

# Programski jezici u nastavi informatike u osnovnoj školi

---

Španec, Jasna

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:119218>

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

Download date / Datum preuzimanja: **2024-11-23**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
PRIRODOSLOVNO–MATEMATIČKI FAKULTET  
MATEMATIČKI ODSJEK

Jasna Španec

**PROGRAMSKI JEZICI U NASTAVI  
INFORMATIKE U OSNOVNOJ ŠKOLI**

Diplomski rad

Voditelj rada:  
izv. prof. dr. sc. Saša Singer

Zagreb, rujan 2019.

Ovaj diplomski rad obranjen je dana \_\_\_\_\_ pred ispitnim povjerenstvom u sastavu:

1. \_\_\_\_\_, predsjednik
2. \_\_\_\_\_, član
3. \_\_\_\_\_, član

Povjerenstvo je rad ocijenilo ocjenom \_\_\_\_\_.

Potpisi članova povjerenstva:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

*Zahvaljujem se svojem mentoru, izv. prof. dr. sc. Saši Singeru na ukazanom povjerenju, vodstvu i savjetima. Kroz studij, upoznala sam predivne ljude i doživotne prijateljice. Cure, hvala vam. Hvala mom dragom Lovri na podršci i strpljivosti. Neizmjerno hvala sestri koja me je svojim savjetima usmjeravala. Od srca zahvaljujem svojoj obitelji na neizmjernoj podršci, posebno svojim roditeljima kojima ovaj rad i posvećujem.*

# Sadržaj

<b>Sadržaj</b>	<b>iv</b>
<b>Uvod</b>	<b>2</b>
<b>1 Peti razred</b>	<b>3</b>
1.1 Odgojno-obrazovni ishodi . . . . .	3
1.2 Programski jezici u petom razredu . . . . .	3
<b>2 Šesti razred</b>	<b>14</b>
2.1 Odgojno-obrazovni ishodi . . . . .	14
2.2 Programski jezici u šestom razredu . . . . .	14
<b>3 Sedmi razred</b>	<b>22</b>
3.1 Odgojno-obrazovni ishodi . . . . .	22
3.2 Programski jezici u sedmom razredu . . . . .	22
<b>4 Osmi razred</b>	<b>38</b>
4.1 Odgojno-obrazovni ishodi . . . . .	38
4.2 Programski jezici u osmom razredu . . . . .	38
<b>5 Zaključak</b>	<b>47</b>
<b>Bibliografija</b>	<b>49</b>

# Uvod

Kurikulum za nastavni predmet Informatike za osnovne škole u Republici Hrvatskoj donesen je 2018. godine. E-Društvo, Digitalna pismenost i komunikacija, Računalno razmišljanje i programiranje te Informacije i digitalna tehnologija su četiri domene kojima se realiziraju ciljevi predmeta Informatika.

Sastavni dio kurikuluma nastavnog predmeta Informatika su prilozi: odgojno-obrazovni ishodi, prikaz godišnjeg broja sati i oblika izvođenja nastavnog predmeta, popis preporučenih kvalifikacija za učitelje i nastavnike Informatike. U prilogu odgojno-obrazovni ishodi nalaze se ishodi, razrada ishoda, razina usvojenosti i preporuka za ostvarivanje odgojno-obrazovnih ishoda po razredima i domenama, s popisom literature. Odgojno-obrazovni ishodi iskazuju se na razini pojedine godine učenja i poučavanja predmeta za svaku domenu. Ishodi su jasni i nedvosmisleni iskaz očekivanja od učenika te mogu biti određeni kao znanja, vještine i stavovi, vrijednosti. Razrada ishoda utvrđuje se na razini pojedinog odgojno-obrazovnog ishoda. Time je precizno određena aktivnost i sadržaj u sklopu pojedinog, ili pak, skupa ishoda. Preporukom se opisuju mogućnosti ostvarivanja ishoda te ona nije obavezan dio ishoda. Razina usvojenosti je opis određenog opsega znanja, dubina razumijevanja i stupanj razvijenosti vještina i usvojenost stavova i vrijednosti.

Novi kurikulum usmjeren je na učenika, on je u središtu procesa poučavanja. Nastavnik mora mijenjati način poučavanja i učenja, stavljati naglasak na rješavanje problema i ishode učenja, a ne na sadržaje. Kao rezultat procesa učenja, od učenika se očekuje da zna, razumije, može napraviti i vrednovati odslušani nastavni sadržaj. Kompetencije koje je učenik stekao učenjem i dokazao nakon postupka učenja su ishod učenja. Skup kompetencija nazivamo kvalifikacijama za budući život. Jasno određenim kriterijima razvijenosti i usvojenosti odgojno-obrazovnih ishoda osiguravamo osnovu za objektivnije i valjano vrednovanje učeničkih postignuća. Učeničko znanje se može pratiti, procjenjivati ili mjeriti, kvalitativno i kvantitativno, i na taj način možemo provjeriti je li učenik usvojio ishode i je li napredovao tijekom procesa učenja.

Kroz domenu Računalno razmišljanje i programiranje učenici razvijaju inovativnost, stvaralaštvo i poduzetnost. Razvijaju logičko i algoritamsko razmišljanje, sposobnost rješavanja problema i programiranja, uče proces stvaranja aplikacije od početka ideje do konačnog proizvoda.

Proučavajući udžbenike i radne bilježnice za nastavu Informatike u osnovnoj školi, uvidjeli smo kako nakladnici *SysPrint* i *Profil Klett* obrađuju programske jezike *Python* i *Logo*, dok *Školska knjiga* obrađuje samo *Python*. Nadalje, *SysPrint* i *Školska knjiga* spominju kornjačinu grafiku te u *SysPrint*ovim udžbenicima programiranje u *Pythonu* uvode pomoću blokova.

U ovom radu analizirat ćemo programske jezike u udžbenicima za osnovnu školu te kritički vrednovati doprinos sadržaja odgojno-obrazovnim ciljevima predmeta Informatika, koji se nalaze u važećem kurikulumu u domeni Računalno razmišljanje i programiranje. Svako poglavlje predstavljat će jedan od viših razreda osnovne škole. Proučavat ćemo na koji način su obrađeni programski jezici *Python* i *Logo* te koliko su uspješno ostvareni zadani odgojno-obrazovni ishodi.

# Poglavlje 1

## Peti razred

### 1.1 Odgojno-obrazovni ishodi

U kurikulumu [1], kao odgojno-obrazovni ishod nakon pete godine učenja predmeta Informatika, u domeni Računalno razmišljanje i programiranje, navodi se kako će se učenik koristiti programskim alatom za stvaranje programa u kojem se koristi ulaznim i izlaznim vrijednostima te ponavljanjem. Učenik će stvarati algoritam za rješavanje jednostavnog zadatka, provjeravati ispravnost algoritma, otkrivati i popravljati greške.

### 1.2 Programski jezici u petom razredu

#### Python

Sadržaj promatranih udžbenika u potpunosti odgovara odgojno-obrazovnim ishodima. Autori udžbenika su na detaljan način objasnili pokretanje *Pythona* te su opisali sučelje i osnovne dijelove alata.

U zadatku prikazanom na slici 1.1, učenici slažu jednostavan niz uputa koristeći se blokovima. Autori su detaljno opisali postupak kako bi učenici mogli samostalno ponoviti postupak.



Napišite program za pozdravljanje. Program treba ispisati *Dobar dan!*

### Rješenje:

U izborniku naredbenih blokova kliknite na izbornik **Izlaz**, pa naredbeni blok **ispiši** odvučite u *slagalicu naredbi*. Zatim iz izbornika **Tekstni** odaberite **“ ”** i odvučite ga do prethodnog bloka tako da se spoje kao slagalica: **ispiši “ ”**. Kliknite u okvir između navodnika i upišite: *Dobar dan!* **ispiši “Dobar dan!”**.

Primijetite da se u programskom okviru Python automatski ispisala naredba:

**Python** (TAB = 2 razmaka)

```
print('Dobar dan!')
```

To je naredba za ispis teksta u programskom jeziku Python. Slaganje naredbi pomoći će vam da ga postupno upoznate.

Izvršenje programa:

**Rezultat**

```
Dobar dan!
```

Kliknite na gumb **izvrši**.

U okviru *Rezultat* ispisat će se pozdrav:

Slika 1.1: Vježba iz udžbenika *Informatika*<sup>+</sup> 5

Primjer zadatka za vježbu pojedinih akcija u sučelju prikazan je na slici 1.2.

Koji je ispravan način spremanja kreiranog programa?

- File* → *New File*
- File* → *Save As*
- File* → *Open*
- File* → *Save*

Koji je ispravan način pokretanja otvorenog programa?

- istovremenim pritiskom na tipke *Shift* i *F5*
- istovremenim pritiskom na tipke *Alt* i *F5*
- istovremenim pritiskom na tipke *Ctrl* i *F5*
- pritiskom na tipku *F5*

Slika 1.2: Zadatak iz radne bilježnice *Informatika* 5

Učenici istražuju djelovanje blokova/naredbi. Na slici 1.3, primjer je zadatka u kojem učenici moraju analizirati naredbu i napisati što ispisuje ta naredba.

Što će ispisati sljedeće naredbe?

a) `>>> print(5)` \_\_\_\_\_

b) `>>> print(5,10)` \_\_\_\_\_

c) `>>> print(5+10)` \_\_\_\_\_

d) `>>> print(5*10)` \_\_\_\_\_

Što će ispisati sljedeće naredbe?

a) `>>> print('August','Cesarec')` \_\_\_\_\_

b) `>>> print('August'+ 'Cesarec')` \_\_\_\_\_

c) `>>> print('August'*2)` \_\_\_\_\_

d) `>>> print('August','Cesarec',sep=':')` \_\_\_\_\_

Slika 1.3: Zadatak iz radne bilježnice *#mojportal5*

U udžbeniku *#mojportal5* [2], na primjeru su pokazani osnovni segmenti izrade programa: ulaz – obrada – izlaz.

Računalno rješenje	Kako radi program?
<code>a=int(input('Upiši stranicu a:'))</code>	unos ulaznih vrijednosti putem tipkovnice (stranice pravokutnika a i b) – ULAZ
<code>b=int(input('Upiši stranicu b:'))</code>	
<code>o=2*a+2*b</code>	izračunavanje traženih vrijednosti (opsega i površine) – OBRADA
<code>p=a*b</code>	
<code>print('Opseg je',o)</code>	ispis izračunatih vrijednosti i odgovarajuće poruke – IZLAZ
<code>print('Površina je',p)</code>	

Slika 1.4: Primjer iz udžbenika *#mojportal5*

U udžbenicima nalazimo velik broj zadataka u kojima učenici mogu analizirati zadani problem i predložiti niz naredbi kao moguće rješenje te, na kraju, pročitati objašnjenje rješenja. Jedan od takvih zadataka se nalazi na slici 1.5.

**\*\*\* VJEŽBA: DVORIŠTE**

Kreirajmo program `dvoriste` koji će imati dvije ulazne varijable: širinu i duljinu. Pretpostavimo da je dvorište pravokutnog oblika. Program će izračunati i ispisati površinu (kvadraturu) cijelog dvorišta.

**Rješenje**

```

sirina = int (input ("Upiši širinu dvorišta:"))
duljina = int (input ("Upiši duljinu
dvorišta:"))

print ("Dvorište ima", sirina*duljina,
"kvadrata.")

```

Primjećujemo da u ovom slučaju trebamo brojčane tipove podataka pa smo ih već pri upotrebi naredbe **INPUT** pretvorili u brojke (funkcija `int()`). Ostatak je jednostavan. Ako je dvorište pravokutnog oblika, njegova se površina dobiva množenjem dviju različitih stranica (`sirina*duljina`).

Nakon pokretanja programa ispisana je sljedeća poruka:

```

Upiši širinu dvorišta:10
Upiši duljinu dvorišta:5
Dvorište ima 50 kvadrata.

```

Ovo je program koji ima dvije ulazne vrijednosti. Može ih imati neograničeno mnogo. Važno ih je pritom dobro označiti.

Slika 1.5: Vježba iz udžbenika *Informatika 5*

U radnim bilježnicama postavljeni su zadaci na način da učenici samostalno razvijaju rješenje problema te se koriste ulaznim i izlaznim podacima, pridružuju vrijednosti i ponavljaju naredbe.



Kreiraj program pod nazivom `kotaci.py`. Program će tražiti da se upiše koliko prometnih sredstava na dva kotača (motor, bicikl, *hoverboard*...) i na četiri kotača (automobil, traktor, autobus...) ima tvoja obitelj. Vrijednosti će spremi u dvije različite varijable naziva `dva` i `cetiri`. Program će nakon pokretanja ispisati koliko je ukupno kotača moguće pobrojati u tvojoj obitelji.

Slika 1.6: Zadatak iz radne bilježnice *Informatika 5*

## Logo


Kao i kod *Pythona*, i kod *Loga*, nastavni sadržaj odgovara zadanim odgojno-obrazovnim ishodima. Učenicima je detaljno opisan postupak pokretanja *Loga* te koji su dijelovi tog programskog alata.

### **Pokrenite FMS Logo**

Kliknite na gumb , pa pronađite mapu *FMSLogo*  u kojoj se nalazi program *FMSLogo*.

Kliknite na  i pokrenite *FMS Logo*.

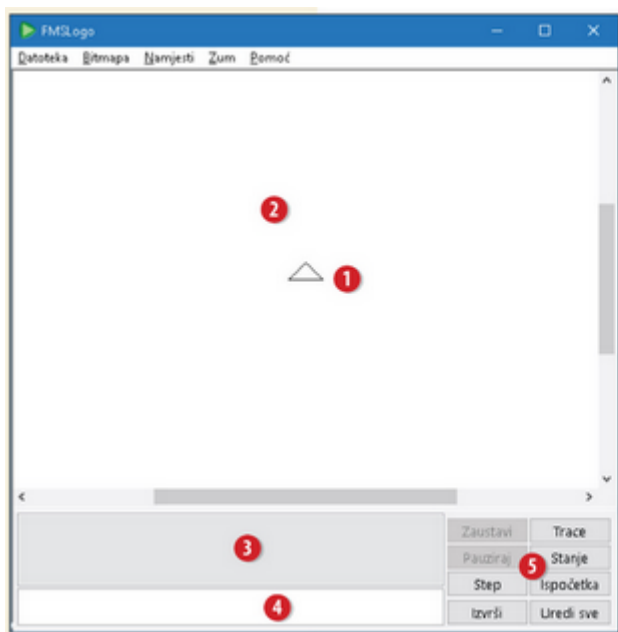
Tijekom instalacije *FMS Loga* na radnoj površini stvori se prečac. Znači, program možete pokrenuti i klikom na prečac.

 Postupak instalacije FMS Loga možete pogledati u prezentaciji **55Ainstal.pptx**.

Sučelje programskog jezika Logo sastoji se od dvaju osnovnih prozora.



Slika 1.7: Upute za instalaciju programskog alata iz udžbenika *Informatika<sup>+</sup> 5*



Trokutić **1** u sredini bijelog prozora zvat ćemo **kornjača**. U gornjem bijelom prozoru **2** crta kornjača. Bijeli prozor jest **grafički prozor**. U grafičkom prozoru crta kornjača.

U donjem sivom prozoru **3** prikazuju se naredbe koje je kornjača upravo izvršila i poruke o pogreškama koje su se dogodile. Sivi prozor jest **naredbeni prozor**.

Na dnu prozora Loga nalazi se bijeli redak **4** u kojemu će pisati naredba koja se trenutno izvodi. To je **naredbeni redak**.

U donjem desnom dijelu prozora Loga **5** nalaze se gumbi koji će vam trebati za pisanje i kontrolu programa. Upoznat ćete ih postupno.



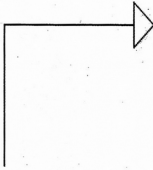
Kao što se jezik kojim govorimo sastoji od

Slika 1.8: Dijelovi programskog alata iz udžbenika *Informatika<sup>+</sup> 5*

Učenici uče crtati geometrijske likove koristeći jednostavan niz uputa. Primjer toga je zadatak koji se nalazi na slici 1.9.


**☛ VJEŽBA: CRTA DO CRTE - LIK**

Naučit ćemo nacrtati osnovne geometrijske likove.

Pokušajmo prvo nacrtati liniju. <b>FD 100</b>	
Okrenut ćemo se udesno koristeći se naredbom. <b>RT 90</b>	
Napravimo još jednu ravnu liniju. <b>FD 100</b>	

Slika 1.9: Zadatak iz udžbenika *Informatika 5*

Pojavljaju se zadaci u kojima učenici povezuju niz naredbi u cjelinu, kako bi došli do traženog rješenja.



**Slika 8.** Niz trokuta

DESNO 45 okreće kornjaču udesno za 45°.




DESNO 30 okreće kornjaču udesno za 30°.

LJEVO 45 okreće kornjaču ulijevo za 45°.

LJEVO 30 okreće kornjaču ulijevo za 30°.

Nacrtajte niz trokuta kao na slici 8.:




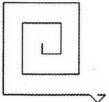
**Rješenje:**

Klikom na gumb  10 nacrtat će se trokut  . Dakle, trokut se crta udesno od položaja kornjače. Zato najprije okrenite kornjaču ulijevo pomoću  90, pa ponavljajući naredbe  10 i  10, šest puta nacrtajte niz trokuta:  . Ostaje obojiti trokute. Pomaknite kornjaču u sredinu prvog trokuta naredbama: **NAZAD 5, DESNO 90, GORE, IDI 5, DOLJE**, pa obojite prvi trokut narančasto pomoću **BOJA 5** i **OBOJI**. Pomaknite se do sljedećeg trokuta u nizu pomoću naredbi: **GORE, IDI 10, DOLJE**, pa obojite drugi trokut narančasto pomoću: **BOJA 14, OBOJI**. Tako dovršite bojenje ostalih trokuta.

Slika 1.10: Vježba iz udžbenika *Informatika<sup>+</sup> 5*

U udžbenicima se nalaze zadaci u kojima su zadani ulazni i izlazni podaci i učenik mora osmisliti kako riješiti zadani problem. U radnim bilježnicama se nalazi zadatak (slika 1.11) u kojem moraju spojiti izlaznu vrijednost s naredbom.

Spoji niz naredbi koje će nacrtati sliku.

<p><b>REPEAT 4 [FD 10 PU FD 10 PD]</b></p>	
<p><b>REPEAT 3 [FD 20 RT 60]</b></p>	
<p><b>REPEAT 10 [REPCOUNT*20 LT 90]</b></p>	
<p><b>REPEAT 4 [BK 40 LT 72]</b></p>	

Slika 1.11: Zadatak iz radne bilježnice *Informatika 5*

Kroz zadatke učenici uvježbavaju korištenje naredbi za ponavljanje. Zadan im je problem te pomoću već poznatih naredbi moraju osmisliti rješenje zadatka.

**\*\*\* VJEŽBA: NIZOVI**

Nacrtajmo niz kvadrata koristeći se *Editorom* i naredbama za ponavljanje.

**TO KVADRAT**

**REPEAT 4 [FD 50 RT 90]**

**END**

**TO NIZ\_KVADRATA**

**KVADRAT**

**FD 50**

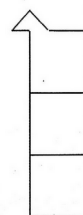
**KVADRAT**

**FD 50**

**KVADRAT**

**FD 50**

**END**

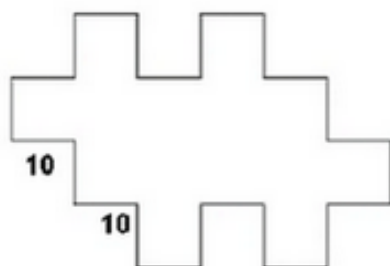


Slika 2a.22. Niz kvadrata

Slika 1.12: Zadatak iz udžbenika *Informatika 5*

Zadaci u radnoj bilježnici su osmišljeni na način da učenici samostalno razvijaju rješenje zadanog problema, koristeći se ulaznim i izlaznim podacima i ponavljanjima.

a) Ivica voli slagati slagalice.  
Nacrtaj Kornijem ovakav oblik:



b) Zatim složi četiri takva oblika i oboji ih:



c) Smislite različite dijelove slagalice pa od njih složite cijelu slagalicu. Pritom pazite da se dijelovi mogu umetati jedan u drugi. *Uputa:* Ako na jednoj strani napravite izbočinu, s druge strane mora biti isto takva udubina.

Slika 1.13: Zadatak iz radne bilježnice *Informatika+ 5*



## Algoritam

Uspoređujući udžbenike i kurikulum, došli smo do zaključka kako nastavni sadržaj **nije** usklađen s odgojno-obrazovnim ishodom. U udžbeniku *Informatika<sup>+</sup> 5* [4] uopće nije objašnjen pojam algoritma. Najviše pažnje algoritmu posvećeno je u udžbeniku *Informatika 5* [6].

Učenici će nakon završenog petog razreda osnovne škole opisivati pojam algoritma i samostalno organizirati neke aktivnosti u oblik algoritma. Jedan od takvih zadataka, u kojem algoritam moraju zapisati riječima, nalazi se u radnoj bilježnici *Informatika 5* [6] (slika 1.14). Na slici 1.15 nalazi se zadatak u kojem algoritam moraju prikazati grafički, to jest, dijagramom tijeka.

Napiši algoritam za pripremu kade/vode za kupanje. Obrati pozornost na to da temperatura vode bude primjerena.

Slika 1.14: Zadatak iz radne bilježnice *Informatika 5*

Nacrtaj dijagram tijeka za problem prebacivanja podataka na memorijski ključić. Stavi naglasak na mogućnost da nema dovoljno prostora za pohranu – u dijagramu tijeka jasno naznači što je potrebno u tom slučaju napraviti. Uključi i radnje poput priključivanja memorijskog ključića, kopiranja podataka, lijepljenja podataka itd.

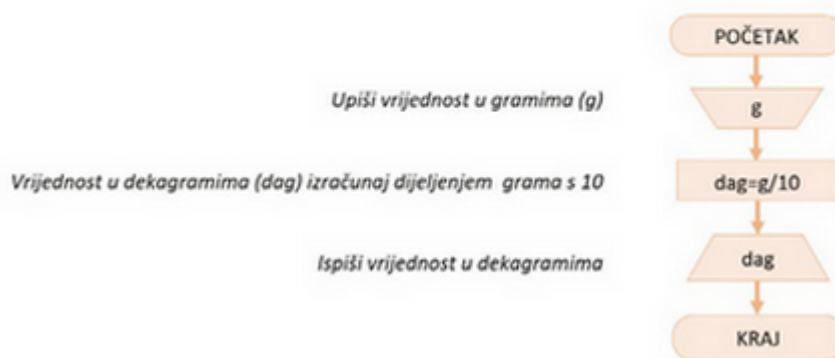
Slika 1.15: Zadatak iz radne bilježnice *Informatika 5*

Odgojno-obrazovni ishod u kojem učenik analizira jednostavan problem, predviđa korake za rješavanje toga problema i prikazuje ih (grafički, usmeno ili tekstom), predviđajući redoslijed njihova izvršavanja, pronašla sam samo u udžbeniku *Informatika 5* [6].

Nastavni sadržaj u odabranim materijalima ne pokriva odgojno-obrazovni ishod u kojem učenici kritički provjeravaju ispravnost svojega algoritma, tako da uspoređuju očekivano rješenje problema s dobivenim rješenjem.

U udžbeniku *#mojportal5* [2] nalazi se zadatak (slika 1.16) u kojem učenici, ovisno o ispravnosti rješenja, preuređuju i ponovo testiraju rješenje.

**Zadatak:** Marija je odlučila iznenaditi mamu te joj ispeći rođendansku tortu. Potražila je jednostavan recept i otišla u nabavu namirnica. Već je na samom početku naišla na problem. Mase svih potrebnih namirnica kao što su šećer, brašno i maslac izražene su u gramima, a u receptu su sve vrijednosti izražene u dekagramima. Pomozimo Mariji i napišimo program koji će izračunati koliko vrijednost mase neke namirnice, izražena u gramima, iznosi u dekagramima.



Napišite računalni program prema algoritmu definiranom na slici.

Slika 1.16: Zadatak iz udžbenika *#mojportal5*

# Poglavlje 2

## Šesti razred

### 2.1 Odgojno-obrazovni ishodi

Učenici će, nakon šeste godine učenja predmeta Informatika, u domeni Računalno razmišljanje i programiranje [1], stvarati, pratiti i preuređivati programe koji sadrže strukture grananja i uvjetnog ponavljanja. Isto tako, učenici će predviđati ponašanje jednostavnih algoritama koji mogu biti prikazani dijagramom, riječima govornog jezika ili programskim jezikom te će razmatrati i rješavati složene probleme, rastavljajući ih na niz potproblema.

### 2.2 Programski jezici u šestom razredu

#### Python

Nastavni sadržaj većinom ispunjava zadane odgojno-obrazovne ishode. Učenik razumije problem i prepoznaje ulazne vrijednosti te algoritamske strukture koje se upotrebljavaju za rješavanje problema. U udžbeniku *Informatika 6* [12], algoritam grananja objašnjen je pomoću kornjačine grafike. Kako bi učenicima bolje objasnili algoritamske strukture slijeda, grananja i ponavljanja, autori udžbenika *Informatika<sup>+</sup> 6* [10] i *#mojportal6* [8] koriste dijagram tijeka. Učenici u radnim bilježnicama imaju zadatke u kojima moraju nadopuniti dijagram tijeka ili pak program, koji odgovara zadanom dijagramu tijeka ili algoritmu (slike 2.1 i 2.2).

Dopuni dijagram toka i program za zadani algoritam. Dopuni tablicu za zadane vrijednosti od x.

Dopuni rečenicu: Ovaj program služi za \_\_\_\_\_.

Slika 2.1: Zadatak iz udžbenika *Informatika<sup>+</sup> 6*

**ulaz(x)**  
**ako je**  $x \% 2 == 0$ :  
     **izlaz**(x je parni.)  
**inače:**  
     **izlaz**(x je neparni.)

x	ispis
14	
24	
20	
37	
10	
	Broj je parni.
	Broj je neparni.

```
x = _____
x = int(x)
if _____ == 0:
    _____ (x, 'je parni.')
else:
    _____ (x, 'je neparni.')
```

Slika 2.2: Zadatak iz udžbenika *Informatika<sup>+</sup> 6*

Zadaci u radnoj bilježnici su osmišljeni na način da učenici samostalno planiraju i pišu niz naredbi te odabiru neku algoritamsku strukturu: slijed, grananje, ponavljanje. U udžbeniku *#mojportal6* [8], na kraju cjeline nalazi se projektni zadatak, slika 2.3. Učenici, uz pomoć nastavnika, zadatak rastavljaju na manje dijelove: analiziraju problem, korake rješavanja prikazuju riječima te zatim pišu program. U radnim bilježnicama *Informatika 6* [13] i *Informatika<sup>+</sup> 6* [11] nalaze se problemi koje učenici samostalno rješavaju, to jest, pišu program.

**PROJEKTI ZADATAK 1**

Učitelji cijelu godinu ocjenjuju učenike različitim elementima i komponentama vrednovanja. Velik se broj škola danas služi mrežnom aplikacijom za vođenje školske dokumentacije koja uključuje i evidentiranje učenikovih ocjena. U sklopu aplikacije svakom se učeniku automatski izračunava srednja ocjena pojedinog predmeta tijekom školske godine.

No, učitelji uz srednju ocjenu predmeta često moraju provoditi razne analize te izračunavati srednje ocjene, npr. pisane provjere znanja, usmenog odgovaranja, izrade projektnog zadatka pa do srednje ocjene izostanaka (opravdanih ili neopravdanih). Bez obzira na to o kojoj je namjeni riječ, uvijek je osnovni zadatak izračunati srednju vrijednost od zadanih brojčanih vrijednosti. Učiteljica te zamolila da joj pomogneš i izradiš računalni program kojim će se brže i lakše izračunati prosječna vrijednost zadanih ocjena, izostanaka i slično.

izrada dije

Slika 2.3: Zadatak iz udžbenika *#mojportal6*

Jedan od odgojno-obrazovnih ishoda je da će učenici predviđati ponašanje algoritma i provjeravati ispravnost algoritma. Zadaci koji se nalaze u radnoj bilježnici *#mojportal6* [9] i *Informatika<sup>+</sup> 6* [11] su osmišljeni na takav način. Učenici imaju zadane izlazne vrijednosti te moraju promatrati ponašanje programa (slike 2.4 i 2.5). U udžbeniku *Informatika 6* [12] detaljno je objašnjena svaka vježba i rezultat programa. Na nastavnicima je da modeliraju vježbe i zadatke, kako bi se ispunio ovaj odgojno-obrazovni ishod.

Napišite program koji za dva unesena broja provjerava da li je prvi broj (a) djelitelj drugog broja (b). Djelovanje algoritma opisano je i s pomoću dvaju testnih primjera.

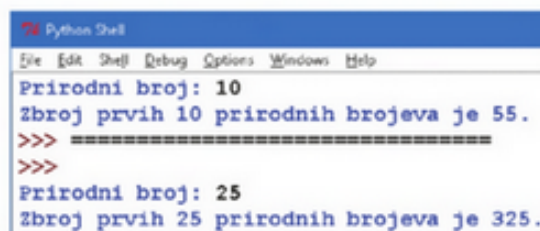
Primjeri testiranja

```
Upiši prvi broj: 7
Upiši drugi broj: 35
7 je djelitelj broja 35
>>>
```

```
Upiši prvi broj: 5
Upiši drugi broj: 26
5 nije djelitelj broja 26
>>>
```

Slika 2.4: Zadatak iz udžbenika *#mojportal6*

Napiši program koji izračunava zbroj prvih  $n$  troznamenkastih brojeva. Na početku programa u varijablu  $n$  upisuje se koliko brojeva treba zbrojiti.



```
Python Shell
File Edit Shell Debug Options Windows Help
Prirodni broj: 10
Zbroj prvih 10 prirodnih brojeva je 55.
>>> *****
>>>
Prirodni broj: 25
Zbroj prvih 25 prirodnih brojeva je 325.
```

Slika 2.5: Zadatak iz radne bilježnice *Informatika<sup>+</sup> 6*

Niti u jednom od promatranih materijala, učenik ne predviđa testne primjere niti kritički provjerava ispravnost rješenja ili ga preuređuje. U prvom projektnom zadatku u *#mojportal6* [8], spominje se testiranje i koji brojevi ne odgovaraju te da je potrebno preuređivanje koda, ali ne potiče se učenika da samostalno to istraži. Ovaj odgojno-obrazovni ishod učenik može ostvariti uz potporu nastavnika.

## Logo

Kod razrade ishoda uočili smo kako **nisu** svi odgojno-obrazovni ishodi ispunjeni. U dostupnim materijalima se nalazi velik broj zadataka i vježbi. Učenici interpretiraju problem i prepoznaju ulazne vrijednosti. U udžbenicima su obrađeni algoritmi slijeda, ponavljanja i grananja. Veća pozornost tim algoritmima pridodana je u udžbeniku *Informatika<sup>+</sup> 6* [10], gdje su objašnjeni pomoću dijagrama tijekom algoritama (slika 2.6).

Dopuni program koji crta pravokutnik ako je zbroj duljina stranica  $A$  i  $B$  manji od 100, a inače crta kvadrat stranice duljine  $A$ .

```

TO PRAVOKUTNIK  ___  ___
REPEAT  ___ [FD  ___  RT 90
           FD  ___  RT  ___ ]
END

TO PRAVI  ___  ___
TEST  :A + :B < 100
IFT [ _____ :A :B ]
IFF [ _____ ]
END

```

*Uputa:*

Pozoveš li program PRAVOKUTNIK s dvjema istim ulaznim vrijednostima, nacrtat će se kvadrat. Npr. PRAVOKUTNIK 40 40 nacrtat će kvadrat stranice duljine 40.

Što će nacrtati PRAVI 55 30?

Što će nacrtati PRAVI 40 70?

Što će nacrtati PRAVI 50 50?

Slika 2.6: Zadatak iz radne bilježnice *Informatika<sup>+</sup> 6*

### \*\*\* ZADATAK: KUTIJE

U poštanskim kamionima uvijek je zbog sigurnosti paketa važno kako se kutije slažu. Nacrtaj niz kutija i posloži ih od manje prema većoj.



Slika 2.7: Zadatak iz udžbenika *Informatika 6*

Učenici kroz vježbe planiraju i slažu naredbe primjenom algoritamskih struktura. Jedna od takvih vježbi prikazana je na slici 2.8. U radnim bilježnicama nalaze se zadaci u kojima učenici samostalno nadopunjuju naredbe. Na kraju lekcije nalaze se zadaci u kojima učenici samostalno rješavaju probleme. Moraju analizirati problem, osmisliti rješenje te napisati program.

**Vježba 6.** Prvi u redu

Na početku sata tjelesne i zdravstvene kulture učenici trebaju stati „u vrstu“ po visini. Tri najbolja prijatelja Matej, Ivan i Luka, među najvišim učenicima u razredu, svaki se sat prepiru tko će stati prvi u red. Napišite program koji će ispisati ime učenika koji treba stati prvi u red ako su ulazne vrijednosti programa PRVI :M :I :L njihove visine koje su izmjerili danas ujutro.

Primjeri poziva programa:

Ulazne vrijednosti programa PRVI :M :I :L redom su visine Mateja, Luke i Ivana. U primjeru ❶ najveća je posljednja ulazna vrijednost, pa se ispisuje da je prvi u redu Luka. U ❷ primjeru najveća je prva ulazna vrijednost, pa se ispisuje da je prvi u redu Matej, a u ❸ primjeru najveća je srednja ulazna vrijednost, pa se ispisuje da je prvi u redu Ivan.

```
❶ PRVI 150 151 152
   PRVI U REDU: LUKA
❷ PRVI 153 151 152
   PRVI U REDU: MATEJ
❸ PRVI 153 154 152
   PRVI U REDU: IVAN
```

Slika 2.8: Vježba iz udžbenika *Informatika<sup>+</sup> 6*

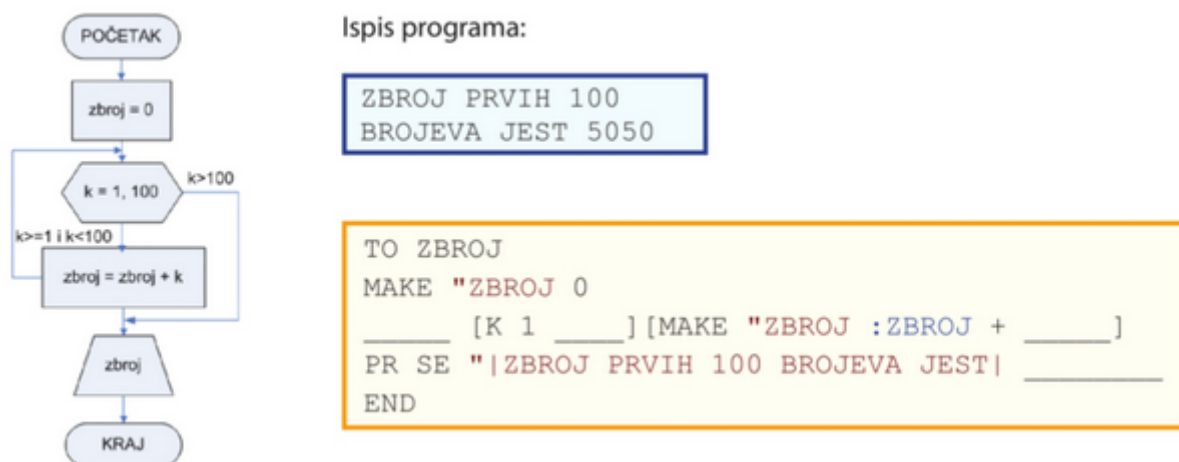
Učenici testiraju program sa zadanim ulaznim vrijednostima, ali samo nastavni sadržaj u *Informatika 6* [12] potiče učenika na predviđanje ponašanja. Jedna lekcija posvećena je testiranju programskih rješenja. Učenika upozoravaju na moguće greške pri pisanju programa. Uz pomoć nastavnika, učenik može ostvariti ovaj odgojno-obrazovni ishod. Potrebno je motivirati učenike na razmišljanje o problemu i da koriste papir i olovku, kako bi provjerili korake programa.

Dopuni program koji računa zbroj prvih 100 prirodnih brojeva.

*Uputa:* Program nema ulaznih vrijednosti. Zbroj prirodnih brojeva računa se slično kao da to radiš „pješke“. Računa se redom:  $1 + 2 = 3$ ,  $3 + 3 = 6$ ,  $6 + 4 = 10$ ,  $10 + 5 = 15$  itd., dok se na kraju zbroju prvih 99 prirodnih brojeva ne doda 100. U varijabli ZBROJ pamti se trenutni zbroj brojeva, a varijabla petlje dodaje se zbroju. Promotri dijagram toka.

Slika 2.9: Zadatak iz radne bilježnice *Informatika<sup>+</sup> 6*



Slika 2.10: Zadatak iz radne bilježnice *Informatika<sup>+</sup> 6*

Proučavajući nastavni sadržaj, niti u jednom udžbeniku (*Informatika<sup>+</sup> 6* [10] i *Informatika 6* [12]) nismo vidjeli zadatke ili pitanja u kojima učenici moraju odabrati odgovarajuće ulazne primjere, već su im oni ponuđeni. Nigdje se ne traži od učenika da kritički provjeri ispravnost rješenja te da ga preuredi.

## Problem

Ovoj skupini odgojno-obrazovnih ishoda najveću pažnju su posvetili autori udžbenika *Informatika<sup>+</sup> 6* [10]. Jedna cjelina posvećena je raščlanjivanju problema, dok je u udžbeniku *Informatika 6* [12] i *#mojportal6* [8], jedna lekcija posvećena tome.

U *#mojportal6* [8], kroz projektni zadatak prikazana je raščlamba problema. Prvi problem se rješava u grupi, tako da svaki član ima svoje zaduženje, a drugi projektni zadatak učenici rješavaju samostalno, na način da podijele poslove kao i dok su radili u grupi. U *Informatika 6* [12] dana uputa kako se rješava problem kroz pet koraka: definiranje problema, prijedlozi rješenja, obrada rješenja, provedba rješenja i pregled rezultata. U udžbeniku *Informatika<sup>+</sup> 6* [10], kroz primjere je prikazano kako se problem može riješiti u četiri koraka: analiza problema, plan rješavanja, raščlanjivanje problema na manje potprobleme, poopćenje postupka rješavanja. Time je ostvaren odgojno-obrazovni ishod, to jest, učenik će opisivati složeni problem i prepoznat će korake u rješavanju problema.

Radne bilježnice *Informatika 6* [13] i *Informatika<sup>+</sup> 6* [11] potiču učenike da vježbaju rješavanje problema. Primjer takvog zadatka je sljedeća vježba, slika 2.11.

**• VJEŽBA: 24**

Neka svatko od nas na papiru zapiše četiri različita broja od 1 do 10, npr. brojeve 1, 2, 4 i 6. Cilj je igre kombiniranjem tih brojeva i osnovnih računskih operacija (zbrajanja, oduzimanja, množenja i dijeljenja) doći do rezultata 24.

Slika 2.11: Zadatak iz udžbenika *Informatika 6*

Kroz zadatke učenik će, uz pomoć nastavnika, razvijati plan rješavanja zadanog problema, tako da problem raščlani na manje potprobleme, s kojim se već susreo i zna ih riješiti. Učenik će razmisliti na koji način implementirati rješenje potproblema u rješenje složenog problema, tako da ga izmijeni ili prilagodi, ako je potrebno. Zadaci u radnoj bilježnici *Informatika 6* [13] i *Informatika<sup>+</sup> 6* [11] te projektni zadaci u udžbeniku *#mojportal6* [8] odnose se na probleme iz stvarnog života. Omogućavaju učeniku da pokaže kako rastaviti problem te da primjenom poznatih koraka riješi zadatak. Kroz igru, slika 2.12, učenici vježbaju rješavanje problema.

Pokreni igricu Sokoban na mrežnim stranicama <https://sokoban-game.com/>. Pomozi čovjeku odgurati sve kamenje. Po potrebi preuredi i preispitaj svoj algoritam dok ne dođeš do rješenja.

Slika 2.12: Zadatak iz radne bilježnice *Informatika 6*

# Poglavlje 3

## Sedmi razred

### 3.1 Odgojno-obrazovni ishodi

Na kraju sedme godine učenja predmeta Informatika [1], učenik će razvijati algoritme za rješavanje različitih problema, koristeći se nekim programskim jezikom, pri čemu se koristi prikladnim strukturama i tipovima podataka. Drugi odgojno-obrazovni ishod je primjenjivanje algoritma (sekvencijalnog) pretraživanja pri rješavanju problema. Učenik će dizajnirati i izraditi modularne programe, koji sadrže potprograme u programskom jeziku te će se koristiti simulacijom pri rješavanju nekog, ne nužno računalnog, problema.

### 3.2 Programski jezici u sedmom razredu

#### Python

Proučavajući dostupni nastavni sadržaj, zaključujemo kako su gotovo svi ishodi ostvareni. Učenici koji koriste *Informatika 7*, jednostavne tipove podataka naučili su već u šestom razredu, a sada su ih ponovili. Jedno poglavlje u udžbeniku posvećeno je ponavljanju naučenog nastavnog sadržaja. Učenici će se na kraju sedme godine osnovnoškolskog obrazovanja služiti složenim tipovima podataka: lista, tuple, set i rječnik. Učenici koji se služe udžbenikom *#mojportal7* [14] i *Informatika<sup>+</sup> 7* [16], upoznati su s jednostavnim tipovima podataka te s jednim složenim tipom podataka – listom (slike 3.1 i 3.2).

**► Jednostavni tipovi podataka u Pythonu**

U Pythonu razlikujemo sljedeće jednostavne tipove podataka:

- **cijeli broj**, npr. 2, 3, 0, -7
- **brojevi s decimalnom točkom**, npr. 3.14, 2.0
- **znakovni niz ili string**, npr. 'mačka', 'Učimo tipove podataka.' Znakovni niz može sadržavati slova, brojeve, druge znakove (., :, ;, ", ...), pa i razmake.

Slika 3.1: Jednostavni tipovi podataka opisani u udžbeniku *Informatika<sup>+</sup> 7*

► Tip podataka koji omogućuje pohranjivanje više podataka različitog tipa (brojeva, riječi, ...) zove se **lista**. Podatci koji čine listu pišu se unutar uglatih zagrada odvojeni zarezom.

Slika 3.2: Složeni tipovi podataka opisani u udžbeniku *Informatika<sup>+</sup> 7*

Učenici su upoznati s tipovima podataka te će moći opisati način rješavanja problema naredbama u *Pythonu*, pomoću različitih tipova podataka.

U radnim bilježnicama *Informatika 7* [19] i *Informatika<sup>+</sup> 7* [17] nalaze se zadaci u kojima učenici analiziraju problem, odabiru strategiju i rješenje zapisuju u obliku programa, koristeći se odgovarajućim tipovima podataka (slika 3.3). U radnoj bilježnici *#mojportal7* [14] nalaze se zadaci u kojima učenici moraju razmisliti o problemu i nadopuniti program odgovarajućim naredbama.

Napiši program koji će tražiti opis sljedećih podataka: ime, prezime, ulica, kućni broj, mjesto stanovanja i poštanski broj. Ti će se podatci upisati u listu `osobni_podatci`.

Slika 3.3: Zadatak iz radne bilježnice *Informatika 7*

Učenici će, koristeći funkcije za rad s listama, riješiti zadatke. Prepoznavat će potrebu za upotrebom složenog tipa podataka. Primjeri zadataka iz udžbenika *#mojportal7* [14] i *Informatika 7* [18] nalaze se na slikama 3.4 i 3.5.

**Zadatak 3.** Napišite računalni program kojim ćemo odrediti broj riječi koje sadržava upisana rečenica.

**Primjer testiranja**

```
Unesi rečenicu: Znanje je moć.  
Rečenica ima 3 riječi.  
>>>
```

**Računalni program**

```
r=input('Unesi rečenicu: ')  
b=0  
for i in range(len(r)):  
    if r[i]==' ':  
        b=b+1  
print('Rečenica ima',b+1,'riječi.')
```

**Objašnjenje**

Promotrimo li nekoliko rečenica možemo zaključiti da će broj riječi neke rečenice uvijek biti za jedan veći od broja praznina (znak ' ') u toj rečenici. Rješenje zadatka svodi se na provjeravanje (`if r[i]==' '`) i prebrojavanje pojavljivanja praznina u rečenici te uvećavanje te vrijednosti pri ispisivanju rezultata.

Slika 3.4: Zadatak iz udžbenika *#mojportal7*

**#### VJEŽBA: ŠIRINE GOLOVA**

U puno sportova imamo dvije ekipe koje pokušavaju zabiti gol protivniku. Dimenzije golova su različite i ovise o sportu. U listu golovi upisano je nekoliko širina gola u metrima (tip brojičanog podatka `FLOAT`). Pokušat ćemo pronaći koja je širina gola povezana s kojim sportom. Potrebno je zbrojiti vrijednosti u listi te ispisati odgovarajuću poruku. Pretpostavlja se da ne znamo broj članova u listi.

**Rješenje:**

```
golovi=[7.32,3,1.8]
uk_sirina=0
cl_liste=len(golovi)
for a in range(cl_liste):
    sirina=golovi[a]
    uk_sirina+=sirina
print('Kad zbrojimo širine svih upisanih golova
dobjemo duljinu od',uk_sirina,'metara.')
```

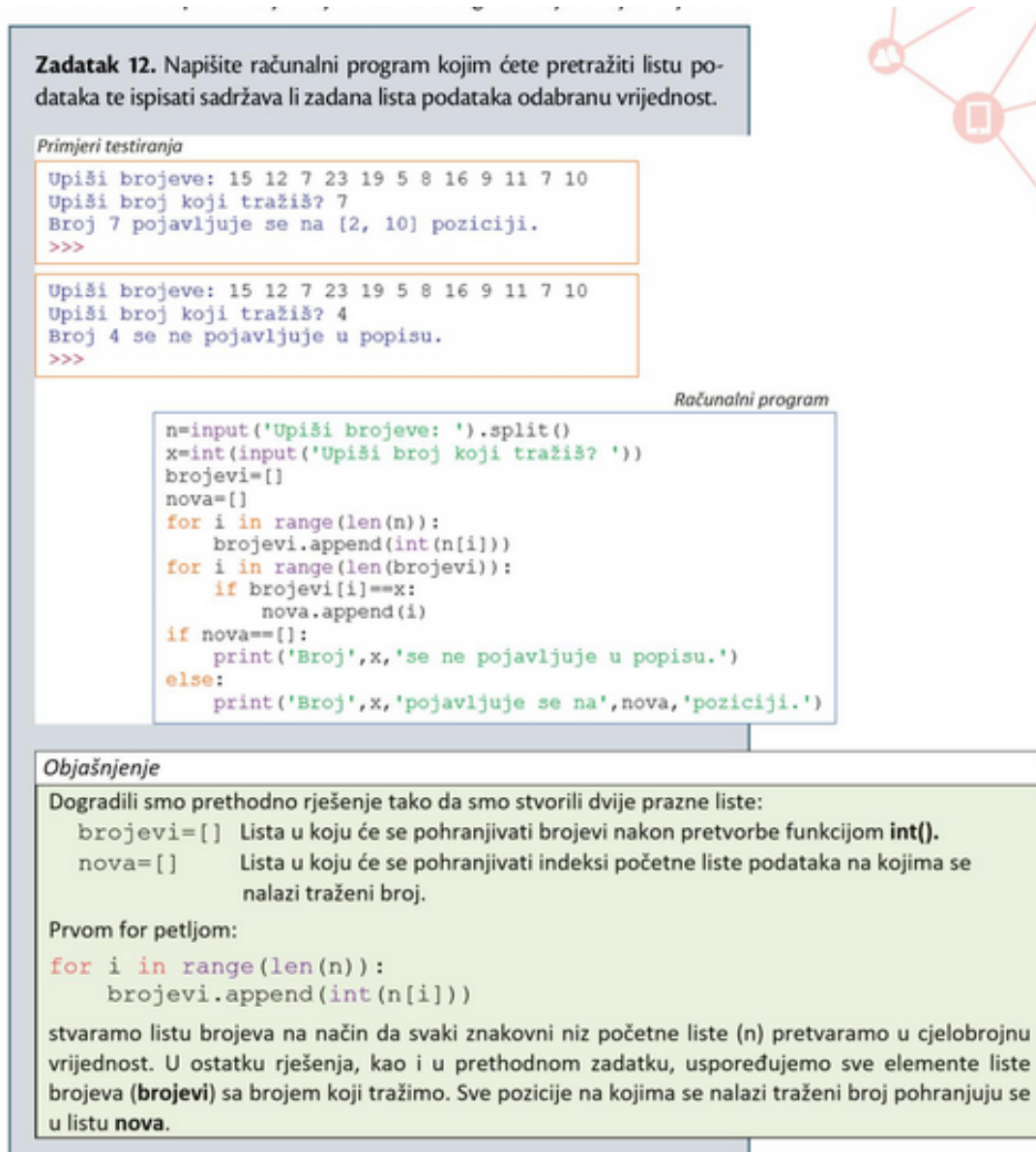
Prvo je potrebno postaviti varijablu u kojoj će se zbrajati vrijednosti na nulu (`uk_sirina=0`) te provjeriti broj članova liste (`cl_liste=len(golovi)`). Nakon toga će program izdvojiti jednog po jednog člana iz liste (`sirina=golovi[a]`) te njegovu vrijednost dodavati varijabli `uk_sirina+=sirina`. To će se ponavljati unutar FOR petlje tako dugo dok ne stigne do zadnjeg člana liste (`for a in range (cl_liste):`).

Slika 3.5: Vježba iz udžbenika *Informatika 7*

U radnoj bilježnici nalaze se zadaci u kojima je zadan program, koji je potrebno nadopuniti, te testni primjer. Ovdje učenici mogu provjeriti ispravnost rješenja te ga preurediti, po potrebi. U radnoj bilježnici *Informatika<sup>+</sup> 7* [17] nalaze se zadaci u kojima učenici moraju napisati program te im je zadan jedan testni primjer. U nastavnom sadržaju koji izdaje Profil Klett ([18], [19]), nisam naišla na primjer ili zadatak u kojem učenik mora provjeriti ispravnost rješenja i, po potrebi, ga preurediti. Ovaj ishod moguće je ostvariti ako se nakon riješenog zadatka postave određena pitanja, koja bi dovela do spoznaje kako je rješenje potrebno provjeriti.

Ishod u kojem učenik primjenjuje algoritam (sekvencijalnog) pretraživanja pri rješavanju problema, u materijalima nakladnika Profil Klett **nije** ispunjen. Učenici se ne susreću s problemom u kojem je potrebno pretraživanje. U udžbeniku *#mojportal7* [14] i *Informatika<sup>+</sup> 7* [18], jedno poglavlje posvećeno je pretraživanju podataka.

Kroz primjere objašnjena je ideja algoritma pretraživanja, kojeg je potrebno primijeniti za rješavanje zadanog problema (slika 3.6).



**Zadatak 12.** Napišite računalni program kojim ćete pretražiti listu podataka te ispisati sadržava li zadana lista podataka odabranu vrijednost.

*Primjeri testiranja*

```
Upiši brojeve: 15 12 7 23 19 5 8 16 9 11 7 10
Upiši broj koji tražiš? 7
Broj 7 pojavljuje se na [2, 10] poziciji.
>>>
```

```
Upiši brojeve: 15 12 7 23 19 5 8 16 9 11 7 10
Upiši broj koji tražiš? 4
Broj 4 se ne pojavljuje u popisu.
>>>
```

*Računalni program*

```
n=input('Upiši brojeve: ').split()
x=int(input('Upiši broj koji tražiš? '))
brojevi=[]
nova=[]
for i in range(len(n)):
    brojevi.append(int(n[i]))
for i in range(len(brojevi)):
    if brojevi[i]==x:
        nova.append(i)
if nova==[]:
    print('Broj',x,'se ne pojavljuje u popisu.')
else:
    print('Broj',x,'pojavljuje se na',nova,'poziciji.')
```

*Objašnjenje*

Dogradili smo prethodno rješenje tako da smo stvorili dvije prazne liste:

- brojevi=[] Lista u koju će se pohranjivati brojevi nakon pretvorbe funkcijom `int()`.
- nova=[] Lista u koju će se pohranjivati indeksi početne liste podataka na kojima se nalazi traženi broj.

Prvom for petljom:

```
for i in range(len(n)):
    brojevi.append(int(n[i]))
```

stvaramo listu brojeva na način da svaki znakovni niz početne liste (n) pretvaramo u cjelobrojnu vrijednost. U ostatku rješenja, kao i u prethodnom zadatku, uspoređujemo sve elemente liste brojeva (**brojevi**) sa brojem koji tražimo. Sve pozicije na kojima se nalazi traženi broj pohranjuju se u listu **nova**.

Slika 3.6: Zadatak iz udžbenika *#mojportal7*

U udžbeniku *Informatika<sup>+</sup> 7* [16] dan je algoritam i program, kako bi učenici lakše razumjeli rješenje problema.

Učenici kroz zadatke razmišljaju o kriterijima za pretraživanje, prepoznaju različite algoritamske strukture kojima se provodi algoritam pretraživanja. Uočavaju kako je lakše zadane podatke zapisati u listu te onda pretraživati listu (slika 3.7). U udžbeniku se spominje metoda `find()`, pomoću koje pronalaze mjesto na kojem se nalazi neka vrijednost u zadanom nizu.

### **Vježba 6.** Prvi u vrsti

Na početku sedmog razreda na satu tjelesne i zdravstvene kulture učenici se moraju poredati u vrstu po visini. Učiteljica je izmjerila njihove visine i bilježi ih u listu redom kako su zapisani u imeniku. Napišite program koji će ispisati redni broj u imeniku najvišeg učenika.

*Primjer:*

- ▶ Ulazni podatci: visine učenika u centimetrima: 170, 165, 180, 177, 182, 162.
- ▶ Izlazni podatak: 5.

*Objašnjenje:* Najviši učenik ima 182 cm, a peti je u imeniku, pa program treba ispisati broj 5.

Ako ima više učenika iste visine, traži se redni broj prvog takvog učenika u razredu.

Slika 3.7: Vježba iz udžbenika *Informatika<sup>+</sup> 7*

U radnim bilježnicama *#mojportal7* [15] i *Informatika<sup>+</sup> 7* [17] nalaze se zadaci u kojima učenici primjenjuju i prilagođavaju algoritam zadanom problemu. U radnoj bilježnici *#mojportal7* [15], učenici moraju nadopuniti program kako bi riješili zadani problem (slika 3.8). Učenici na kraju sedme godine osnovnoškolskog obrazovanja utvrđuju važna obilježja algoritma (sekvencijalnog) pretraživanja. Rješavaju probleme iz svakodnevnog života, ali ne moraju samostalno pronalaziti probleme u kojima se može primijeniti algoritam pretraživanja.



Napiši program kojim će se za upisani znakovni niz ispisati pozicija elementa niza na kojem se nalazi zadnje slovo 'a'. Doprni računalno rješenje tako da na predviđena prazna mjesta upišeš naredbe iz tablice s desne strane.

*Primjer testiranja*

```
Unesi znakovni niz: Svako zlo za neko dobro
Zadnje pojavljivanje slova a je na poziciji 11
>>>
```

*Računalni program*

```
a= _____('Unesi znakovni niz: ')
poz=-1
for i in range(_____):
    if a[i]==_____:
        poz=_____
if poz==_____:
    print('Slovo a se ne pojavljuje u riječi.')
else:
    print('Zadnje pojavljivanje slova a je na poziciji',_____)
```

len(a)
'a'
input
poz
i
-1

Slika 3.8: Zadatak iz radne bilježnice *#mojportal7*

U svakom od promatranih udžbenika, potprogramima je posvećeno jedno poglavlje. U *Informatika<sup>+</sup> 7* [16] potprogrami su pojašnjeni pomoću kornjačine grafike. Učenici će opisivati problem i prepoznati u njemu potprobleme te pomoću niza uputa u *Pythonu* opisati potproblem (slike 3.9 i 3.10).

### ✿ VJEŽBA: IZLET

Učenici sedmih razreda krenuli su na izlet. Na tom izletu imali su dogovoren ručak. Mogli su birati između triju različitih menija. S obzirom da su se gotovo svima svidjela sva tri menija, odlučili su napraviti program koji će odabrati jedan meni umjesto njih.

#### Rješenje:

```
def meni1():
    print("Predjelo: juha od šparoga")
    print("Glavno jelo: pohana piletina, mlinci, zelena salata")
    print("Desert: palačinke s čokoladom")

def meni2():
    print("Predjelo: bistra pileća juha s rezancima")
    print("Glavno jelo: vratina s roštilja, riža, miješana salata")
    print("Desert: domaće štrukle")

def meni3():
    print("Predjelo: govedska juha s krpicama")
    print("Glavno jelo: orada, pekarski krumpir, salata od rajčice")
    print("Desert: fritule s čokoladom")

from random import *
sl_broj = randrange(1, 4)
if sl_broj==1:
    meni1()
elif sl_broj==2:
    meni2()
else:
    meni3()
```

Slika 3.9: Zadatak iz udžbenika *Informatika 7*

Potprogram je dio kôda/naredbi glavnog programa koji se izvršava samo kada ga se pozove/aktivira. U vježbi izlet potprogrami su meni 1, 2 i 3. Potprograme se može pozvati koliko puta želimo.

Potprogrami koje smo napravili za potrebe rješavanja zadanog zadatka nazivaju se još i procedure. Procedura je potprogram koji na kraju svojeg izvršavanja ne izbacuje nekakvu vrijednost nego samo izvršava naredbe koje su u njemu.

Svaki potprogram započinjemo pisati s ključnom riječi `def`. Nakon toga slijedi naziv programa te otvorena i zatvorena okrugla zagrada popraćena dvotočjem. Potprogram se poziva/pokreće tako da se upiše njegov naziv i pripadajuće mu zagrade.

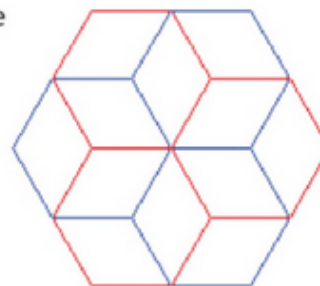
Slika 3.10: Opis potprograma iz udžbenika *Informatika 7*

Zadaci u radnim bilježnicama i udžbenicima koncipirani su na način da učenici analiziraju problem i povezuju module potprograma odgovarajućim parametrima. Zadaci u radnoj bilježnici *#mojportal7* [15] zadani su programom, kojeg učenici moraju nadopuniti, tako da učenici nemaju slobodu da samostalno rastave program na manje potprograme. Zadani su im testni primjeri, tako da mogu provjeriti ispravnost rješenja i preurediti ga, ako je potrebno. U radnim bilježnicama *Informatika<sup>+</sup> 7* [17] i *Informatika 7* [19], zadaci su postavljeni na način da učenici sami odabiru način rješavanja problema, ali ne moraju provjeriti ispravnost rješenja. Da bi se do kraja ostvario ishod, nastavnik mora svojim smjernicama dati do znanja kako je rješenje potrebno provjeriti i, po potrebi, preurediti.

**Napiši program koji crta likove sa zajedničkom točkom tako da naizmjenice mijenjaš boje. Duljina stranice lika  $a$  upisuje se na početku programa.**

**Uputa:** Nazive boja možeš staviti u listu pa pri svakom prolazu petljom uzeti sljedeći član liste.

a) šest šesterokuta sa zajedničkom točkom, naizmjenice crveni i plavi



Slika 3.11: Zadatak iz radne bilježnice *Informatika<sup>+</sup> 7*

## Logo

Kao i kod programskog jezika *Python*, tako i u programskom jeziku *Logo*, učenici sedmog razreda koji koriste udžbenik *Informatika 7* [18], već su se susreli s jednostavnim tipovima podataka i sada ih ponavljaju. U sedmom razredu naučili su liste kao jedan od složenih tipova podataka. Učenici koji koriste udžbenik *Informatika<sup>+</sup> 7* [16], u sedmom se razredu susreću s jednostavnim tipovima podataka te s listom kao složenim tipom podataka. Na kraju godine, učenici će prepoznavati različite tipove podataka programskog jezika *Logo*, kojima se mogu koristiti za pohranjivanje različitih vrsta podataka pri rješavanju problema. Na primjerima uče na koji način riješiti problem naredbama programskog jezika *Logo*, koristeći se različitim tipovima podataka.

**Vježba 4.**

Napišite program `PRVA.SLOVA :L` koji od zadane liste `L` stvara novu listu čiji su članovi prva slova članova liste `L`. Npr. ako pozovemo `PRVA.SLOVA [MATEJ LUKA DOMINIK IVAN]`, onda program treba vratiti listu `[M L D I]`.

**Rješenje:**

Na početku zadajete praznu listu (`MAKE "NOVA [ ]`). U listu `NOVA` dodajete prvo slovo prvog člana liste `L` i brišete ga iz liste. Postupak ponavljate onoliko puta koliko lista ima članova.

```
TO PRVA.SLOVA :L
MAKE "NOVA [ ]
REPEAT COUNT :L [MAKE "NOVA LPUT FIRST FIRST :L :NOVA MAKE "L BF :L]
SHOW :NOVA
END
```

Slika 3.12: Vježba iz udžbenika *Informatika<sup>+</sup> 7*

Učenici rješavaju zadatke koji se nalaze u radnim bilježnicama i na taj način analiziraju problem, odabiru strategiju rješavanja i rješenje prikazuju u obliku programa te se koriste odgovarajućim tipovima podataka.

Zadana je riječ `R` koja ima barem tri slova. Napiši program `ISPIS :L` koji:

- ispisuje novu riječ koja se sastoji od prvog i zadnjeg slova riječi `R`
- ispisuje novu riječ koja se sastoji od prvog, drugog i zadnjeg slova riječi `R`
- ispisuje novu riječ koja se sastoji od predzadnjeg i zadnjeg slova riječi `R`
- ispisuje riječ `R` počevši od trećeg slova
- ispisuje koliko riječ `R` ima slova
- ispisuje riječ koja se sastoji od drugog i četvrtog slova riječi `R`.

Slika 3.13: Zadatak iz radne bilježnice *Informatika<sup>+</sup> 7*

U udžbenicima su obrađene funkcije koje se koriste kod listi. Kroz zadatke, učenici prepoznaju kako je lakše zadane podatke zapisati u listu te, pomoću funkcija koje su namijenjene za rad s listama, riješiti problem (slika 3.14).

**\*\*\* ZADATAK: LISTA PREDMETA U ŠKOLI**

Kreiraj listu predmeta u školi:

- a) ispiši ukupan broj članova liste
- b) ispiši prvi i zadnji član liste
- c) ispiši listu bez zadnjeg člana
- d) unesi n-ti broj i ispiši n-ti član liste.

Slika 3.14: Zadatak iz udžbenika *Informatika 7*

Vidjeli smo kod razrade ishoda kako **nisu** svi odgojno-obrazovni ishodi ispunjeni. Od učenika se ne traži da provjeravaju ispravnost rješenja i da ga, prema potrebi, preuređuju. Ishod je moguće ostvariti uz nastavnikovu asistenciju.

Autori udžbenika *Informatika<sup>+</sup> 7* [16] u jednom poglavlju su opisali i kroz primjere prikazali korištenje algoritma za pretraživanje. Učenik će prepoznati i objasniti ideju algoritma koji je primijenio, kako bi riješio zadatak. Autori udžbenika *Informatika 7* [18] samo su objasnili što je to sekvencijalno pretraživanje i postavili zadatak. Na nastavniku je na koji će način obraditi ovaj zadatak, kako bi učenici ostvarili zadani ishod.

Zadatak koji se nalazi u radnoj bilježnici i udžbeniku *Informatika 7* [19], [18] napominje kako učenik mora koristiti liste kako bi riješio problem. Učenik razmišlja koji kriterij pretraživanja odgovara ovom zadatku. U materijalima *Informatika<sup>+</sup> 7* [17] zadan je veći broj zadataka i učenik mora razmisliti o kriteriju pretraživanja te odlučiti kojim će algoritamskim strukturama provoditi algoritam pretraživanja (slike 3.15 i 3.16).

**Vježba 5. Traži se učenik**

U školu je došao redatelj dječjeg filma i traži od ravnatelja da mu za potrebe filma pronađe učenika sedmog razreda koji je visok 172 cm. Ravnatelj je zamolio učenike informatike da napišu program koji će im pomoći pronaći takvog učenika. Kako biste vi pristupili rješavanju tog zadatka?

Slika 3.15: Vježba iz udžbenika *Informatika<sup>+</sup> 7*

**Algoritam:**

```

postoji = laž
k = 1
n = duljina(visine)
sve dok je k < n i nije postoji:
    ako je traži = k-tom članu liste:
        postoji = istina
        k = k + 1
    ako je postoji = istina:
        izlaz(Učenik je pronađen.)
inače:
    izlaz(Nema učenika te visine.)

```

Slika 3.16: Vježba iz udžbenika *Informatika*<sup>+</sup> 7

Kroz zadatke učenici proučavaju, primjenjuju i prilagođavaju algoritam (sekvencijalnog) pretraživanja zadanom problemu. Primjer takvog zadatka nalazi se na slici 3.17.

### \*\*\* ZADATAK: LISTA PREDMETA

Napiši program koji kreira listu tvojih predmeta u školi. Program kao rezultat treba ispisati koliko imaš predmeta. Neka se svaki izborni predmet pri ispisu liste ne prikazuje.

Slika 3.17: Zadatak iz udžbenika *Informatika* 7

Učenici rješavaju zadatke i na taj način utvrđuju važna obilježja algoritma pretraživanja. U dostupnim materijalima nismo uočili kako se od učenika očekuje da pronađu slične probleme kod kojih se može primijeniti algoritam pretraživanja.

Učenici su već u šestom razredu naučili raščlaniti problem na manje probleme koje znaju riješiti. U ovoj godini učenja nadopunjuju svoje znanje. Kroz zadatke ponavljaju, opisuju problem i prepoznaju potprobleme. Potproblem opisuju naredbama u programskom jeziku *Logo* (slike 3.18 i 3.19).

**Vježba 7.** Pogodi broj

Računalo je zamislilo neki broj od 1 do 100. Napiši program u kojem ćeš pogoditi broj koji je računalo zamislilo. Treba prebrojiti u koliko je pokušaja broj pogođen.

Nakon svakog broja koji se pogađa program ispisuje poruku je li upisani broj manji ili veći od zamišljenog broja. Promotrite primjer na slici 5. Program je zamislio broj 64. Prvi broj koji je upisan tipkovnicom jest 50. Program je ispisao: 50 JE MANJI. jer je  $50 < 64$ . Znači, treba pokušati s većim brojem. Zatim je upisan broj 75. Program je ispisao: 75 je VEĆI. jer je  $75 > 64$ . Postupak se nastavlja sve dok se ne upiše 64. Tada program ispisuje poruku:

BRAVO! ZAMISLIO SAM 64.

POGOĐENO U 7 POKUŠAJA.

Slika 3.18: Vježba iz udžbenika *Informatika<sup>+</sup> 7*

*Igra:*

- Treba zamisliti neki broj.
- Netko od učenika pokušat će pogoditi broj.
- To je prvi pokušaj.
- Sve dok se ne pogodi broj, treba ponavljati:
- Je li broj veći od zamišljenog?  
Reći da je izrečeni broj veći od zamišljenog.
- Je li broj manji od zamišljenog?  
Reći da je izrečeni broj manji od zamišljenog.
- Povećati broj pokušaja za 1.
- Bravo! Zamislio sam... (okrenuti papir sa zamišljenim brojem). Reći u koliko je pokušaja pogođeno.

*Algoritam:*

```

x = slučajan (1, 100)
ulaz(moj)

br = 1
sve dok moj ≠ x:

    ako je moj > x:
        izlaz(moj je veći)

    ako je moj < x:
        izlaz(moj je manji)

br = br + 1
izlaz(Bravo! Zamislio sam x)
izlaz(br)

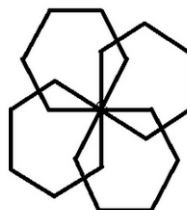
```

Slika 3.19: Vježba iz udžbenika *Informatika<sup>+</sup> 7*



Kroz vježbe i primjere učenici analiziraju problem, raščlanjuju da na manje potprobleme i te potprobleme prikazuju naredbama u programskom jeziku *Logo*. Složeni program rastavljaju na jednostavnije probleme koje zovu potprogramima. Povezuju te potprograme odgovarajućim parametrima.

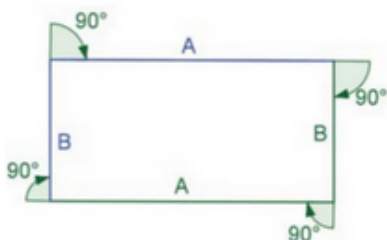
Nacrtaj vjetrenjaču.



Slika 3.20: Zadatak iz radne bilježnice *Informatika 7*

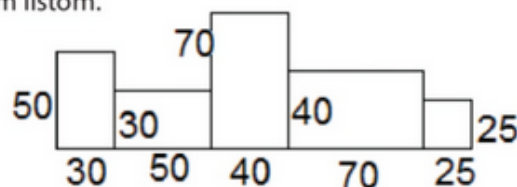
Rješavajući zadatke iz radne bilježnice, učenik samostalno odabire strategiju rješavanja problema, rastavlja ga na manje potprobleme. Rješenje problema prikazuje u obliku programa koji može sadržavati više od jednog modula (slika 3.21). Nismo uočili da se od učenika traži provjera ispravnosti rješenja i preuređenje programa, ako je potrebno, te stoga zaključujemo kako ovaj ishod **nije** do kraja ostvaren.

Dopuni program za crtanje pravokutnika ako su duljine susjednih stranica pravokutnika A i B.



```
TO PRAVOKUTNIK :A :B
REPEAT 2 [FD ____ RT ____ ____ ____]
END
```

Koristeći se programom PRAVOKUTNIK :A :B, dopuni program NIZ :L koji crta niz pravokutnika. Duljine stranica pravokutnika zadane su listom L koja sadržava dvočlane liste. U svakoj dvočlanoj listi dva su broja: duljine susjednih stranica pravokutnika. Dopuni program koji crta niz pravokutnika čije su stranice zadane tom listom.



Slika 3.21: Zadatak iz radne bilježnice *Informatika 7*



## Program

Zadaci su koncipirani na način da učenik opisuje problem i određuje ulazne i izlazne podatke. Kako bi mogao riješiti zadatak, učenik analizira problem, prepoznaje važna obilježja koja opisuju i utječu na rješavanje zadanog problema i razvija model za rješavanje problema (slike 3.22 i 3.23).

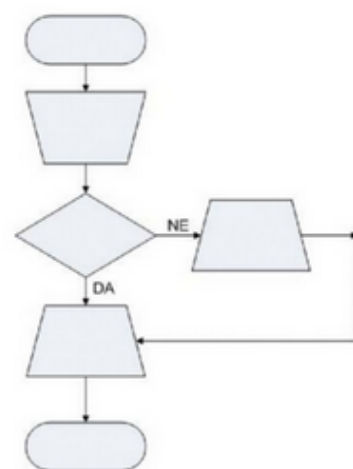
Dopuni program i dijagram toka koji provjerava je li upisani prirodni broj  $n$  parni. Broj je parni ako je njegov ostatak pri dijeljenju s 2 jednak 0.

```

ulaz(x)
ako je  $x \% 2 == 0$ :
    izlaz(x je parni.)
inače:
    izlaz(x je neparni.)
  
```


```

x = _____
x = int(x)
if _____ == 0:
    _____ (x, ' je parni. ')
else:
    _____ (x, ' je neparni. ')
  
```



Slika 3.22: Zadatak iz radne bilježnice *Informatika*<sup>+</sup> 7

Kako bi riješio problem učenik razmišlja o pravilima, koristi se grafičkim prikazima koji opisuju, ali moguće je da i predviđaju, ponašanje njegovog modela. Ishod **nije** do kraja ispunjen, jer nismo vidjeli da se od učenika očekuje da analizira ponašanje modela ili da ga kritički vrednuje i predviđa rješenje problema.



**Zadatak 2.** Napišite računalni program koji će provjeriti koliko se puta u nekoj zadanoj riječi pojavljuje slovo 'a'.

**Primjer testiranja**

```
Unesi riječ: Matematika
Slovo a pojavljuje se 3 puta.
>>>
```

**Računalni program**

```
r=input('Unesi riječ: ')
b=0
for i in range(len(r)):
    if r[i]=='a':
        b=b+1
print('Slovo a pojavljuje se',b,'puta.')
```

**Objašnjenje**

S obzirom na to da zadatak traži prebrojavanje slova 'a', u rješenju zadatka moramo upotrijebiti varijablu (b) koja će se ponašati kao brojač. Vrijednost brojača početno postavljamo na nulu (inicijalizacija brojača). Daljnjim izvršavanjem programa, uporabom for petlje, osigurali smo ispitivanje svih slova upisane riječi. Naredbom grananja (if r[i]=='a': ) uspoređujemo svako slovo riječi sa zadanim slovom 'a' te prebrojavamo sve slučajeve u kojima je uvjet ispunjen. Nakon završetka petlje, vrijednost brojača upravo je rješenje našeg zadatka, broj pojavljivanja slova 'a' unutar upisane riječi.

Slika 3.23: Zadatak iz udžbenika #mojportal7

# Poglavlje 4

## Osmi razred

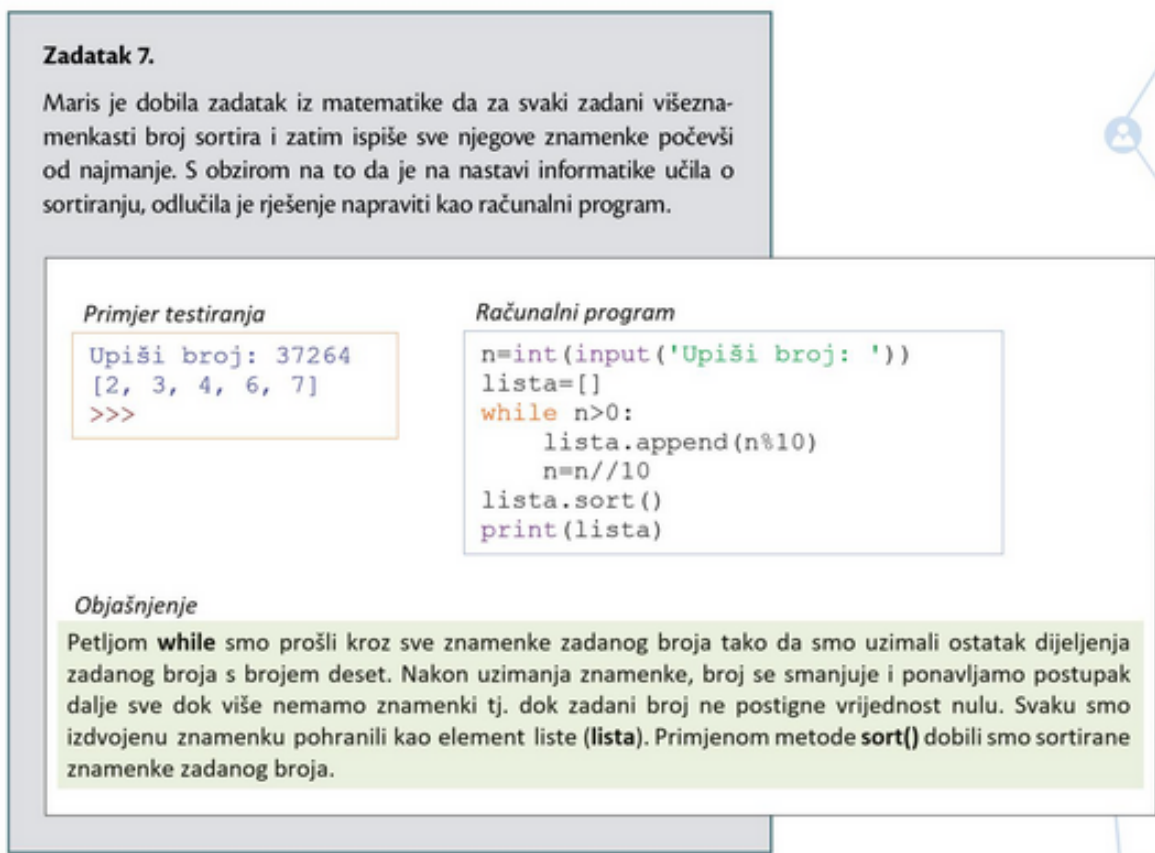
### 4.1 Odgojno-obrazovni ishodi

Odgojno-obrazovni ishodi nakon osme godine učenja predmeta Informatika, u domeni Računalno razmišljanje i programiranje [1], su identificiranje nekih problema, stvaranje programa za rješavanje, dokumentiranje rada programa i predstavljanje djelovanja programa drugima. Učenik će prepoznavati i opisivati algoritam sortiranja, primjenjivati jedan algoritam sortiranja za rješavanje zadanog problema u programskom jeziku. Učenik će prepoznavati i opisivati mogućnost primjene rekurzivnih postupaka pri rješavanju odabranih problema te istraživati daljnje mogućnosti primjene rekurzije.

### 4.2 Programski jezici u osmom razredu

#### Python

U primjerima i vježbama učenici prepoznaju kako je problem učinkovitije riješiti s podacima koji su sortirani. Algoritam sortiranja opisan je riječima i algoritmom. Rješavajući zadatke u radnim bilježnicama, učenici uočavaju potproblem sortiranja. U udžbeniku *Informatika 8* [24] opisano je više načina sortiranja: sortiranje izborom (selection sort) i sortiranje umetanjem (insertion sort). U preostalim udžbenicima obrađuje se samo selection sort te se spominje metoda `sort()` za rad s listama (slika 4.1).



**Zadatak 7.**

Maris je dobila zadatak iz matematike da za svaki zadani višeznamenkasti broj sortira i zatim ispiše sve njegove znamenke počevši od najmanje. S obzirom na to da je na nastavi informatike učila o sortiranju, odlučila je rješenje napraviti kao računalni program.

**Primjer testiranja**

```
Upiši broj: 37264
[2, 3, 4, 6, 7]
>>>
```

**Računalni program**

```
n=int(input('Upiši broj: '))
lista=[]
while n>0:
    lista.append(n%10)
    n=n//10
lista.sort()
print(lista)
```

**Objašnjenje**

Petljom **while** smo prošli kroz sve znamenke zadanog broja tako da smo uzimali ostatak dijeljenja zadanog broja s brojem deset. Nakon uzimanja znamenke, broj se smanjuje i ponavljamo postupak dalje sve dok više nemamo znamenki tj. dok zadani broj ne postigne vrijednost nulu. Svaku smo izdvojenu znamenku pohranili kao element liste (**lista**). Primjenom metode **sort()** dobili smo sortirane znamenke zadanog broja.

Slika 4.1: Zadatak iz udžbenika *#mojportal8*

Zadaci u radnim bilježnicama osmišljeni su na način da učenici opisuju postupak sortiranja (riječima ili grafički) i samostalno primjenjuju algoritam sortiranja za rješavanje zadanog problema u programskom jeziku *Python* (slika 4.2).

Napiši program koji upisuje imena  $n$  učenika u razredu. Za svakog učenika treba upisati ime učenika i postignuti broj bodova. Program treba ispisati učenike poredane prema postignutom broju bodova od onog s najvećim brojem bodova prema onom s najmanjim brojem bodova u obliku tablice koja ima tri stupca: redni broj, ime i broj bodova. Na početku programa upisuje se broj učenika u varijablu  $n$ .

#### Upute:

Podatke o učenicima najbolje je zapamtiti u obliku liste koja sadržava dvočlane liste (ime i broj bodova svakog učenika). Npr. za podatke sa slike 8 lista je: `lista = [['Ivan', 45], ['Matej', 50], ['Ana', 56], ['Maja', 90], ['Biba', 29]]`. Sada pojedini  $k$ -ti učenik ima ime zapisano u članu `lista[k][0]`, a broj bodova u `lista[k][1]`.



```
Python Shell
File Edit Shell Debug Options Window
n = 5
Učenik: Ivan
Bodovi: 45
Učenik: Matej
Bodovi: 50
Učenik: Ana
Bodovi: 56
Učenik: Maja
Bodovi: 90
Učenik: Biba
Bodovi: 29
r.br   ime   bodovi
1 .    Maja  90
2 .    Ana   56
3 .    Matej 50
4 .    Ivan  45
5 .    Biba  29
```

Slika 4.2: Zadatak iz radne bilježnice *Informatika*<sup>+</sup> 8

Učenike se uvodi u rekurzije pomoću primjera iz stvarnog života: školjka morskog puža, Babuška, igra tenisa. Oni promatraju i opisuju njihova zajednička obilježja. Rješavajući primjer, prepoznaju korake rekurzivnog postupka.

U funkciju `ispis(x)` postavljen je uvjet:

```
if x==0:
    return 1
else:
    return ispis (x-1)
```

Kada će  $x$  biti nula (if `x==0`), funkcija prestaje s radom (return 1).

Ako uvjet nije ispunjen, funkcija `ispis(x)` nastavlja s radom (ispisom vrijednosti  $x$ ), ali na način da se vrijednost varijable  $x$  smanjuje za 1 (return `ispis (x-1)`).

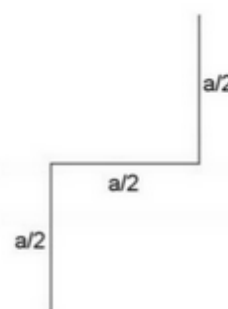
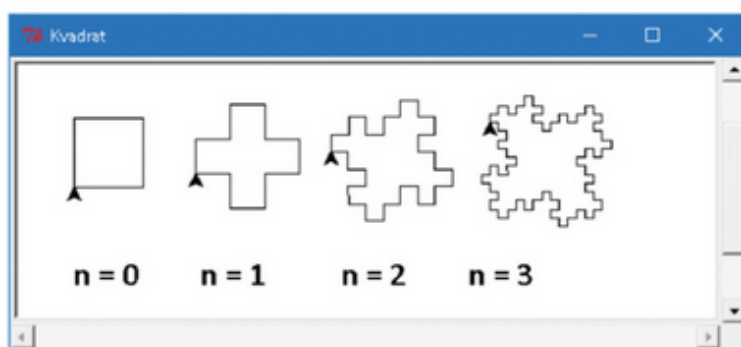
Slika 4.3: Korak rekurzivnog postupka opisan u udžbeniku *Informatika* 8

Velika pažnja je posvećena rekurzijama. Kako bi se učenicima lakše približile, u *#mojportal8* [20] nalaze se primjeri riješeni pomoću kornjačine grafike. U radnim

bilježnicama nalaze se zadaci u kojima učenici moraju analizirati problem, prepoznati rekurzije i na koji će način pozvati rekurzije. Moraju pronaći i predložiti rješenje zadanog problema primjenom rekurzivnog postupka (slika 4.4).

Napiši program koji rekurzivnom procedurom `kvadrat (n, a)` crta kvadrat na sljedeći način:

- $n = 0$ : crta se kvadrat stranice duljine  $a$
- $n = 1$ : svaka stranica kvadrata podijeli se na tri dijela čija je duljina  $a/2$ , kao na slici 18
- $n = 2$ : svaka linija nacrtana na razini  $n = 1$  dijeli se na isti način itd.



Slika 4.4: Zadatak iz radne bilježnice *Informatika*<sup>+</sup> 8

Od učenika se ne traži da istražuju problem u kojem mogu primijeniti rekurzivni postupak. Zaključujemo kako ova skupina odgojno-obrazovnih ishoda **nije** u potpunosti ostvarena.

## Logo

Učenici najprije uče korištenje metode `sort()`, a zatim i `selection sort`. U primjerima prepoznaju kako je problem lakše riješiti ako su podaci već sortirani. Raščlambom problema, dolaze do zaključka kako je jedan od potproblema sortiranje podataka.

### \*\*\* ZADATAK: MJERNE JEDINICE

Napiši program pod nazivom mjerne jedinice koji će kreirati listu većih i listu manjih mjernih jedinica od osnovne jedinice i koji će kao rezultat ispisati osnovnu, najmanju i najveću mjernu jedinicu.

Slika 4.5: Zadatak iz udžbenika *Informatika* 8

Kroz primjere su naučili kako radi algoritam sortiranja te ga mogu opisati riječima. U radnim bilježnicama samostalno rješavaju zadatke, primjenjujući algoritam sortiranja te to zapisuju u obliku programa u programskom jeziku *Logo*.

**Vježba 3.** Tajne poruke (prema zadatku s natjecanja Dabar 2016.)

Bliži se Ivičin rođendan. Odlučio je pozvati samo neke prijatelje, a poziv želi poslati u „razrednu grupu“. Zato je smislio sljedeći način šifriranja pomoću tablice od četiriju stupaca:

Poruku TAJANSTVENA PORUKA upisao je u tablicu po redcima, a u zadnje dvije prazne ćelije stavio je \*. Poruku će poslati tako da je čita po stupcima:

T	A	J	A	• prvi stupac: TNEPK
N	S	T	V	• drugi stupac: ASNOA
E	N	A		• treći stupac: JTAR*
P	O	R	U	• četvrti stupac: AV U*
K	A	*	*	

Ivica će prijateljima poslati poruku: TNEPKASNOAJTAR\*AV U\*.

Prijateljima koje želi pozvati na rođendan objasnio je kako će dešifrirati poruku. Kada je proslava rođendana ako je Ivica poslao poruku: RN UB 7TODJ OU IĐAEST S\*EN UU1A\*? Napišite program za šifriranje poruka koje Ivica šalje prijateljima koje želi pozvati na rođendan. Zatim napišite program za dešifriranje poruka kojim će prijatelji otkriti tajnu poruku.

Slika 4.6: Vježba iz udžbenika *Informatika*<sup>+</sup> 8

Rekurzije se uvode pomoću primjera. Jedan od takvih primjera iz stvarnog života je školjka, tenis, rekurzije u ogledalu. Učenici promatraju i primjećuju zajednička obilježja rekurzivnih fenomena. Kroz primjere, uče korake rekurzivnog postupka. U radnoj bilježnici nalaze se zadaci u kojima moraju analizirati problem, prepoznati osnovni slučaj rekurzije i na koji će način pozvati rekurziju. Učenici smišljaju rješenje, koristeći se rekurzivnim postupkom.

### ► **Rekurzivni program**

Program ili potprogram koji u sebi ima naredbu koja poziva isti taj program (ili potprogram) zove se **rekurzivni program** ili **rekurzija**.

Naredba **STOP** prekida izvršavanje samo onog potprograma u kojem je navedena, a ne i glavni program ili neki drugi, viši potprogram iz kojeg je pozvan potprogram u kojem se nalazi **STOP**. Kažemo da je **STOP** logički kraj programa, a **END** fizički kraj programa (iza **END** nema više naredbi).

Slika 4.7: Definicija rekurzije iz udžbenika *Informatika<sup>+</sup> 8*

Napiši program koji će zbrojiti sve brojeve u prethodnoj listi.

Slika 4.8: Zadatak iz radne bilježnice *Informatika 8*

Ova skupina odgojno-obrazovnih ishoda **nije** do kraja ostvarena. Od učenika se ne traži da istražuje u kojim problemima može, pri rješavanju, koristiti rekurzivni postupak.

## Program

Ovoj skupini odgojno-obrazovnih ishoda najveću pažnju posvetili su autori udžbenika *Informatika 8* [24]. U jednom poglavlju opisali su metodu rješavanja zadatka. U *#mojportal8* [20], kroz projektni zadatak opisana je metoda rješavanja, dok u *Informatika<sup>+</sup> 8* [22] nije puno pažnje posvećeno ovom ishodu. U *Informatika<sup>+</sup> 8* [22] i *#mojportal8* [20], učenik, kroz zadatak, opisuje odabrani problem te predlaže i prikazuje osnovne korake za rješavanje problema. U *Informatika 8* [24], učenik uči kako definirati problem i planiranje. Odlučuje kako će rješavati problem i donositi odluke (slike 4.9 i 4.10).

### ☆ **ZADATAK: PROBLEMI, PROBLEMI**

Razmisli i definiraj problem. Problem može uključivati i gradivo nekih drugih predmeta pa čak i gradivo iz nekih drugih godina.

Slika 4.9: Definiranje problema i planiranje rješavanja iz udžbenika *Informatika 8*



Određuje se ideja projekta. Svatko iznosi svoju ideju projekta/ problema. Kritički se preispituje svaki prijedlog (svatko iznosi pozitivne i negativne strane projekta). Na kraju se utvrđuje tema.

### ☆☆ ZADATAK: PLANIRANJE

Nakon što je tvoj tim odabrao problem, potrebno je planirati. Dogovorite u skupini osnovna pravila rada.

- Kako će skupina rješavati problem?
- Kako će se donositi odluke?
- Prema kojim će se kriterijima nadzirati rad učenika i tko će ga nadzirati? Hoće li to biti skala procjene rada u skupini?

Check lista može biti popis ponašanja koja se opažaju i bilježe kvačicom. Svaki član tima može na taj način za sebe napraviti samoprocjenu.

Slika 4.10: Definiranje problema i planiranje rješavanja iz udžbenika *Informatika 8*

Kroz zadatke, učenici analiziraju problem, predviđaju (planiraju) ulazne vrijednosti programa i razvijaju algoritamsko rješenje u programskom jeziku. U *Informatika 8* [24], učenici uče kako izraditi plan nakon odabira projekta. Razmišljaju koji im je sljedeći korak, prikupljaju informacije potrebne za rješavanje problema, razmišljaju jesu li već negdje vidjeli problem u istom ili malo izmijenjenom obliku te znaju li za sličan problem. Gledaju što im je nepoznato i razmišljaju mogu li iskoristiti sličan problem, koji je već riješen (slika 4.11).

### ☆☆☆ ZADATAK: IZRADA PLANA

Nakon utvrđivanja teme i odabira projekta čiji će se problem rješavati potrebno je izraditi plan.

- Što radimo? Prvo slijedi oluja ideja (engl. *brainstorming*) mogućih rješenja i odabir najprihvatljivijeg rješenja.
- Prikupite informacije potrebne za rješavanje problema i razmijenite ih unutar skupine.
- Jesmo li problem već negdje vidjeli u istom ili malo izmijenjenom obliku?
- Znamo li za sličan problem?
- Pogledajmo što nam je nepoznato. Znamo li za sličan problem koji ima slične nepoznanice?
- Možemo li iskoristiti sličan problem koji je već riješen? Možemo li iskoristiti njegove rezultate ili strategiju rješavanja?

Slika 4.11: Izrada plana iz udžbenika *Informatika 8*

Ovisno o nastavniku, učenik provjerava ispravnost algoritamskog rješenja i, prema potrebi, ga preuređuje. U *Informatika 8* [24], učenici provode plan, provjeravaju svaki korak te pregledavaju rezultat. Provjeravaju što radi, a što ne. Popravljaju program, ako je potrebno (slika 4.12).

#### ☆☆☆☆ ZADATAK: PROVEDBA PLANA I REZULTATI

Nakon detaljne izrade plana provedi plan i pregledaj rezultate.

- Provedi plan rješenja problema.
- Napiši program (nacrtaj prometni znak, riješi fizikalni problem ili izradi aplikaciju).
- Provjeri svaki korak.
- Pregledaj rezultate (što radi, a što ne).

Slika 4.12: Provedba plana iz udžbenika *Informatika 8*

Učenici stvaraju program, ali nije u svakom udžbeniku navedena popratna dokumentacija. U *Informatika 8* [24], kroz primjer, učenici pišu tehničku dokumentaciju koja objašnjava način rada programskog koda te korisničku dokumentaciju, koja na najjednostavniji mogući način objašnjava brzo i učinkovito korištenje svih programa za krajnjeg korisnika. U udžbeniku *#mojportal8* [20], učenicima je, kroz projektni zadatak, objašnjeno na koja pitanja mora odgovoriti dokumentacija (slika 4.13):

- Koji je osnovni cilj aplikacije?
- Kako se aplikacija pokreće?
- Čemu služe pojedini gumbi na definiranim prozorima?
- Kako zatvoriti aplikaciju?

**Zadatak 4.** Pripreмимо dokumentaciju kojom ćemo jednostavnim, a opet jasnim prikazom predstaviti našu aplikaciju. Dokumentaciju možemo pripremati u programu za obradu teksta, programu za izradu prezentacija ili nekom alatu za izradu mrežnih stranica. Dokumentacijom se moraju prikazati osnovne mogućnosti aplikacije, to jest odgovoriti na pitanja:

- Koji je osnovni cilj aplikacije?
- Kako se aplikacija pokreće?
- Čemu služe pojedini gumbi na definiranim prozorima?
- Kako odabrati drugi zadatak za vježbanje?
- Kako zatvoriti aplikaciju?

Poželjno je u dokumentaciji prikazati prozore koji se koriste u aplikaciji. Prema želji, ovisno o načinu prezentiranja, u dokumentaciju se može uključiti i prikaz cjelovitoga računalnog rješenja.

Slika 4.13: Zadatak iz udžbenika *#mojportal8*

U udžbeniku *Informatika<sup>+</sup> 8* [22] ovaj ishod **nije** ispunjen, od učenika se ne traži argumentirano predstavljanje i obrazloženje programskog rješenja problema. U *#mojportal8* [20] spominje se kako se dokumentacija može prezentirati te da je poželjno uključiti i prikaz cjelovitog računalnog rješenja. U *Informatika 8* [24], učenik prezentira rješenje svojim kolegama: prezentacijom, plakatom, video uratkom, multimedijским sadržajem, mrežnom stranicom ili mobilnom aplikacijom (slika 4.14).

**\*\*\*\*\* ZADATAK: PREZENTACIJA RJEŠENJA DRUGIM TIMOVIMA**

- Prezentiraj rješenje problema prezentacijom, plakatom, videouratkom, multimedijalnim sadržajem, mrežnom stranicom ili mobilnom aplikacijom.

Slika 4.14: Prezentacija problema iz udžbenika *Informatika 8*

# Poglavlje 5

## Zaključak

U novom kurikulumu nastavnog predmeta Informatika [1], postoje četiri domene: E-Društvo, Digitalna pismenost i komunikacija, Računalno razmišljanje i programiranje, Informacije i digitalna tehnologija. U svakoj domeni navedeni su odgojno-obrazovni ishodi. Ishodi su iskazani na razini pojedine godine učenja i poučavanja te jasno i nedvosmisleno određuju očekivanja od učenika, a mogu biti određeni kao znanja, vještine, stavovi i vrijednosti.

Analizirali smo programske jezike *Python* i *Logo* u udžbenicima za osnovnu školu i kritički vrednovali nastavni sadržaj. Provjeravali smo usklađenost nastavnog sadržaja s odgojno-obrazovnim ishodima koji se nalaze u domeni Računalno razmišljanje i programiranje.

Nastavni sadržaj u petom razredu potpuno ispunjava zadane odgojno-obrazovne ishode povezane s programskim jezicima *Python* i *Logo*. Skup odgojno-obrazovnih ishoda koji se odnose na algoritme **nije** u potpunosti ispunjen. Jedino je u udžbeniku *Informatika 5* [6] analiziran problem. U udžbenicima *Informatika<sup>+</sup> 5* [4] i *Informatika 5* [6] ne pronalazimo da se od učenika zahtijeva kritička provjera ispravnosti algoritma.

Odgojno-obrazovni ishodi nakon završenog šestog razreda osnovne škole **nisu** u potpunosti ispunjeni. Kod programskog jezika *Python*, ne nalazimo zadatke u kojima učenici moraju predviđati odgovarajuće ulazne primjere i kritički provjeravati ispravnost rješenja te, prema potrebi, preuređivati svoje rješenje. Isti problem je i kod programskog jezika *Logo*. U udžbeniku *Informatika 6* [12] naišli smo na zadatak u kojem učenici moraju predvidjeti ponašanje programa.

Ishodi za sedmi razred osnovne škole **nisu** u potpunosti ispunjeni. Kod programskog jezika *Python*, učenici koji koriste udžbenik *Informatika 7* [18] ne moraju provjeravati rješenje i preuređivati ga (prema potrebi) te se tamo ne nalazi problem

u kojem je potrebno pretraživanje. Nastavni sadržaj ne potiče učenike da samostalno pronalaze problem u kojem mogu koristiti algoritam pretraživanja te ne moraju kritički provjeravati ispravnost rješenja i preuređivati ga. Kao i kod *Pythona*, tako i kod programskog jezika *Logo*, javljaju se isti neispunjeni odgojno-obrazovni ishodi. U skupini odgojno-obrazovnih ishoda koja se odnosi na problem, nismo naišli na primjere/zadatke u kojima učenici analiziraju ponašanje modela, kritički ga vrednuju i predviđaju rješenje problema.

Skupina odgojno-obrazovnih ishoda za osmi razred osnovne škole, koja se odnosi na programske jezike *Logo* i *Python*, **nije** u potpunosti ispunjena u promatranim nastavnim sadržajima. Učenici ne istražuju i ne predlažu primjere problema pri čijem se rješavanju mogu primijeniti rekurzivni postupci. Nastavni sadržaj udžbenika *Informatika 8* [24] najbolje ispunjava skupinu odgojno-obrazovnih ishoda koja se donosi na probleme, dok nastavni sadržaj udžbenika *Informatika<sup>+</sup> 8* [22] najmanje ispunjava zadane ishode. Tamo se ne nalaze primjeri koji potiču učenike na pisanje potrebne dokumentacije za rješenje svojeg problema te na argumentirano predstavljanje i obrazlaganje svojeg programskog rješenja problema.

Nastavnici mogu povezati ishode učenja s metodama poučavanja i vrednovanja. Moraju motivirati učenike na razmišljanje o problemu te korištenju papira i olovke kao pomagala. Moguće je ostvariti sve odgojno-obrazovne ishode uz nastavničku pomoć. Kroz primjere, učenici mogu uočiti važnost valjanih ulaznih primjera. Nastavnik mora poticati učenike na provjeravanje rješenja, preuređivanje rješenja i kritičko vrednovanje.

# Bibliografija

- [1] Ministarstvo znanosti i obrazovanja, *Kurikulum nastavnog predmeta informatika za osnovne i srednje škole*, 2018. [https://mzo.hr/sites/default/files/dokumenti/2018/OBRAZOVANJE/Nacionalni-kurikulumi/informatika/kurikulum\\_nastavnoga\\_predmeta\\_informatika.pdf](https://mzo.hr/sites/default/files/dokumenti/2018/OBRAZOVANJE/Nacionalni-kurikulumi/informatika/kurikulum_nastavnoga_predmeta_informatika.pdf), posjećeno 25. 6. 2019.

## 5. razred

- [2] M. Babić, N. Bubica, S. Leko, Z. Dimovski, M. Stančić, N. Mihočka, I. Ružić, B. Vejnović, *#mojportal5, Udžbenik informatike u petom razredu osnovne škole*, Školska knjiga, 2018.
- [3] M. Babić, N. Bubica, S. Leko, Z. Dimovski, M. Stančić, N. Mihočka, I. Ružić, B. Vejnović, *#mojportal5, Radna bilježnica za informatiku u petom razredu osnovne škole*, Školska knjiga, 2018.
- [4] I. Kniewald, V. Galešev, G. Sokol, V. Vlahović, H. Kovač, *Informatika<sup>+</sup> 5, Udžbenik iz informatike za 5. razred osnovne škole*, SysPrint, 2018.
- [5] I. Kniewald, V. Galešev, G. Sokol, V. Vlahović, H. Kovač, *Informatika<sup>+</sup> 5, Radna bilježnica za 5. razred osnovne škole*, SysPrint, 2018.
- [6] V. Gregurić, N. Hajdinjak, M. Jakšić, B. Počuča, D. Rakić, S. Svetličić, D. Šokac, D. Vlačinić, *Informatika 5, Udžbenik informatike za 5. razred osnovne škole*, Profil Klett, 2018.
- [7] V. Gregurić, N. Hajdinjak, M. Jakšić, B. Počuča, D. Rakić, S. Svetličić, D. Šokac, D. Vlačinić, *Informatika 5, Radna bilježnica iz informatike za 5. razred osnovne škole*, Profil Klett, 2018.

## 6. razred

- [8] M. Babić, N. Bubica, S. Leko, Z. Dimovski, M. Stančić, N. Mihočka, I. Ružić, B. Vejnović, *#mojportal6, Udžbenik informatike u šestom razredu osnovne škole*, Školska knjiga, 2018.

- [9] M. Babić, N. Bubica, S. Leko, Z. Dimovski, M. Stančić, N. Mihočka, I. Ružić, B. Vejnović, *#mojportal6, Radna bilježnica za informatiku u šestom razredu osnovne škole*, Školska knjiga, 2018.
- [10] I. Kniewald, V. Galešev, G. Sokol, V. Vlahović, H. Kovač, *Informatika<sup>+</sup> 6, Udžbenik iz informatike za 6. razred osnovne škole*, SysPrint, 2018.
- [11] I. Kniewald, V. Galešev, G. Sokol, V. Vlahović, H. Kovač, *Informatika<sup>+</sup> 6, Radna bilježnica za 6. razred osnovne škole*, SysPrint, 2018.
- [12] S. Deljac, V. Gregurić, N. Hajdinjak, B. Počuča, D. Rakić, S. Svetličić, *Informatika 6, Udžbenik informatike za 6. razred osnovne škole*, Profil Klett, 2018.
- [13] S. Deljac, V. Gregurić, N. Hajdinjak, B. Počuča, D. Rakić, S. Svetličić, *Informatika 6, Radna bilježnica iz informatike za 6. razred osnovne škole*, Profil Klett, 2018.

## 7. razred

- [14] M. Babić, N. Bubica, S. Leko, Z. Dimovski, M. Stančić, N. Mihočka, I. Ružić, B. Vejnović, *#mojportal7, Udžbenik informatike u sedmom razredu osnovne škole*, Školska knjiga, 2018.
- [15] M. Babić, N. Bubica, S. Leko, Z. Dimovski, M. Stančić, N. Mihočka, I. Ružić, B. Vejnović, *#mojportal7, Radna bilježnica za informatiku u sedmom razredu osnovne škole*, Školska knjiga, 2018.
- [16] I. Kniewald, V. Galešev, G. Sokol, V. Vlahović, H. Kovač, D. Kager, N. Kunštek, *Informatika<sup>+</sup> 7, Udžbenik iz informatike za 7. razred osnovne škole*, SysPrint, 2018.
- [17] I. Kniewald, V. Galešev, G. Sokol, V. Vlahović, H. Kovač, D. Kager, N. Kunštek, *Informatika<sup>+</sup> 7, Radna bilježnica za 7. razred osnovne škole*, SysPrint, 2018.
- [18] S. Deljac, V. Gregurić, N. Hajdinjak, B. Počuča, D. Rakić, S. Svetličić, *Informatika 7, Udžbenik informatike za 7. razred osnovne škole*, Profil Klett, 2018.
- [19] S. Deljac, V. Gregurić, N. Hajdinjak, B. Počuča, D. Rakić, S. Svetličić, *Informatika 7, Radna bilježnica iz informatike za 7. razred osnovne škole*, Profil Klett, 2018.

**8. razred**

- [20] M. Babić, N. Bubica, S. Leko, Z. Dimovski, M. Stančić, N. Mihočka, I. Ružić, B. Vejnović, *#mojportal8, Udžbenik informatike u osmom razredu osnovne škole*, Školska knjiga, 2018.
- [21] M. Babić, N. Bubica, S. Leko, Z. Dimovski, M. Stančić, N. Mihočka, I. Ružić, B. Vejnović, *#mojportal8, Radna bilježnica za informatiku u osmom razredu osnovne škole*, Školska knjiga, 2018.
- [22] I. Kniewald, V. Galešev, G. Sokol, D. Kager, H. Kovač, J. Purgar, N. Kunštek, *Informatika<sup>+</sup> 8, Udžbenik iz informatike za 8. razred osnovne škole*, SysPrint, 2018.
- [23] I. Kniewald, V. Galešev, G. Sokol, D. Kager, H. Kovač, J. Purgar, N. Kunštek, *Informatika<sup>+</sup> 8, Radna bilježnica za 8. razred osnovne škole*, SysPrint, 2018.
- [24] S. Deljac, V. Gregurić, N. Hajdinjak, B. Počuča, D. Rakić, S. Svetličić, *Informatika 8, Udžbenik informatike za 8. razred osnovne škole*, Profil Klett, 2018.
- [25] S. Deljac, V. Gregurić, N. Hajdinjak, B. Počuča, D. Rakić, S. Svetličić, *Informatika 8, Radna bilježnica iz informatike za 8. razred osnovne škole*, Profil Klett, 2018.



# Sažetak

U važećem kurikulumu nastavnog predmeta Informatika postoje četiri domene kojima se realiziraju ciljevi predmeta: E-Društvo, Digitalna pismenost i komunikacija, Računalno razmišljanje i programiranje te Informacije i digitalna tehnologija. Proučavali smo odgojno-obrazovne ishode navedene u domeni Računalno razmišljanje i programiranje. Analizirali smo programske jezike u obaveznim udžbenicima i pripadajućim dopunskim nastavnim sredstvima za osnovnu školu te kritički vrednovali doprinos sadržaja odgojno-obrazovnim ciljevima predmeta Informatika.

# Summary

There are four domains in the current curriculum of the Informatics that fulfil the objectives of the course: E-Society, Digital Literacy and Communication, Computer Thinking and Programming, Information and Digital Technology. We studied the educational outcomes outlined in the Computer Thinking and Programming domain. We analyzed the programming languages in the compulsory textbooks and associated supplementary teaching aids for elementary school and critically evaluated the contribution of the content to the educational goals of the Informatics course.

# Životopis

Rođena sam 1. 11. 1993. u Zaboku. Pohađala sam OŠ Oroslavje, a zatim i Opću gimnaziju Oroslavje. U srpnju 2012. godine, nakon završetka gimnazije, upisala sam Prirodoslovno-matematički fakultet u Zagrebu. Godine 2017. završila sam preddiplomski studij matematike, smjer nastavnički, i iste godine upisala diplomski studij matematike i informatike, smjer nastavnički, na istom fakultetu.