

CRNA GORA

ZAVOD ZA ŠKOLSTVO

Predmetni program

**UVOD U PROGRAMIRANJE**

Izborni predmet za IX razred osnovne škole

Podgorica

2017.

SADRŽAJ

[**A.** **NAZIV PREDMETA** 3](#_Toc493592631)

[**B.** **ODREĐENJE PREDMETA** 3](#_Toc493592632)

[**C.** **CILJEVI PREDMETA** 3](#_Toc493592633)

[**D.** **POVEZANOST SA DRUGIM PREDMETIMA I MEĐUPREDMETNIM TEMAMA** 4](#_Toc493592634)

[**E.** **OBRAZOVNO-VASPITNI ISHODI PREDMETA** 5](#_Toc493592635)

[**F.** **DIDAKTIČKE PREPORUKE ZA REALIZACIJU PREDMETA** 9](#_Toc493592636)

[**G.** **PRILAGOĐAVANJE PROGRAMA DJECI SA POSEBNIM OBRAZOVNIM POTREBAMA I NADARENIM UČENICIMA** 10](#_Toc493592637)

[**H.** **VREDNOVANJE OBRAZOVNO-VASPITNIH ISHODA** 10](#_Toc493592638)

[**I.** **USLOVI ZA REALIZACIJU PREDMETA** 11](#_Toc493592639)

# **NAZIV PREDMETA**

**UVOD U PROGRAMIRANJE**

# **ODREĐENJE PREDMETA**

Kroz izborni predmet Uvod u programiranje učenici[[1]](#footnote-1) treba da steknu osnovnu algoritamsku pismenost i afinitet za upotrebu računara kao alata za razvoj softvera.

Nastavni predmet Uvod u programiranje izučava se u devetom razredu osnovne škole s fondom od jednog časa sedmično, odnosno 34 časa na nivou nastavne godine. U sljedećoj tabeli dat je broj časova obaveznog (80‒85%) i otvorenog dijela (15‒20%) predmetnog programa, kao i broj časova za određene oblike nastave (teorijska nastava, vježbe i ostali vidovi). Otvoreni dio programa (15‒20%) je sadržaj koji planira škola/nastavnik. Može se realizovati kao cjelina ili periodično, između pojedinih tema obaveznog programa, u saglasnosti sa sadržajima tih tema.

NAPOMENA: Uzimajući u obzir niz specifičnosti ovog u odnosu na ostale nastavne predmete u osnovnoj školi, neophodno je da se poštuje princip da na jednom računaru radi jedan učenik.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Razred** | **Sedmični broj časova** | **Broj časova – obavezni dio**  (80‒85%) | **Broj časova – otvoreni dio**  (15‒20%) | **Ukupno časova** | **Teorijska**  **nastava** | **Vježbe i ostali**  **vidovi nastave** |
| **IX** | 1 | 26 | 5 | 31 | 7 | 19 |

# **CILJEVI PREDMETA**

* 1. **Saznajni ciljevi predmeta**

Učenjem predmeta Uvod u programiranje učenici treba da:

* + - razumiju koncepte i elemente savremenih aplikacija;
    - razumiju koncept grafičkog interfejsa i upotrebljavaju instalirane aplikacije;
    - razumiju načine upotrebe računara kao uređaja za kreiranje softvera;
    - primjenjuju softverske alate za implementiranje programa u izabranom programskom jeziku;
    - shvate pravna i etička načela upotrebe ICT i raspravljaju o posljedicama njihovih narušavanja.
  1. **Procesni ciljevi predmeta**

Učenjem predmeta Uvod u programiranje učenici treba da:

* + - razvijaju informatičku pismenost i proširuju svoje mogućnosti za upotrebu računara u svakodnevnom životu, učenju i radu;
    - razvijaju interesovanje za moderne tehnologije, kreativnost i inovativnost kroz razvoj algoritama i implementaciju programa;
    - razvijaju sposobnost algoritamskog načina razmišljanja u različitim sferama života;
    - stiču sposobnost da kritički odaberu izvore informacija i prikažu ih na razumljiv i uvjerljivo oblikovan način;
    - razvijaju tačnost, pravovremenost i preciznost u radu, upornost i istrajnost u rješavanju zadataka;
    - razvijaju interesovanje za različita zanimanja iz oblasti računarskih nauka i pozitivan stav prema preduzetništvu i samozapošljavanju u ovoj oblasti;
    - razvijaju stvaralačke sposobnosti (kreativnost, originalnost i individualnost) i praktične vještine za svakodnevni život;
    - razvijaju sposobnost za samostalno rješavanje problema, pronalaženje racionalnih rješenja i funkcionalno i estetsko oblikovanje;
    - stiču navike i osposobe se za međusobnu saradnju u radu.

# **POVEZANOST SA DRUGIM PREDMETIMA I MEĐUPREDMETNIM TEMAMA**

Povezanost sa predmetima:

* Matematika;
* Fizika;
* Informatika sa tehnikom;
* Crnogorski-srpski, bosanski, hrvatski jezik i književnost;
* Strani jezik (engleski jezik).

Povezanost sa međupredmetnim temama:

* Učenje učenja;
* Preduzetništvo;
* Zdravi stilovi života;
* Likovna kultura;
* Matematičke kompetencije;
* Kompetencija iz oblasti tehničkih nauka;
* Upotreba informacijskih i komunikacijskih tehnologija.

1. **OBRAZOVNO-VASPITNI ISHODI PREDMETA**

|  |
| --- |
| **Obrazovno-vaspitni ishod 1**  ***Na kraju učenja učenik će moći da opiše šta je algoritam i razlikuje načine zapisivanja algoritama.*** |
| **Ishodi učenja**  *Tokom učenja učenik će moći da:*   * objasni intuitivni pojam algoritma i prepozna ga u svakodnevnom životu; * ispiše rezultate izvršavanja algoritma korak po korak; * kreira jednostavan algoritam sa zadatim skupom instrukcija; * implementira jednostavan program koji izvršava zadate instrukcije i objasni razliku između algoritma i programa. |
| **Didaktičke preporuke za realizaciju obrazovno-vaspitnog ishoda**:   1. **Sadržaji/pojmovi**   **Sadržaji**:  Radi se:   * prikazivanje algoritma kao niza instrukcija zadatih na prirodnom jeziku i kao algoritamske sheme (npr. naći put u lavirintu ako su instrukcije: GORE, DOLJE, LIJEVO, DESNO); * izvršavanje instrukcija zadatog algoritma korak po korak i zapisivanje dobijenih vrijednosti u svakom koraku; * zadavanje jednostavnog problemskog zadatka za koji treba napraviti algoritam (na prirodnom jeziku i/ili algoritamskom shemom); * pokazivanje radnog okruženja za izradu programa (Python/Logo/Scratch…); * zadavanje jednostavnog problemskog zadatka za koji treba napraviti program (npr. naći površinu pravougaonika, obim trougla i sl.) i implementiranje algoritma u nekom programskom jeziku.   **Pojmovi:**   * algoritam, algoritamska shema (flowchart), pseudo-kod, instrukcija.  1. **Aktivnosti učenja**  * Podrazumijeva se da učenici rade samostalno, po jedan učenik za računarom. * Učenici se dijele u parove. Svaki učenik pojedinačno ispisuje rezultate izvršavanja algoritma zadatog algoritamskom shemom, pseudo-kodom ili prirodnim jezikom, koje zatim kontroliše drugi učenik. * Učenici rješavaju npr. lavirinit u obliku pravougaonika ako su dozvoljene instrukcije GORE, DOLJE, LIJEVO, DESNO. * Učenici pišu algoritam/crtaju shemu za zadatak sa instrukcijama koje uključuju samo aritmetičke operacije i ulazno/izlazne operacije.  1. **Broj časova realizacije (okvirno): 4 časa** |
| **Obrazovno-vaspitni ishod 2**  ***Na kraju učenja učenik će biti sposoban da implementira jednostavan program****.* |
| **Ishodi učenja**  *Tokom učenja učenik će moći da:*   * koristi razvojno okruženje za kreiranje programa; * upotrebljava promjenljive u programu; * snimi program na zadatu lokaciju; * izračuna vrijednosti jednostavnih aritmetičkih izraza; * primjenjuje jednostavne komande za ulaz i izlaz; * dodaje komentare u program. |
| **Didaktičke preporuke za realizaciju obrazovno-vaspitnog ishoda**:   1. **Sadržaji/pojmovi**   **Sadržaji:**  Radi se:   * opisivanje nekog radnog okruženja za razvoj programa (npr. Python IDLE, FMSLogo, Scratch…); * opisivanje promjenljive kao mehanizma čuvanja vrijednosti u programu i opisivanje mogućih vrijednosti promjenljivih (npr. u jeziku Python: broj, string, lista/niz); * objašnjavanje aritmetičkih operacija i naredbe dodjeljivanja; * crtanje algoritamske sheme i pisanje koda programa (implementiranje) na računaru, gdje se algoritam svodi na primjenu aritmetičkih operacija (npr. izračunati površinu poda sobe, koliko je kanti boje potrebno za farbanje učionice ako se jednom kantom može ofarbati 8m2 …); * opisivanje komandi programskog jezika za ulaz i izlaz; * objašnjavanje potrebe za komentarima u programu.   Ne rade se:   * složene aritmetičke operacije koje uključuju matematičke funkcije kao što su stepenovanje i korjenovanje.   **Pojmovi:**   * integrisano razvojno okruženje (IDE), instrukcija, promjenljiva, vrijednost promjenljive, program, lista/niz (ako jezik podržava taj tip), naredba dodjeljivanja, aritmetički operatori, ulaz/izlaz, komentar.  1. **Aktivnosti učenja:**  * upoznavanje sa osnovnim elementima radnog okruženja (meni, toolbar, način pokretanja i snimanja...); * definisanje promjenljivih i zadavanje vrijednosti; * pisanje komenatara u programu: unijeti ime učenika, datum i dr.; * pisanje programa sa instrukcijama koje sadrže aritmetičke operacije ulazno-izlazne operacije.  1. **Broj časova realizacije (okvirno): 5 časova** |
| **Obrazovno-vaspitni ishod 3**  ***Na kraju učenja učenik će biti osposobljen da implementira uslovne naredbe i cikluse u programu.*** |
| **Ishodi učenja**  *Tokom učenja učenik će moći da:*   * upotrebljava uslovne i logičke operatore programskog jezika; * kreira uslovne naredbe sa jednim kriterijumom odlučivanja; * implementira uslovne naredbe sa više kriterijuma odlučivanja; * razumije potrebu za ciklusima u programu i kreira programe sa ciklusima. |
| **Didaktičke preporuke za realizaciju obrazovno-vaspitnog ishoda**:   * + - 1. **Sadržaji/pojmovi**   **Sadržaji:**  Radi se:   * primjenjivanje uslovnih operatora (<, >, <=, >=,=, !=) i opisivanje razlike između matematičkih operatora i operatora programskog jezika; * objašnjavanje i primjenjivanje logičkih operatora (AND, OR, NOT, XOR); * opisivanje strukture naredba uslovnog skoka (if) i navođenje primjera algoritama i programa sa naredbom if; * opisivanje strukture ciklusa u programskom jeziku (for, while…) i implementacija programa sa ciklusima (npr. štampati ime škole 100 puta, štampati sve brojeve od a do b…).   Ne radi se:   * naredba switch...case.   **Pojmovi:**   * Uslovni operatori (<, >, <=, >=,=, !=), logički operatori (AND, OR, NOT, XOR), if, while, for.   + - 1. **Aktivnosti učenja:** * pisanje algoritama i programa u kojima se primjenjuju uslovni operatori (<, >, <=, >=,=, !=); * pisanje algoritama i programa u kojima se primjenjuju logički operatori (AND, OR, NOT, XOR); * rješavanje zadataka u kojima se koristi naredba uslovnog skoka; * rješavanje zadataka u kojima se koriste ciklusi.   + - 1. **Broj časova realizacije (okvirno): 6 časova** |
| **Obrazovno-vaspitni ishod 4**  ***Na kraju učenja učenik će biti osposobljen da upotrijebi postojeći kod i organizuje svoj kod za buduću upotrebu.*** |
| **Ishodi učenja**  *Tokom učenja učenik će moći da:*   * objasni potrebu za grupisanjem koda u funkcije; * koristi ugrađene funkcije jezika; * istražuje sistem za pomoć u razvojnom okruženju; * kreira jednostavnije sopstvene funkcije. |
| **Didaktičke preporuke za realizaciju obrazovno-vaspitnog ishoda**:   1. **Sadržaji/pojmovi**   **Sadržaji:**  Radi se:   * objašnjavanje potrebe za grupisanjem koda u funkcionalne cjeline; * objašnjavanje ugrađenih (tzv. built-in) funkcija programskog jezika (npr. max, kvadratni korijen, stepen...) i njihova primjena u zadacima (npr. izračunati dužinu dijagonale pravougaonika ako su date dužine stranica...); * prikazivanje sistema za pomoć (help) u radnom okruženju i pronalaženje ugrađenih funkcija; * pisanje sopstvenih funkcija, sa navođenjem primjera: funkcija bez parametara/argumenata (npr. funkcija koja štampa ime škole), funkcija sa više parametara/argumenata (npr. funkcija sa dva argumenta koja vraća zbir kvadrata argumenata), funkcija koja vraća vrijednost, funkcija koja ne vraća vrijednost (npr. funkcija sa jednim argumentom koja štampa argument 100 puta).   Ne radi se:   * primjena rekurzije; primjena ugrađenih funkcija koje nijesu odgovarajuće uzrastu učenika (npr.trigonometrijske funkcije, logaritamska funkcija...).   **Pojmovi:**   * funkcija (procedura, metod, modul…), ugrađene (built-in) funkcije, korisničke (user defined) funkcije, sistem za pomoć u radnom okruženju (Help).  1. **Aktivnosti učenja:**  * diskutovanje o primjerima u kojima se isti programski kod ponavlja i uočavanje potrebe da se takav kod grupiše; * pisanje programa u kojima se koriste ugrađene funkcije programskog jezika; * pronalaženje podataka o ugrađenoj funkciji primjenom sistema za pomoć; * implementiranje korisničkih funkcija sa zadatim imenom i argumentima.  1. **Broj časova realizacije (okvirno): 6 časova** |
| **Obrazovno-vaspitni ishod 5**  ***Na kraju učenja učenik će biti osposobljen da implementira programe koji crtaju grafičke objekte i reaguju na korisničke događaje***. |
| **Ishodi učenja**  *Tokom učenja učenik će moći da:*   * koristi funkcije za crtanje elementarnih geometrijskih figura; * primjenjuje metode promjene boje i promjene debljine linije; * implementira metode popunjavanja oblasti datom bojom; * razumije koncept korisničkog događaja; * kreira kod koji reaguje na korisničke događaje; * otklanja jednostavnije greške u programu. |
| **Didaktičke preporuke za realizaciju obrazovno-vaspitnog ishoda**:   1. **Sadržaji/pojmovi**   **Sadržaji:**  Radi se:   * objašnavanje koncepta korisničkog prozora i radne površine prozora (npr. u jeziku Python se naziva Canavas); * podešavanje boja i debljine linije; * pozivanje ugrađenih funkcija za crtanje linija, krugova i mnogouglova; * popunavanje figura zadatom bojom; * objašnjavanje koncepta reakcije na događaj (event) u programskom jeziku; * objašnjavanje i implementacija rekacije na događaje mouse-click i key-press (npr. na klik miša nacrtati krug, ili klikom na slovo a sa tastature) ‒ na primjer, nacrtati krug u prozoru na poziciji gdje je kliknuto mišem; * pokazivanje načina uklanjanja grešaka iz programa; * predstavljanje projektnog zadatka: formiranje grupa (npr. parovi), upućivanje na izvore informacija (npr. literatura, internet...), zadavanje pravila za pisanje algoritma i implementiranje programa, definisanje pravila za prezentaciju projekta.   Ne radi se:   * primjenjivanje naprednih funkcija; implementiranje reakcije na druge tipove događaja (MouseMove, timer...).   **Pojmovi:**   * korisnički prozor, radna površina prozora (kanvas), događaj (event), debug.  1. **Aktivnosti učenja:**  * pisanje programa koji kreiraju korisnički prozor; * pisanje programa u kojima se koriste linije različitih boja i debljina; * implementiranje programa u kojima se na radnoj površini prozora crtaju geometrijski oblici (linije, elipse/krugovi, mnogouglovi); * popunavanje figura zadatom bojom; * pisanje programskog koda za reakcije na događaje mouse-click i key-press; * pronalaženje grešaka u datom programskom kodu; * izrada projektnog zadatka.  1. **Broj časova realizacije (okvirno): 5 časova** |

# **DIDAKTIČKE PREPORUKE ZA REALIZACIJU PREDMETA**

Akcenat ovog predmeta stavlja se na praktično znanje pa je suština aktivnosti u učenju bazirana na praktičnim vježbama uz asistenciju nastavnika.

Većina softvera koji će biti korišćeni u okviru nastave uglavnom imaju podršku za jezike s našeg govornog područja, ali ipak treba insistirati na engleskom jeziku, a pojedine termine treba po potrebi ispisati i prevesti (npr. save – sačuvati, help – pomoć...). Smatramo da će na ovaj način u početku efekti učenja biti malo slabiji, ali globalno u smislu sticanja informatičkih znanja prednost je nesumnjivo na strani interfejsa na engleskome jeziku.

Pripremanje za realizaciju nastavnih sadržaja predmeta podrazumijeva precizno definisanje učenikovih aktivnosti, uz navođenje nastavnikovih i prema realnim resursima za izvođenje nastave.

U skladu s dinamikom razvoja tehnoloških procesa i izvjesnih tehničkih inovacija, planirani nastavni sadržaji predmeta Uvod u programiranje biće podložni izmjenama i dopunama, tako da ovaj program ostavlja slobodu u njegovoj realizaciji svim učesnicima procesa.

Predviđeni sadržaji zahtijevaju raznovrstan metodički pristup, odnosno primjenu postupaka:

* + - modularne nastave (učenici prolaze kroz nastavnu oblast realizacijom projekta po svome opredjeljenju; nastavnik organizuje njihov rad, dajući im potpunu slobodu iskazivanja kreativnosti, znanja i vještina);
    - individualizacije (učenik bira aktivnosti po sopstvenome planu);
    - diferencijacije (na osnovu ličnih sposobnosti i interesa učenika za određene oblasti);
    - aktivne nastave (učenik je subjekat u svim fazama realizacije nastavnoga procesa);
    - algoritamske nastave (podrazumijeva sistem pravila i uputstava za obavljanje aktivnosti učenika i nastavnika pri realizaciji praktičnih vježbi).

Neprekidno treba naglašavati vezu između algoritma i programa.

Za programiranje treba da se koristi programski jezik koji nastavnik poznaje. Preporučuje se da se koriste neki od sljedećih programskih jezika: Python, Logo, Scratch, Basic. Mogu se koristiti alati kao što su Python IDLE, FMSLogo ili Scratch ili neki od online alata (npr. <https://turtleacademy.com/>, <http://www.calormen.com/jslogo/>).

# **PRILAGOĐAVANJE PROGRAMA DJECI SA POSEBNIM OBRAZOVNIM POTREBAMA I NADARENIM UČENICIMA**

1. **Prilagođavanje programa djeci sa posebnim obrazovnim potrebama**

Treba koristiti asistivne i informaciono/komunikacione tehnologije u nastavi, kao i opremu i pomagala koja omogućavaju realizaciju nastave u skladu sa obrazovnim i razvojnim ciljevima i karakteristikama učenikove smetnje u razvoju.

Računar kao sredstvo za prilagođavanje programa djeci sa posebnim obrazovnim potrebama može poslužiti ne samo za ovaj predmet nego za školovanje uopšte. Treba koristiti specijalne ulazno-izlazne uređaje i programe za osobe sa posebnim obrazovnim potrebama.

1. **Prilagođavanje programa nadarenim učenicima**

Sigurno će se pojaviti učenici koji posjeduju znanje iz dijela programa koji se obrađuje, čak i učenici koja znaju i posjeduju vještine koje prevazilaze ovaj program, pa ih treba usmjeriti na pomaganje drugim učenicima u savladavanju programa i sticanju vještina i demonstraciji određenih aktivnosti u toku časa. Na taj način će se učenici stimulisati da rade na času, pomažu ostalim učenicima i usmjere svoju energiju na kreativni dio koji prevazilazi učenje osnovnih vještina i pomaže ostalim učenicima u shvatanju svrhe znanja i vještina koje usvajaju.

# **VREDNOVANJE OBRAZOVNO-VASPITNIH ISHODA:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ishod** | **Ocjena 2** | **Ocjena 3** | **Ocjena 4** | **Ocjena 5** |
| Ishod 1 | Učenik prepoznaje različite načine predstavljanja algoritma i izvršava korake algoritma zadatog prirodnim jezikom. | Učenik prepoznaje i izvršava korake algoritma zadatog pseudo-kodom ili shemom i kreira jednostavniji algoritam uz pomoć nastavnika. | Učenik kreira jednostavniji algoritam i uspijeva da koriguje greške uz pomoć nastavnika. | Učenik kreira jednostavniji algoritam i samostalno uočava i otklanja greške. |
| Ishod 2 | Učenik prepoznaje elemente razvojnog okruženja, zna da snimi program i da pravilno kreira promjenljive i piše komentare. | Učenik primjenjuje naredbu dodjeljivanja, koristi promjenljive različitih tipova (string, broj, lista) i koristi naredbe ulaza i izlaza. | Učenik umije da na osnovu datog algoritma napiše program uz pomoć nastavnika. | Učenik umije da na osnovu datog algoritma napiše program. |
| Ishod 3 | Učenik prepoznaje i razlikuje logičke i uslovne operatore programskog jezika. | Učenik prepoznaje i zna da koristi logičke operatore, prepoznaje situacije u kojima se određene naredbe ponavljaju i kreira uslovne operatore sa jednim kriterijumom. odlučivanja | Učenik zna da kreira uslovne operatore sa više kriterijuma odlučivanja i kreira programe sa jednim ciklusom. | Učenik samostalno kreira uslovne operatore sa više kriterijuma odlučivanja i kreira programe sa ugniježdenim ciklusima. |
| Ishod 4 | Učenik prepoznaje u postojećem programu funkciju i zna da objasni potrebu za istom. | Učenik prepoznaje i uz pomoć nastavnika koristi built-in funkcije i poziva ih u glavni program. | Učenik samostalno koristi built-in funkcije i sistem za pomoć i poziva ih u glavni program. | Učenik samostalno kreira sopstvene funkcije i poziva ih u glavni program. |
| Ishod 5 | Učenik razumije pojam korisničkog događaja i prepoznaje funkcije za crtanje elementarnih geometrijskih figura. | Učenik prepoznaje šta radi zadati grafički program i uz pomoć nastavnika koristi funkcije za crtanje elementarnih geometrijskih figura. | Učenik samostalno koristi funkcije za crtanje elementarnih geometrijskih figura i primjenjuje metode promjene boje i debljine linija uz pomoć nastavnika. | Učenik implementira metode popunjavanja oblasti datom bojom i kreira programe koji reaguju na korisničke događaje. |

# **USLOVI ZA REALIZACIJU PREDMETA**

Nastavu mogu da izvode nastavnici koji su završili PMF, ETF, FON, Nastavnički fakultet a imali su predmet Programiranje u obimu od najmanje 20 kredita (4 jednosemestralna predmeta ili dva dvosemestralna) i koji ispunjavaju uslove propisane Zakonom o osnovnom vaspitanju i obrazovanju i Pravilnikom o profilu obrazovanja nastavnika predmetne nastave.

Predmetni program **UVOD U PROGRAMIRANJE**,izborni predmet za IX razred osnovne škole uradila je Komisija u sastavu:

prof. dr Goran Šuković, predsjednik

Aleksandra Račić, član

U izradi predmetnih programa korišćeni su *predmetni programi  (2004, 2009,* *2011 i 2013) i Metodološko uputstvo za pisanje predmetnih programa zasnovanih na ishodima učenja, 2017.*

Pored naših u izradi predmetnih programa konsultovani su i predmetni programi iz više evropskih država (Malta, Velika Britanija, Hrvatska), kao i sljedeći obrazovni dokumenti, kursevi i udžbenici:

1. Computing Our Future – Computer programming and coding: Priorities, school curricula and initiatives across Europe (October 2015, European Schoolnet)
2. Computing in the national curriculum – A guide for primary teachers ([www.computingatschool.org.uk](http://www.computingatschool.org.uk)) ICT Syllabus for Primary Schools Rationale (Malta) <https://curriculum.gov.mt/en/Curriculum/Year-1-to-6/Pages/default.aspx>
3. Udžbenici Digital Teens (<http://www.binarylogic.net/first/computing-ict/secondary-courses/digital-teens-intl-english/> )
4. Jason R. Briggs – Python for Kids: Playful Introduction to Programming (Dec. 2012, ISBN-13: 978-1-59327-407-8)
5. Majed Marji – Learn to Program with Scratch : A Visual Introduction to Programming with Games, Art, Science, and Math (Feb. 2014, ISBN-13:978-1-59327-543-3)
6. Logo – online učenje programiranja (<https://tesla.carnet.hr/course/view.php?id=81>)

Nacionalni savjet za obrazovanje (II saziv) na 15. sjednici, održanoj 03. jula 2017. godine, utvrdio je predmetni program **UVOD U PROGRAMIRANJE,** izborni predmet za IX razred osnovne škole.

1. Svi izrazi koji se u ovom dokumentu koriste u muškom rodu obuhvataju iste izraze u ženskom rodu. [↑](#footnote-ref-1)