

Programiranje (2. razred Informatičke gimnazije)

Ishodi učenja: po odslušanom i položenom predmetu učenik će imati slijedeća znanja, kompetencije i vještine:

- razumije napredne koncepte imperativnog programiranja
- razumije rad sa složenijim kompozitnim tipovima: višedimenzionalna polja (nizovi, arrays), strukture, te poseban naglasak na rad sa tekstualnim tipovima podataka (stringovi)
- posjeduje detaljno i napredno poznavanje korištenja petlji (višestrukog-ugniježdene petlje)
- sposoban je prepoznati probleme i specifične korake u rješavanju problema i kreirati kod koji pretražuje prostor rješenja. (primjer problem Hanojskih tornjeva)
- poznaje sortiranje i algoritme za prolazak kroz podatkovne strukture i pretraživanje
- koristi bibliotečne funkcije i klase za rješavanje programskih problema uz razumijevanje kako iste rade, njihovih prednosti i nedostataka
- zna koristiti debugging alate u cilju otkrivanja grešaka u kodu
- sposobnost dekompozicije problema na manje složene probleme
- sposobnost pisanja dobro organizovanog i čitljivog programskog koda uz pravilno imenovanje konstanti, varijabli i funkcija, te organizaciju programskog koda u funkcije
- poznaje rad sa datotekama
- poznaje pojmove klasa i objekat

Pregled nastavnih cjelina koje će se obraditi u toku nastavne godine:

REALIZACIJA PROGRAMA	ČAS
Upoznavanje, uvod u predmet, literatura i pribor	1
Ponavljanje gradiva 1. razreda	6
Funkcije	5
Strategije rješavanja problema	2
Strukturalni tip podataka	7
Rad sa datotekama	8
Uvod u objektno-orijentisano programiranje	4
Tekstualni tip podataka	8
Algoritmi i strukture podataka	10
Razvoj složenijih programskih rješenja i projektni zadatak	19
UKUPNO:	70

NAPOMENA: Nastavni plan i program u drugom razredu zasniva se na izučavanju jednog od programskih jezika: C, C++, C#, Java – programiranje osnovni nivo (struktuirano programiranje). U ovom Nastavnom planu i programu dat je primjer za programski jezik C++, ali osnovna struktura predmeta se može slijediti i sa drugim programskim jezicima.

Prijedlog forme obrazaca za razvijanje nastavnog plana i programa

nastavnih predmeta modula IT za gimnaziju izbornog područja Informacionih tehnologija

Škola: GIMNAZIJA

Izborno područje/zanimanje: INFORMACIONE TEHNOLOGIJE

Nastavni predmet: PROGRAMIRANJE 2

Razred: II (drugi)

Broj sati sedmično: 2

Broj sati za školsku godinu: 70

CILJI I ZADACI:

Cilj: Usvojiti osnovna znanja i vještine o informatici i njenom razvoju radi stjecanja opće računarske pismenosti i kulture te razumijevanja civilizacijskog razvoja. Usvojiti i osposobiti učenike za samostalnu izradu jednostavnih programa u jednom od proceduralnih (objektno orijentisanih) programskih jezika.

Zadaci

Omogućiti učeniku da:

- bude osposobljen da razlikuje i zna implementirati različite tipove grananja, operatore poređenja, logičke operatore, višestruke naredbe grananja kao što su switch-case i goto naredba,
- bude osposobljen da razlikuje i zna implementirati različite strukture (naredbe break i continue) ponavljanja kao i vrste (for, while i do-while) ponavljanja,
- bude osposobljen da koristi različite tipovi nizova (nizovi cijelih brojeva, realnih brojeva i znakova (string)) i njihovu strukturu, jednodimenzionalne i višedimenzionalne (matrice) s algoritmima,
- nauči i razumije pojam funkcije i bibliotečke funkcije u programiranju, bude osposobljen za kreiranje vlastite funkcije,
- shvati i razumije razliku između funkcija i procedura,
- upozna funkcije koje nemaju parametre i funkcije koje ne vraćaju ništa,
- shvata i razumije pojam lokalne promjenjive, poziva po vrijednosti i kompoziciju funkcija,
- shvata i razumije primjenu strategije rješavanja problema kroz prikaz problema Hanojskog tornja,
- bude osposobljen za odabire odgovarajući algoritam za rješanje konkretnog problema,
- nauči i razumije definirati vlastite tipove podatak u programskom jeziku kao što su: typedef, pobrojani tip (enum) i strukturni tip,
- shvata, razumije i bude osposobljen za izradu strukturnih podataka: datuma i vremena, gometrijskih tijela i učenika,
- razumije i shvata polja i strukture kao članova struktura, polje strukturnih objekata,
- nauči i razumije princip rada datotečnog sistema, imenovanje, kreiranje i brisanje datoteka,
- spozna postojanje različitih tipova datoteka kao što su: tekstualne, binarne, strukturirane,
- shvata i razumije funkcije za rad sa datotekama,
- upozna funkcije za rad sa datotekama kroz primjere programskih zadatata sa datotekama (prebrojavanje znakova, pronalazak uzorka teksta...),
- bude osposobljen za strukturiranje datotke i povezivanje (implementiranje) jednostavne baze podataka kroz tekstualnu datoteku,
- nauči i razumije osnovne pojmove objektno-orientisanog programiranja: klase, objekat, definiciju objektno-orientisanog programiranja,
- bude osposobljen za primjene klase kao strukture sa funkcijama i implementirane enkapsulacije
- bude osposobljen za kodiranje podataka na računaru, ASCII kod, kodne stranice i predstavljanje naših slova,
- shvata, razumije i bde osposobljen za implementaciju različitih znakovnih tipova podataka kao što su: char, string, wchar, wstring, Qstring, bude osposobljen za konverzije između tipova podataka,
- shvata, razumije i bude osposobljen za implementiranje složenijih primjere programskih zadataka sa stringovima i tekstualnim datotekama,
- razumije pojam algoritam i nauči analizirati primere jednostavnih algoritama,
- bude osposobljen za implementaciju jednog (ili više) algoritama sortiranja i bibliotečne funkcije za sortiranje, u programskim zadacima u kojima je potrebno sortiranje,

- nauči i razumije binarnu pretragu,
- bude osposobljen da analizira (odabere) i kreiranje/primjene binarnu pretragu (ili neke drugu) u bazi podataka učenik, efikasnom pretragom (recimo po imenu i prezimenu),
- nauči i razumije bibliotečne strukture (niz, vektor, lista, riječni) princip rada, prednosti i nedostatke,
- bude osposobljen da analizira i primjeni zadatke sa odabirom optimalne strukture,
- bude osposobljen da analizira i primejmi algoritam za rad na datotekama, pretraživanju dataoteka koristeći indeks,
- spozna način odabira zadataka (projekta/programa) na kojem će učenici raditi,
- bude osposobljen da analizira i dizajnira/planira izradu pseudo kod za odabrani program,
- bude osposobljen da odabere i koristi algoritme i algoritamske strukture za rješavanje problema,
- bude osposobljen da procjenjuje prednosti i ograničenja algoritamskog pristupa u rješavanju problema,
- bude osposobljen da organizuje kod, dekompoziciju (datoteke zaglavlja /header/, razdvajanje deklaracije od dfinicije klase) u samoodabranog programu,
- bude osposobljen da koristi pravila pisanja čitljivog koda: imenovanje konstanti, promojenljivih i funkcija u samoodabranom programu.
- bude osposobljen da koristi osnovne pojmove refactoringa (code smellsa) u samoodabranog programu,
- bude osposobljen da povezuje i koristi dosadašnja znanja u svrhu razvoja jednostavne baze podata; povezivanja sa strukturnom datotekom u primjeni samoodabrnog programu,
- upozna primjenu binarne datoteke, pojma serijalizacije, XMLa,
- bude osposobljen da predstavi, diskutuje, upoređuje svoje algoritme i prateću dokumentaciju sa algoritmima drugih učenika iz razreda i da pomože drugim učenicima po potrebi,
- bude osposobljen da nastavniku/profesoru/mentoru predstavi, diskutuje sa njim i donosi važne odluke kako na najbolji način izradi samoodabranom programu,
- bude osposobljen da shvati, razumije i razvije strategiju testiranja i otkrivanja grešaka u kodu, kao i primjena jediničnih testova (unit testing), pokrivenosti koda testvoima (coverage), test driven development i korištenje debuggera na izradi samoodabranom programu.
- Bude osposobljen da obavlja mentora tokom izrade programa i da na kraju predstavi svoj završeni samoosabrani program pred razredom.

NASTAVNI PROGRAM

Nastavne oblasti / Nastavne cjeline	Sati	Nastavni sadržaj / Nastavne jedinice	Po završetku nastavne cjeline učenik će imati sljedeća znanja, vještine i kompetencije:
Ponavljanje gradiva 1. razreda	7	<ul style="list-style-type: none"> ✓ Upoznavanje, uvod u predmet, literature i pribor ✓ Grananje, tipovi grananja ✓ Petlje, tipovi petlji, višestruke (ugniježdene petlje) ✓ Nizovi ✓ Matrice (polja) 	<ul style="list-style-type: none"> ✓ Razumije i razlikuje različite tipove grananja, operatore poređenja, logičke operatore, višestruke naredbe grananja kao što su switch-case i goto naredba. ✓ Razumije i razlikuje strukture (naredbe break i continue) ponavljanja kao i vrste (for, while i do-while) ponavljanja. ✓ Razumije šta su to nizovi, različite tipovi nizova (nizovi cijelih brojeva, realnih brojeva i znakova (string)) i njihovu strukturu (jednodimenzionalni i višedimenzionalni). ✓ Razumije algoritama kao što su maksimum/minimum, provjera da li članovi niza zadovoljavaju uslove, izbacivanje člana iz niza, determinanta, glavna/sporedna dijagonala, razdvajanje rečenice na riječi, prebrojavanje slova, zamjena velikih i malih slova, itd. ✓ Razumije strukturu višedimenzionalnog nizova (matrica) s algoritmima.
Funkcije	5	<ul style="list-style-type: none"> ✓ Pojam funkcije u programiranju. Primjeri bibliotечnih funkcija. Sintaksa za pravljenje vlastite funkcije. Funkcije vs. procedure. ✓ Funkcija koja nema parametre. Funkcija koja ne vraća ništa. ✓ Pojam lokalne promjenljive i primjeri. Poziv po vrijednosti. Kompozicija funkcija. 	<ul style="list-style-type: none"> ✓ Zna šta je to funkcija i bibliotечka funkcija u programiranju. ✓ Sposobnost izrade (kreiranje) vlastite funkcije. ✓ Razumjevanje razlike između funkcije i procedure. ✓ Razumjevanje funkcija koje nemaju parametre i funkcija koje ne vraćaju ništa (vrijednost). ✓ Razumjevanje pojma lokalne promjenljive, poziva po vrijednosti i kompozicije funkcija.
Strategija rješavanja	2	<ul style="list-style-type: none"> ✓ Pretraživanje problemskog prostora (primjer problem Hanojskih tornjeva) 	<ul style="list-style-type: none"> ✓ Razumije i pridružuje se u raspravi primjene strategije rješavanja problema kroz prikaz problema

programskih problema		<ul style="list-style-type: none"> Definisanje vlastitih tipova podataka u programskom jeziku: typedef, pobrojani tip (enum), strukturalni tip Primjeri strukturiranih podataka (datum i vrijeme) Primjeri strukturiranih podataka (geometrijska tijela, učenik) Polja i strukture kao članovi struktura, polje strukturalnih objekata 	<p>Hanojskog tornja.</p> <ul style="list-style-type: none"> Odabire odgovarajući algoritam za rješenje konkretnog problema.
Strukturalni tip podataka	7	<ul style="list-style-type: none"> Definisanje vlastitih tipova podataka u programskom jeziku: typedef, pobrojani tip (enum), strukturalni tip Primjeri strukturiranih podataka (datum i vrijeme) Primjeri strukturiranih podataka (geometrijska tijela, učenik) Polja i strukture kao članovi struktura, polje strukturalnih objekata 	<ul style="list-style-type: none"> Zna definisati vlastite tipove podataka u programskom jeziku kao što su: typedef, pobrojani tip (enum) i strukturalni tip. Razumijeti primjere strukturiranih podataka datuma i vrijeme, geometrijskih tijela i učenika. Sposobnost izradi jednog od navedenih primjera strukturiranih podataka: datum i vrijeme, geometrijska tijela ili učenik. Razumijevanje polja i strukture kao članova struktura, polje strukturiranih objekata Sposobnost izrade drugog od navedenih primjera strukturiranih podataka (polja i struktura kao članova struktura): datum i vrijeme, geometrijska tijela ili učenik.
Rad sa datotekama	8	<ul style="list-style-type: none"> Princip rada datotečnog sistema, imenovanje, kreiranje i brisanje datoteka Tipovi datoteka: tekstualne, binarne, strukturirane. Funkcije za rad sa datotekama Primjeri programskih zadataka sa datotekama (prebrojavanje znakova, pronalazak uzorka teksta...) Strukturiranje datoteke, primjeri implementacije jednostavne baze podataka kroz tekstualnu datoteku 	<ul style="list-style-type: none"> Razumije princip rada datotečnog sistema, imenovanje, kreiranje i brisanje datoteka. Razumije postojanje različitih tipova datoteka kao što su: tekstualne, binarne, strukturirane. Zna funkcije za rad sa datotekama. Razlikuje funkcije za rad sa datotekama kroz primjere programskih zadataka sa datotekama (prebrojavanje znakova, pronalazak uzorka teksta...). Sposobnost strukturiranja datotke i povezivanje (implementiranje) jednostavne baze podataka kroz tekstualnu datoteku.
Uvod u objektno-orientisano programiranje	4	<ul style="list-style-type: none"> Osnovni pojmovi objektno-orientisanog programiranja: Klase, objekat, definicija OOP Klasa kao struktura sa funkcijama: primjeri (dodati funkcije u neke do sada obrađene 	<ul style="list-style-type: none"> Zna i razumije osnovne pojmove objektno-orientisanog programiranja: klase, objekat, definiciju objektno-orientisanog programiranja. Sposobnost primjene klase kao strukture sa funkcijama (dodati funkcije u neke do sada obrađene

		strukture), enkapsulacija	strukture) <ul style="list-style-type: none"> ✓ Razumjeti i implementirati enkapsulaciju ✓ Sposobnost kodiranja podataka na računaru, ASCII kod, kodne stranice i predstavljanje naših slova. ✓ Razumjeti, analizirati i implementirati različite znakovne tipove podataka kao što su: char, string, wchar, wstring, Qstring ✓ Sposobnost konverzije između tipova podataka. ✓ Razumjeti, analizirati i implementirati složenije primjere programskih zadataka sa stringovima i tekstualnim datotekama.
Tekstualni tip podatka	8	<ul style="list-style-type: none"> ✓ Kodiranje podataka na računaru, ASCII kod, kodne stranice i predstavljanje naših slova, pojam Unicode ✓ Različiti znakovni tipovi podataka, char, string, wchar, wstring, Qstring, konverzija između tipova podataka ✓ Složeniji primjeri programskih zadataka sa stringovima ✓ Primjeri programskih zadataka sa tekstualnim datotekama 	<ul style="list-style-type: none"> ✓ Razumjeti pojam algoritam i analizirati primere jednostavnih algoritama ✓ Sposobnost kreiranja/implementacije jednog (ili više) algoritama sortiranja i bibliotečne funkcije za sortiranje, u programskim zadacima u kojima je potrebno sortiranje. ✓ Razumjeti binarnu pretragu. ✓ Sposobnost analize (odabira) i kreiranje/primjene binarne pretrage (ili neke druge) u bazi podataka o učenika (sa sortiranjem) i efikasnom pretragom (recimo po imenu i prezimenu). ✓ Razumjeti bibliotečne strukture (niz, vektor, lista, riječni) princip rada, prednosti i nedostatke. ✓ Sposobnost analize i primjene zadatak sa odabirom optimalne strukture. ✓ Sposobnost analize i primjene algoritama za rad na datotekama, pretraživanju dataoteka koristeći indeks.
Algoritmi i strukture podataka	10	<ul style="list-style-type: none"> ✓ Pojam algoritma, primjeri jednostavnih algoritama ✓ Sortiranje: implementacija jednog jednostavnog algoritma sortiranja, bibliotečna funkcija za sortiranje, programski zadaci u kojima je potrebno koristiti sortiranje ✓ Binarna pretraga; primjer: baza podataka o učenicima sa sortiranjem i efikasnom pretragom po imenu i prezimenu ✓ Bibliotečne strukture (niz, vektor, lista, riječnik) princip rada, prednosti i nedostaci, primjeri zadatka sa odabirom optimalne strukture ✓ Algoritmi za rad na datotekama, pretraživanje datoteka koristeći indeks 	<ul style="list-style-type: none"> ✓ Razumjeti pojam algoritam i analizirati primere jednostavnih algoritama ✓ Sposobnost kreiranja/implementacije jednog (ili više) algoritama sortiranja i bibliotečne funkcije za sortiranje, u programskim zadacima u kojima je potrebno sortiranje. ✓ Razumjeti binarnu pretragu. ✓ Sposobnost analize (odabira) i kreiranje/primjene binarne pretrage (ili neke druge) u bazi podataka o učenika (sa sortiranjem) i efikasnom pretragom (recimo po imenu i prezimenu). ✓ Razumjeti bibliotečne strukture (niz, vektor, lista, riječni) princip rada, prednosti i nedostatke. ✓ Sposobnost analize i primjene zadatak sa odabirom optimalne strukture. ✓ Sposobnost analize i primjene algoritama za rad na datotekama, pretraživanju dataoteka koristeći indeks.
Razvoj složenih programskih rješenja i projektni zadatak	19	<ul style="list-style-type: none"> ✓ Diskusija o projektnim zadacima i podjela zadataka ✓ Organizacija koda, dekompozicija (datoteke zaglavljaja /header/, razdvajanje deklaracije 	<ul style="list-style-type: none"> ✓ Diskutovati i odabrati zadatak (projekta/programa) na kojem će učenici raditi. ✓ Analizirati i dizajnirati/planirati izradu pseudo kod izabranog programa.

	<p>od definicije klase)</p> <ul style="list-style-type: none"> ✓ Pravila pisanja čitljivog koda: imenovanje konstanti, promjenljivih i funkcija, osnovni pojmovi refactoringa (code smells) ✓ Povezivanje dosadašnjih znanja u svrhu razvoja jednostavne baze podataka: <i>predstavljanje tabelarnih podataka u grafičkom interfejsu</i>, povezivanje sa strukturalnom datotekom ✓ Binarne datoteke, pojam serijalizacije, XML ✓ Strategije testiranja i otkrivanja grešaka u kodu, jedinični testovi (unit testing), pojam pokrivenosti koda testovima (coverage), test driven development, korištenje debuggera ✓ Konsultacije u vezi projektnih zadataka ✓ Odbrana projektnih zadataka 	<ul style="list-style-type: none"> ✓ Odabire i upotrebljava algoritme i algoritamske strukture za rješavanje problema. ✓ Procjenjuje prednosti i ograničenja algoritamskog pristupa u rješavanju problema. ✓ Organizuje kod, dekompoziciju (datoteke zaglavljia /header/, razdvajanje deklaracije od dfinicije klase). (samoodabranog projekta) ✓ Sposobnost kreiranja pravila pisanja čitljivog koda: imenovanje konstanti, promjenljivih i funkcija. (samoodabranog projekta) ✓ Sposobnost implementiranja osnovnih pojmova refactoringa (code smells). (samoodabranog projekta) ✓ Sposobnost povezivanja i implementacij dosadašnjih znanja u svrhu razvoja jednostavne baze podata; povezivanja sa strukturalnom datotekom. (primjena u samoodabranog projektu) ✓ Razumjevanje primjene binarne datoteke, pojma serijalizacije, XMLa. ✓ Pomaže drugima, predstavlja, diskutuje i upoređuje svoje složene algoritme sa algoritmima drugih učenika iz razreda. ✓ Predstavlja, diskutuju i upoređuje svoje algoritme i prateću dokumentaciju sa algoritmima drugih učenika iz razreda. ✓ Profesoru predstavlja, diskutuje sa njim i donosi važne odluke kako na najbolji način izradi samoodabranom programu/projektu ✓ Razumjevanje i sposobnost razvijanja strategije testiranja i otkrivanja grešaka u kodu, kao i primjena jediničnih testova (unit testing), pokrivenosti koda testovima (coverage), test driven development i korištenje debuggera na izradi samoodabranom
--	---	--

			<p>✓ projektu. Objašnjava i predstavlja rad svog samoodabranog projekat pred razredom.</p>
--	--	--	---

NAČINI OSTVARIVANJA PROGRAMA¹

Na početku školske godine upoznati učenike sa ciljevima i ishodima nastave, odnosno učenja, planom rada i načinima ocjenjivanja.

Oblici nastave: Nastava se realizuje kroz laboratorijske vježbe/praktični rad na računaru primjenom kombiniranih oblika rada.

Mjesto realizacije nastave: Nastava na predmetu Informatika se realiziraje u kabinetima informatike.

Podjela odjeljenja u grupe: Prilikom realizacije nastavnog procesa na predmetu Informatika odjeljenje se dijeli na dvije grupe u skladu sa važećim Pedagoškim standardima za srednje obrazovanje.

MEĐUPREDMETNA KORELACIJA

Bosanski, hrvatski, srpski jezik i književnost – komunikacione vještine, obrada teksta

Engleski jezik – komunikacione vještine, obrada teksta, korištenje literature i interneta

Matematika – numerička obrada podataka

Fizika – kreativno računarstvo

Ostale prirodne nauke – kreativno računarstvo

Likovna kultura i medijska kultura – grafička obrada podataka

Psihologija – kreativno računarstvo, grafička obrada podataka, prezentacijske vještine

Sociologija – kreativno računarstvo, grafička obrada podataka, prezentacijske vještine

Svi ostali predmeti – napredno pretraživanje interneta, prezentacijske vještine, grafička obrada podataka

¹ NPP Informatika za gimnaziju, MONKS avgust 2016. godine

PRAĆENJE, VREDNOVANJE I OCJENJIVANJE

Nastavnik treba evaluirati učenicka postignuća na različite načine. Pri tome treba da koristi pozitivna pedagoška iskustva i dostignuća u nastavi. Evaluacija treba biti kontinuirana, javna i podsticajna.

Preporuke:

- ocjenjivanje teoretskih znanja treba obavljati usmeno i/ili primjenom testova koje nastavnik sam kreira na osnovu svojih planova, literature i sličnih testova pronađenih na internetu;
- ocjenjivanje praktičnih znanja treba biti na osnovu pripremljenih zadataka za rješavanje problema na računaru u toku ili izvan nastave, koje će učenik prezentirati pred ostalim učenicima;
- ocjenjivanje se može primijeniti i na aktivnost iz domena informatike koje su provedene za druge predmete, projekte ili za potrebe stručnih službi škole itd. (izrada prezentacije, videa, aplikacije i sl.) pri čemu učenik ne treba da zanemaruje ostale obaveze u nastavi informatike;
- ocjena treba da uključuje i teoretsko i praktično znanje učenika kao i njegov interes, trud i pomoć drugim učenicima u savladavanju gradiva;

PROFIL I STRUČNA SPREMA NASTAVNIKA²

U skladu sa Zakonom o srednjem obrazovanju („Službene novine“ Kantona Sarajevo broj: 23, od 15. juna 2017. godine), Član 120. (Profil i stručna sprema nastavnika), Stav 3. zakona stoji:

Općebrazovnu, stručno-teorijsku, praktičnu i nastavu u okviru laboratorijskog rada, u skladu sa stavom (2) ovog člana, u srednjoj školi izvode lica:

- a) sa završenim najmanje VII stepenom stručne spreme, sa zvanjem profesora, ili završenim drugim fakultetom i položenom pedagoško-psihološkom i metodičko-didaktičkom grupom predmeta i
- b) sa završenim II, odnosno III ciklusom bolonjskog visokoobrazovnog procesa na nastavničkom fakultetu ili drugom fakultetu i položenom pedagoško-psihološkom i metodičko-didaktičkom grupom predmeta.

Nastavu informatike u gimnaziji izborno područje informacionih tehnologija mogu izvoditi lica koja su završila:

1. Prirodno-matematički fakultet:

- Diplomirani matematičar-informatičar
- Magistar softverskog inženjerstva
- Magistar matematike, nastavnički smjer
- Magistar matematičkih nauka, smjer teorijska kompjuterska nauka,
- Svršenici Prirodno-matematičkog fakulteta informatičkog i/ili računarskog usmjerenja.

2. Elektrotehnički fakultet:

- Diplomirani inženjer informatike i računarstva,
- Svršenici Elektrotehničkog fakulteta informatičkog i/ili računarskog usmjerenja.

² Komisija za izradu Nastavnog plana i programa daje prijedlog za Profil i stručnu spremu nastavnika, a Ministarstvo za obrazovanje, nauku i mlade Kantona Sarajevo je nadležno da navedene profile uskladi sa zakonskom regulativom

3. Fakultet informatičkog i/ili računarskog usmjerenja sa završenim četverogodišnjim studijem u skladu sa gore navedenim članom Zakona, tačkom a) odnosno tačkom b) sa stečenim zvanjima iz sljedećih oblasti:

- Matematike i informatike
- Informatike i/ili računarstva
- Softverskog inženjerstva
- Kumpjuterskih/Računarskih nauka
- Informativnih tehnologija
- Informatike i tehničkog odgoja

NAPOMENA: Profil i stručna sprema profesora na predmetu Informatika ne odnosi se na prosvjetni kadar koji predaje predmet Informatika u gimnazijama po ugovoru na neodređeno vrijeme prije donošenja ovog Nastavnog plana i programa u skladu sa prethodnim Nastavnim planom i programom za predmet Informatika u gimnazijama iz 2003. godine.³

PREPORUČENI IZVOR INFORMACIJA ZA UČENJE

1. J. Štribar i B. Motik, *Demistificirani C++*, Zagreb 2001 (ili bilo koje drugo izdanje npr. 2014)
2. Dennis M. Ritchie i Brian W. Kernighan, *Programski jezik C*, Drugo izdanje
3. Željko Jurić, *Principi programiranja /kroz programski jezik C++/, Radni materijali – akademska godina 2009/10*

³ NPP Informatika za gimnaziju, MONKS avgust 2016. godine