



Bosna i Hercegovina
Federacija Bosne i Hercegovine
ZENIČKO-DOBOJSKI KANTON
MINISTARSTVO ZA OBRAZOVANJE, NAUKU, KULTURU I SPORT

KURIKULUM NASTAVNOG PREDMETA
PROGRAMIRANJE
ZA GIMNAZIJE

Zenica, juni 2023.



Bosna i Hercegovina
Federacija Bosne i Hercegovine
ZENIČKO-DOBOJSKI KANTON
MINISTARSTVO ZA OBRAZOVANJE, NAUKU, KULTURU I SPORT

KURIKULUM NASTAVNOG PREDMETA

PROGRAMIRANJE

ZA GIMNAZIJE

Zenica, juni 2023.

Kurikulum nastavnog predmeta Programiranje za gimnazije

Izdavač: Ministarstvo za obrazovanje, nauku, kulturu i sport Zeničko-dobojskog kantona

Za izdavača: Draženka Subašić, ministrica

Voditeljica Stručnog tima:

Aida Salkić, direktorica Pedagoškog zavoda Zenica

Grupa za izradu predmetnog kurikulumuma:

mr.sc. Adin Begić, voditelj

mr.sc. Edin Hadžikadunić, koordinator

dr.sc. Almir Sivro, koordinator

Mujo Zeničanin, dipl.ing., član

Zerina Šehović, prof., član

mr.sc. Mehmed Ahmetagić, član

Recenzenti:

prof.dr. Edin Berberović

Ajla Halilović, MA

Tehnička priprema i uređenje:

Pedagoški zavod Zenica

SADRŽAJ

A/ OPIS PREDMETA	4
B/ CILJEVI UČENJA I PODUČAVANJA PREDMETA	6
C/ OBLASNA STRUKTURA PREDMETNOG KURIKULUMA	6
D/ ODGOJNO-OBRAZOVNI ISHODI	9
GIMNAZIJA	9
3. razred gimnazije	9
4. razred gimnazije	13
E/ UČENJE I PODUČAVANJE	16
F/ VREDNOVANJE U PREDMETNOM KURIKULUMU	18
G/ PROFIL I STRUČNA SPREMA NASTAVNIKA	20

A/ OPIS PREDMETA

Sve veća upotreba računara, pametnih uređaja i drugih uređaja koji u sebi imaju integrirane pametne module, zahtijeva dobro poznavanje programiranja i programerskih tehnika. Programiranje je disciplina koja ima svoja specifična pravila, uključuje logiku i matematiku, te obiluje svojim posebnim pristupom i načinom razmišljanja. Danas na tržištu postoji veliki broj programskih jezika koji se razlikuju u nijansama i načinu pisanja koda. Bez obzira koji programski jezik odaberemo, suština je ista. Učenici će proučavanjem ovog predmeta imati mogućnost upoznati se sa strukturnim i objektnim programiranjem koje će im pomoći u sveukupnom razumijevanju tehnika programiranja.

Kroz predmet programiranje učenici razvijaju sposobnost prilagodbe različitim programskim jezicima. Učenici će biti u mogućnosti da ih međusobno razlikuju i formiraju kritički stav prema odabranom programskom jeziku. Također, učenjem ovog predmeta učenici će shvatiti razliku između računarstva i informatike, prepoznat će značaj poznavanja programiranja kroz razvoj vlastitih aplikacija u različitim alatima, te će vladati pojmovima sekvenca, selekcija i iteracija i pravilno ih upotrebljavati u razvoju aplikacija, bez obzira na odabir programskog jezika.

Znanje i vještine stečene na časovima programiranja mogu se primjenjivati u različitim sferama društva. Sekvenca, selekcija i iteracija se provlače kroz cijeli život. Svakodnevno donosimo različite odluke, svakodnevno ponavljamo neke operacije, a pojedine operacije se pojavljuju periodično i rješavamo ih jednokratno. Isto se provlači i kroz programiranje. Predmet programiranje predstavlja jedan od najsloženijih dijelova informatike i predstavlja jedan od najvećih izazova u informatičkom svijetu.

Učenjem programiranja, učenici razvijaju sposobnost kreativnog i kritičkog razmišljanja kroz analizu zadatih problema. Učenici će shvatiti da za jedan problem postoji potencijalno mnogo rješenja, ali bez obzira koji način odaberu, rezultat uvijek mora biti jasan i tačan. Iz ovoga proizilazi da će učenici proučavanjem ovog predmeta steći širok spektar kompetencija koje su primjenjive kroz cijeli život, a najvažnije su primjenjive za sve programerske tehnike i programske jezike.

Informatička i digitalna pismenost se stavlja u prvi plan. Učenici se ciljano koriste različitim programerskim tehnikama, te se njima služe u različitim situacijama i koriste ih u realizaciji različitih zadataka i problema. Pravilno se koriste programskim jezicima i koriste njihovu sintaksu, čime se potiče stvaranje znanja koja su lahko primjenjiva u situacijama učenja novih stvari i novih predmeta.

Programiranje kod učenika razvija istraživački duh, te obradom ulaznih podataka razvija kritički stav prema odabranoj programerskoj tehnici, prilagođavanju programa i prihvatanju izmjena.

Programiranje se naslanja na algoritamski način razmišljanja i nadovezuje se na vještine stečene u matematici. Također, poznavanje engleskog i maternjeg jezika je primjenjivo i poželjno u razumijevanju programiranja kroz poznavanje sintakse.

Programiranje kao dio rješavanja problema pomoću IKT-a je direktno povezano sa matematikom i informatikom kao općeobrazovnim predmetima, te bazama podataka kao stručno-teorijskim predmetom. Elementarno poznavanje matematike je neophodno prilikom realizacije matematičkih i logičkih zadataka, pa je poznavanje matematike neophodno u realizaciji predmeta Programiranje. Informatika kao predmet treba da postavi temelje za predmet Programiranje, ali u isto vrijeme ne mora ulaziti u dubinu tehnika programiranja.

Učenje predmeta programiranje se odvija interakcijom sa drugim učenicima te se podstiče rad u grupama. U podučavanju ovog predmeta je poželjno posjedovati informatički kabinet, jer programiranje je najefikasnije uz korištenje računarske opreme.

Programiranje pripada oblastima rješavanje problema primjenom IKT-a. Podučava se u trećem i četvrtom razredu gimnazija u informaciono-komunikacionom izbornom području.

B/ CILJEVI UČENJA I PODUČAVANJA PREDMETA

Ciljevi učenja i podučavanja nastavnog predmeta Programiranje su sljedeći:

1. Korištenje algoritama i dijagrama toka za kreiranje programskih rješenja kroz sekvencu, selekciju i iteraciju.
2. Korištenje programskih jezika za kreiranje strukturnih i objektno orijentisanih programa, a kroz to će primjenjivati tačnost i urednost pisanja koda, inicirati promjene u kodu, razvijati kritički stav i učestvovati u timskom radu.
3. Razvijanje znanja i vještina za ispravnu upotrebu raznih programskih jezika, s ciljem razvijanja ličnosti i stalnog usavršavanja kroz cjeloživotno učenje.
4. Povezivanje i praktična primjena usvojenih znanja kroz strukturno i objektno orijentisano programiranje.
5. Razumijevanje i primjena različitih programskih struktura i struktura podataka kao što su sekvenca, selekcija, iteracija, nizovi, matrice, stringovi, datoteke, klase, objekti, metode i programiranje baza podataka.
6. Primjena pravila programiranja kroz praksu, izrada programske dokumentacije samostalno ili u timu, provjera autorskih prava i zaštita autorskih prava vlastitog programa.

C/ OBLASNA STRUKTURA PREDMETNOG KURIKULUMA

Uvod

III razred:

1. Programski jezici i historijski razvoj programskih jezika
2. Algoritamsko rješavanje problema i strukture podataka
3. Osnovne programske strukture
4. Funkcije
5. Stringovi, slogovi i datoteke
6. Pisanje dokumentacije

IV razred:

1. Strukture podataka i pointeri
2. Primjena struktura podataka i pointera
3. Objektno orijentisano programiranje
4. Alati za brzi razvoj aplikacija
5. Programiranje baza podataka
6. Pisanje dokumentacije za programe povezane sa bazom podataka

A. Informacione i komunikacione tehnologije

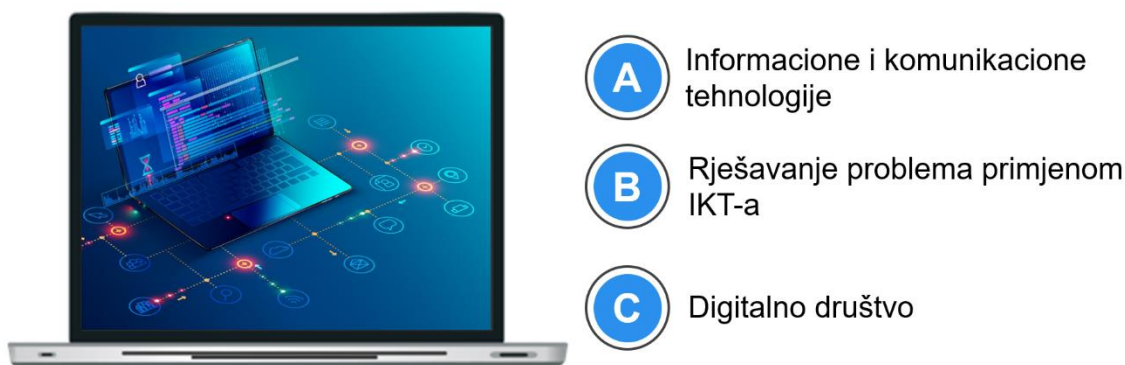
IKT predstavlja najbolji alat koji nam je dostupan u svakodnevnom učenju, kao i za učenje i rad u online okruženju. Potrebno je stalno praćenje i usavršavanje u korištenju novih hardverskih i softverskih rješenja, da bi se pravilno izabrala tehnologija i odgovarajući softver za primjenu u nekoj oblasti. Da bi se efikasno koristila IKT potrebno je poznavati osnovne informatičke pojmove, kao što su vrste memorija i dugotrajnost zapisa u njima, količina memorije koju zauzima neki fajl, brzina prenosa podataka, kreiranje rezervnih kopija. Od brzine koja nam je dostupna na mreži zavisi da li ćemo koristiti tekst, zvuk ili video u komunikaciji sa drugima, pa je potrebna pravilna procjena koja se temelji na navedenim osnovnim informatičkim pojmovima.

B. Rješavanje problema primjenom IKT-a

Za rješavanje problema koristimo razna softverska rješenja zavisno od toga da li radimo sa brojevima, tekstom, slikama, zvukom, videom. Poznavanje softvera i podataka koje on koristi jako je bitno za pravilan izbor metode za rješavanje konkretnog zadatka. Informacije su svima dostupne na internetu, ali ih je potrebno pronaći. Traženje informacija i postavljanje pravilnog upita je osnovni korak u pronalaženju rješenja za neki problem. Poznavanjem osnovnih pojmova, načela i zakonitosti zajedno sa vještinom pronalaženja informacija na internetu i logičkim povezivanjem i zaključivanjem može se doći do rješenja većine problema koji se postavljaju učeniku. Algoritamsko rješavanje problema predstavlja prevođenje nekog problema iz našeg okruženja u niz koraka koji su prilagođeni računaru. Rješavanje ovakvih problema razvija kod učenika logiku, modeliranje problema, indukciju, dedukciju i apstrakciju. Od korisnika učenik postaje kreator programa koje može dalje usavršavati i dijeliti. Stalno ispravljanje grešaka u programu i usavršavanje i poboljšavanje programa razvija samokritičnost i upornost kod učenika.

C. Digitalno društvo

Prelazak u digitalno društvo se desio toliko brzo da ga još nisu adekvatno regulisale ni države ni obrazovne institucije. Pristup digitalnom društvu bi trebao biti omogućen svakom odraslom čovjeku kao i učenicima, ali pod nadzorom roditelja. Potrebno je imati znanje i vještine za razmjenu informacija, ali i za zaštitu svojih prava i lične sigurnosti. Digitalno društvo olakšava mnoge aspekte života: učenje, podučavanje, bankarstvo, izdavaštvo, rad od kuće, informisanje. S druge strane potrebna je stalna edukacija da bi se zaštitili od raznih zloupotreba i prevara poput krađe identiteta, phishing-a, nasilja na internetu i raznih drugih opasnih radnji na internetu.



Oblasna struktura predmetnog kurikuluma Programiranje

U nastavku slijedi dio koji se odnosi na odgojno-obrazovne ishode koji su okosnica predmetnog kurikuluma i razrađeni su za svaku od tri oblasti (domene) na kojima se temelji. Odgojno-obrazovni ishodi pomažu nastavnicima u praćenju napretka učenika i u vrednovanju učeničkih postignuća. Tokom pripremanja procesa učenja i podučavanja nastavnik treba povezati odgojno-obrazovne ishode sa sadržajima navedenim u kurikulumu i metodama podučavanja. U tabelama su odgojno-obrazovni ishodi označeni šiframa. Skraćenice poput A.III.1. ili B.IV.3. označavaju redom: oblast kojoj ishod pripada (A. Informacione i komunikacione tehnologije, B. Rješavanje problema primjenom IKT-a i C. Digitalno društvo), godinu podučavanja predmeta (III. – treći razred i IV. – četvrti razred u gimnaziji), te redni broj odgojno-obrazovnog ishoda koji se podučava u sklopu navedene oblasti (1. – prvi ishod, 2. – drugi ishod, ...). Skraćenice TIT 3.1.1. ili npr. TIT 4.1.2. označavaju poveznice sa Zajedničkom jezgrom nastavnih planova i programa za tehniku i informacione tehnologije definisanoj na ishodima učenja.

D/ ODGOJNO-OBRAZOVNI ISHODI**GIMNAZIJA****3. razred gimnazije /70 nastavnih sati godišnje/**

Oblast: A/Informacione i komunikacione tehnologije	
Ishod učenja	Razrada ishoda
A.III.1. Analizira historijski razvoj programskih jezika.	<ul style="list-style-type: none"> • Definiše osnovne pojmove programiranja. • Analizira programske jezike prema namjeni i podjeli. • Objašnjava historijski razvoj programskih jezika.
Poveznice sa ZJNPP	TIT 3.1.1.
Ključni sadržaji	
Program. Programski jezik. Mašinski jezici. Asemblerski jezici. Strukturno orijentisani programski jezici. Objektno orijentisani programski jezici.	
Preporuke za ostvarenje ishoda	
Prilikom realizacije oblasti na temu programskih jezika i historijskog razvoja programskih jezika, učenicima pojasniti pojam programa i definisati programski jezik. Omogućiti učenicima da samostalno analiziraju razvoj programskih jezika kroz historiju. Dati učenicima zadatak da pronađu sličnosti i razlike između programskih jezika u skladu sa njihovim historijskim razvojem. Vrednovati učenike kroz aktivnost i usmeni odgovor.	

Oblast: B/Rješavanje problema primjenom IKT-a	
Ishod učenja	Razrada ishoda
B.III.1. Rješava jednostavne i složene probleme korištenjem algoritamskih struktura.	<ul style="list-style-type: none"> • Objašnjava algoritamsku strukturu. • Kreira algoritme upotrebom dijagrama toka. • Koristi sekvencu, selekciju i iteraciju. • Opisuje strukture podataka. • Rješava jednostavne i složene probleme upotrebom dijagrama toka.
Poveznice sa ZJNPP	TIT 4.1.1. TIT 4.1.2. TIT 4.1.3.
Ključni sadržaji	
Algoritam. Osobine algoritama. Faze razvoja algoritma. Načini predstavljanja algoritama. Sekvenca. Selekcija. Logika iskaza (Not, And, Or, ...). Iteracija. Strukture podataka.	
Preporuke za ostvarenje ishoda	
Prilikom realizacije oblasti na temu algoritamsko rješavanje problema i struktura podataka, poželjno je raditi u grupi i omogućiti učenicima da samostalno rješavaju različite zadatke. Obraditi sa učenicima faze razvoja algoritma te načine njihovog predstavljanja. Kroz praktične primjere obraditi sa učenicima sekvencu, selekciju i iteraciju. Također, obraditi sa učenicima različite strukture podataka. Kroz različite vrste vježbi i projekte provjeriti stečeno znanje iz algoritamskog rješavanja problema. Pratiti učenike kroz aktivnost i kroz rad u grupi.	

<p>B.III.2. Upotrebljava programski jezik za pisanje programa pomoću osnovnih programskih struktura.</p>	<ul style="list-style-type: none"> • Koristi komentare, ugrađene biblioteke, tipove podataka, varijable i konstante u odabranom programskom jeziku. • Koristi sekvencu, selekciju i iteraciju. • Kreira nizove i matrice. • Rješava i piše jednostavne i složene programe upotrebom odabranog programskog jezika.
<p>Poveznice sa ZJNPP</p>	<p>TIT 4.2.1. TIT 4.2.2.</p>
<p>Ključni sadržaji</p>	
<p>Sintaksa programskog jezika. Standardne biblioteke. Komentari. Komande. Tipovi podataka. Varijable i konstante. Matematičke operacije i funkcije. Sekvenca. Selekcija (IF, THEN, ELSE, Switch Case). Iteracija (For, While, Do While...). Nizovi i matrice. Dokumentacija.</p>	
<p>Preporuke za ostvarenje ishoda</p>	
<p>Za realizaciju ove oblasti potrebno je da učenici rade u kabinetu za informatiku vodeći računa da najviše dva učenika mogu koristiti jedan računar. U realizaciji ove oblasti obraditi sa učenicima historijski razvoj programskih jezika. Upoznati učenike sa sintaksom odabranog programskog jezika, pojasniti im biblioteke, komentare i značaj pisanja komentara. Obraditi sa učenicima komande koje se koriste u odabranom programskom jeziku i omogućiti učenicima da uvide razliku između komande i drugih dijelova koda. Obraditi sa učenicima tipove podataka i njihov značaj te kreirati varijable i konstante. Dati učenicima zadatak da analiziraju razlike između varijable i konstante. Obraditi osnovne programske strukture: sekvencu, selekciju (If-Then-Else, Switch-Case, For, While, Do While). Obraditi matematičke operacije i funkcije koje dolaze u nekoj od biblioteka odabranog programskog jezika. Dati učenicima zadatke da samostalno rješavaju neke probleme, na primjer neke matematičke zadatke. Dati im mogućnost da porede tehnike koje su koristili i omogućiti im da obrazlože kako su došli do rješenja. Obraditi sa učenicima nizove i matrice, objasniti im razlike i omogućiti učenicima da samostalno rješavaju probleme korištenjem nizova i matrica. Kroz različite vrste vježbi i projekata provjeriti stečeno znanje iz osnovnih programskih struktura. Pratiti učenike kroz aktivnost i kroz rad u grupi. Koristiti neki od programskih jezika: C++, Python, C#, Java i sl.</p>	
<p>B.III.3. Upotrebljava programski jezik za pisanje i izradu svojih funkcija.</p>	<ul style="list-style-type: none"> • Definiše funkcije. • Kreira samostalno funkcije prema sintaksi programskog jezika. • Koristi i poziva samostalno kreirane funkcije. • Kombinuje funkcije. • Rješava i piše jednostavne i složene programe upotrebom odabranog programskog jezika.
<p>Poveznice sa ZJNPP</p>	<p>TIT 4.2.1. TIT 4.2.2.</p>
<p>Ključni sadržaji</p>	
<p>Pojam funkcije. Osnovni dijelovi funkcije. Argumenti. Vraćanje rezultata. Funkcije koje ne vraćaju rezultate (void). Poziv funkcije u glavnom programu. Poziv funkcije unutar druge funkcije. Rekurzivna funkcija.</p>	

Preporuke za ostvarenje ishoda	
<p>Za realizaciju ove oblasti potrebno je da učenici rade u kabinetu za informatiku vodeći računa da najviše dva učenika mogu koristiti jedan računar. Objasniti učeniku značaj pisanja samostalnih funkcija i obraditi sa učenicima načine kreiranja funkcija. Također, učenicima pojasniti razliku između komande i funkcije. Zatim, pokazati učenicima kako se funkcije pozivaju u glavnom programu ili nekoj drugoj funkciji i kako funkcija vraća rezultat. Objasniti učenicima void funkcije i njihov značaj. Učenicima objasniti kada i gdje se koriste rekurzivne funkcije i koja su ograničenja kod korištenja istih. Dati učenicima niz problema u kojima će oni realizovati vlastite funkcije i vršiti poziv istih. Kroz različite vrste vježbi i projekata provjeriti stečeno znanje iz funkcija. Pratiti učenike kroz aktivnost i kroz rad u grupi. Koristiti neki od programskih jezika: C++, Python, C#, Java i sl.</p>	
<p>B.III.4. Upotrebljava programski jezik za pisanje programa korištenjem stringova i datoteka.</p>	<ul style="list-style-type: none"> • Definiše stringove i datoteke. • Koristi biblioteke i funkcije za rad sa stringom. • Razlikuje vrste datoteka (tekstualne i binarne). • Koristi biblioteke i funkcije za rad sa datotekama. • Rješava i piše jednostavne i složene programe upotrebom odabranog programskog jezika.
<p>Poveznice sa ZJNPP</p>	<p>TIT 4.2.1. TIT 4.2.2.</p>
Ključni sadržaji	
<p>Pojam string-a. Funkcije za rad sa stringom. Pretraživanje stringa, dodavanje u string, spajanje stringova, promjena sadržaja stringa. Pojam sloga. Datoteke, osnovne operacije sa datotekama (čitanje, pisanje, brisanje). Vrste datoteka. Ulazne i izlazne datoteke. Formiranje niza na osnovu sloga. Formiranje datoteke na osnovu sloga.</p>	
Preporuke za ostvarenje ishoda	
<p>Za realizaciju ove oblasti potrebno je da učenici rade u kabinetu za informatiku vodeći računa da najviše dva učenika mogu koristiti jedan računar. Obraditi sa učenicima stringove i funkcije nad stringovima u odabranom programskom jeziku. Pojasniti učenicima značaj korištenja stringova. Naučiti učenike korištenje slogova. Zatim, objasniti učenicima pojmove vezane za datoteku te vrste datoteka. Koristiti tehnike kreiranja datoteka i vršiti upis sadržaja u datoteku. Dati učenicima zadatak da analiziraju tekstualne i binarne datoteke te izvršiti analizu tekstualnih i binarnih datoteka. Također, naučiti učenike kako se podaci čitaju iz datoteke. Obraditi sa učenicima zadatke u kojima će uzimati određene podatke iz datoteke, vršiti njihovu obradu i davati rezultate ili ih upisivati u neku drugu datoteku. Naučiti učenike kako se formira niz i datoteka na osnovu sloga. Dati učenicima niz zadataka u kojima će koristiti stringove i kreirati datoteke. Kroz različite vrste vježbi i projekata provjeriti stečeno znanje iz stringova i datoteka. Pratiti učenike kroz aktivnost i kroz rad u grupi i kroz praktične vježbe u radu sa stringovima, slogovima i datotekama. Koristiti neki od programskih jezika: C++, Python, C#, Java i sl.</p>	

Oblast: C/ Digitalno društvo	
Ishod učenja	Razrada ishoda
<p>C.III.1. Samostalno kreira dokumentaciju za upotrebu i objavljivanje programa.</p>	<ul style="list-style-type: none"> • Razlikuje elemente dokumentacije programa. • Kreira potrebnu dokumentaciju.
<p>Poveznice sa ZJNPP</p>	<p>TIT 3.4.1.</p>
Ključni sadržaji	
<p>Elementi dokumentacije. Pisanje dokumentacije. Sistemski zahtjevi. Dokumentacija izvornog koda.</p>	

Preporuke za ostvarenje ishoda

Prilikom realizacije oblasti na temu pisanja dokumentacije programa, poželjno je da nastavnik pojasni učenicima elemente dokumentacije i da ih uvede u pravilno pisanje dokumentacije. Na osnovu nekog ranije urađenog programa, dati učenicima praktičan rad da opišu načine korištenja programa, sistemske zahtjeve i ograničenja u korištenju. Kroz praktičnu vježbu vrednovati kvalitet napisane dokumentacije analizom unaprijed dogovorenih elemenata.

4. razred gimnazije /90 nastavnih sati godišnje/

Oblast: A/Informacione i komunikacione tehnologije	
Ishod učenja	Razrada ishoda
A.IV.1. Analizira strukture podataka i pointere.	<ul style="list-style-type: none"> • Definiše strukture podataka. • Definiše pointere. • Razlikuje elemente strukture podataka.
Poveznice sa ZJNPP	TIT 3.1.1.
Ključni sadržaji	
Definicija strukture. Elementi strukture. Pointeri. Primjena struktura podataka i pointera.	
Preporuke za ostvarenje ishoda	
Za realizaciju ovog ishoda, potrebno je da nastavnik definiše strukture podataka i pointere. Potrebno je objasniti učenicima od kojih dijelova se sastoji struktura podataka i koja je njena primjena. Detaljno obrazložiti pointere i upotrebu pointera. Dati učenicima zadatak da definišu neku strukturu i da odrede elemente strukture. Povezati strukture i pokazivače. Kroz različite vrste vježbi i projekata provjeriti stečeno znanje iz struktura podataka i pointera. Pratiti učenike kroz aktivnost i kroz rad u grupi.	

Oblast: B/ Rješavanje problema primjenom IKT-a	
Ishod učenja	Razrada ishoda
B.IV.1. Upotrebljava programski jezik za pisanje i izradu struktura podataka.	<ul style="list-style-type: none"> • Opisuje strukture podataka. • Kreira strukturu podataka. • Kreira pointere. • Kreira vezane liste. • Rješava jednostavne i složene probleme upotrebom struktura podataka.
Poveznice sa ZJNPP	TIT 4.2.1. TIT 4.2.2.
Ključni sadržaji	
Kreiranje struktura. Deklaracija strukture. Pristup dijelovima strukture. Deklaracija pointera. Dodjela memorijske adrese pointeru. Pointeri na nizove i funkcije. Vezane liste. Dijelovi liste. Operacije nad listama.	
Preporuke za ostvarenje ishoda	
Za realizaciju ovog ishoda potrebno je da učenici rade u kabinetu za informatiku vodeći računa da najviše dva učenika mogu koristiti jedan računar. Obraditi sa učenicima strukture, deklaraciju struktura i objasniti im značaj kreiranja struktura. Zatim, naučiti učenike kako se koriste i kako se kreiraju pointeri i obrazložiti njihov značaj. Kroz različite vježbe, primijeniti znanja koja su učenici stekli iz oblasti struktura i pokazivača. Naučiti učenike kako se kreiraju vezane liste, koji su dijelovi liste i koje su operacije nad listom. Dati učenicima niz zadataka u kojima će rješavati zadate probleme korištenjem struktura, pokazivača i listi. Oslanjati se na prethodna stečena znanja, kao, na primjer, snimati rezultate u datoteku i slično. Kroz različite vrste vježbi i projekata provjeriti stečeno znanje iz struktura. Pratiti učenike kroz aktivnost i kroz rad u grupi.	

<p>B.IV.2. Upotrebljava programski jezik za objektno orijentisano programiranje.</p>	<ul style="list-style-type: none"> • Objašnjava pojam OOP i prednosti korištenja OOP. • Kreira klasu, metodu i objekte klase. • Rješava jednostavne i složene probleme upotrebom klasa, objekata i metoda.
<p>Poveznice sa ZJNPP</p>	<p>TIT 4.2.1. TIT 4.2.2.</p>
<p>Ključni sadržaji</p>	
<p>Objektno orijentisano programiranje. Klasa. Objekat. Metoda. Osnovne operacije sa klasama. Apstrakcija i enkapsulacija. Kontrola prava pristupa. Zajednički članovi klase. Konstruktori i destruktori klase. Operatorske funkcije. Polimorfizam.</p>	
<p>Preporuke za ostvarenje ishoda</p>	
<p>Za realizaciju ovog ishoda potrebno je da učenici rade u kabinetu za informatiku vodeći računa da najviše dva učenika mogu koristiti jedan računar. Uvesti učenike u objektno orijentisano programiranje. Potrebno je detaljno uraditi teorijski osvrt na klasu, objekat, metodu i osnovne operacije nad klasama. Praktično realizovati sve vezano za kreiranje klase, metoda i objekata. Obraditi sa učenicima apstrakciju i enkapsulaciju, kontrolu prava pristupa, zajedničke članove klase, konstruktore i destruktore klase (ukoliko ih podržava odabrani programski jezik), operatorske funkcije i polimorfizam. Uraditi sa učenicima niz zadataka vezanih za objektno orijentisano programiranje. Analizirati sa učenicima značaj korištenja objektno-orijentisanog programiranja. Kroz različite vrste vježbi i projekata provjeriti stečeno znanje iz OOP. Pratiti učenike kroz aktivnost i kroz rad u grupi. Koristiti neki od programskih jezika: C++, Python, C#, Java i sl.</p>	
<p>B.IV.3. Upotrebljava programski jezik za GUI programiranje.</p>	<ul style="list-style-type: none"> • Razlikuje dijelove alata za programiranje GUI aplikacija. • Kreira okruženje za aplikaciju korištenjem kontrola. • Kreira programski kod za elemente forme. • Kreira GUI programe.
<p>Poveznice sa ZJNPP</p>	<p>TIT 4.2.1. TIT 4.2.2.</p>
<p>Ključni sadržaji</p>	
<p>Prednosti korištenja GUI alata. Forma. Kontrole i njihove osobine (komandno dugme, text box, labela, radio dugme, check box, list box, combo box, frame). Programiranje elemenata forme. Programiranje miša. Programiranje tastature. Kontrole i događaji.</p>	
<p>Preporuke za ostvarenje ishoda</p>	
<p>Za realizaciju ovog ishoda potrebno je da učenici rade u kabinetu za informatiku vodeći računa da najviše dva učenika mogu koristiti jedan računar. Uvesti učenike u programiranje Windows aplikacija i pojasniti im prednosti u korištenju GUI aplikacija. Kreirati različite forme korištenjem različitih kontrola kao što su text box, label, command button, combo box, list box, radio button i druge. Objasniti učenicima kako se programiraju kontrole. Obraditi sa učenicima programiranje događaja nad mišem i tastaturom. Obraditi kontrole i događaje. U ovoj oblasti bi bilo poželjno programirati neku Windows aplikaciju ili programirati neku zabavnu aplikaciju, odnosno igru. Programiranje igre bi moglo posebno motivisati učenike, jer su oni uglavnom zainteresirani ovim sadržajima na internetu. Također, programiranje igre omogućava programiranje vještačke inteligencije i raznih akcija koje su unaprijed definisane u zavisnosti od uslova i okruženja u kojim se nalaze likovi. Kroz različite vrste vježbi i projekata provjeriti stečeno znanje GUI programiranja. Pratiti učenike kroz aktivnost i kroz rad u grupi. Napisati GUI aplikaciju. Koristiti neki od programskih jezika: C++, Python, C#, Java, VisualStudio i sl. koji imaju podršku za razvoj GUI aplikacija.</p>	

B.IV.4. Upotrebljava programski jezik za povezivanje i manipulaciju podacima unutar baze podataka.	<ul style="list-style-type: none"> • Objasnjava elemente baze podataka. • Kreira konekciju za bazom podataka. • Kreira programski kod za manipulaciju podacima unutar baze podataka. • Kreira programe korištenjem konekcije sa bazom podataka.
Poveznice sa ZJNPP	TIT 4.2.1. TIT 4.2.2.
Ključni sadržaji	
Baza podataka. Konekcija i ODBC driver. Tabele. Upiti. Izveštaji. Manipulacija podacima unutar baze.	
Preporuke za ostvarenje ishoda	
<p>Za realizaciju ovog ishoda potrebno je da učenici rade u kabinetu za informatiku vodeći računa da najviše dva učenika mogu koristiti jedan računar. Pošto su učenici već upoznati sa predmetom Baze podataka, objasniti im značaj korelacije Programiranja i Baza podataka. Pojasniti im prednosti povezivanja baze podataka i programskih jezika. Naučiti učenike kako se kreira konekcija i korištenje ODBC driver-a. Omogućiti poziv tabele u programskom jeziku iz unaprijed kreirane baze podataka. Također, u programskom jeziku kreirati tabelu u odabranoj bazi. Vršiti upis podataka u tabele. Kreirati različite upite i povlačiti rezultate iz baze podataka. Izvršiti grupisanje i sortiranje podataka. Kreirati izvještaje koje je moguće printati. Pokazati učenicima kako se vrši manipulacija sa podacima unutar baze podataka. Dati učenicima niz zadataka u kojima će samostalno vršiti unos podataka u bazu i zadatke u kojima će kreirati različite upite i izvještaje. Kroz različite vrste vježbi i projekata provjeriti stečeno znanje iz oblasti programiranja baza podataka. Pratiti učenike kroz aktivnost i kroz rad u grupi. Koristiti neki od programskih jezika: C++, Python, C#, Java i sl.</p>	

Oblast: C/Digitalno društvo	
Ishod učenja	Razrada ishoda
C.IV.1. Samostalno kreira dokumentaciju za upotrebu i objavljivanje programa koji su povezani sa bazom podataka.	<ul style="list-style-type: none"> • Razlikuje elemente dokumentacije programa. • Kreira potrebnu dokumentaciju. • Kreira dokumentaciju za opis baze u projektu.
Poveznice sa ZJNPP	TIT 3.4.1.
Ključni sadržaji	
Elementi dokumentacije. Pisanje dokumentacije. Sistemske zahtjevi. Dokumentacija izvornog koda. Dokumentacija baze podataka. Opis načina povezivanja programa i baze podataka.	
Preporuke za ostvarenje ishoda	
Prilikom realizacije oblasti na temu pisanja dokumentacije programa, poželjno je da nastavnik pojasni učenicima elemente dokumentacije i da ih uvede u pravilno pisanje dokumentacije. Na osnovu nekog prethodno urađenog programa u kojem se povezuju program i baza podataka, dati učenicima praktičan rad da opišu načine korištenja programa, sistemske zahtjeve, ograničenja u korištenju i način na koji su baza podataka i program povezani. Kroz praktičnu vježbu vrednovati kvalitet napisane dokumentacije analizom unaprijed dogovorenih elemenata.	

E/ UČENJE I PODUČAVANJE

Učenje i podučavanje predmeta Programiranje organizira se prema zadanim odgojno-obrazovnim ciljevima i ishodima učenja. Nastavnici imaju mogućnost odabira različitih pristupa u skladu sa potrebama, interesima i nivoima znanja i vještina učenika kao i uslovima rada. Uvažavajući postavljena načela učenja i podučavanja, svaki nastavnik programiranja može osmisliti izvedbu kurikuluma u najboljem interesu učenika.

Učenje i podučavanje usmjereno je na kreativnost učenika, samostalno istraživanje, prikupljanje podataka i povezivanje sadržaja. Učenici prikupljaju podatke, predstavljaju algoritme, obrađuju materijale, osmišljavaju interface i pišu kodove. Uslovi podučavanja predmeta usmjereni su na metodičku raznovrsnost svih raspoloživih metodičkih sistema i metoda koje će osigurati konkretno i potpuno znanje. Učenje i podučavanje programiranja ostvaruje se procesima:

- definisanja problema,
- prikupljanja podataka za rješavanje problema,
- definisanja promjenljivih za zadati problem,
- definisanja entiteta i veza među entitetima,
- kreiranja grafikona za zadati problem,
- korištenja prikladnih alata za realizaciju i izradu dijagrama,
- ponovno razmatranje definisanog problema i upoređivanje s prethodnim rješenjima,
- kodiranje definisanih zahtjeva,
- uklanjanja grešaka,
- vođenja dokumentacije.

Učenici su aktivni kreatori znanja koji uz pomoć nastavnika pronalaze, razumiju i koriste se znanjem kako bi donosili bolje odluke u realizaciji problema. Učenici međusobno razmjenjuju iskustva i pomažu jedni drugima u realizaciji problema. Učenici mogu raditi individualno, u paru ili u grupama u zavisnosti od teme. Podjelu u manje grupe je moguće primijeniti u projektnom radu, problemskoj nastavi, te timskom radu.

Nastavnik u ostvarivanju odgojno-obrazovnih ishoda uzima u obzir interese učenika i njihove sposobnosti. Aktivno prati napredak svakog učenika i metodu rada prilagođava potrebama učenika i njihovim sposobnostima. Prilikom osmišljavanja zadataka, nastavnik može davati matematičke, logičke i problemske zadatke. U III razredu preporučuje se više matematičkih i logičkih i dijelom problemskih zadataka, dok u IV razredu davati više problemskih zadataka. U odabiru zadataka voditi računa o trenutnom znanju učenika iz drugih oblasti. Ukoliko se daje zadatak iz oblasti koju učenici nisu prešli u dosadašnjem obrazovanju, nastavnik je dužan da učenicima objasni metode kojim će doći do rješenja. Nastavnik potiče učenike na istraživanje, daje im praktične vježbe u kojima će iskustvenim učenjem ostvarivati zadane odgojno-obrazovne ishode učenja. Također, nastavnik savjetuje učenike i prati ih tokom realizacije zadataka te im pomaže u otklanjanju grešaka.

Poželjno je koristiti i razne edukativne platforme, poput npr. eTwinning-a koji nudi priliku za razvoj i jačanje novih vještina i kompetencija za 21. stoljeće (<https://www.etwinning.net/bs>).

Načini realizacije učenja i podučavanja:

- projektno učenje i podučavanje,
- problemsko učenje i podučavanje,
- praktično učenje i podučavanje.

Posebno je važno da nastavnik pokaže pravilnu upotrebu i primjenu odabranog programskog jezika i pravilnu upotrebu kabinetske opreme. Vježbe se primjereno biraju od jednostavnih ka složenim. Bitno je da nastavnik na kraju rada razgovara sa učenicima, razmijeni utiske sa učenicima i da omogući učenicima da zajednički analiziraju potencijalna rješenja. Pojasniti učenicima koji nisu najbolje razumjeli zadatak ili koji nisu izveli zadatak do kraja gdje su napravili grešku i kako izbjeći istu grešku u narednoj vježbi. Prilikom realizacije nastavnih sadržaja izbjegavati dominantno frontalni rad. Potrebno je stvoriti radno okruženje gdje će učenici raditi u parovima ili će grupno rješavati zadate probleme.

Za učenje i podučavanje predmeta Programiranje potrebni su materijalni resursi koji uključuju opremljen kabinet informatike. Kabinet treba imati priključak na internet. Na svakom računaru treba biti instaliran odabrani programski jezik za realizaciju odgojno-obrazovnih ishoda. Predloženi programski jezici su: C++, Python, C#, Java i drugi programski jezik koji zadovoljava potrebe kurikuluma. Poželjno bi bilo da jedan učenik sjedi za jednim računarem, ali ako to nije izvodivo, omogućiti da maksimalno dva učenika sjede za jednim računarem. Izvori učenja su udžbenici, radni materijali i razni izvori sa interneta.

Nastava iz predmeta programiranje se izvodi dva nastavna sata sedmično u III razredu (70 sati godišnje) i tri nastavna sata sedmično u IV razredu (90 sati godišnje). U III razredu poželjno bi bilo predmet Programiranja izvoditi kao blok čas. U IV razredu bi bilo poželjno izvoditi nastavu kao blok čas i jedan posebno čas u toku nastavne sedmice. Vrijeme potrebno za ostvarivanje postavljenih odgojno-obrazovnih ishoda unutar pojedine oblasti određuje nastavnik, vodeći računa o tome da obradi ključne sadržaje definisane kurikulumom.

F/VREDNOVANJE U PREDMETNOM KURIKULUMU

Vrednovanje je proces kojim se kontinuirano prati ostvarivanje postavljenih ciljeva učenja i podučavanja i odgojno obrazovnih ishoda. Programiranje koje se izučava u III i IV razredu srednje škole u gimnazijama u informaciono-komunikacionom izbornom području je jako kompleksno i pristup vrednovanju bi trebao biti kontinuiran i precizan. Poseban akcenat treba staviti na vrednovanje praktičnih radova i analize urađenih zadataka.

Postoje tri vrste vrednovanja:

Vrednovanje za učenje (formativno vrednovanje) bi trebalo biti povratna informacija o kvaliteti urađenog kojoj je svrha unaprijediti proces učenja i podučavanja. Ova vrsta vrednovanja podstiče saradnju između nastavnika, učenika i roditelja.

Vrednovanje kao učenje podrazumijeva aktivno uključivanje učenika u proces vrednovanja uz stalnu podršku nastavnika, kako bi se podstakao razvoj samoregulisanog učenja, učeničke samoprocjene, samovrednovanja i samoocjenjivanja. Da bismo to postigli kriteriji za vrednovanje i ocjenjivanje moraju biti precizni, jasni i transparentni.

Vrednovanje naučenog (sumativno vrednovanje) podrazumijeva procjenu nivoa postignuća učenika nakon određenog perioda (nakon određene teme, polugodišta i sl.). Po pravilu se iskazuje zaključnom ocjenom.

Vrednovanje pomaže da se što bolje ostvare ishodi znanja, ali i direktno podstiče učenike za daljnje napredovanje. Učenici se manje trude i pasivniji su u radu ukoliko vrednovanje rješenja njihovih zadataka nije učestalo. Vrednovanje učeničkih postignuća je kontinuirana djelatnost. Informacija o onome što učenici nisu dobro uradili za učenike može biti korisna u smislu napredovanja za ubuduće.

Programiranje podrazumijeva teorijska i dominantno praktična znanja, stoga je neophodno da nastavnik daje što više praktičnih vježbi koje bi radili u učionici i projekata koje bi učenici radili kod kuće. Koristiti matematičke, logičke i problemske zadatke. Posebno je potrebno obratiti pažnju na vrednovanje učenika kroz rad u paru ili grupi. Također, nastavnik bi trebao poticati učenike na samokritičnost i omogućiti učenicima da samostalno evaluiraju svoja znanja.

U svakoj oblasti predmeta Programiranje potrebno je u vrednovanju dati veću važnost u ocjenjivanju praktičnih radova kroz:

- vrednovanje odabranih primjera za entitete, atribute i podatke,
- vrednovanje pravilnog odabira strukture problema i pravilnog predstavljanja istih,
- vrednovanje pravilnog korištenja odabranih ili kreiranih funkcija od strane učenika,
- vrednovanje odabranih operacija,
- vrednovanje efikasno korištenih resursa,
- vrednovanje složenosti zadatka,
- vrednovanje krajnjih poruka i rezultata,
- vrednovanje cjelokupnih rezultata,
- vrednovanje rada u paru ili grupi,
- vrednovanje individualnih analiza zadataka i uklanjanja grešaka.

Veoma bitna stavka u vrednovanju je uključenost samog učenika u proces vrednovanja. Sistem ocjenjivanja treba biti transparentan i trebamo dati mogućnost učeniku samoprocjene po definisanom sistemu. Preporučuje se da prije evaluacije praktičnih radova učenici imaju na raspolaganju sve elemente vrednovanja da bi se bolje pripremali za nastavu i praktične zadatke. Uključivati učenike kao one koji će vršiti vrednovanje i procjenu praktičnih radova drugih učenika ili drugih grupa i timova. Vršnjačko vrednovanje je posebno dobro kod grupnog i timskog rada na praktičnim vježbama jer možemo ujedno vrednovati i učenike koji vrednuju tuđi rad. Učenici u tom slučaju trebaju poštovati definisana pravila i kriterije vrednovanja i ocjenjivanja. Programiranje kao predmet upravo to treba omogućiti.

Tehnike i indikatori kvaliteta vrednovanja:

- usmene provjere znanja,
- praktične vježbe,
- grupni projekti,
- aktivnost učenika,
- pismene provjere.

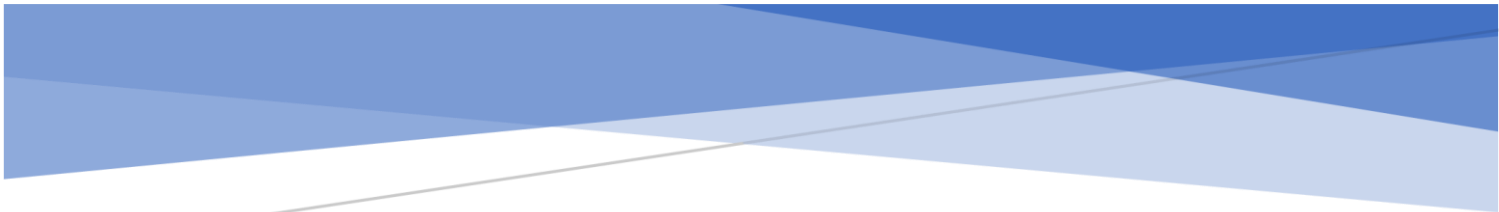
U zavisnosti od ciljeva oblasti biramo i načine vrednovanja i tipove zadataka. Prilikom zaključivanja ocjena treba obratiti pažnju na omjer reprodukcije teorijskog znanja, praktičnog rada i konačnog rezultata. Odnos bi trebao biti:

- 20% reprodukcija teorijskog znanja,
- 60% praktični radovi,
- 20% konačni rezultat.

Opći utisak nastavnika prilikom izvođenja ocjena ne smije biti subjektivan i treba se obrazložiti pred odjeljenjem u skladu sa detaljnim objašnjenjima onoga što je učenik u toku školske godine uspio da postigne kroz sve oblasti. Neophodno je da nastavnik vodi evidenciju o postignućima svakog učenika u toku školske godine kako bi mogao transparentno, precizno i objektivno iskoristiti svoja zapažanja u donošenju zaključne ocjene.

G/PROFIL I STRUČNA SPREMA NASTAVNIKA

- Nastavu programiranja mogu izvoditi lica koja su završila odgovarajući četverogodišnji studij i stekla zvanje:
 - profesor informatike,
 - profesor matematike i informatike,
 - profesor matematike, smjer matematika s informatikom,
 - diplomirani inženjer informatike, s položenom pedagoško-psihološko-didaktičko-metodičkom grupom predmeta,
 - diplomirani inženjer elektrotehnike, smjer informatika ili računarstvo, s položenom pedagoško-psihološko-didaktičko-metodičkom grupom predmeta,
 - softver inženjer, s položenom pedagoško-psihološko-didaktičko-metodičkom grupom predmeta,
 - diplomirani inženjer informacijskih tehnologija, s položenom pedagoško-psihološko-didaktičko-metodičkom grupom predmeta,
 - diplomirani ekonomista, smjer informatika, s položenom pedagoško-psihološko-didaktičko-metodičkom grupom predmeta,
 - profesor ostalih predmeta uz završen dvogodišnji kurs Informatike na fakultetu koji obrazuje informatički kadar (kurs mora verifikovati Nastavno naučno vijeće fakulteta).
- Nastavu programiranja mogu izvoditi i lica koja imaju završen najmanje II (drugi) ciklus Bolonjskog sistema studiranja u trajanju od jedne godine (60 ECTS bodova) ili dvije godine (120 ECTS bodova) – ukupno 300 ECTS bodova sa bodovima prvog ciklusa, koja su stekla akademsku titulu i zvanje magistra ili ekvivalenta za određenu oblast.
- Lica koja u toku studija nisu polagala ispite iz pedagoško-psihološko-didaktičko-metodičke grupe predmeta, dužna su ove ispite položiti u roku od godinu dana od dana stupanja na posao nastavnika.



[The main body of the page contains a large, faint watermark or ghosted text that is mostly illegible due to its low opacity and large size. It appears to be a long, multi-line paragraph of text.]

