Nastavni plan i program za predmet: Programiranje 4

Izborno područje: Informacione tehnologije
4. razred

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

Programiranje

ISHODI UČENJA: po odslušanom i položenom predmetu učenik će imati slijedeća znanja, kompetencije i vještine:

- Detaljno poznaje napredne koncepte jednog popularnog statičkog objektno-orjentisanog programskog jezika (C++, C#, Java).
- Poznaje način na koji je organizirana memorija računara, te može ručno (kroz programski kod) da upravlja memorijom.
- Napredno poznavanje razumijevanje *complier*-a, *linker*-a i interpretera, razumije šta je asemblerski kod i kako se programski kod izvršava na hardverskom nivou.
- Poznaje pojmove generičkog programiranja, metaprogramiranja i funkcionalnog programiranja, koristi u programima osnovne koncepte iz funkcionalnog programiranja (lambda-funkcije).
- Prati savremene trendove odabranog programskog jezika i nove verzije specifikacije/standarda za odabrani programski jezik (C++17, Java 8...), razumije šta je novo u objavljenoj specifikaciji i kako to iskoristiti u praktičnim programerskim zadacim.

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

REALIZACIJA PROGRAMA	**ČAS**
Upoznavanje, uvod u predmet, literatura i pribor	1
Osvrt na napredne koncepte jednog popularnog statičkog OOP jezika (C++, C#, Java)	7
Organizacija memorija u računaru	11
Compiler, linker, interpreter i asemblersi kod	2
Generičko programiranja, metaprogramiranje i funkcionalno programiranje	12
Generičko programiranja, metaprogramiranje i funkcionalno programiranje	7
Savremeni trendovi odabranog programskog jezika	5
Samostalan rad učenika na projektu uz mentorstvo profesora	14
Ponavljanje gradiva, provjera usvojenosti nastavnih sadržaja, vrednovanje rada učenika i zaključivanje ocjena	1
UKUPNO:	**60**

NAPOMENA: Nastavni plan i program u četvrtom 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 jezici

MJESEC	BROJ ČASOVA		REALIZACIJA PROGRAMA
PRVO POLUGODIŠTE			
SEPTEMBAR	8	1.	Upoznavanje, uvod u predmet, literatura i pribor
		Osvrt na napredne koncepte jednog popularnog statičkog OOP jezika (C++, C#, Java)	
		2.	Objektno-orjentisano i objektno-zasnovano programiranja
		3.	Vježba
		4.	Kompozicija, nasljeđivanje, enkapsulacija i polimorfizam
		5.	Vježba
		6.	Oporavak od greške
		7.	Vježba
		8.	Ponavljanje gradiva i praktična provjera usvojenosti nastavnih sadržaja
OKTOBAR	8	**Organizacija memorija u računaru**	
		9.	Stack memorija (lokalne varijable i parametri funkcija)
		10.	Stack memorija (lokalne varijable i parametri funkcija)
		11.	Vježbe
		12.	Vježbe
		13.	Dinamička alokacija varijable i Heap memorija
		14.	Dinamička alokacija varijable i Heap memorija
		15.	Dinamička alokacija varijable i Heap memorija
		16.	Vježbe
NOVEMBAR	9	17.	Vježbe
		18.	Vježbe
		19.	Ponavljanje gradiva i praktična provjera usvojenosti nastavnih sadržaja
		Compiler, linker, interpreter i asemblersi kod	
		20.	Razumjevanje compiler-a, linker-a i interpretera
		21.	Razumjevajne asemblerskog koda i kako se programski kod izvršava na hardverskom nivou
		Generičko programiranja, metaprogramiranje i funkcionalno programiranje	
		22.	Generičko programiranje (uvod i primjeri)
		23.	Generičko programiranje (generičke funkcije)
DECEM	8	24.	Generičko programiranje (generičke klase)

		25.	Generičko programiranje (generičke klase)
		26.	Vježbe
		27.	Vježbe
		28.	Metaprogramiranje (C# i Java refleksija/ C++ generičke klase)
		29.	Metaprogramiranje (C# i Java refleksija/ C++ generičke klase)
		30.	Vježbe
		31.	Vježbe
		32.	Ponavljanje gradiva
		33.	Provjera usvojenosti nastavnih sadržaja, vrednovanje rada učenika i zaključivanje ocjena

DRUGO POLUGODIŠTE			
FEBRUAR	8	**Generičko programiranja, metaprogramiranje i funkcionalno programiranje**	
		34.	Funkcionalno programiranje (uvod)
		35.	Funkcionalno programiranje (funkcije višeg nivoa)
		36.	Funkcionalno programiranje (funkcije višeg nivoa)
		37.	Funkcionalno programiranje (lambda funkcije)
		38.	Vježbe
		39.	Vježbe
		40.	Ponavljanje gradiva i praktična provjera usvojenosti nastavnih sadržaja
		Savremeni trendovi odabranog programskog jezika	
		41.	Nove verzije specifikacije/standarda za odabrani programski jezik
MART	9	42.	Nove verzije specifikacije/standarda za odabrani programski jezik
		43.	Vježbe (kako iskoristiti u praktičnim programskim zadacima)
		44.	Vježbe (kako iskoristiti u praktičnim programskim zadacima)
		45.	Ponavljanje gradiva i praktična provjera usvojenosti nastavnih sadržaja
		Samostalan rad učenika na projketu 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
		49.	Samostalan rad učenika na projektu uz mentorstvo profesora
		50.	Samostalan rad učenika na projektu uz mentorstvo profesora

APRIL	9	51.	Samostalan rad učenika na projektu uz mentorstvo profesora
		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
		55.	Samostalan rad učenika na projektu uz mentorstvo profesora
		56.	Prezentacija i ocjenjivanje projektnih zadataka
		57.	Prezentacija i ocjenjivanje projektnih zadataka
		58.	Prezentacija i ocjenjivanje projektnih zadataka
		59.	Prezentacija i ocjenjivanje projektnih zadataka
MAJ	1	60.	Ponavljanje gradiva, provjera usvojenosti nastavnih sadržaja, vrednovanje rada učenika i zaključivanje ocjena

Nastavni plan i program

Škola: GIMNAZIJA

Izborno područje/zanimanje: INFORMACIONE TEHNOLOGIJE

Natavni predmet: PROGRAMIRANJE 4

Razred: IV (četvrti)

Broj časova sedmično: 2

Broj časova za školsku godinu: 60

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:

- bude osposobljen da razliku između objektno-orjentisanog i objektno-zasnovanog programiranja,
- shvati, razumije i bude osposobljen za primjenu kompozicija, nasljeđivanja, enkapsulacije i polimorfizma pri izradi programa,
- 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,
- shvati i razumije šta je to stack memorija,
- shvati, razumije i bude osposobljen za implementaciju lokalne varijable i parametre funkcija,
- shvati, razumije i bude osposobljen za implementaciju dinamičke alokacije varijable i Heap memorije,
- shvati i razumije šta je compiler, linker i interpreter,
- shvati i razumije šta je to asemblerski kod i kako se programski kod izvršava na hardverskom nivou,
- shvati i razumije šta je to generičko programiranje,
- shvati i razuije šta su to generičke funkcije,
- bude osposobljen za primjenu generičkih funkcija pri izradi jednostavnih programa,
- shvati i razumije šta su to generičke klase,

- shvati i razumije šta je to metaprogramiranje (C# i Java refleksija a C++ generičke klase),
- bude osposobljen za primjenu refleksija/generičkih klasa (C#, Java ili C++) pri izradi jednostavnih programa,
- shvati i razumije šta je to funkcionalno programiranje,
- shvati i razumije šta su to funkcije višeg nivoa,
- shvati i razumije šta je to lambda funkcije,
- bude osposobljen za primjenu funkcionalnog programiranja pri izradi jednostavnih programa (npr. lamda funkcija),
- nauči i razumije nove verzije secifikacija/standarda za odabrani programski jezik,
- bude osposobljen za primjenu novih verzija specifikacija/standarda pri izradi programa u odabranom programskom jeziku,
- 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:**
Osvrt na napredne koncepte jednog popularnog statičkog OOP jezika (C++, C#, Java)	8	• Upoznavanje, uvod u predmet, literature i pribor • Objektno-orjentisano i objektno-zasnovano programiranja • Kompozicija, nasljeđivanje, enkapsulacija i polimorfizam • Oporavak od greške	✓ Razumije razliku između objektno-orjentisanog i objektno-zasnovanog programiranja. ✓ Sposobnost primjene kompozicija, nasljeđivanja, enkapsulacije i polimorfizma pri izradi programa. ✓ 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.
Organizacija memorija u računaru	11	• Stack memorija (lokalne varijable i parametri funkcija) • Dinamička alokacija varijable i Heap memorija	✓ Zna šta je to stack memorija. ✓ Razumije implementaciju lokalne varijable i parametre funkcija. ✓ Sposobnost implementacije lokalnih vraijabli i parametara funcija. ✓ Zna šta je dinamička alokacija varijable i Heap memorija. ✓ Razumije implementaciju dinamičke alokacije varijable i Heap memorija. ✓ Sposobnost primjene dinamičke alokacije varijable i Heap memorije.
Compiler, linker, interpreter i asemblersi kod	2	• Razumjevanje compiler-a, linker-a i interpretera • Razumjevajne asemblerskog koda i kako se programski kod izvršava na hardverskom nivou	✓ Zna i razumije šta je compiler, linker i interpreter. ✓ Zna i razumije šta je to asemblerski kod i kako se programski kod izvršava na hardverskom nivou.
Generičko programiranja, metaprogramiranje	12	• Generičko programiranje (uvod i primjeri) • Generičko programiranje (generičke funkcije)	✓ Zna i razumije šta je to generičko programiranje. ✓ Zna i razuije šta su to generičke funkcije.

i funkcionalno programiranje		• Generičko programiranje (generičke klase) • Metaprogramiranje (C# i Java refleksija/ C++ generičke klase)	✓ Sposobnost primjene generičkih funkcija pri izradi jednostavnih programa. ✓ Zna i razumije šta su to generičke klase. ✓ Zna i razumije šta je to metaprogramiranje (C# i Java refleksija a C++ generičke klase). ✓ Sposobnost primjene refleksija/generičkih klasa (C#, Java ili C++) pri izradi jednostavnih programa.
Generičko programiranja, metaprogramiranje i funkcionalno programiranje	7	• Funkcionalno programiranje (uvod) • Funkcionalno programiranje (funkcije višeg nivoa) • Funkcionalno programiranje (lambda funkcije)	✓ Zna i razumije šta je to funkcionalno programiranje. ✓ Zna i razumije šta su to funkcije višeg nivoa. ✓ Zna i razumije šta je to lambda funkcije. ✓ Sposobnost primjene funkcionalnog programiranja pri izradi jednostavnih programa (npr. lamda funkcija).
Savremeni trendovi odabranog programskog jezika	5	• Nove verzije specifikacije/standarda za odabrani programski jezik	✓ Zna i razumije nove verzije secifikacija/standarda za odabrani programski jezik. ✓ Sposobnost primjene novih verzija specifikacija/standarda pri izradi programa u odabranom programskom jeziku.
Samostalan rad učenika na projektu uz mentorstvo profesora	15	• Samostalan rad učenika na projektu uz mentorstvo profesora • Prezentacija i ocjenjivanje projektnih zadataka	✓ Sposobnost izrade samoizabranog programa/projekata koji mora biti usko vezan za usvojeno gradivo tokom tekuće školske godine. ✓ Objašnjava i predstavlja rad svog samoodabranog projekat pred razredom.

NAČINI OSTVARIVANJA PROGRAMA[1]

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 4 se realiziraje u kabinetima informatike.
Podjela odjeljenja u grupe: Prilikom realizacije nastavnog procesa na predmetu Programiranje 4 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

Profesor 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 ili kroz pračan rad;
- 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;
- ocjena treba da uključuje i teoretsko i praktično znanje učenika kao i njegov interes, trud i pomoć drugim učenicima u savladavanju gradiva;

[1] NPP Informatika za gimnaziju, MONKS avgust 2016. godine

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 strušne 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 inžinjerstva
- Magistar matematike, nastavnički smjer
- Magistar matematičkih nauka, smjer teorijska kompjuterska nauka,
- Svršenici Prirodno-matematičkog fakulteta informatičkog i/ili računarskog usmjerenja.

2. Elektrotehnički fakultet:

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

3. Fakulet informatičkog i/ili računarskog usmjerenja sa završenim **<u>četverogodišnjim studijem</u>** 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 inžinjerstva
- Kumpjuterskih/Računarskih nauka
- Informacionih tehnologija
- Informatike i tehničkog odgoja

NAPOMENA: Profil i stručna sprema profesora na predmetu Programiranje 4 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

PREPORUČENI IZVOR INFORMACIJA ZA UČENJE

1. Ne postoji adekvatan udžbenik ili skup udžbenika na našem jeziku koji pokriva gradivo predmeta.

2. Literatura iz vanjskih izvora:

    1. Julijan Šribar i Boris Motik, *Demistificirani C++*, Tisak Nova promocija, Zagreb 2001
    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 20017/18