

Nastavni plan i program za
predmet:
Softver inženjering

**Izorno područje: Informacione tehnologije
4. razred**

KANTON SARAJEVO
Ministarstvo za obrazovanje, nauku i mlade
August, 2020.

Softver inženjering

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

- razumije vezu između softver inženjeringa i drugih informatičkih predmeta koji su izučavani ranije
- dobro zna pojmove iz oblasti softver inženjeringa (na BHS i engleskom jeziku)
- upoznat je sa karakteristikama životnog ciklusa softvera i osnovnim osobinama industrijski najrasprostranjenijih izabranih modela životnog ciklusa softvera
- sposoban je postavljati (prava) pitanja u analizi problema (traženje dodatnih informacija i pojašnjenja)
- ima vještinu jasnog i konciznog izražavanja misli u sklopu softver inženjeringa. Ima sposobnost rada u grupi. Prepoznaje suštinu rečenog/napisanog.
- upoznat je sa konceptom softverske arhitekture
- poznaje elemente organizacije softverskog projekta i ima sposobnost da kao član tima učestvuje na projektu
- poznaje najčešće korištene uzorke ili primjere dizajna softvera (*Software Design Patterns*)
- upoznat je sa prednostima i značajem upravljanja softverskom konfiguracijom i motivacijom za postojanje
- zna koristiti softver za verzioniranje, kao i platformu za timski rad (npr. GitHub)
- snalazi se u programima koji se sastoje od više izvornih datoteka, ima sposobnost organizovanja koda i znanje pripreme izvornog koda za produkciju (kompiliranje).
- posjeduje znanja za automatizaciju procesa generisanja izvršnog koda od izvornog (*build*)
- poznaje jedno razvojno okruženje i postojanje alternativa,
- upoznat je sa pojmom SW okvir (*framework*) i zna navesti primjere i poznaje neke najvažnije osobine/ograničenja istih.
- upoznat je sa značajem pisanja automatiziranih testova (*unit tests*).
- zna prezentirati dio softverskog projekta na kom radi
- prati novosti iz oblasti softver inženjeringa, čuo je za nove ideje u oblasti.
- zna prepoznati pouzdane izvore informacija iz oblasti softver inženjeringa

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

REALIZACIJA PROGRAMA	ČAS
Upoznavanje, uvod u predmet, literatura i oprema	1
Ponavljanje gradiva iz stručnih predmeta iz 3. razreda	2
Pojam i uloga softver inženjeringa	3
Životni ciklus razvoja softvera i industrijski najrasprostranjeniji modeli životnog ciklusa softvera, evolucija softverskog proizvoda	9
Organizacija softverskog projekta – aspekt učesnika projekta	3
Pripreme za praktičan rad na izradi srednje složene aplikacije, u timovima od po tri člana ili individualno	6
Specifikacija softverskih zahtjeva	9
Upoznavanje sa konceptom softverske arhitekture	3
Dizajn i arhitektura softvera – Uzorci dizajna	9
Upravljanje softverskom konfiguracijom	6
Razvoj i implementacija: Postojeća razvojna okruženja, uvod u korištenje razvojnih SW okvira (<i>frameworks</i>), standardi kodiranja, implementacija, testiranje	24
Prezentacija napravljene aplikacije i diskusije o otvorenim pitanjima tokom razvoja softvera	9
Novosti u Softver inženjeringu	3
Provjera usvojenosti nastavnih sadržaja, vrednovanje rada učenika i zaključivanje ocjena	3
UKUPNO:	90

MJESEC	BROJ ČASOVA	REALIZACIJA PROGRAMA
PRVO POLUGODIŠTE		
SEPTEMBAR	12	1. Upoznavanje, uvod u predmet, literatura i pribor
		Ponavljjanje gradiva
		2. Veza sa gradivom Programiranja i Baza podataka (gradivni elementi softverskog sistema: baza podataka, programiranje osnovne funkcionalnosti)
		3. Veza sa gradivom Matematskih osnova računarskih nauka i Napredne primijenjene informatike (gradivni elementi softverskog sistema: mehanizmi nadzora rada sistema, mehanizmi administracije sistema, izvješatavanje, ...)
		Pojam i uloga softver inženjeringa
		4. Uvod u softver inženjering
		5. Uloga softver inženjering
		6. Značaj pouzdanosti softvera
		Životni ciklus razvoja softvera i industrijski najrasprostranjeniji modeli životnog ciklusa softvera
		7. Životni ciklus razvoja softvera
		8. Osnovne osobine industrijski najrasprostranjenijih modela životnog ciklusa razvoja softvera
		9. Klasični model: <i>waterfall</i>
10. Klasični model: spiralni		
11. Evolucija softvera		
12. Manifest agilnih metoda razvoja softvera		
OKTOBAR	12	13. Jedna agilna metoda (preporučeno SCRUM)
		14. Alati za saradnju
		15. Samostalan rad učenika na prikupljanju informacija uz mentorstvo profesora
		Organizacija softverskog projekta – aspekt učesnika projekta
		16. Organizacija softverskog projekta
		17. Učesnici u projektu i njihove uloge
		18. Samostalan rad učenika na prikupljanju informacija uz mentorstvo profesora
		Pripreme za praktičan rad na izradi složenije aplikacije
		19. Razmatranje ideja i mogućnosti
		20. Izbor platforme i elemenata
		21. Analiza mogućih praktičnih projekata
		22. Formiranje timova od tri člana - uloge
23. Podjela projektnih zadataka učenicima		
24. Presentacija plana realizacije projekta		
Z O V E	12	Specifikacija softverskih zahtjeva

		25.	Specifikacija i validacija softverskih zahtjeva		
		26.	Softverski zahtjevi (<i>software requirements</i>)		
		27.	Utvrđivanje i analiza softverskih zahtjeva		
		28.	Softverski zahtjevi - traženje dodatnih informacija i pojašnjenja		
		29.	Samostalan rad učenika na projektu uz mentorstvo profesora		
		30.	Samostalan rad učenika na projektu uz mentorstvo profesora		
		31.	Samostalan rad učenika na projektu uz mentorstvo profesora		
		32.	Samostalan rad učenika na projektu uz mentorstvo profesora		
		33.	Samostalan rad učenika na projektu uz mentorstvo profesora		
		34.	Samostalan rad učenika na projektu uz mentorstvo profesora		
		35.	Samostalan rad učenika na projektu uz mentorstvo profesora		
		36.	Samostalan rad učenika na projektu uz mentorstvo profesora		
		Upoznavanje sa konceptom softverske arhitekture			
		37.	Koncept softverske arhitekture		
		38.	Uloga softver arhitekta		
		39.	Samostalan rad učenika na projektu uz mentorstvo profesora		
		DECEMBAR	12	Dizajn i arhitektura softvera – Tipovi dizajna	
				40.	Softverski dizajn (Software Design)
				41.	Strategije, notacija i analiza softverskog dizajna
42.	Priroda i karakteristike dizajna				
43.	Softverski arhitekturni stilovi				
44.	Samostalan rad učenika na projektu uz mentorstvo profesora				
45.	Samostalan rad učenika na projektu uz mentorstvo profesora				
46.	Samostalan rad učenika na projektu uz mentorstvo profesora				
47.	Samostalan rad učenika na projektu uz mentorstvo profesora				
48.	Samostalan rad učenika na projektu uz mentorstvo profesora				

MJESEC	BROJ ČASOVA	REALIZACIJA PROGRAMA	
DRUGO POLUGODIŠTE			
FEBRUAR	12	Upravljanje softverskom konfiguracijom	
		49.	Upravljanje softverskom konfiguracijom (SCM - <i>software configuration management</i>)
		50.	Planiranje i upravljanje SCM procesom
		51.	Upoznavanje sa softverom za verzioniranje, kao i platformom za timski rad (npr. GitHub)
		52.	Samostalan rad učenika na projektu uz mentorstvo profesora
		53.	Samostalan rad učenika na projektu uz mentorstvo profesora

		54.	Samostalan rad učenika na projektu uz mentorstvo profesora
		Razvoj i implementacija: Postojeća razvojna okruženja, uvod u korištenje razvojnih SW okvira (frameworks), standardi kodiranja, implementacija, testiranje	
		55.	Modeli procesa razvoja softvera
		56.	Savremena metodologija razvoja softvera
		57.	SW okviri i najvažnije osobine
		58.	Upoznavanje sa razvojnim okruženjem
		59.	Automatizacija procesa pretvaranja izvornog koda u izvršni kod (pojmovi kontinuirana integracija i kontinuirana isporuka)
MART	12	60.	Samostalan rad učenika na projektu uz mentorstvo profesora
		61.	Implementacija
		62.	Samostalan rad učenika na projektu uz mentorstvo profesora
		63.	Samostalan rad učenika na projektu uz mentorstvo profesora
		64.	Samostalan rad učenika na projektu uz mentorstvo profesora
		65.	Samostalan rad učenika na projektu uz mentorstvo profesora
		66.	Samostalan rad učenika na projektu uz mentorstvo profesora
		67.	Testiranje, automatizovani testovi
		68.	Samostalan rad učenika na projektu uz mentorstvo profesora
		69.	Samostalan rad učenika na projektu uz mentorstvo profesora
		70.	Samostalan rad učenika na projektu uz mentorstvo profesora
		71.	Samostalan rad učenika na projektu uz mentorstvo profesora
APRIL	12	72.	Samostalan rad učenika na projektu uz mentorstvo profesora
		73.	Podrška poslije implementacije
		74.	Samostalan rad učenika na projektu uz mentorstvo profesora
		75.	Samostalan rad učenika na projektu uz mentorstvo profesora
		Prezentacija napravljene aplikacije	
		76.	Prezentacija i ocjenjivanje razvijenog softvera
		77.	Prezentacija i ocjenjivanje razvijenog softvera
		78.	Prezentacija i ocjenjivanje razvijenog softvera
		79.	Prezentacija i ocjenjivanje razvijenog softvera
		80.	Prezentacija i ocjenjivanje razvijenog softvera
		81.	Prezentacija i ocjenjivanje razvijenog softvera
		82.	Prezentacija i ocjenjivanje razvijenog softvera
83.	Prezentacija i ocjenjivanje razvijenog softvera		
84.	Prezentacija i ocjenjivanje razvijenog softvera		

MAJ	6	Novosti u Softver inženjeru	
		85.	Pouzdana izvori informacija iz oblasti softver inženjeringa
		86.	Vježba – istraživanje novosti iz oblasti
		87.	Vježba – istraživanje novosti iz oblasti
		Provjera usvojenosti nastavnih sadržaja, vrednovanje rada učenika i zaključivanje ocjena	
		88.	Ponavljjanje gradiva, provjera usvojenosti nastavnih sadržaja, vrednovanje rada učenika i zaključivanje ocjena
		89.	Ponavljjanje gradiva, provjera usvojenosti nastavnih sadržaja, vrednovanje rada učenika i zaključivanje ocjena
		90.	Ponavljjanje gradiva, provjera usvojenosti nastavnih sadržaja, vrednovanje rada učenika i zaključivanje ocjena

Nastavni plan i program

Škola: GIMNAZIJA

Izorno područje/zanimanje: INFORMACIONE TEHNOLOGIJE

Nastavni predmet: Softver inženjering

Razred: IV (četvrti)

Broj sati sedmično: 3

Broj sati za školsku godinu: 90

CILJ I ZADACI:

Cilj: Upoznati se cjelokupnim procesom razvoja softvera. Naučiti da je programiranje samo dio postupka razvoja softvera koji rješava stvarne probleme. Razumjeti ostale aktivnosti tokom izrade aplikacija, kao što su prepoznavanje potreba, definisanje zahtjeva, modeliranje, upotrebu razvojnih alata i okruženja, održavanje verzija.

Zadaci

Omogućiti učeniku da:

- Razumije vezu između softver inženjeringa i drugih informatičkih predmeta koji su izučavani ranije
- Zna pojmove iz oblasti softver inženjeringa (na BHS i engleskom jeziku)
- Razumije se sa karakteristikama životnog ciklusa softvera i osnovnim osobinama industrijski najrasprostranjenijih izabranih modela životnog ciklusa softvera
- Zna postavljati (prava) pitanja u analizi problema (traženje dodatnih informacija i pojašnjenja)
- Zna koristiti vještinu jasnog i konciznog izražavanja misli u sklopu softver inženjeringa. Ima sposobnost rada u grupi. Prepoznaje suštinu rečenog/napisanog.
- Razumije koncept softverske arhitekture
- Zna elemente organizacije softverskog projekta i ima sposobnost da kao član tima učestvuje na projektu
- Zna najčešće korištene uzorke ili primjere dizajna softvera (*Software Design Patterns*)
- Razumije prednosti i značenje upravljanjem softverskom konfiguracijom i motivacijom za postojanje
- Zna koristiti softver za verzioniranje, kao i platformu za timski rad (npr. GitHub)
- Analizira i snalazi se u programima koji se sastoje od više izvornih datoteka, ima sposobnost organizovanja koda i znanje pripreme izvornog koda za produkciju (kompiliranje).
- Posjeduje znanja za automatizaciju procesa generisanja izvršnog koda od izvornog (*build*)
- Zna jedno razvojno okruženje i postojanje alternativa,
- Razumije pojam SW okvir (framework) i zna navesti primjere i poznaje neke najvažnije osobine/ograničenja istih.
- Razumije značaj pisanja automatiziranih testova (*unit tests*).
- Zna prezentirati dio softverskog projekta na kom radi
- Zna prati novosti iz oblasti softver inženjeringa., čuo je za nove ideje u oblasti.
- Zna prepoznati pouzdane izvore informacija iz oblasti softver inženjeringa

NASTAVNI PROGRAM

Nastavne oblasti / Nastavne cjeline	Sati	Nastavni sadržaj / Nastavne jedinice	Po završetku nastavne cjeline učenik će imati sljedeća znanja, vještine i kompetencije:
Ponavljjanje gradiva iz stručnih predmeta iz 3. razreda (Programiranje, Baze podataka, Matematske osnove računarskih nauka, Napredna primijenjena informatika)	3	<ul style="list-style-type: none"> • Upoznavanje, uvod u predmet, literatura i oprema • Veza sa gradivom Programiranja • Veza sa gradivom Baza podataka • Veza sa gradivom Matematskih osnova računarskih nauka • Veza sa gradivom Napredne primijenjene informatike 	<ul style="list-style-type: none"> ✓ Razumije vezu između softver inženjeringa i drugih informatičkih predmeta koji su izučavani ranije
Pojam i uloga softver inženjeringa	3	<ul style="list-style-type: none"> • Uvod u softver inženjering • Uloga softver inženjeringa • Značaj pouzdanosti softvera 	<ul style="list-style-type: none"> ✓ Zna pojmove iz oblasti softver inženjeringa (na BHS i engleskom jeziku)
Životni ciklus razvoja softvera i industrijski najrasprostranjeniji modeli životnog ciklusa softvera, evolucija softverskog proizvoda	9	<ul style="list-style-type: none"> • Životni ciklus razvoja softvera • Osnovne osobine industrijski najrasprostranjenijih modela životnog ciklusa razvoja softvera • Klasični model: <i>waterfall</i> • Klasični model: spiralni • Evolucija softvera • Manifest agilnih metoda razvoja softvera • Jedna agilna metoda (preporučeno SCRUM) • Alati za saradnju 	<ul style="list-style-type: none"> ✓ Razumije karakteristike životnog ciklusa softvera i osnovnim osobinama industrijski najrasprostranjenijih izabranih modela životnog ciklusa softvera

Organizacija softverskog projekta – aspekt učesnika projekta	3	<ul style="list-style-type: none"> • Organizacija softverskog projekta • Učesnici u projektu i njihove uloge 	✓ Zna elemente organizacije softverskog projekta i ima sposobnost da kao član tima učestvuje na projektu)
Pripreme za praktičan rad na izradi srednje složene aplikacije, u timovima od po tri člana ili individualno	6	<ul style="list-style-type: none"> • Razmatranje ideja i mogućnosti • Izbor platforme i elemenata • Analiza mogući praktičnih projekata • Formiranje timova - uloge • Podjela projektnih zadataka učenicima • Presentacija plana realizacije projekta 	✓ Zna koristi vještinu jasnog i konciznog izražavanja misli u sklopu softver inženjeringa. Ima sposobnost rada u grupi. Prepoznaje suštinu rečenog/napisanog
Specifikacija softverskih zahtjeva	9	<ul style="list-style-type: none"> • Softverski zahtjevi (<i>Software Requirements</i>) • Specifikacija i validacija softverskih zahtjeva • Utvrđivanje i analiza softverskih zahtjeva • Samostalan rad učenika na projektu uz mentorstvo profesora 	✓ Sposoban je primjeniti (prava) pitanja u analizi specifikacije (traženje dodatnih informacija i pojašnjenja).
Upoznavanje sa konceptom softverske arhitekture.	3	<ul style="list-style-type: none"> • Koncept softverske arhitekture • Uloga softver arhitekta na projektu • Samostalan rad učenika na projektu uz mentorstvo profesora 	✓ Razumije koncept softverske arhitekture
Dizajn i arhitektura softvera – Uzorci dizajna	9	<ul style="list-style-type: none"> • Softverski dizajn (<i>Software Design</i>) • Strategije, notacija i analiza softverskog dizajna • Priroda i karakteristike dizajna • Softverski arhitekturni stilovi • Samostalan rad učenika na projektu uz mentorstvo profesora 	✓ Zna najčešće korištene uzorke ili primjere dizajna softvera (<i>Software Design Patterns</i>)
Upravljanje softverskom konfiguracijom	6	<ul style="list-style-type: none"> • Upravljanje softverskom konfiguracijom (SCM - <i>software configuration management</i>) 	✓ Razumije prednosti i značenje upravljanjem softverskom konfiguracijom i motivacijom za postojanje

		<ul style="list-style-type: none"> • Planiranje i upravljanje SCM procesom • Upoznavanje sa softverom za verzioniranje, kao i platformom za timski rad (npr. GitHub) • Samostalan rad učenika na projektu uz mentorstvo profesora 	<ul style="list-style-type: none"> ✓ Zna koristiti softver za verzioniranje, kao i platformu za timski rad (npr. GitHub)
Razvoj i implementacija: Postojeća razvojna okruženja, uvod u korištenje razvojnih SW okvira (<i>frameworks</i>), standardi kodiranja, implementacija, testiranje	24	<ul style="list-style-type: none"> • Modeli procesa razvoja softvera • Savremena metodologija razvoja softvera • SW okviri i najvažnije osobine • Upoznavanje sa razvojnim okruženjem • Automatizacija procesa pretvaranja izvornog koda u izvršni kod (pojmovi kontinuirana integracija i kontinuirana isporuka) • Implementacija • Testiranje, automatizovani testovi • Podrška poslije implementacije • Samostalan rad učenika na projektu uz mentorstvo profesora 	<ul style="list-style-type: none"> ✓ Analizira i snalazi se u programima koji se sastoje od više izvornih datoteka, ima sposobnost organizovanja koda i znanje pripreme izvornog koda za produkciju (kompiliranje). ✓ Posjeduje znanja za automatizaciju procesa generisanja izvršnog koda od izvornog (<i>build</i>) ✓ Zna jedno razvojno okruženje i postojanje alternativa, ✓ Razumije pojmom SW okvir (<i>framework</i>) i zna navesti primjere i poznaje neke najvažnije osobine/ograničenja istih. ✓ Razumije značaj pisanja automatiziranih testova (<i>unit tests</i>)
Prezentacija napravljene aplikacije i diskusije o otvorenim pitanjima tokom razvoja softvera	9	<ul style="list-style-type: none"> • Prezentacija i ocjenjivanje razvijenog softvera 	<ul style="list-style-type: none"> ✓ Zna prezentirati dio softverskog projekta na kom radi
Novosti u Softver inženjeringu	3	<ul style="list-style-type: none"> • Pouzdani izvori informacija iz oblasti softver inženjeringa • Vježba – istraživanje novosti iz oblasti 	<ul style="list-style-type: none"> ✓ Zna prati novosti iz oblasti softver inženjeringa., čuo je za nove ideje u oblasti. ✓ Zna prepoznati pouzdane izvore informacija iz oblasti softver inženjeringa

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 teoretska predavanja laboratorijske vježbe/praktični rad na računaru primjenom kombiniranih oblika rada.

Mjesto realizacije nastave: Nastava na predmetu Softver inženjering se izvodi u kabinetima informatike.

Podjela odjeljenja u grupe: Prilikom realizacije nastavnog procesa na predmetu Softver inženjering 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

PRAĆENJE, VREDNOVANJE I OCJENJIVANJE

Nastavnik treba evaluirati učenička postignuća na različite načine. Pri tome trebe 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;

¹ NPP Informatika za gimnaziju, MONKS avgust 2016. godine

- 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 - programiranja koje su provedene za druge predmete, razvoj programa za potrebe stručnih službi škole itd. pri čemu učenik ne treba da zanemaruje ostale obaveze u nastavi softver inženjeringa;
- 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 NASTAVNIKA

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:

Općeobrazovnu, 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.

3. Fakulet 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
- Informacionih 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. ²

² NPP Informatika za gimnaziju, MONKS avgust 2016. godine