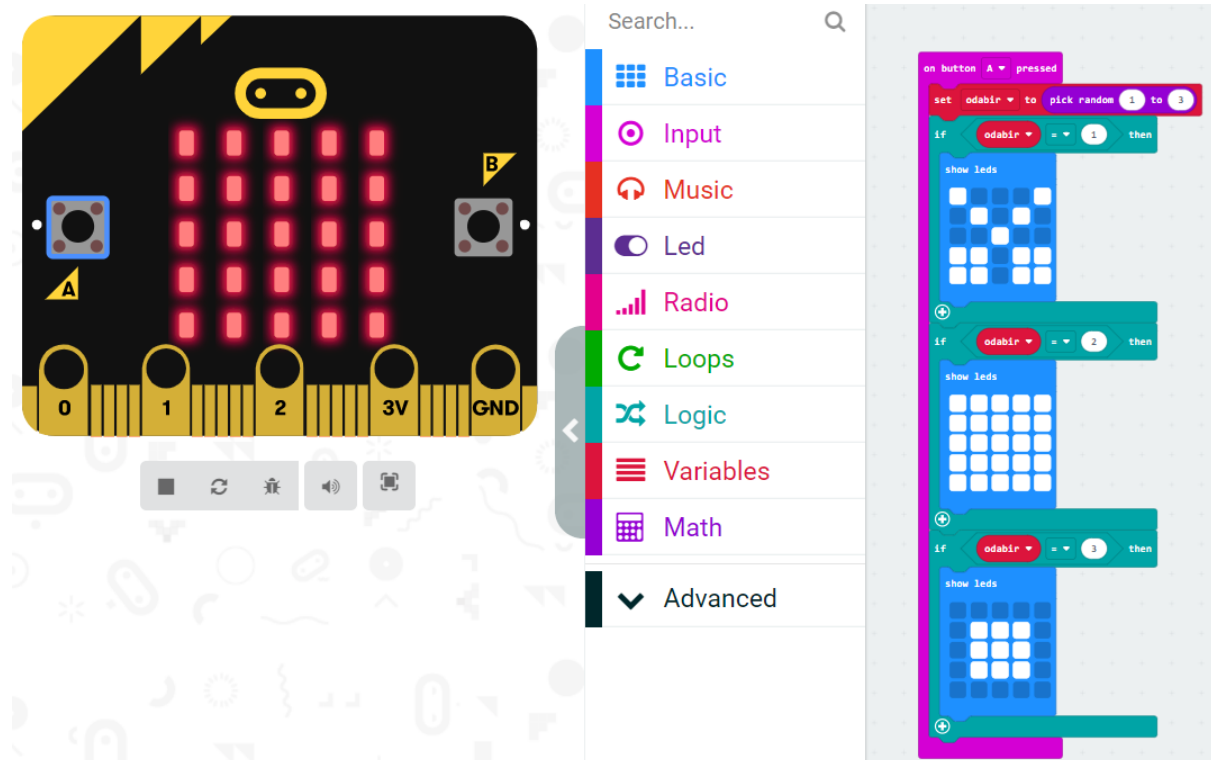
-Scratch program (poslije ispravka)



4. prilog- Materijal za 4. razred

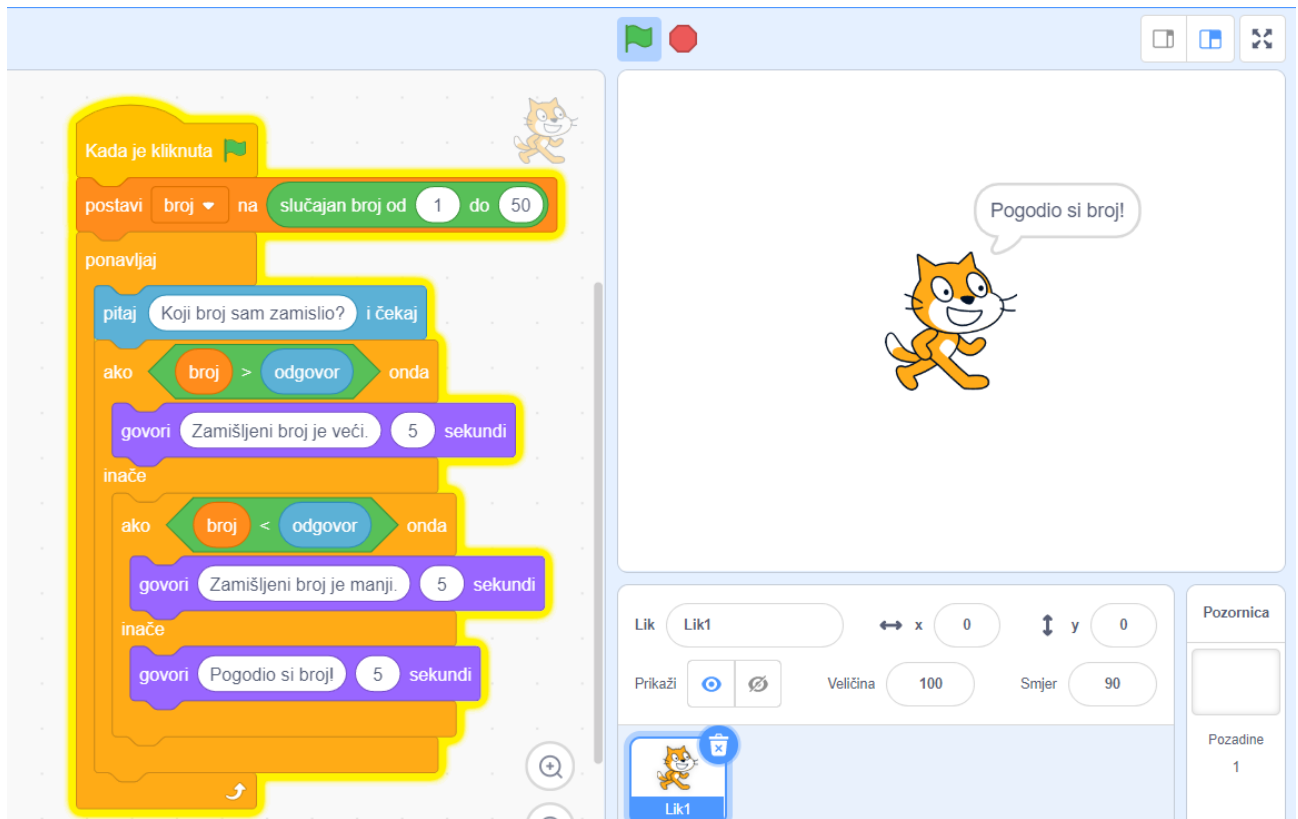
a) *Micro:bit*

-rješenje za igru *Kamen-škare-papir*



### c) Igre u Scratchu 2

-rješenje za igru *Pogodi broj*



### 5. prilog: Materijal za 5. razred

#### a) Kreiraj algoritam

-zadatci za učenike

ZADATCI:

#### 1. Algoritam spremanja kreveta

Ivan se ujutro probudio i kasnio u školu. Kako bi mu olakšali osmislićemo algoritam za spremanje kreveta. Imaj na umu da se na krevetu nalaze: jastuk, plahta i pokrivač.

#### 2. Algoritam izrade muffina

Janja slavi rođendan. Na proslavu rođendana dolazi 11 prijatelja. Janja želi ispeći za svakog po jedan muffin. Izradi algoritam izrade muffina.

Pomoć: potrebno je izmiješati sastojke iz vrećice s jajetom i uljem, miješati 2 minute, rasporediti u zasebne kalupe i posuti čokoladnim mrvicama.

### 3. Algoritam dodjeljivanje ocjena

Učenici su iz Geografije pisali ispit. Učiteljica im je na ispit napisala bodove, a na ploču napisala bodovnu ljestvicu te pridodala ocjene. Učenici trebaju izraditi algoritam koji će ispisivati ocjenu za dobiveni broj bodova.

Ocjena	Broj bodova
5	18-20
4	16-17
3	13-15
2	10-12
1	0-9

-rješenja

1.

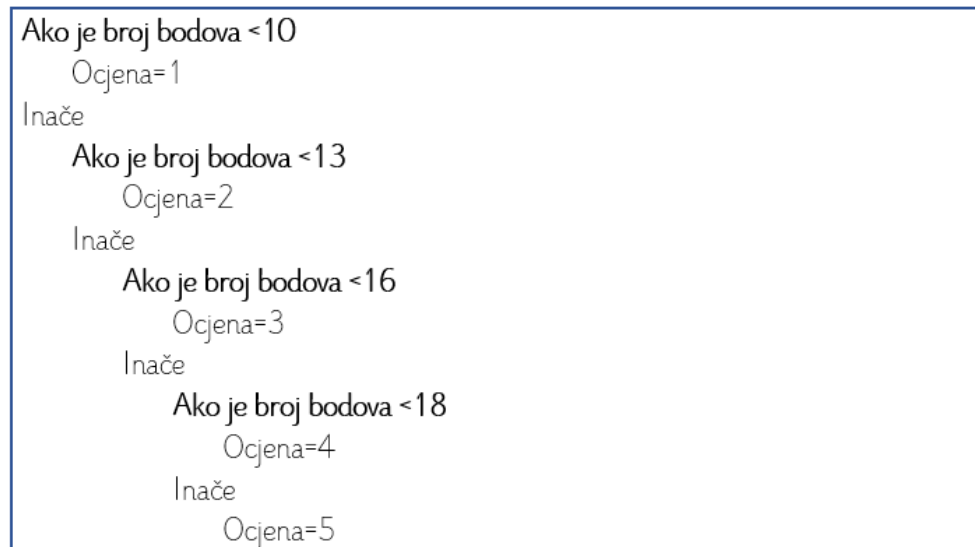
Makni s kreveta jastuk i pokrivač  
Stegni plahtu u rubove kreveta  
Vrati jastuk na sredinu uzglavlje kreveta  
Raširi pokrivač po krevetu

2.

Uzmi posudu za miješanje  
Usipaj sadržaj vrećice u posudu  
Razbij jaje u posudu  
Dodaj ulje  
Uzmi električnu miješalicu  
Miješaj sastojke 2 minute  
Uzmi žlicu  
Uzmi kalupe za muffine  
**Stavi papirnatu košaricu u kalup (12x)**  
**Žlicom stavi smjesu u kalup (12x)**  
Uzmi čokoladne mrvice  
**Posipaj muffin čokoladnim mrvicama (12x)**  
Stavi muffine u pećnicu

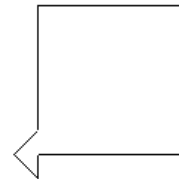


3.



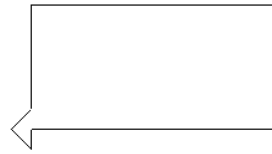
b) *Crtaj geometrijske likove*

-kvadrat bez petlje



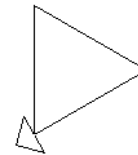
```
FD 100
RT 90
FD 100
RT 90
FD 100
RT 90
FD 100
```

-pravokutnik bez petlje



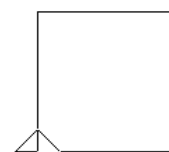
```
fd 100  
rt 90  
fd 200  
rt 90  
fd 100  
rt 90  
fd 200
```

-jednakostranični trokut bez petlje



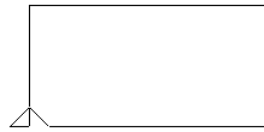
```
fd 100  
rt 360/3  
fd 100  
rt 360/3  
fd 100
```

-kvadrat s petljom



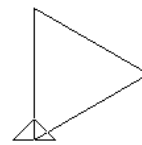
```
repeat 4 [ FD 100 RT 90 ]
```

-pravokutnik s petljom



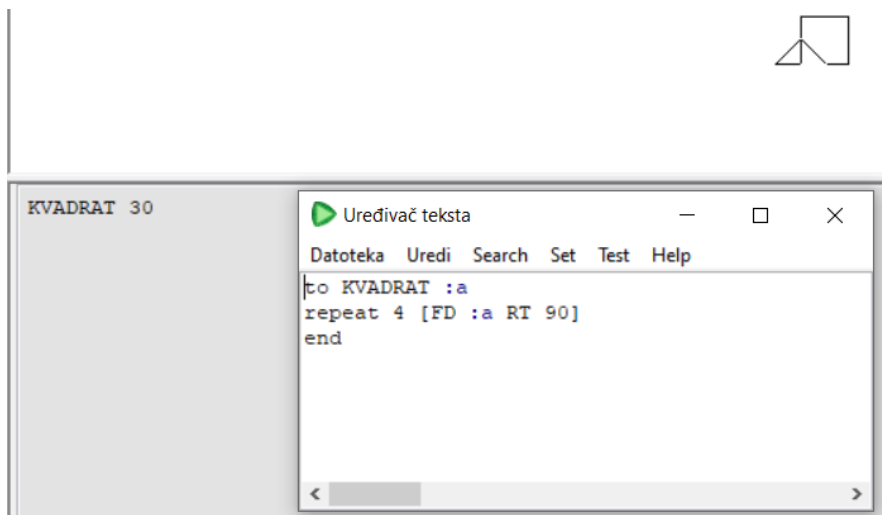
```
repeat 2 [ FD 100 RT 90 FD 200 RT 90]
```

-jednakostranični trokut s petljom



```
repeat 3 [ FD 100 RT 360/3]
```

-program za crtanje kvadrata s ulaznom vrijednošću



The screenshot shows a turtle graphics environment. On the left, a window titled "KVADRAT 30" is visible. On the right, a text editor window titled "Uređivač teksta" is open, displaying the following code:

```
to KVADRAT :a  
repeat 4 [FD :a RT 90]  
end
```

-program za crtanje kvadrata s ulaznom vrijednošću



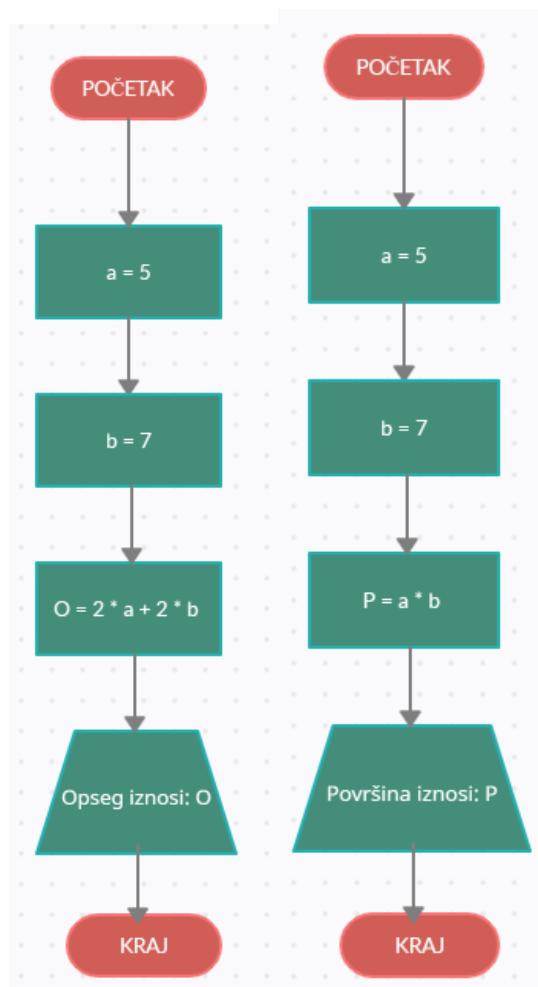
```
TROKUT 40
```

```
Uređivač teksta
Datoteka Uredi Search Set Test Help
to TROKUT :a
repeat 3 [FD :a RT 360/3]
end
```

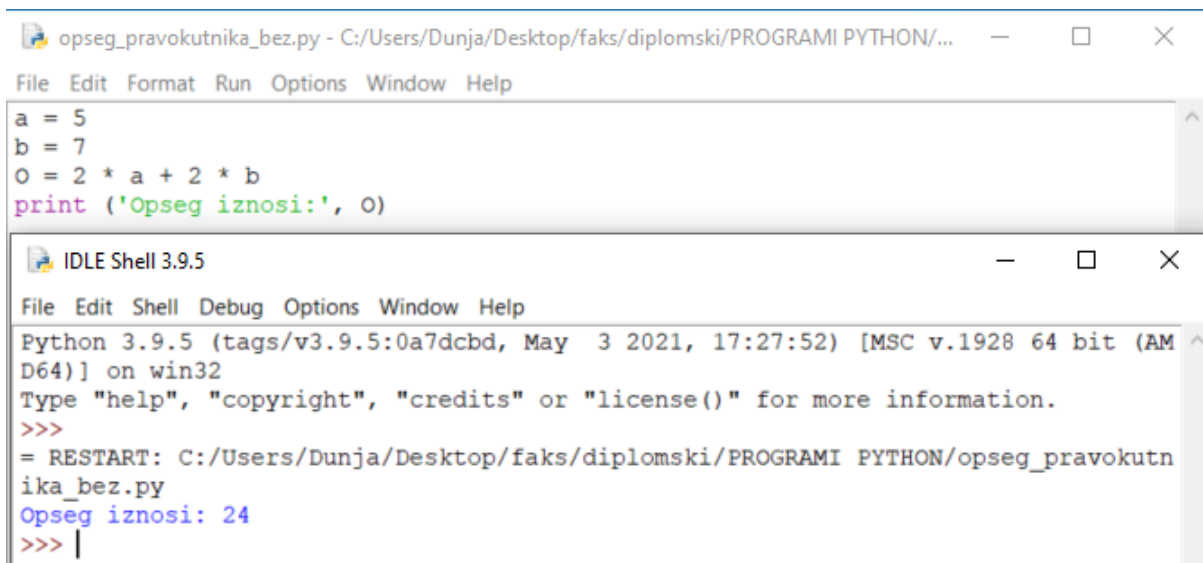
6. Prilog: Materijal za 6. razred

a) *Ulaz-obrađ-izlaz*

-dijagrami toka za izračun opsega i površine



-program za izračun opsega



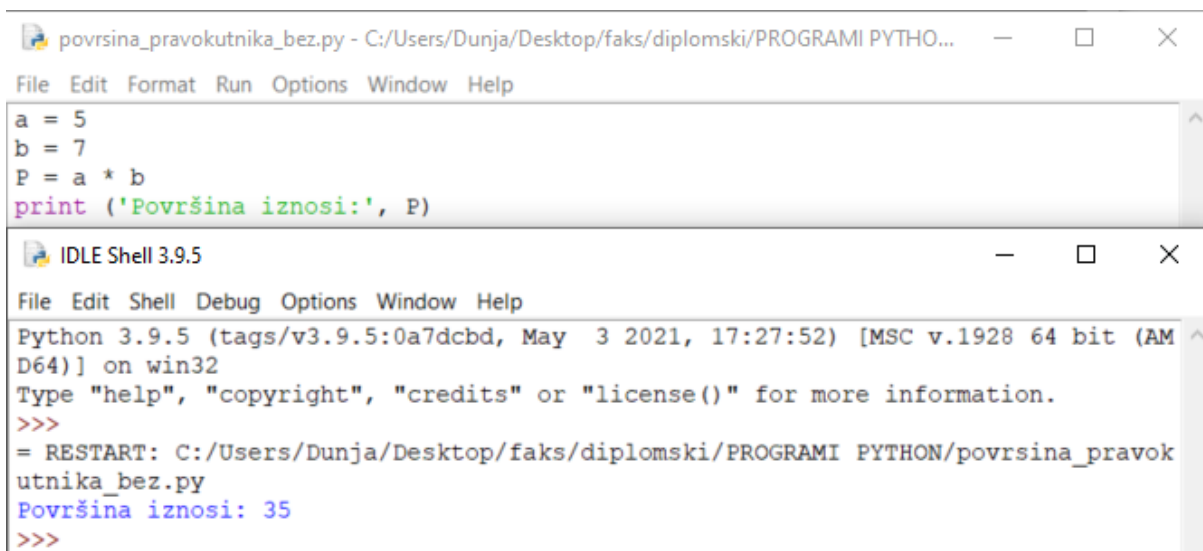
The screenshot shows a Python IDE window titled 'opseg\_pravokutnika\_bez.py'. The code in the editor is:

```
a = 5
b = 7
O = 2 * a + 2 * b
print ('Opseg iznosi:', O)
```

Below the editor is the 'IDLE Shell 3.9.5' window. It shows the execution output:

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/opseg_pravokutnika_bez.py
Opseg iznosi: 24
>>> |
```

-program za izračun površine



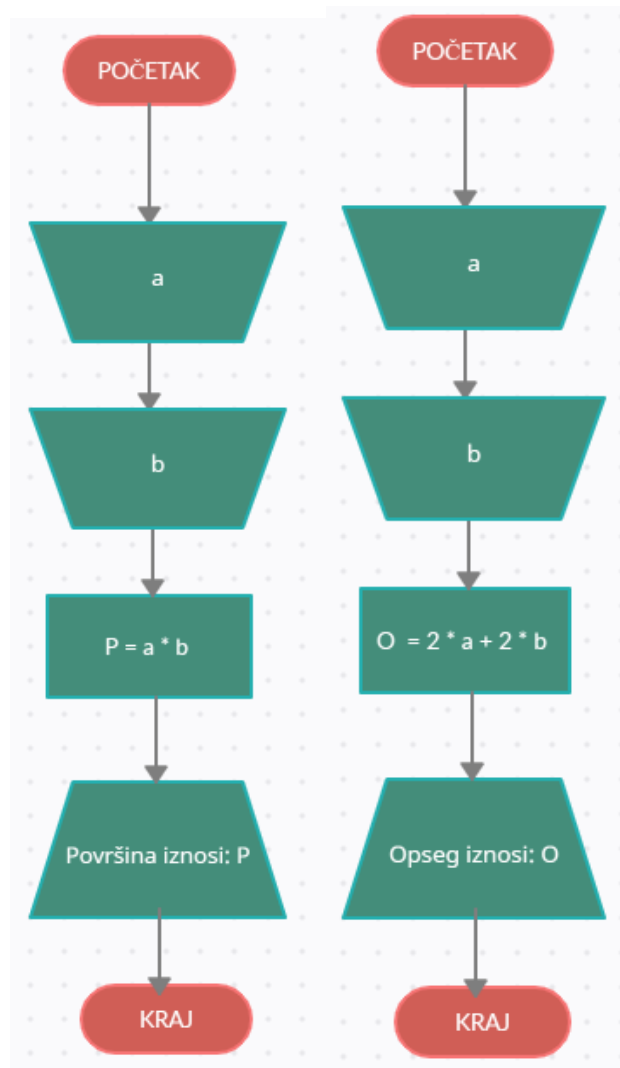
The screenshot shows a Python IDE window titled 'povrsina\_pravokutnika\_bez.py'. The code in the editor is:

```
a = 5
b = 7
P = a * b
print ('Površina iznosi:', P)
```

Below the editor is the 'IDLE Shell 3.9.5' window. It shows the execution output:

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/povrsina_pravokutnika_bez.py
Površina iznosi: 35
>>>
```

-dijagrami toka za izračun površine i opsega s unosom ulaznih vrijednosti



-program za izračun površine s unosom ulaznih vrijednosti

```
površina_pravokutnika_sa_ulaz.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYT...
File Edit Format Run Options Window Help
a = int (input ('Unesi duljinu stranice a:'))
b = int (input ('Unesi duljinu stranice b:'))
P = a * b
print ('Površina iznosi:', P)

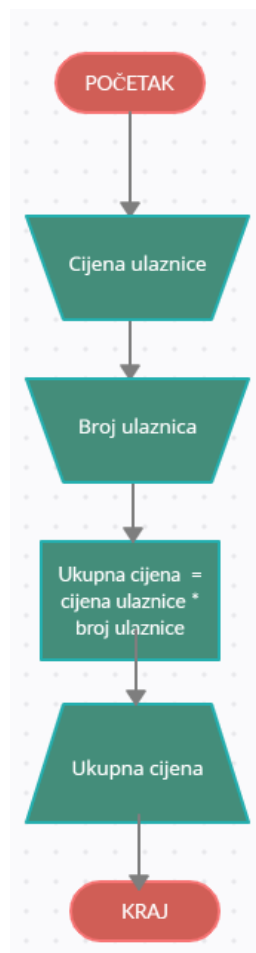
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/površina_pravokutnika_sa_ulaz.py
Unesi duljinu stranice a:10
Unesi duljinu stranice b:15
Površina iznosi: 150
>>> |
```

-program za izračun opsega s unosom ulaznih vrijednosti

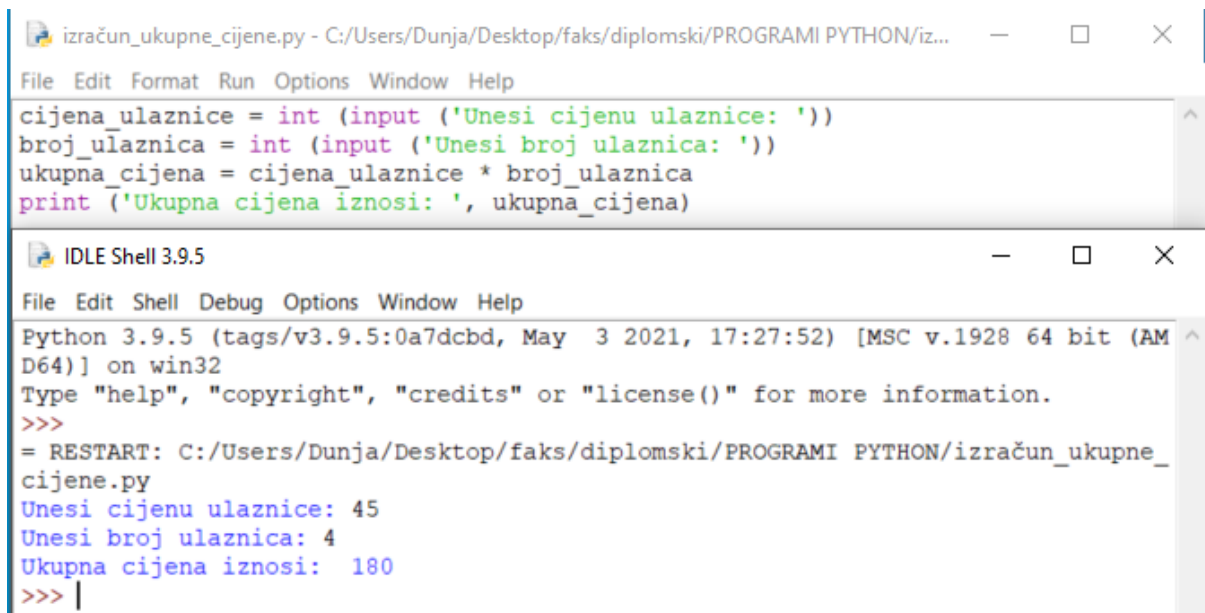
```
opseg_pravokutnika_sa_ulaz.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTH...
File Edit Format Run Options Window Help
a = int (input ('Unesi duljinu stranice a:'))
b = int (input ('Unesi duljinu stranice b:'))
O = 2 * a + 2 * b
print ('Opseg iznosi:', O)

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PYTHON/opseg_pravokutn
ika_sa_ulaz.py
Unesi duljinu stranice a:10
Unesi duljinu stranice b:15
Opseg iznosi: 50
>>> |
```

-dijagram toka za izračun ukupne cijene ulaznica za kino



-program za izračun ukupne cijene ulaznica za kino



The screenshot shows two windows from the Python IDE. The top window is a text editor titled 'izračun\_ukupne\_cijene.py' containing the following Python code:

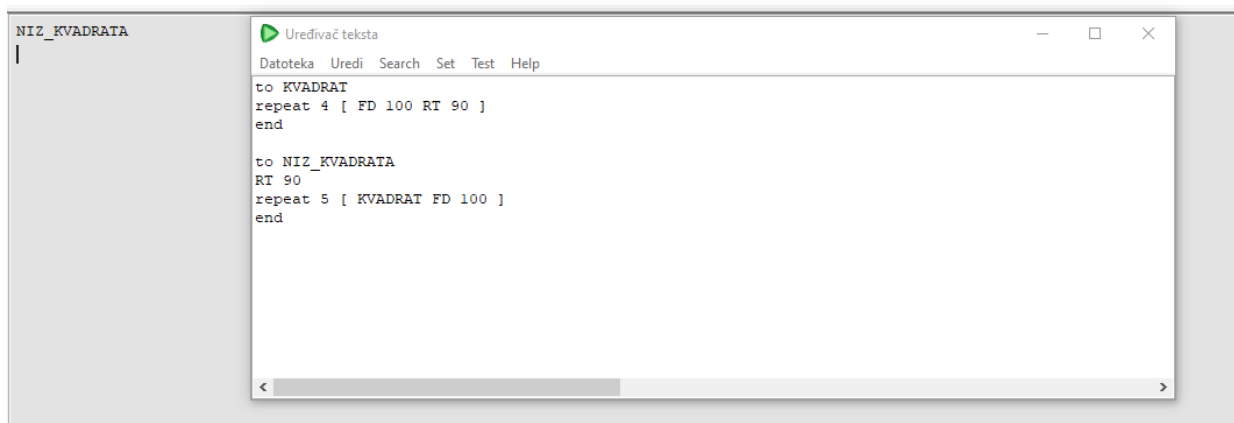
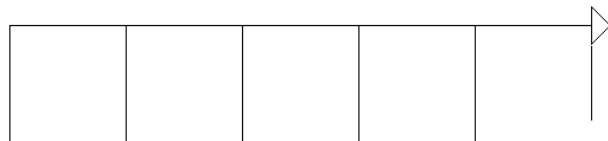
```
cijena_ulaznice = int (input ('Unesi cijenu ulaznice: '))
broj_ulaznica = int (input ('Unesi broj ulaznica: '))
ukupna_cijena = cijena_ulaznice * broj_ulaznica
print ('Ukupna cijena iznosi: ', ukupna_cijena)
```

The bottom window is the 'IDLE Shell 3.9.5' showing the execution of the script. The output is:

```
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/izračun_ukupne_cijene.py
Unesi cijenu ulaznice: 45
Unesi broj ulaznica: 4
Ukupna cijena iznosi: 180
>>> |
```

b) *Potproblem u problemu*

-niz kvadrata bez razmaka



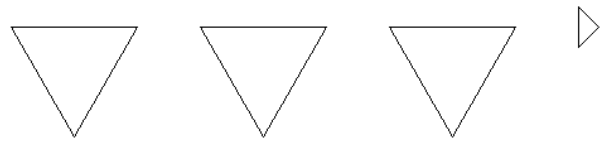
The screenshot shows a text editor window titled 'Uređivač teksta' with the following code:

```
to KVADRAT
repeat 4 [ FD 100 RT 90 ]
end

to NIZ_KVADRATA
RT 90
repeat 5 [ KVADRAT FD 100 ]
end
```



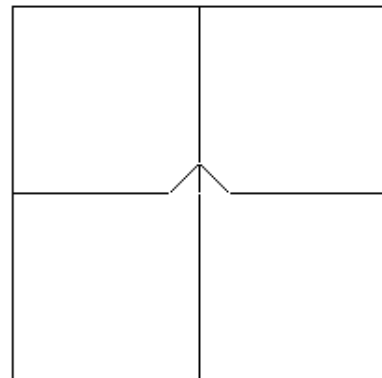
-niz jednakostaničnih trokuta s razmakom



```
NIZ_TROKUTA
Uređivač teksta
Datoteka Uredi Search Set Test Help
to TROKUT
repeat 3 [ FD 100 RT 120 ]
end

to NIZ_TROKUTA
RT 90
repeat 3 [ PD TROKUT PU FD 150 ]
end
```

-4 kvadrata koji čine jedan veći kvadrat



```
VELIKI_KVADRAT
Uređivač teksta
Datoteka Uredi Search Set Test Help
to KVADRAT
repeat 4 [ FD 100 RT 90 ]
end

to VELIKI_KVADRAT
repeat 4 [ RT 90 KVADRAT ]
end
```

## 7. prilog: Materijal za 7. razred

### a) Sekvencijalno pretraživanje

-program- godišnja doba

```
godisnja_doba.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/godisnja_...
File Edit Format Run Options Window Help
godisnja_doba = input ('Upiši godišnje doba: ').split()
print ('Lista godišnjih doba:', godisnja_doba)
trazeno_godisnje_doba = input ('Pretraži godišnje doba: ')
print ('Pozicija unutar liste', godisnja_doba.index(trazeno_godisnje_doba))

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/godisnja_doba.py
Upiši godišnje doba: ljetoproljece
Lista godišnjih doba: ['ljetoproljece']
Pretraži godišnje doba: ljetoproljece
Pozicija unutar liste 0
>>> |

godisnja_doba.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/godisnja_...
File Edit Format Run Options Window Help
godisnja_doba = input ('Upiši godišnje doba: ').split()
print ('Lista godišnjih doba:', godisnja_doba)
trazeno_godisnje_doba = input ('Pretraži godišnje doba: ')
print ('Pozicija unutar liste', godisnja_doba.index(trazeno_godisnje_doba))

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/godisnja_doba.py
Upiši godišnje doba: ljetoproljece
Lista godišnjih doba: ['ljetoproljece']
Pretraži godišnje doba: jesen
Pozicija unutar liste 1
>>> |
```

```
godisnja_doba.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/godisnja_...
File Edit Format Run Options Window Help
godisnja_doba = input ('Upiši godišnje doba: ').split()
print ('Lista godišnjih doba:', godisnja_doba)
trazeno_godisnje_doba = input ('Pretraži godišnje doba: ')
print ('Pozicija unutar liste', godisnja_doba.index(trazeno_godisnje_doba))

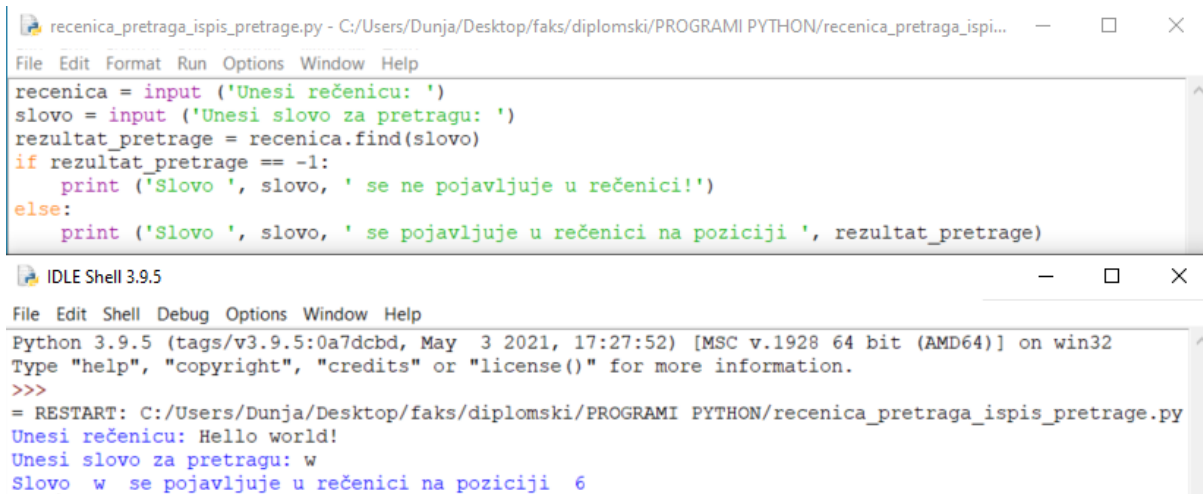
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/godisnja_doba.py
Upiši godišnje doba: ljeto jesen zima proljece
Lista godišnjih doba: ['ljeto', 'jesen', 'zima', 'proljece']
Pretraži godišnje doba: ponedjeljak
Traceback (most recent call last):
  File "C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/godisnja_doba.py",
    line 4, in <module>
      print ('Pozicija unutar liste', godisnja_doba.index(trazeno_godisnje_doba))
ValueError: 'ponedjeljak' is not in list
>>> |
```

-program- pretraga slova u rečenici

```
recenica_pretraga.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/receni...
File Edit Format Run Options Window Help
recenica = input ('Unesi rečenicu: ')
slovo = input ('Unesi slovo za pretragu: ')
rezultat_pretrage = recenica.find(slovo)
print ('Pozicija slova: ', rezultat_pretrage)

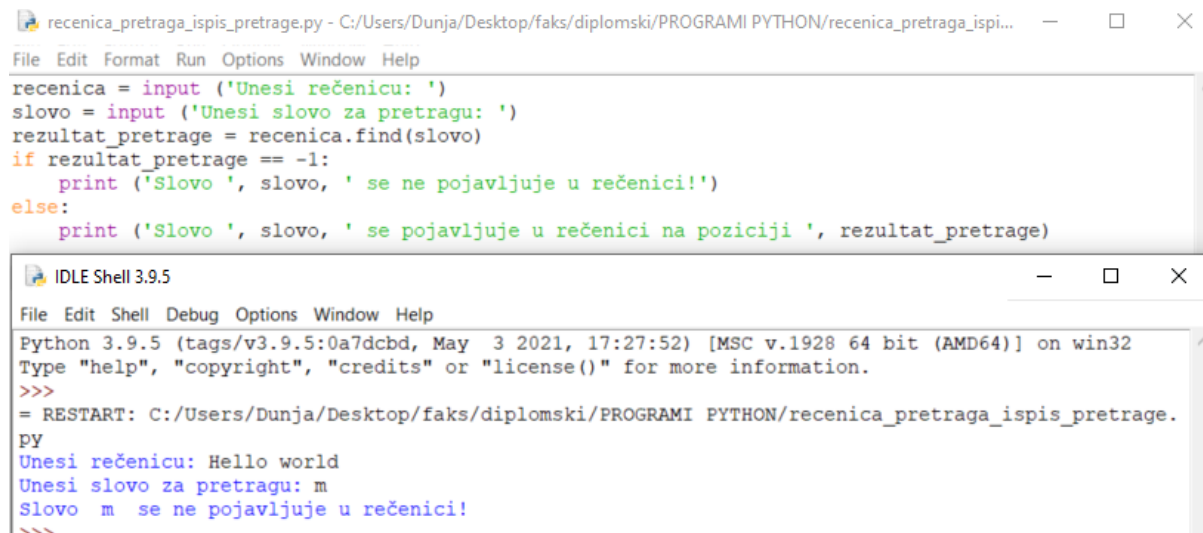
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/recenica_pretraga.py
Unesi rečenicu: Hello world!
Unesi slovo za pretragu: l
Pozicija slova: 2
```

## -program- nadogradnja programa



```
recenica_pretraga_ispis_pretrage.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/recenica_pretraga_ispis...
File Edit Format Run Options Window Help
recenica = input ('Unesi rečenicu: ')
slovo = input ('Unesi slovo za pretragu: ')
rezultat_pretrage = recenica.find(slovo)
if rezultat_pretrage == -1:
    print ('Slovo ', slovo, ' se ne pojavljuje u rečenici!')
else:
    print ('Slovo ', slovo, ' se pojavljuje u rečenici na poziciji ', rezultat_pretrage)

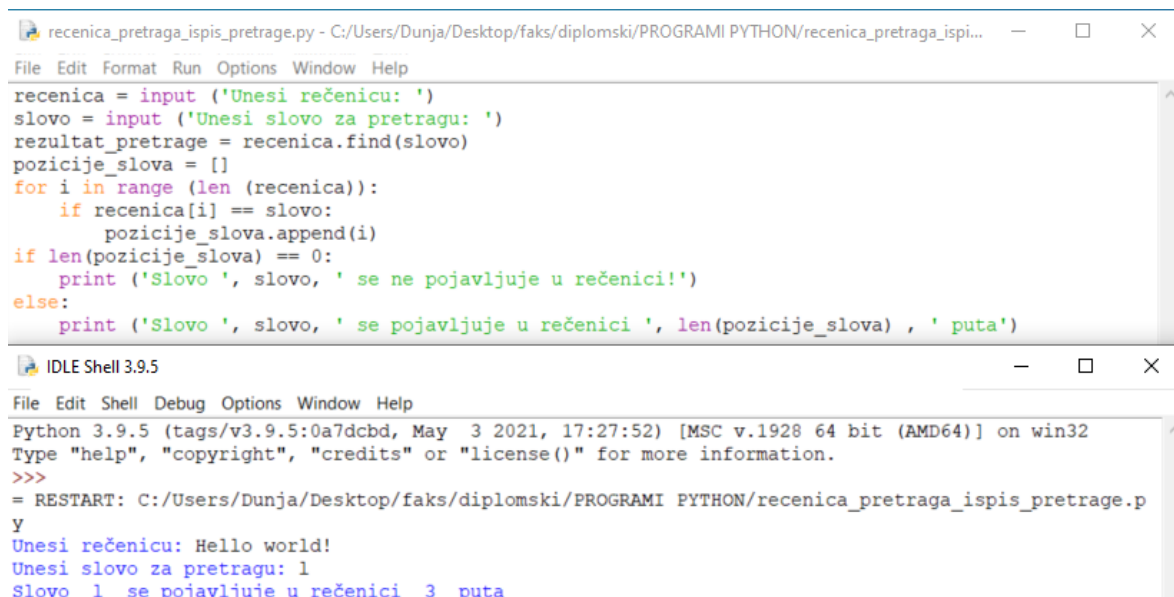
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/recenica_pretraga_ispis_pretrage.py
Unesi rečenicu: Hello world!
Unesi slovo za pretragu: w
Slovo w se pojavljuje u rečenici na poziciji 6
```



```
recenica_pretraga_ispis_pretrage.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/recenica_pretraga_ispis...
File Edit Format Run Options Window Help
recenica = input ('Unesi rečenicu: ')
slovo = input ('Unesi slovo za pretragu: ')
rezultat_pretrage = recenica.find(slovo)
if rezultat_pretrage == -1:
    print ('Slovo ', slovo, ' se ne pojavljuje u rečenici!')
else:
    print ('Slovo ', slovo, ' se pojavljuje u rečenici na poziciji ', rezultat_pretrage)

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/recenica_pretraga_ispis_pretrage.py
Unesi rečenicu: Hello world
Unesi slovo za pretragu: m
Slovo m se ne pojavljuje u rečenici!
```

## -program- nadogradnja programa 2



```
recenica_pretraga_ispis_pretrage.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/recenica_pretraga_ispis...
File Edit Format Run Options Window Help
recenica = input ('Unesi rečenicu: ')
slovo = input ('Unesi slovo za pretragu: ')
rezultat_pretrage = recenica.find(slovo)
pozicije_slova = []
for i in range (len (recenica)):
    if recenica[i] == slovo:
        pozicije_slova.append(i)
if len(pozicije_slova) == 0:
    print ('Slovo ', slovo, ' se ne pojavljuje u rečenici!')
else:
    print ('Slovo ', slovo, ' se pojavljuje u rečenici ', len(pozicije_slova) , ' puta')

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/recenica_pretraga_ispis_pretrage.py
Unesi rečenicu: Hello world!
Unesi slovo za pretragu: l
Slovo l se pojavljuje u rečenici 3 puta
```

```
recenica_pretraga_ispis_pretrage.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/recenica_pretraga_ispis...
File Edit Format Run Options Window Help
recenica = input ('Unesi rečenicu: ')
slovo = input ('Unesi slovo za pretragu: ')
rezultat_pretrage = recenica.find(slovo)
pozicije_slova = []
for i in range (len (recenica)):
    if recenica[i] == slovo:
        pozicije_slova.append(i)
if len(pozicije_slova) == 0:
    print ('Slovo ', slovo, ' se ne pojavljuje u rečenici!')
else:
    print ('Slovo ', slovo, ' se pojavljuje u rečenici ', len(pozicije_slova) , ' puta')
```

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/recenica_pretraga_ispis_pretrage.p
y
Unesi rečenicu: Hello world!
Unesi slovo za pretragu: m
Slovo m se ne pojavljuje u rečenici!
```

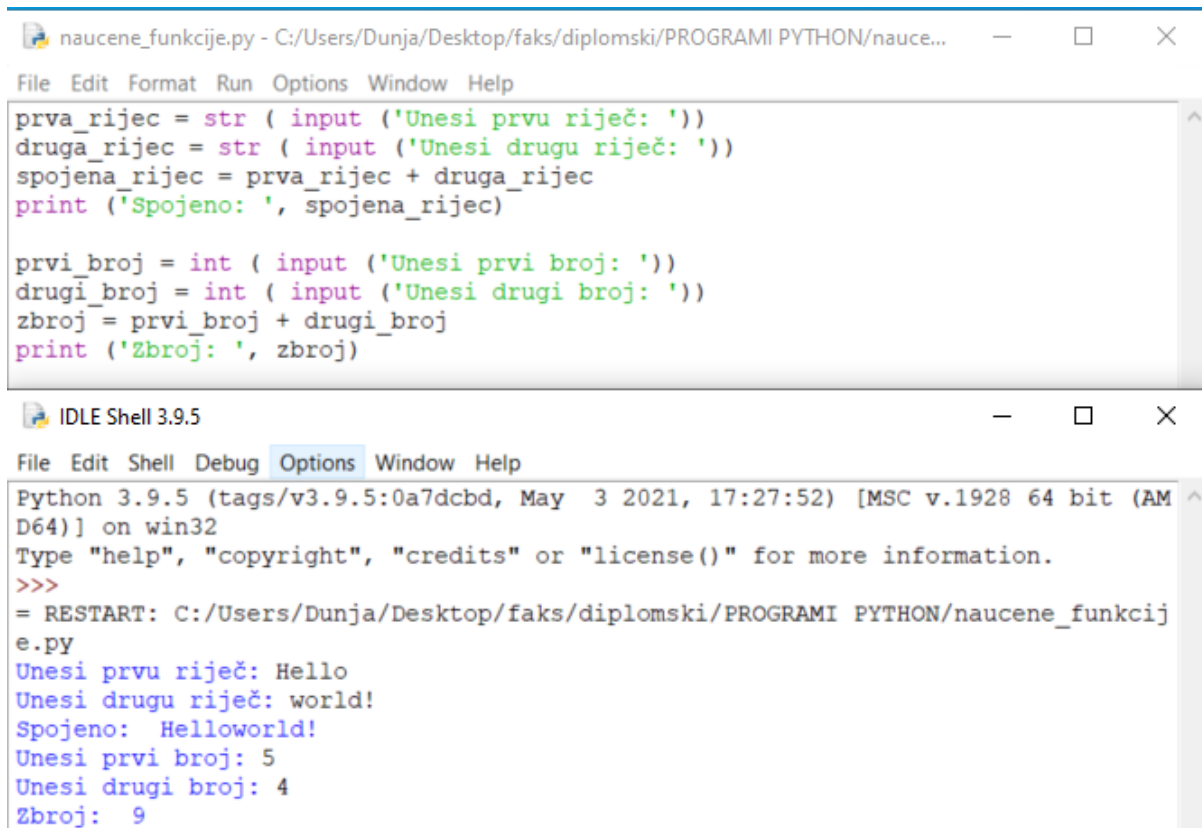
-program za ispis liste parnih brojeva

```
lista_parnih_brojeva.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/lista_...
File Edit Format Run Options Window Help
uneseni_brojevi = input ('Unesi brojeve: ').split()
pretvoreni_brojevi = []
parni_brojevi = []
for i in range (len (uneseni_brojevi)):
    pretvoreni_brojevi.append(int (uneseni_brojevi[i]))
for i in range (len (pretvoreni_brojevi)):
    if pretvoreni_brojevi[i] % 2 == 0:
        parni_brojevi.append(pretvoreni_brojevi[i])
print ('Uneseni parni brojevi su: ', parni_brojevi)
```

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/lista_parnih_br
ojeva.py
Unesi brojeve: 1 2 3 4 5 6 7 8 9 10
Uneseni parni brojevi su: [2, 4, 6, 8, 10]
```

b) Potprogrami u programu

- naučene funkcije

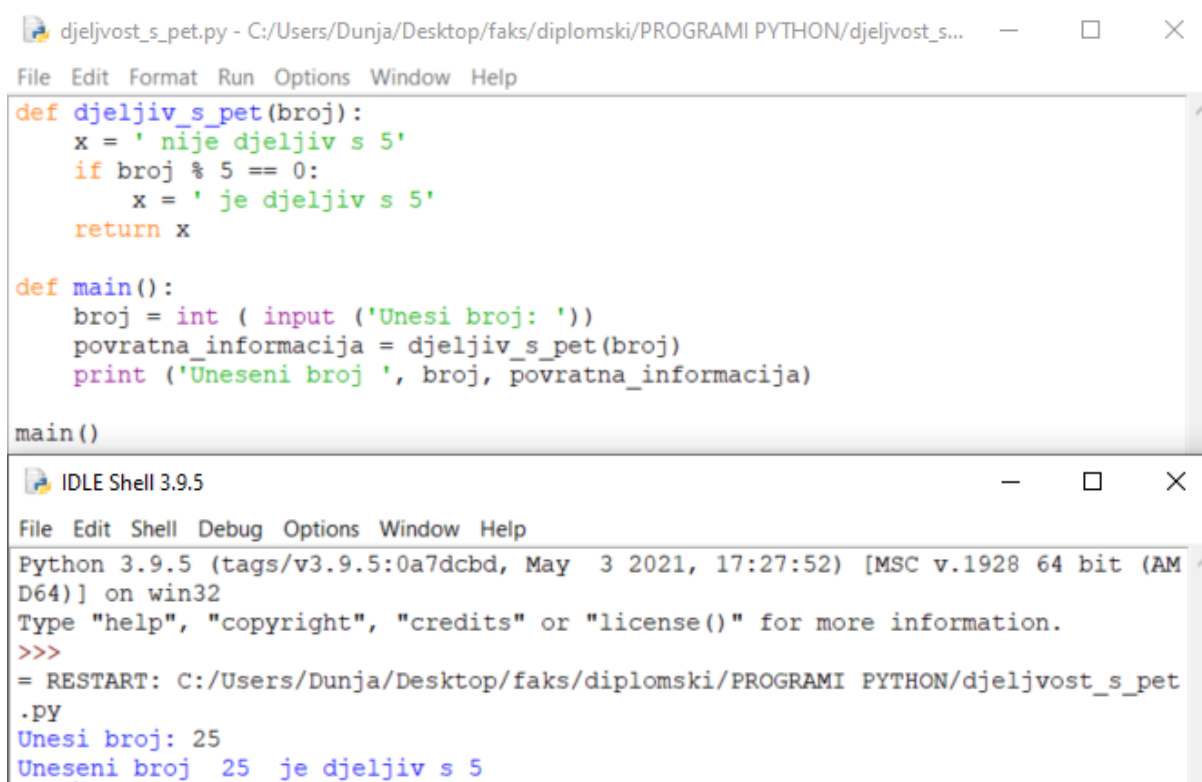


```
naucene_funkcije.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/nauce...
File Edit Format Run Options Window Help
prva_rijec = str ( input ('Unesi prvu riječ: '))
druga_rijec = str ( input ('Unesi drugu riječ: '))
spojena_rijec = prva_rijec + druga_rijec
print ('Spojeno: ', spojena_rijec)

prvi_broj = int ( input ('Unesi prvi broj: '))
drugi_broj = int ( input ('Unesi drugi broj: '))
zbroj = prvi_broj + drugi_broj
print ('Zbroj: ', zbroj)

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/naucene_funkcije.py
Unesi prvu riječ: Hello
Unesi drugu riječ: world!
Spojeno: Helloworld!
Unesi prvi broj: 5
Unesi drugi broj: 4
Zbroj: 9
```

-program za određivanje djeljivosti s 5



```
djeljvost_s_pet.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/djeljvost_s...
File Edit Format Run Options Window Help
def djeljiv_s_pet(broj):
    x = ' nije djeljiv s 5'
    if broj % 5 == 0:
        x = ' je djeljiv s 5'
    return x

def main():
    broj = int ( input ('Unesi broj: '))
    povratna_informacija = djeljiv_s_pet(broj)
    print ('Uneseni broj ', broj, povratna_informacija)

main()

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/djeljvost_s_pet.py
Unesi broj: 25
Uneseni broj 25 je djeljiv s 5
```

```
djeljivost_s_pet.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/djeljivost_s...
File Edit Format Run Options Window Help
def djeljivost_s_pet(broj):
    x = ' nije djeljiv s 5'
    if broj % 5 == 0:
        x = ' je djeljiv s 5'
    return x

def main():
    broj = int ( input ('Unesi broj: '))
    povratna_informacija = djeljivost_s_pet(broj)
    print ('Uneseni broj ', broj, povratna_informacija)

main()
```

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
1= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/djeljivost_s_pet.py
Unesi broj: 17
Uneseni broj 17 nije djeljiv s 5
>>> |
```

-program za oduzimanje dva broja bez ulaznih vrijednosti i bez vraćanja vrijednosti

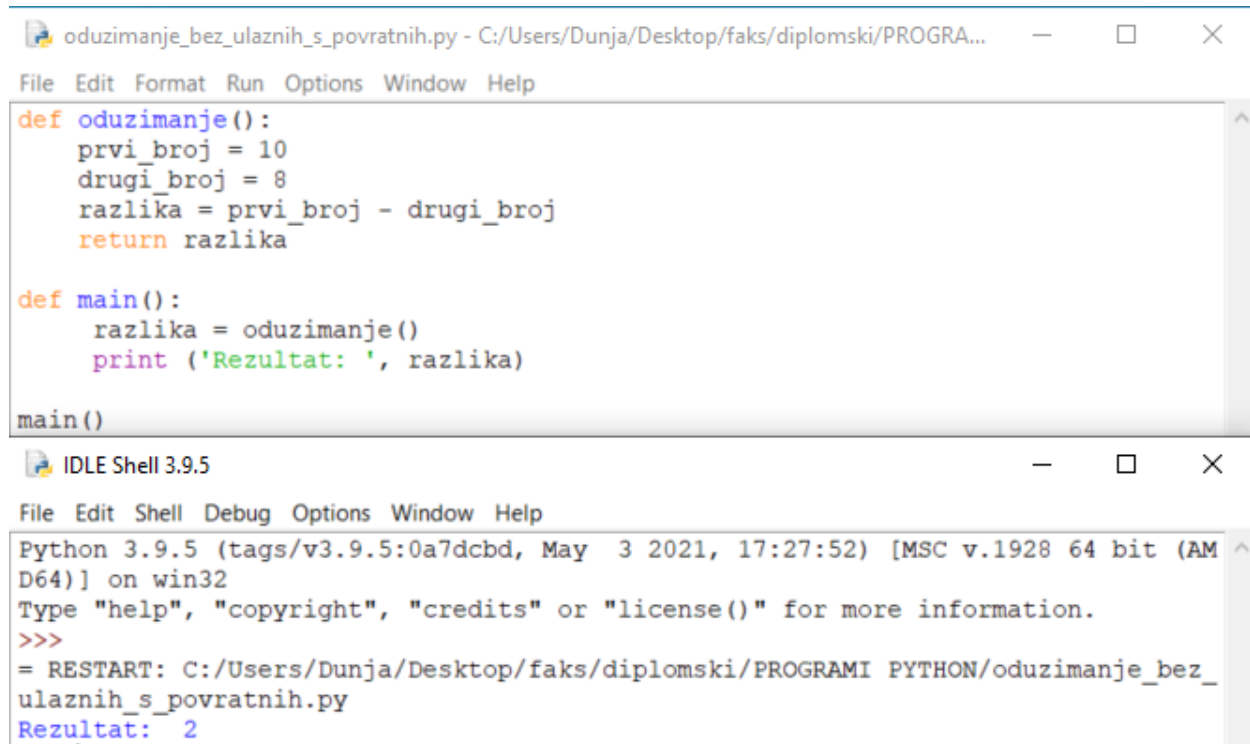
```
oduzimanje_bez_ulaznih_bez_povratnih.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGR...
File Edit Format Run Options Window Help
def oduzimanje():
    prvi_broj = 10
    drugi_broj = 8
    razlika = prvi_broj - drugi_broj
    print ('Rezultat: ', razlika)

def main():
    oduzimanje()

main()
```

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/oduzimanje_bez_ulaznih_bez_povratnih.py
Rezultat: 2
```

-program za oduzimanje dva broja bez ulaznih vrijednosti s vraćanjem vrijednosti



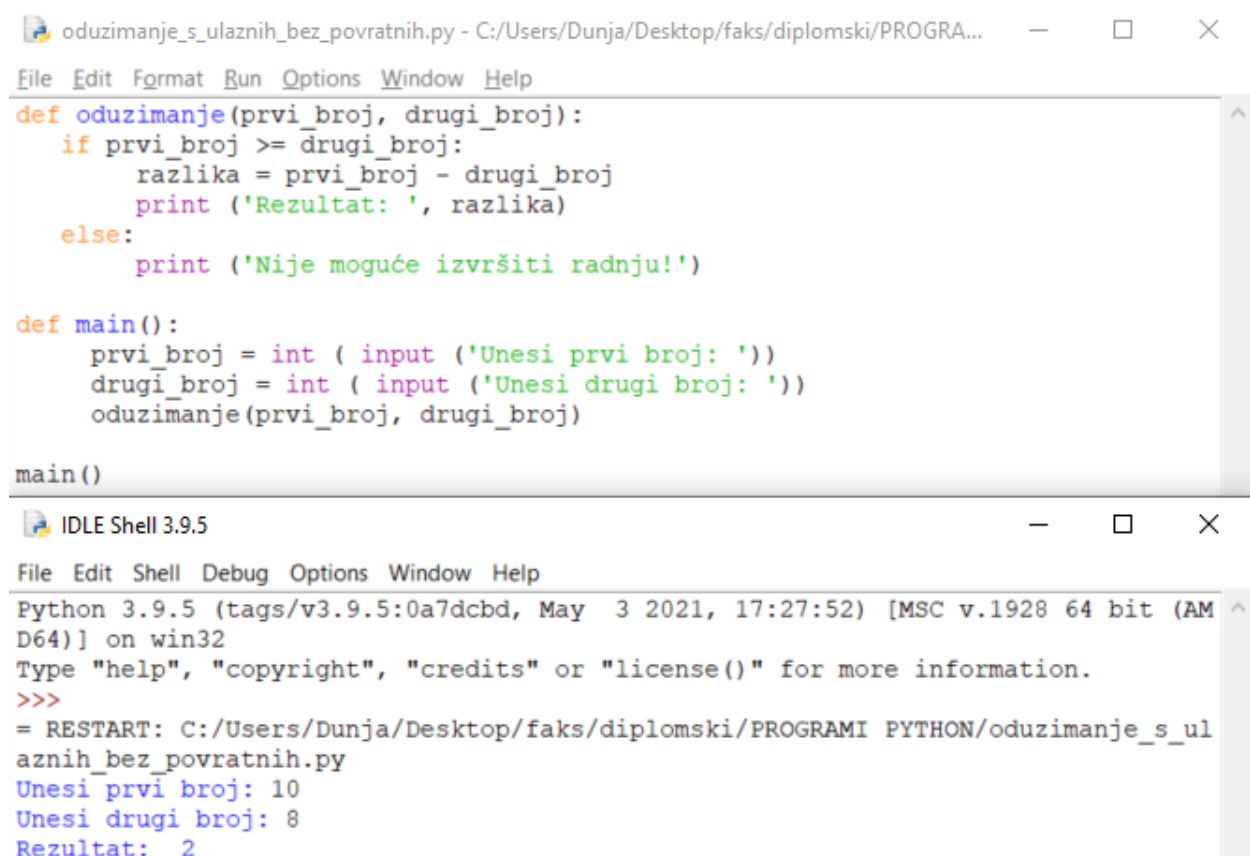
```
oduzimanje_bez_ulaznih_s_povratnih.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRA...
File Edit Format Run Options Window Help
def oduzimanje():
    prvi_broj = 10
    drugi_broj = 8
    razlika = prvi_broj - drugi_broj
    return razlika

def main():
    razlika = oduzimanje()
    print ('Rezultat: ', razlika)

main()

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/oduzimanje_bez_
ulaznih_s_povratnih.py
Rezultat: 2
```

-program za oduzimanje dva broja s ulaznim vrijednostima i bez povratnih vrijednosti



```
oduzimanje_s_ulaznih_bez_povratnih.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRA...
File Edit Format Run Options Window Help
def oduzimanje(prvi_broj, drugi_broj):
    if prvi_broj >= drugi_broj:
        razlika = prvi_broj - drugi_broj
        print ('Rezultat: ', razlika)
    else:
        print ('Nije moguće izvršiti radnju!')

def main():
    prvi_broj = int ( input ('Unesi prvi broj: '))
    drugi_broj = int ( input ('Unesi drugi broj: '))
    oduzimanje(prvi_broj, drugi_broj)

main()

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/oduzimanje_s_ul
aznih_bez_povratnih.py
Unesi prvi broj: 10
Unesi drugi broj: 8
Rezultat: 2
```



```
oduzimanje_s_ulaznih_bez_povratnih.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRA...
File Edit Format Run Options Window Help
def oduzimanje(prvi_broj, drugi_broj):
    if prvi_broj >= drugi_broj:
        razlika = prvi_broj - drugi_broj
        print ('Rezultat: ', razlika)
    else:
        print ('Nije moguće izvršiti radnju!')

def main():
    prvi_broj = int ( input ('Unesi prvi broj: '))
    drugi_broj = int ( input ('Unesi drugi broj: '))
    oduzimanje(prvi_broj, drugi_broj)

main()

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/oduzimanje_s_ulaznih_bez_povratnih.py
Unesi prvi broj: 8
Unesi drugi broj: 10
Nije moguće izvršiti radnju!
```

-program za oduzimanje dva broja s ulaznim vrijednostima i povratnom vrijednošću

```
oduzimanje_s_ulaznih_s_povratnih.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI ...
File Edit Format Run Options Window Help
def oduzimanje(prvi_broj, drugi_broj):
    razlika = -1
    if prvi_broj >= drugi_broj:
        razlika = prvi_broj - drugi_broj
    return razlika

def main():
    prvi_broj = int ( input ('Unesi prvi broj: '))
    drugi_broj = int ( input ('Unesi drugi broj: '))
    rezultat = oduzimanje(prvi_broj, drugi_broj)
    if rezultat == -1:
        print ('Nije moguće izvršiti radnju!')
    else:
        print ('Rezultat:' , rezultat)

main()

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/oduzimanje_s_ulaznih_s_povratnih.py
Unesi prvi broj: 10
Unesi drugi broj: 8
Rezultat: 2
```

```

oduzimanje_s_ulaznih_s_povratnih.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI ...
File Edit Format Run Options Window Help
def oduzimanje(prvi_broj, drugi_broj):
    razlika = -1
    if prvi_broj >= drugi_broj:
        razlika = prvi_broj - drugi_broj
    return razlika

def main():
    prvi_broj = int ( input ('Unesi prvi broj: '))
    drugi_broj = int ( input ('Unesi drugi broj: '))
    rezultat = oduzimanje(prvi_broj, drugi_broj)
    if rezultat == -1:
        print ('Nije moguće izvršiti radnju!')
    else:
        print ('Rezultat:' , rezultat)

main()

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/oduzimanje_s_ulaznih_s_povratnih.py
Unesi prvi broj: 8
Unesi drugi broj: 10
Nije moguće izvršiti radnju!

```

## 8. Prilog: Materijal za 8. razred

### a) *Sortiranje*

-sortiranje od najmanjeg do najvećeg broja

```

izracun_najmanjeg_od_3.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/izracun_najmanjeg...
File Edit Format Run Options Window Help
def sortiraj (a,b,c):
    if a > b:
        a,b=b,a
    if a > c:
        a,c=c,a
    if b > c:
        b,c=c,b
    print('Sortirani brojevi od najmanjeg do najvećeg ', a,b,c)

def main ():
    a = float (input('Prvi broj:'))
    b = float (input('Drugi broj:'))
    c = float (input('Treci broj:'))
    sortiraj(a,b,c)

main()

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Dunja\Desktop\faks\diplomski\PROGRAMI PYTHON\izracun_najmanjeg_od_3.py
Prvi broj:21.35
Drugi broj:22.12
Treci broj:21.40
Sortirani brojevi od najmanjeg do najvećeg 21.35 21.4 22.12

```

-sortiranje od najvećeg do najmanjeg broja

```
najveci_najmanji.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/najveci_najmanji.py (3.9.5)
File Edit Format Run Options Window Help
def sortiraj (a,b,c):
    if a > b:
        a,b=b,a
    if a > c:
        a,c=c,a
    if b > c:
        b,c=c,b
    print('Sortirani bojevi od najvećeg do najmanjeg ', c,b,a)

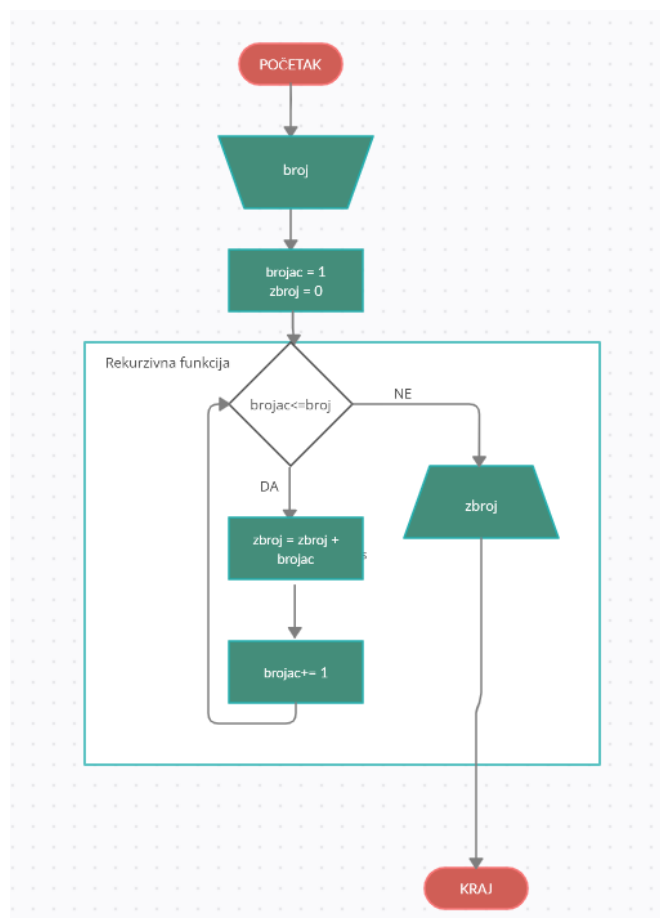
def main ():
    a = float (input('Prvi broj:'))
    b = float (input('Drugi broj:'))
    c = float (input('Treci broj:'))
    sortiraj(a,b,c)

main()

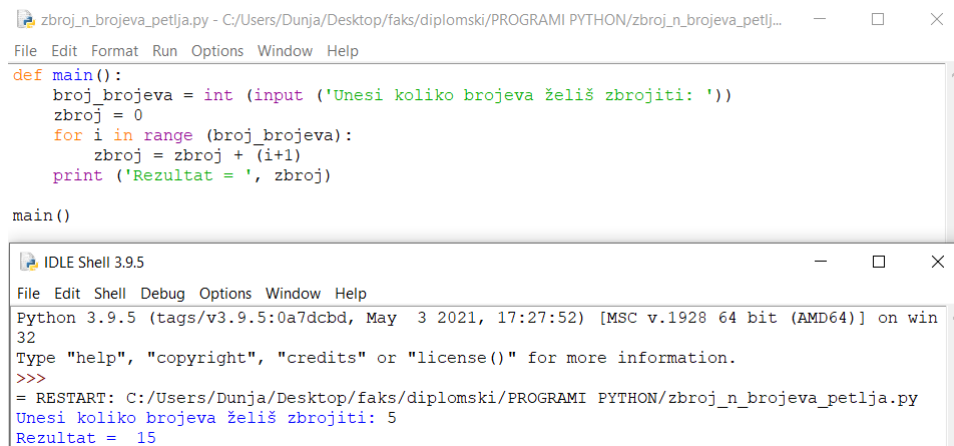
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/najveci_najmanji.py
Prvi broj:21.35
Drugi broj:22.12
Treci broj:21.40
Sortirani bojevi od najvećeg do najmanjeg  22.12 21.4 21.35
```

b) *Primjena rekurzije u Pythonu*

-dijagram toka za zbroj prvih n brojeva



## -program za zbroj prvih n brojeva bez rekurzije

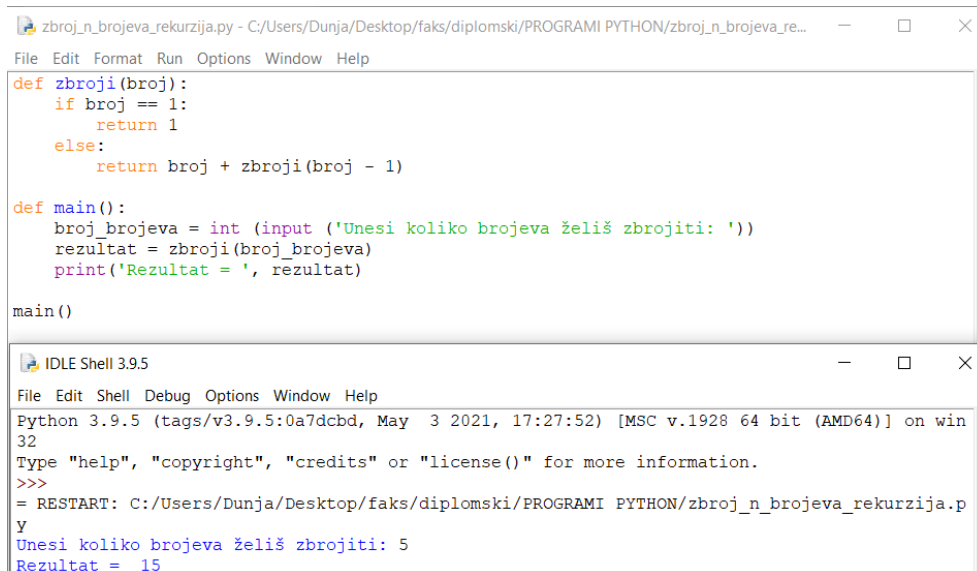


```
zbroj_n_brojeva_petlja.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/zbroj_n_brojeva_petlj...
File Edit Format Run Options Window Help
def main():
    broj_brojeva = int (input ('Unesi koliko brojeva želiš zbrojiti: '))
    zbroj = 0
    for i in range (broj_brojeva):
        zbroj = zbroj + (i+1)
    print ('Rezultat = ', zbroj)

main()

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/zbroj_n_brojeva_petlja.py
Unesi koliko brojeva želiš zbrojiti: 5
Rezultat = 15
```

## -program za zbroj prvih n brojeva s rekurzijom



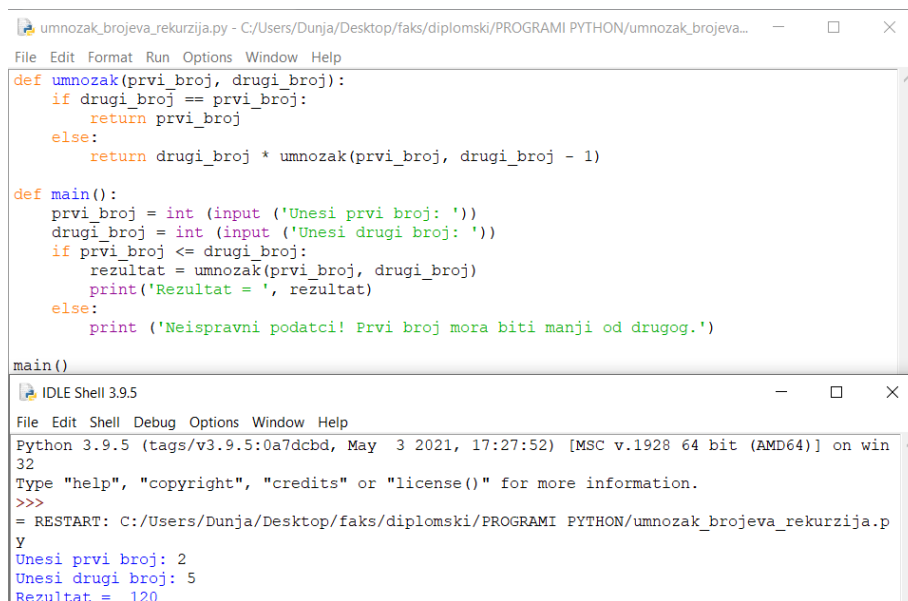
```
zbroj_n_brojeva_rekurzija.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/zbroj_n_brojeva_re...
File Edit Format Run Options Window Help
def zbroji(broj):
    if broj == 1:
        return 1
    else:
        return broj + zbroji(broj - 1)

def main():
    broj_brojeva = int (input ('Unesi koliko brojeva želiš zbrojiti: '))
    rezultat = zbroji(broj_brojeva)
    print ('Rezultat = ', rezultat)

main()

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/zbroj_n_brojeva_rekurzija.p
y
Unesi koliko brojeva želiš zbrojiti: 5
Rezultat = 15
```

## -program za računanje umnoška brojeva između dva unesena broja



```
umnozak_brojeva_rekurzija.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/umnozak_brojeva...
File Edit Format Run Options Window Help
def umnozак(prvi_broj, drugi_broj):
    if drugi_broj == prvi_broj:
        return prvi_broj
    else:
        return drugi_broj * umnozак(prvi_broj, drugi_broj - 1)

def main():
    prvi_broj = int (input ('Unesi prvi broj: '))
    drugi_broj = int (input ('Unesi drugi broj: '))
    if prvi_broj <= drugi_broj:
        rezultat = umnozак(prvi_broj, drugi_broj)
        print ('Rezultat = ', rezultat)
    else:
        print ('Neispravni podatci! Prvi broj mora biti manji od drugog.')

main()

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/umnozak_brojeva_rekurzija.p
y
Unesi prvi broj: 2
Unesi drugi broj: 5
Rezultat = 120
```

## 9. Prilog: Materijal za projektnu nastavu u 4. razredu- *Programi za množenje* -zadatci

### 1. skupina

Izradite program za vježbanje množenja pri čemu će program zadati 2 broja jedan četveroznamenkast, a drugi dvoznamenkasti. Program treba nuditi mogućnost prikaza umnoška (kada želimo provjeriti jesmo li točno riješili zadatak). Koristite *Scratch*.

### 2. skupina

Izradite program za vježbanje množenja pri čemu mi unosimo faktore. Program nam ne daje odmah rješenje već kada mi želimo (odnosno kada želimo provjeriti jesmo li točno riješili zadatak). Koristite *Scratch* |

### 3. skupina

Izradite program za vježbanje množenja pri čemu će program zadati 2 broja jedan četveroznamenkast, a drugi dvoznamenkasti. Program treba nuditi mogućnost prikaza umnoška (kada želimo provjeriti jesmo li točno riješili zadatak). Koristite *Micro:bit*.

## -listić za plan

1. Što je potrebno izraditi?

---

2. Koji programski jezik ćemo koristiti?

---

3. Što program treba raditi?

---

---

---

4. Plan:

---

---

---

---

5. Problemi kod izrade programa:

---

---

-prijevod blokova naučenih blokova u *Micro:bitu*

Blokovi u Micro:Bit-u koje smo koristili:

- On start- na početku
- Show number- prikaži broj
- Show leds- prikaži led svjetlima
- Show icon- prikaži ikonu
- Show string- prikaži string (za prikaz teksta)
- On button a pressed- na pritisak gumba A
- On shake- kada protresemo
- Repeat (4) times do- ponovi (4) puta
- If (true) then- ako je (istina) onda
- If (true) then- else- ako je (istina) onda- inače
- Pick random (0) to (10)- odaberi nasumičan broj od 1 do 6

-rješenje 1. skupine

The screenshot displays the Micro:bit Scratch editor interface. On the left, the script area contains the following code blocks:

- When clicked** (yellow)
- Set 1. faktor to random number from 1000 to 9999** (orange)
- Say 1. faktor je 1. faktor** (purple)
- Wait 3 seconds** (orange)
- Set 2. faktor to random number from 10 to 99** (orange)
- Say 2. faktor je 2. faktor** (purple)
- When key pressed: spacebar** (yellow)
- Set Umnožak to 1. faktor \* 2. faktor** (orange)
- Say Umnožak je Umnožak** (purple)

On the right, the stage area shows a yellow ball with a speech bubble that says "Umnožak je 236180". Above the stage, the current values are displayed: 1. faktor: 6748, 2. faktor: 35, and Umnožak: 236180. At the bottom, the object properties for the 'Ball' are shown: Lik: Ball, x: -36, y: -14, Veličina: 100, Smjer: 90. The 'Pozornica' (Stage) area shows 'Pozadine' (Background) set to 1.

## -rješenje 2. skupine

The image shows a Scratch code editor with a script for a multiplication calculator. The script starts with a 'when clicked' event, followed by two prompts for factors, each with a 'wait' block. The factors are stored in variables '1. faktor' and '2. faktor'. A 'when spacebar is pressed' event triggers the calculation: 'Umnožak' is set to '1. faktor' multiplied by '2. faktor'. Finally, the result is spoken using a 'say' block.

```
when clicked  
  ask "Unesi 1. faktor:" and wait  
  set 1. faktor to answer  
  ask "Unesi 2. faktor:" and wait  
  set 2. faktor to answer  
  
when spacebar is pressed  
  set Umnožak to 1. faktor * 2. faktor  
  say "Umnožak je " + Umnožak for 2 seconds
```

The right side of the editor shows the current state: '1. faktor' is 2943, '2. faktor' is 32, and 'Umnožak' is 94176. A speech bubble from the Scratch cat says 'Umnožak je 94176'.

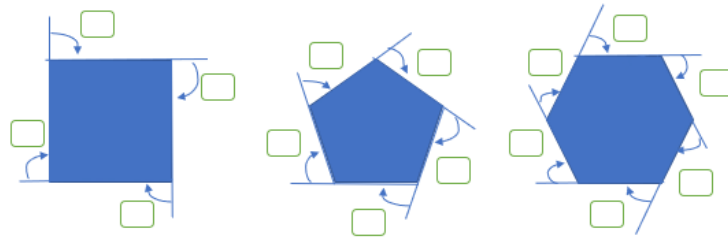
## -rješenje 3. skupine

The image shows a Scratch code editor with a script for a random number multiplier. It features three event-driven blocks: 'on button A pressed', 'on button B pressed', and 'on shake'. Each event sets a factor to a random number and shows it. The 'on shake' event calculates the product of the two factors and shows the result.

```
on button A pressed  
  set 1. faktor to pick random 1000 to 9999  
  show number 1. faktor  
  
on button B pressed  
  set 2. faktor to pick random 10 to 99  
  show number 2. faktor  
  
on shake  
  set umnožak to 1. faktor * 2. faktor  
  show number umnožak
```

10. Prilog: Materijal za projektnu nastavu u 5. razredu- *Program za mnogokute*

PROGRAMI ZA MNOGOKUTE



PRAVILNI MNOGOKUT	Broj stranica	Veličina unutarnjeg kuta	Veličina vanjskog kuta
četverokut			
peterokut			
šesterokut			
sedmerokut			
osmerokut			

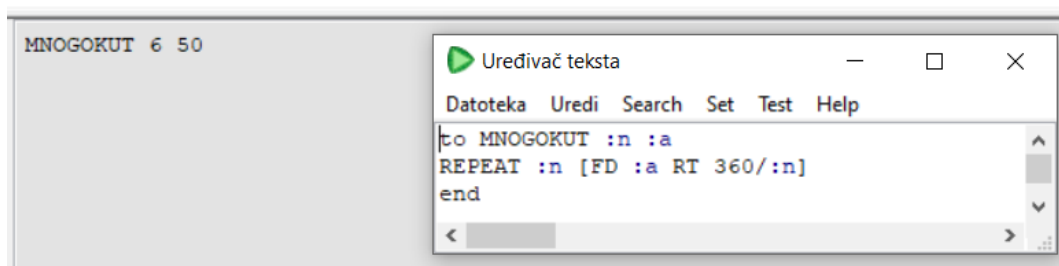
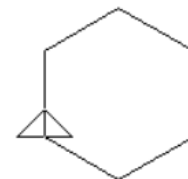
S povećanjem broja stranica veličina unutarnjeg kuta se povećava/smanjuje.

S povećanjem broja stranica veličina vanjskog kuta se povećava/smanjuje.

LT/RT određen je veličinom unutarnjeg/vanjskog kuta.

PROSTOR ZA ALGORITAM:

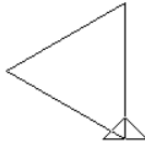
-programsko rješenje



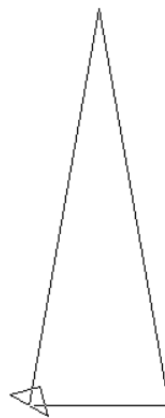


# 11. Prilog: Materijal za projektну nastavu u 6. razredu- *Trokut*

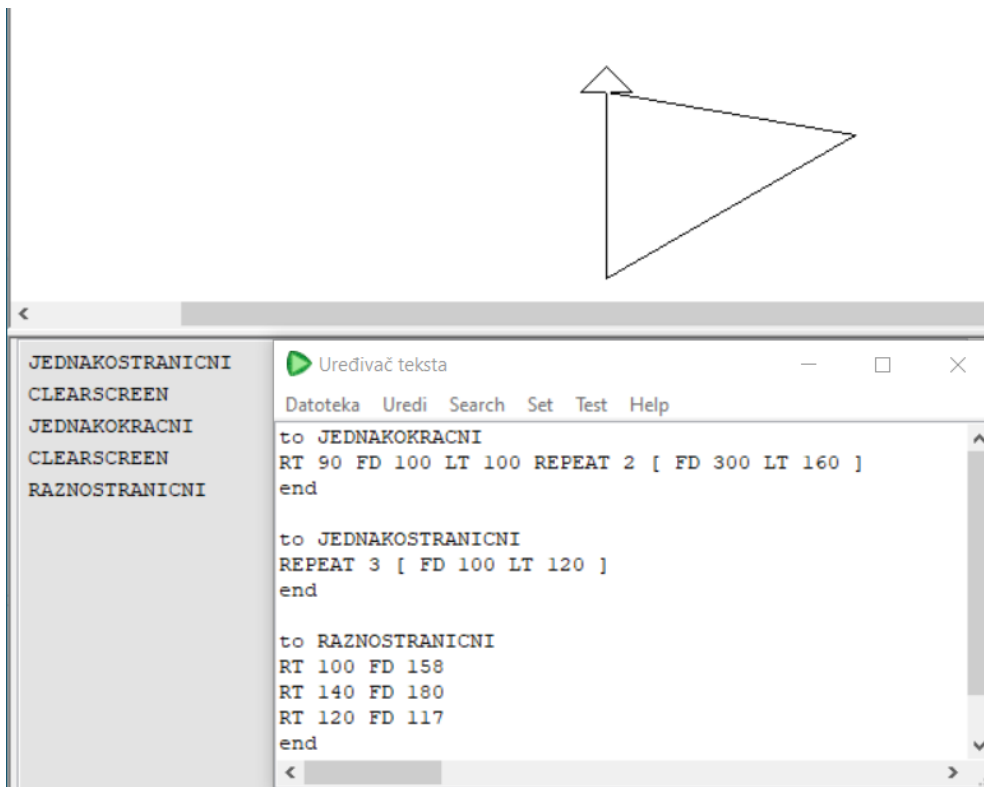
-Logo program za crtanje trokuta



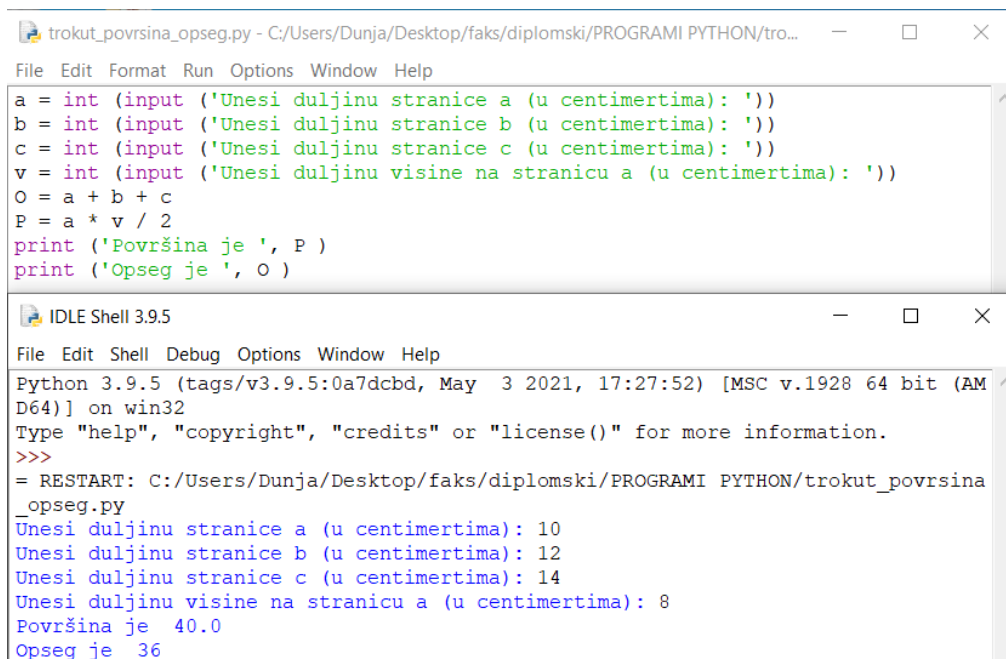
```
JEDNAKOSTRANICNI
Uređivač teksta
Datoteka Uredi Search Set Test Help
to JEDNAKOKRACNI
RT 90 FD 100 LT 100 REPEAT 2 [ FD 300 LT 160 ]
end
to JEDNAKOSTRANICNI
REPEAT 3 [ FD 100 LT 120 ]
end
to RAZNOSTRANICNI
RT 100 FD 158
RT 140 FD 180
RT 120 FD 117
end
```



```
JEDNAKOSTRANICNI
CLEARSCREEN
JEDNAKOKRACNI
Uređivač teksta
Datoteka Uredi Search Set Test Help
to JEDNAKOKRACNI
RT 90 FD 100 LT 100 REPEAT 2 [ FD 300 LT 160 ]
end
to JEDNAKOSTRANICNI
REPEAT 3 [ FD 100 LT 120 ]
end
to RAZNOSTRANICNI
RT 100 FD 158
RT 140 FD 180
RT 120 FD 117
end
```



-Python program za računanje površine i opsega



```

trokut_povrsina_opseg1.py - C:\Users\Dunja\Desktop\faks\diplomski\PROGRAMI PYTHON\tr...
File Edit Format Run Options Window Help
Odabir = input ('Upiši A za jednakostranični, B za jednakokračni ili C za raznos
if Odabir == 'A':
    a = int (input ('Unesi duljinu stranice a (u centimertima): '))
    v = int (input ('Unesi duljinu visine na stranicu a (u centimertima): '))
    O = 3 * a
    P = a * v / 2
    print ('Površina je ', P )
    print ('Opseg je ', O )
if Odabir == 'B':
    a = int (input ('Unesi duljinu stranice a (u centimertima): '))
    b = int (input ('Unesi duljinu stranice b (u centimertima): '))
    v = int (input ('Unesi duljinu visine na stranicu a (u centimertima): '))
    O = a + 2 * b
    P = a * v / 2
    print ('Površina je ', P )
    print ('Opseg je ', O )
if Odabir == 'c':
    a = int (input ('Unesi duljinu stranice a (u centimertima): '))
    b = int (input ('Unesi duljinu stranice b (u centimertima): '))
    c = int (input ('Unesi duljinu stranice c (u centimertima): '))
    v = int (input ('Unesi duljinu visine na stranicu a (u centimertima): '))
    O = a + b + c
    P = a * v / 2
    print ('Površina je ', P )
    print ('Opseg je ', O )

```

```

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Dunja\Desktop\faks\diplomski\PROGRAMI PYTHON\trokut_povrsina_opseg1.py
Upiši A za jednakostranični, B za jednakokračni ili C za raznostranični trokut: B
Unesi duljinu stranice a (u centimertima): 15
Unesi duljinu stranice b (u centimertima): 10
Unesi duljinu visine na stranicu a (u centimertima): 18
Površina je 135.0
Opseg je 35

```

```

trokut_povrsina_opseg1.py - C:\Users\Dunja\Desktop\faks\diplomski\PROGRAMI PYTHON\tr...
File Edit Format Run Options Window Help
Odabir = input ('Upiši A za jednakostranični, B za jednakokračni ili C za raznos
if Odabir == 'A':
    a = int (input ('Unesi duljinu stranice a (u centimertima): '))
    v = int (input ('Unesi duljinu visine na stranicu a (u centimertima): '))
    O = 3 * a
    P = a * v / 2
    print ('Površina je ', P )
    print ('Opseg je ', O )
if Odabir == 'B':
    a = int (input ('Unesi duljinu stranice a (u centimertima): '))
    b = int (input ('Unesi duljinu stranice b (u centimertima): '))
    v = int (input ('Unesi duljinu visine na stranicu a (u centimertima): '))
    O = a + 2 * b
    P = a * v / 2
    print ('Površina je ', P )
    print ('Opseg je ', O )
if Odabir == 'c':
    a = int (input ('Unesi duljinu stranice a (u centimertima): '))
    b = int (input ('Unesi duljinu stranice b (u centimertima): '))
    c = int (input ('Unesi duljinu stranice c (u centimertima): '))
    v = int (input ('Unesi duljinu visine na stranicu a (u centimertima): '))
    O = a + b + c
    P = a * v / 2
    print ('Površina je ', P )
    print ('Opseg je ', O )

```

```

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Dunja\Desktop\faks\diplomski\PROGRAMI PYTHON\trokut_povrsina_opseg1.py
Upiši A za jednakostranični, B za jednakokračni ili C za raznostranični trokut:
C
Unesi duljinu stranice a (u centimertima): 12
Unesi duljinu stranice b (u centimertima): 10
Unesi duljinu stranice c (u centimertima): 20
Unesi duljinu visine na stranicu a (u centimertima): 8
Površina je 48.0
Opseg je 42

```

## 12. Prilog: Materijal za projektnu nastavu u 7. razredu- *Kalkulator*

```
*program_pretvaranje.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/program_pretvaranje...
File Edit Format Run Options Window Help
def odabrana_pretvorba():
    print ('Unesite broj željene pretvorbe:')
    print ('(1) CM u DM')
    print ('(2) CM u M')
    print ('(3) DM u CM')
    print ('(4) DM u M')
    print ('(5) M u DM')
    print ('(6) M u CM')
    print ('(0) Izlaz iz programa')
    ulazna_komanda = int (input ('Unesi broj:'))
    return ulazna_komanda

def pretvori_cm_u_dm():
    broj = float (input ('Unesi željeni broj:'))
    rjesenje = broj / 10
    print (broj, ' cm = ', rjesenje, ' dm')

def pretvori_cm_u_m():
    broj = float (input ('Unesi željeni broj:'))
    rjesenje = broj / 100
    print (broj, ' cm = ', rjesenje, ' m')

def pretvori_dm_u_cm():
    broj = float (input ('Unesi željeni broj:'))
    rjesenje = broj * 10
    print (broj, ' dm = ', rjesenje, ' cm')

def pretvori_dm_u_m():
    broj = float (input ('Unesi željeni broj:'))
    rjesenje = broj / 10
    print (broj, ' dm = ', rjesenje, ' m')

def pretvori_m_u_dm():
    broj = float (input ('Unesi željeni broj:'))
    rjesenje = broj * 10
    print (broj, ' m = ', rjesenje, ' dm')
    def pretvori_m_u_cm():
        broj = float (input ('Unesi željeni broj:'))
        rjesenje = broj * 100
        print (broj, ' m = ', rjesenje, ' cm')

def pretvori(ulazna_komanda):
    if ulazna_komanda == 1:
        pretvori_cm_u_dm()
    if ulazna_komanda == 2:
        pretvori_cm_u_m()

    if ulazna_komanda == 3:
        pretvori_dm_u_cm()
    if ulazna_komanda == 4:
        pretvori_dm_u_m()
    if ulazna_komanda == 5:
        pretvori_m_u_dm()
    if ulazna_komanda == 6:
        pretvori_m_u_cm()

def main():
    print ('Dobrodošli u program za pretvaranje mjernih jedinica')
    ulazna_komanda = -1
    while (ulazna_komanda != 0):
        ulazna_komanda = odabrana_pretvorba()
        pretvori(ulazna_komanda)
    print ('Kraj pretvaranja!')

main()
```

```

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/program_pretvaranje.py
Dobrodošli u program za pretvaranje mjernih jedinica
Unesite broj željene pretvorbe:
(1) CM u DM
(2) CM u M
(3) DM u CM
(4) DM u M
(5) M u DM
(6) M u CM
(0) Izlaz iz programa
Unesi broj:1
Unesi željeni broj:30
30.0 cm = 3.0 dm
Unesite broj željene pretvorbe:
(1) CM u DM
(2) CM u M
(3) DM u CM
(4) DM u M
(5) M u DM
(6) M u CM
(0) Izlaz iz programa
Unesi broj:2
Unesi željeni broj:12
12.0 cm = 0.12 m
Unesite broj željene pretvorbe:
(1) CM u DM
(2) CM u M
(3) DM u CM
(4) DM u M
(5) M u DM
(6) M u CM
(0) Izlaz iz programa
Unesi broj:3
Unesi željeni broj:40
40.0 dm = 400.0 cm
Unesite broj željene pretvorbe:
(1) CM u DM
(2) CM u M
(3) DM u CM
(4) DM u M
(5) M u DM
(6) M u CM
(0) Izlaz iz programa
Unesi broj:0
Kraj pretvaranja!

```

### 13. Prilog: Materijal za projektnu nastavu u 8. razredu- *Matematika*

-program za izračun kvadrata zbroja i razlike

```

projekt_matematika_osmi.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/projekt_matematik...
File Edit Format Run Options Window Help
def kvadrat_zbroja():
    prvi_broj = int(input('Unesi prvi broj:'))
    drugi_broj = int(input('Unesi drugi broj:'))
    rjesenje = prvi_broj * prvi_broj + (2 * prvi_broj * drugi_broj) + drugi_broj * drugi_broj
    return rjesenje

def kvadrat_razlike():
    prvi_broj = int(input('Unesi prvi broj:'))
    drugi_broj = int(input('Unesi drugi broj:'))
    rjesenje = prvi_broj * prvi_broj - (2 * prvi_broj * drugi_broj) + drugi_broj * drugi_broj
    return rjesenje

def obradi_komandu(ulazna_komanda):
    if ulazna_komanda == 1:
        print('Rezultat = ', kvadrat_zbroja())
    if ulazna_komanda == 2:
        print('Rezultat = ', kvadrat_razlike())

def main():
    print('Dobrodošli u program')
    ulazna_komanda = -1
    while (ulazna_komanda != 0):
        print('Unesite broj željene naredbe:')
        print('(1) Izračun kvadrata zbroja')
        print('(2) Izračun kvadrata razlike')
        print('(0) Izlaz iz programa')
        ulazna_komanda = int(input('Unesi broj:'))
        obradi_komandu(ulazna_komanda)
    print('Kraj programa!')

main()

```

```

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/projekt_matematika_osmi.py
Dobrodošli u program
Unesite broj željene naredbe:
(1) Izračun kvadrata zbroja
(2) Izračun kvadrata razlike
(0) Izlaz iz programa
Unesi broj:1
Unesi prvi broj:4
Unesi drugi broj:2
Rezultat = 36
Unesite broj željene naredbe:
(1) Izračun kvadrata zbroja
(2) Izračun kvadrata razlike
(0) Izlaz iz programa
Unesi broj:2
Unesi prvi broj:4
Unesi drugi broj:2
Rezultat = 4
Unesite broj željene naredbe:
(1) Izračun kvadrata zbroja
(2) Izračun kvadrata razlike
(0) Izlaz iz programa
Unesi broj:0
Kraj programa!
>>> |

```

-program za izračun duljine dijagonale pravokutnika/kvadrata

```

projekt_matematika_osmi_dijagonala.py - C:\Users\Dunja\Desktop\faks\diplomski\PROGRAMI PYTHON\projekt...
File Edit Format Run Options Window Help
from math import *

def dijagonala_pravokutnika():
    stranica_a = int(input('Unesi duljinu stranice a:'))
    stranica_b = int(input('Unesi duljinu stranice b:'))
    rjesenje = sqrt((stranica_a * stranica_a) + (stranica_b * stranica_b))
    return rjesenje

def dijagonala_kvadrata():
    stranica_a = int(input('Unesi duljinu stranice a:'))
    rjesenje = sqrt((stranica_a * stranica_a) + (stranica_a * stranica_a))
    return rjesenje

def obradi_komandu(ulazna_komanda):
    if ulazna_komanda == 1:
        print('Duljina dijagonale iznosi ', dijagonala_pravokutnika())
    if ulazna_komanda == 2:
        print('Duljina dijagonale iznosi', dijagonala_kvadrata())

def main():
    print('Dobrodošli u program')
    ulazna_komanda = -1
    while (ulazna_komanda != 0):
        print('Unesite broj željene naredbe:')
        print('(1) Izračun dijagonale pravokutnika')
        print('(2) Izračun dijagonale kvadrata')
        print('(0) Izlaz iz programa')
        ulazna_komanda = int(input('Unesi broj:'))
        obradi_komandu(ulazna_komanda)
    print('Kraj programa!')

main()

```

```

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Dunja\Desktop\faks\diplomski\PROGRAMI PYTHON\projekt_matematika_osmi_dij
agonalna.py
Dobrodošli u program
Unesite broj željene naredbe:
(1) Izračun dijagonale pravokutnika
(2) Izračun dijagonale kvadrata
(0) Izlaz iz programa
Unesi broj:1
Unesi duljinu stranice a:4
Unesi duljinu stranice b:3
Duljina dijagonale iznosi 5.0
Unesite broj željene naredbe:
(1) Izračun dijagonale pravokutnika
(2) Izračun dijagonale kvadrata
(0) Izlaz iz programa
Unesi broj:2
Unesi duljinu stranice a:4
Duljina dijagonale iznosi 5.656854249492381
Unesite broj željene naredbe:
(1) Izračun dijagonale pravokutnika
(2) Izračun dijagonale kvadrata
(0) Izlaz iz programa
Unesi broj:0
Kraj programa!
>>> |

```

## -program- pravilna četverostrana piramida

```

projekt_matematika_osmi_piramida.py - C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/projekt_matematika_osmi_piramida.py (3.9.5)
File Edit Format Run Options Window Help
def izracun_povrsine_baze(stranica_a):
    povrsina_baze = stranica_a * stranica_a
    return povrsina_baze

def izracun_povrsine_pobocja(stranica_a, visina_a):
    povrsina_pobocja = 2 * stranica_a * visina_a
    return povrsina_pobocja

def izracun_oplosja_piramide(povrsina_baze, povrsina_pobocja):
    oplosje = povrsina_baze + povrsina_pobocja
    return oplosje

def izracun_volumena_piramide(povrsina_baze, visina):
    volumen = 1/3 * povrsina_baze * visina
    return volumen

def obradi_komandu(ulazna_komanda, stranica_a, visina, visina_a):
    if ulazna_komanda == 1:
        print ('Površina baze = ', izracun_povrsine_baze(stranica_a))
    if ulazna_komanda == 2:
        print ('Površina pobočja = ', izracun_povrsine_pobocja(stranica_a, visina_a))
    if ulazna_komanda == 3:
        oplosje = izracun_oplosja_piramide(izracun_povrsine_baze(stranica_a), izracun_povrsine_pobocja(stranica_a, visina_a))
        print ('Oplošje = ', oplosje)
    if ulazna_komanda == 4:
        print ('Volumen = ', izracun_volumena_piramide(izracun_povrsine_baze(stranica_a), visina))

def main():
    print ('Dobrodošli u program - Pravilna četverostrana piramida')
    stranica_a = float (input ('Unesi duljinu stranice a:'))
    visina = float (input ('Unesi duljinu visine na bazu:'))
    visina_a = float (input ('Unesi duljinu visine na stranica a:'))
    ulazna_komanda = -1
    while (ulazna_komanda != 0):
        print ('Unesite broj željene naredbe:')
        print ('(1) Površina baze')
        print ('(2) Površina pobočja')
        print ('(3) Oplošje piramide')
        print ('(4) Volumen piramide')
        print ('(0) Izlaz iz programa')
        ulazna_komanda = int (input ('Unesi broj:'))
        obradi_komandu(ulazna_komanda, stranica_a, visina, visina_a)
    print ('Kraj programa!')

main()

```

IDLE Shell 3.9.5

File Edit Shell Debug Options Window Help

```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dunja/Desktop/faks/diplomski/PROGRAMI PYTHON/projekt_matematika_osmi_piramida.py
Dobrodošli u program - Pravilna četverostrana piramida
Unesi duljinu stranice a:4
Unesi duljinu visine na bazu:8
Unesi duljinu visine na stranica a:12
Unesite broj željene naredbe:
(1) Površina baze
(2) Površina pobočja
(3) Oplošje piramide
(4) Volumen piramide
(0) Izlaz iz programa
Unesi broj:1
Površina baze = 16.0
Unesite broj željene naredbe:
(1) Površina baze
(2) Površina pobočja
(3) Oplošje piramide
(4) Volumen piramide
(0) Izlaz iz programa
Unesi broj:2
Površina pobočja = 96.0
Unesite broj željene naredbe:
(1) Površina baze
(2) Površina pobočja
(3) Oplošje piramide
(4) Volumen piramide
(0) Izlaz iz programa
Unesi broj:3
Oplošje = 112.0
Unesite broj željene naredbe:
(1) Površina baze
(2) Površina pobočja
(3) Oplošje piramide
(4) Volumen piramide
(0) Izlaz iz programa
Unesi broj:4
Volumen = 42.666666666666664
Unesite broj željene naredbe:
(1) Površina baze
(2) Površina pobočja
(3) Oplošje piramide
(4) Volumen piramide
(0) Izlaz iz programa
Unesi broj:0
Kraj programa!
```



### **Izjava o izvornosti diplomskog rada**

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristila drugim izvorima osim onih koji su u njemu navedeni.

---

(vlastoručni potpis studenta)