

Nastavni plan i program za  
predmet:  
Programiranje

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

**KANTON SARAJEVO**  
**Ministarstvo za obrazovanje, nauku i mlade**  
Juni, 2019.

## Programiranje

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

- poznavanje osnovnih pojmova i principa objektno-orjentisanog programiranja
- razumijevanje razlike između objektno-orjentisanog i objektno-zasnovanog programiranja
- poznavanje pojmova enkapsulacije, polimorfizma i kompozicije sa primjenom na praktičnim programima
- koristi osnovne grafičke primitive (tačka, linija, pravougaonik)
- kreira aplikacije sa grafičkim vizuelnim interfejsom (GUI), poznaje osnovne elemente GUIja (prozor, dugme, meni, tekstualno polje...)
- može da grupiše metode, srodne datoteka u direktorije.
- zna da organizuje kod u logičke cjeline (klase, foldere).
- konzistentno daje imena za datoteka, konstanti, varijabli, klasa, metode i sl.
- radi provjeru argumenata ili kritičnih stanja i ugrađuje programski kod koje ispituje pretpostavke za uspješno izvršenje koda. (defensive coding)
- prepoznaje situacija greške u izvršenju programa i kodira oporavak od greške.

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

<b>REALIZACIJA PROGRAMA</b>	<b>ČAS</b>
Upoznavanje, uvod u predmet, literatura i pribor	1
Objektno-orjentisano i objektno-zasnovano programiranja	8
Kompozicija, nasljeđivanje, enkapsulacija i polimorfizam	16
Osnovi grafičke primitive	2
Grafički korisnički interfejs (GUI)	6
Organizacija koda	8
Konzistentno dodjeljivanje imena	4
Pretpostvke uspješnosti izvršenja koda	6
Oporavak od greške	8
Samostalan rad učenika na projektu uz mentorstvo profesora	10
Ponavljjanje gradiva, provjera usvojenosti nastavnih sadržaja, vrednovanje rada učenika i zaključivanje ocjena	1
<b>UKUPNO:</b>	<b>70</b>

**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.

MJESEC	BROJ ČASOVA	REALIZACIJA PROGRAMA
<b>PRVO POLUGODIŠTE</b>		
SEPTEMBAR	8	1. Upoznavanje, uvod u predmet, literatura i pribor
		<b>Objektno-orjentisano i objektno-zasnovano programiranje</b>
		2. Osnovni pojmovi i principi objektno-orjentisanog programiranja
		3. Osnovni pojmovi i principi objektno-orjentisanog programiranja
		4. Strukture i klase (primjeri u odabranom programskom jeziku)
		5. Strukture i klase (primjeri u odabranom programskom jeziku)
		6. Razlika između objektno-orjentisanog i objektno-zasnovanog programiranja
		7. Vježba (osnovni pojmovi)
OKTOBAR	8	8. Vježba (npr. odrediti ime atributa i tipova na osnovu nekih zahtjeva (auto: makra auta, boja, ...))
		9. Ponavljanje gradiva i praktična provjera usvojenosti nastavnih sadržaja
		<b>Kompozicija, nasljeđivanje, enkapsulacija i polimorfizam</b>
		10. Kompozicija (unošenje jednog objekta u drugi objekat npr. boja kao objekat postaje klasa (koliko koje boje ima na autu))
		11. Vježba
		12. Nasljeđivanje i kompozicija
		13. Nasljeđivanje (inheritance)
		14. Nasljeđivanje
NOVEMBAR	9	15. Vježba
		16. Enkapsulacija
		17. Nasljeđivanje i enkapsulacija
		18. Vježba
		19. Polimorfizam (primjeri polimorfizma)
		20. Polimorfizam (primjeri polimorfizma)
		21. Vježba
		22. Polimorfizam (primjeri implementacije, Vtable)
23. Vježba		
24. Vježba		

		25.	Ponavljjanje gradiva i praktična provjera usvojenosti nastavnih sadržaja
DECEMBAR	8	<b>Osnovi grafiče primitive</b>	
		26.	Uvod u grafičke primitive (tačka, linija, pravougaonik)
		27.	Vježba
		<b>Grafički korisnički interfejs (GUI)</b>	
		28.	Osnovni pojmovi grafičkog korisničkog interfejsa, podešavanje okruženja, elementi grafičkog korisničkog interfejsa: prozor, dugme, tekstualno polje...
		29.	Kreiranje grafičkog korisničkog interfejsa koristeći vizuelni dizajner, osnove dobrog dizajna
		30.	Vježba
		31.	Povezivanje grafičkog korisničkog interfejsa sa programskim kodom: događaji, osnovna manipulacija grafičkim elementima iz koda, struktura automatski generisanog koda
		32.	Vježba
		33.	Ponavljjanje gradiva, provjera usvojenosti nastavnih sadržaja, vrednovanje rada učenika i zaključivanje ocjena

DRUGO POLUGODIŠTE			
FEBRUAR	8	<b>Organizacija koda</b>	
		34.	Organizacija koda u module/pakete/foldere
		35.	Organizacija kod u module/pakete/foldere
		36.	Primjeri open source projekta
		37.	Vježba (npr. na papiru organizovati kod (koji je učenicima priložen)
		38.	Organizacija koda u klase
		39.	Primjeri open source projekta
		40.	Vježba (idejni projekat na papiru kako bi učenici organizovali kod u logičke cjeline-klase)
		41.	Vježba
		MART	9
42.	Konzistentno dodjeljivanje imena za datoteke, konstante, varijable, klase, metode i sl.		
43.	Vježba		
44.	Ponavljjanje gradiva		
45.	Ponavljjanje gradiva i praktična provjera usvojenosti nastavnih sadržaja		
<b>Pretpostvke usješnosti izvršenja koda</b>			
46.	Provjera argumenata i kritičkih stanja		
47.	Provjera argumenata i kritičkih stanja		
48.	Vježba (npr. da li je nazivnik prije djeljenja jednak nuli)		

		49.	Vježba (da li je null pointer)	
		50.	Vježba (kako prepoznati situaciju beskonačne petlje i riješiti problem, petlja (rekurzija))	
APRIL	9	51.	Vježba (granični slučajevi npr. da li je lista prazna, da li niz ima dovoljno prostora)	
		<b>Oporavak od greške</b>		
		52.	Idea iza try catch blokova	
		53.	Rad sa try catch blokovima u program	
		54.	Vježba (oporavak od nepravilnog zauzimanja (alokacija) memorije)	
		55.	Vježba (greška pri nepravilnom korištenju fajlova)	
		56.	Vježba (greška pri radu sa bazom podataka)	
		57.	Vježba (greška pri type casting/promjene tipa)	
		58.	Ponavljjanje gradiva	
		59.	Ponavljjanje gradiva i praktična provjera usvojenosti nastavnih sadržaja	
MAJ	8	<b>Samostalan rad učenika na projektu uz mentorstvo profesora</b>		
		60.	Samostalan rad učenika na projektu uz mentorstvo profesora	
		61.	Samostalan rad učenika na projektu uz mentorstvo profesora	
		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.	Prezentacija i ocjenjivanje projektnih zadataka			
JUNI	3	68.	Prezentacija i ocjenjivanje projektnih zadataka	
		69.	Prezentacija i ocjenjivanje projektnih zadataka	
		70.	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: PROGRAMIRANJE 3**

**Razred: III (treći)**

**Broj sati sedmično: 2**

**Broj sati za školsku godinu: 70**

## **CILJ 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:

- shvati i razumije osnovne pojmove i principe objektno orijentisanog programiranja,
- shvati i razumije šta su strukture i klase,
- bude osposobljen da razlikuje između objektno-orijentisanog i objektno-zasnovanog programiranja,
- bude osposobljen za implementaciju programa gdje će određivati ime atributa i tipova na osnovi nekih zahtjeva (auto: marka auta, boja, itd),
- shvati, razumije i bude osposobljen za primjenu kompozicija pri izradi programa (kao što je npr. unošenje jednog objekta u drugi objekat (boja kao objekat postaje klasa, koliko koje boje ima na autu))
- shvati, razumije i bude osposobljen za implementaciju nasljeđivanja pri izradi programa,
- shvati, razumije i bude osposobljen za implementaciju enkapsulacije pri izradi programa,
- shvati, razumije i bude osposobljen za implementaciju polimorfizma pri izradi programa,
- bude osposobljen za implementaciju Vtable-a u polimorfizmu,
- shvata i razumije šta su to grafičke primitive,
- bude osposobljen za implementaciju programa koji prikazuje npr. tačku/tačke, linije i pravougaonik,
- nauči i razumije šta su to osnovni pojmovi grafičkog korisničkog interfejsa,
- bude osposobljen da podešava okruženje grafičkog korisničkog interfejsa,
- nauči i razumije elemente grafičkog korisničkog interfejsa kao što su prozor, dugme, tekstualno polje itd.,
- bude osposobljen za implementaciju grafičkog korisničkog interfejsa koristeći vizuelni dizajner,
- bude osposobljen za povezivanje grafičkog korisničkog interfejsa sa programskim kodom,

- shvati i razumije šta je to organizacija koda u module/pakete/foldere,
- bude osposobljen za organizaciju koda (kod je na papiru priložen učenicima) u module/pakete/foldera,
- shvati i razumije šta je to organizacija koda u klase,
- bude osposobljen za organizaciju koda (idejni projekat priložen je učenicima na papiru) u logičke cjeline-klase,
- bude osposobljen za implementaciju konzistentnog dodjeljivanja imena za datoteke, konstante, varijable, klase, metode i sl.,
- shvata, razumije i bude osposobljen za implementaciju pojma Provjera argumenata i kritičkog stanja kroz vježbe kao što su npr.: da li je nazivnik prije djeljenja jednak nuli, da li je null pointer, kako prepoznati situaciju beskonačne petlje i rješavanje problema, petlje (rekurzija), granični slučajevi npr. da li je lista prazna, da li niz ima dovoljno prostora,
- nauči i razumije pojam Idea iza try catch blokova,
- bude osposobljen za implementaciju pojma try catch blokova kroz vježbe kao što su: oporavak od nepravilnog zauzimanja (alokacija) memorije, greška pri nepravilnom korištenju fajlova, greška pri radu sa bazom podataka, greška pri type casting/promjene tipa,
- bude osposobljen za implementaciju samoizabranog programa/projekata koji mora biti usko vezan za usvojeno gradivo tokom tekuće školske godine,
- bude osposobljen da objašnjava i predstavi rad svog samoodabranog projekat 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:
Objektno- orjentisano i objektno- zasnovano programiranje	9	<ul style="list-style-type: none"> <li>- Upoznavanje, uvod u predmet, literature i pribor</li> <li>- Osnovni pojmovi i pricipi objektno-orjentisanog programiranja</li> <li>- Strukture i klase (primjeri u odabranom programskom jeziku)</li> <li>- Razlika između objektno-orjentisanog i objektno-zasnovanog programiranja</li> </ul>	<ul style="list-style-type: none"> <li>✓ Zna i razumije osnovne pojmove i principe objektno orjentisanog programiranja.</li> <li>✓ Zna i razumije šta su strukture i klase.</li> <li>✓ Razumije razliku između objektno-orjentisanog i objektno-zasnovanog programiranja.</li> <li>✓ Sposobnost pravljenja programa kao što je npr. program gdje učenici određuju ime atributa i tipova na osnovi nekih zahtjeva (auto: marka auta, boja, itd)</li> </ul>
Kompozicija, nasljeđivanje, enkapsulacija i polimorfizam	16	<ul style="list-style-type: none"> <li>- Kompozicija (unošenje jednog objekta u drugi objekat npr. boja kao objekat postaje klasa (koliko koje boje ima na autu))</li> <li>- Nasljeđivanje i kompozicija</li> <li>- Nasljeđivanje (inheritance)</li> <li>- Enkapsulacija</li> <li>- Nasljeđivanje i enkapsulacija</li> <li>- Polimorfizam (primjeri polimorfizma)</li> <li>- Polimorfizam (primjeri implementacije, Vtable)</li> </ul>	<ul style="list-style-type: none"> <li>✓ Razumije pojam Kompozicija u OOP-u.</li> <li>✓ Sposobnost primjene kompozicija pri izradi programa (kao što je npr. unošenje jednog objekta u drugi objekat (boja kao objekat postaje klasa, koliko koje boje ima na autu))</li> <li>✓ Zna i razumije pojam Nasljeđivanje.</li> <li>✓ Sposobnost primjene nasljeđivanja pri izradi programa.</li> <li>✓ Razumije pojam Enkapsulacije.</li> <li>✓ Sposobnost primjene enkapsulacije pri izradi programa.</li> <li>✓ Zna šta je to Polimorfizam.</li> <li>✓ Razumije implementaciju polimorfizma.</li> <li>✓ Sposobnost primjene polimorfizma.</li> <li>✓ Sposobnost primjene Vtable-a u polimorfizmu.</li> </ul>



Osnovi grafičke primitive	2	<ul style="list-style-type: none"> <li>- Uvod u grafičke primitive (tačka, linija, pravougaonik)</li> </ul>	<ul style="list-style-type: none"> <li>✓ Zna i razumije šta su to grafičke primitive.</li> <li>✓ Sposobnost izrade programa koji prikazuje npr. tačku/tačke, linije i pravougaonik.</li> </ul>
Grafički korisnički interfejs (GUI)	6	<ul style="list-style-type: none"> <li>- Osnovni pojmovi grafičkog korisničkog interfejsa, podešavanje okruženja, elementi grafičkog korisničkog interfejsa: prozor, dugme, tekstualno polje...</li> <li>- Kreiranje grafičkog korisničkog interfejsa koristeći vizuelni dizajner, osnove dobrog dizajna</li> <li>- Povezivanje grafičkog korisničkog interfejsa sa programskim kodom: događaji, osnovna manipulacija grafičkim elementima iz koda, struktura automatski generisanog koda</li> </ul>	<ul style="list-style-type: none"> <li>✓ Zna i razumije šta su to osnovni pojmovi grafičkog korisničkog interfejsa.</li> <li>✓ Sposobnost podešavanja okruženja grafičko korisničkog interfejsa.</li> <li>✓ Zna i razumije elemente grafičkog korisničkog interfejsa kao što su prozor, dugme, tekstualno polje itd.</li> <li>✓ Sposobnost kreiranja grafičkog korisničkog interfejsa koristeći vizuelni dizajner.</li> <li>✓ Sposobnost povezivanja grafičkog korisničkog interfejsa sa programskim kodom.</li> </ul>
Organizacija koda	8	<ul style="list-style-type: none"> <li>- Organizacija kod u module/pakete/foldere</li> <li>- Primjeri open source projekta</li> <li>- Organizacija koda u klase</li> <li>- Primjeri open source projekta</li> </ul>	<ul style="list-style-type: none"> <li>✓ Zna i razumije šta je to organizacija koda u module/pakete/foldere.</li> <li>✓ Sposobnost organizacije koda (kod je na papiru priložen učenicima) u module/pakete/foldera.</li> <li>✓ Zna i razumije šta je to organizacija koda u klase.</li> <li>✓ Sposobnost organizacije koda (idejni projekat priložen je učenicima na papiru) u logičke cjeline-klase.</li> </ul>
Konzistentno dodjeljivanje imena	4	<ul style="list-style-type: none"> <li>- Konzistentno dodjeljivanje imena za datoteke, konstante, varijable, klase, metode i sl.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Sposobnost konzistentnog dodjeljivanja imena za datoteke, konstante, varijable, klase, metode i sl.</li> </ul>
Pretpostavke uspješnosti izvršenja koda	6	<ul style="list-style-type: none"> <li>- Provjera argumenata i kritičkih stanja</li> </ul>	<ul style="list-style-type: none"> <li>✓ Razumije pojam Provjera argumenata i kritičkih stanja.</li> <li>✓ Sposobnost implementacije pojma Provjera argumenata i</li> </ul>

			<p>kritičkog stanja kroz vježbe kao što su npr.: da li je nazivnik prije djeljenja jednak nuli, da li je null pointer, kako prepoznati situaciju beskonačne petlje i rješavanje problema, petlje (rekurzija), granični slučajevi npr. da li je lista prazna, da li niz ima dovoljno prostora.</p>
Oporavak od greške	8	<ul style="list-style-type: none"> <li>- Idea iza try catch blokova</li> <li>- Rad sa try catch blokovima u program</li> </ul>	<ul style="list-style-type: none"> <li>✓ Razumije pojam Idea iza try catch blokova.</li> <li>✓ Sposobnost implementacije pojma try catch blokova kroz vježbe kao što su: oporavak od nepravilnog zauzimanja (alokacija) memorije, greška pri nepravilnom korištenju fajlova, greška pri radu sa bazom podataka, greška pri type casting/promjene tipa.</li> </ul>
Samostalan rad učenika na projektu uz mentorstvo profesora	10	<ul style="list-style-type: none"> <li>- Samostalan rad učenika na projektu uz mentorstvo profesora</li> <li>- Prezentacija i ocjenjivanje projektnih zadataka</li> </ul>	<ul style="list-style-type: none"> <li>✓ Sposobnost izrade samoizabranog programa/projekata koji mora biti usko vezan za usvojeno gradivo tokom tekuće školske godine.</li> <li>✓ Objašnjava i predstavlja rad svog samoodabranog projekat pred razredom.</li> </ul>

## NAČINI OSTVARIVANJA PROGRAMA<sup>1</sup>

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 Programiranje 3 se realiziraje u kabinetima informatike.

**Podjela odjeljenja u grupe:** Prilikom realizacije nastavnog procesa na predmetu Programiranje 3 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;
- 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, 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;

---

<sup>1</sup> NPP Informatika za gimnaziju, MONKS avgust 2016. godine

- 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. 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
- Kompjuterskih/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. <sup>2</sup>

### **PREPORUČENI IZVOR INFORMACIJA ZA UČENJE**

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

---

<sup>2</sup> NPP Informatika za gimnaziju, MONKS avgust 2016. godine