

Programiranje (1. razred)

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

- razumije pojmove “programiranje”, “programski kod”, “program”
- ima osnovno razumijevanje kompilera, linkera i interpretera
- koristi IDE u programiranju
- posjeduje osnovno znanje jednog popularnog statičkog objektno-orijentisanog programskog jezika (C++, C#, Java)
- poznaje osnovne tipove podataka
- koristi tekstualni ulaz i izlaz koristeći konzolu (terminal)
- poznaje osnovne programske strukture za kontrolu toka: uslovi i grananja, petlje
- poznaje kompozitni tip podatak polje (niz, array)

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

| REALIZACIJA PROGRAMA | ČAS |
|--|------------|
| Upoznavanje, uvod u predmet, literatura i pribor | 1 |
| Uvod u programiranje | 6 |
| Programsko okruženje i jednostavan program | 7 |
| Osnovni pojmovi programiranja i ulaz/izlaz | 13 |
| Programske strukture grananja | 11 |
| Programske strukture ponavljanja | 17 |
| Polja (nizovi, arrays) | 15 |
| UKUPNO: | 70 |

NAPOMENA: Nastavni plan i program u prvom razredu zasniva se na izučavanju jednog od programskih jezika: C++, C#, Java – programiranje osnovni nivo (strukturirano 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 | |
|---|--|---|--|
| PRVO POLUGODIŠTE | | | |
| SEPTEMBAR | 8 | 1. | Upoznavanje, uvod u predmet, literatura i pribor |
| | | Uvod u programiranje | |
| | | 2. | Šta je to program? Šta znači znati programirati? Zašto učimo programirati? Zanimanje "programer" |
| | | 3. | Programski jezici |
| | | 4. | Pojam algoritma, pseudo-kod |
| | | 5. | Dijagram toka |
| | | 6. | Dijagram toka |
| | | 7. | Ponavljjanje gradiva |
| Programsko okruženje i jednostavan program | | | |
| 8. | Upoznavanje sa programskim okruženjem | | |
| OKTOBAR | 8 | 9. | Prvi program: struktura "Hello world" programa, unos i pokretanje, naredba za izlaz teksta na ekran, sekvencijalno izvršenje |
| | | 10. | Šta se dešava kada pokrenemo program? Pojmovi: kompajler, linker, interpreter, debugger |
| | | 11. | Šta se dešava kada imamo sintaksnu grešku u programu i kako je otkloniti? |
| | | 12. | Drugi program: deklaracija, ulaz i izlaz cjelobrojnih promjenljivih, matematička operacija i dodjela, česte greške |
| | | 13. | Vježba |
| | | 14. | Ponavljjanje gradiva i praktična provjera usvojenosti nastavnih sadržaja |
| | | Osnovni pojmovi programiranja i ulaz/izlaz | |
| | | 15. | Tipovi podataka, deklaracija promjenljivih |
| 16. | Pojam operatora i operanda, operator dodjele, najčešći matematički operatori | | |
| NOVEMBAR | 9 | 17. | Konstante, komentari, ulaz i izlaz numeričkih vrijednosti |
| | | 18. | Opšti princip rješavanja programskih zadataka na primjeru rješavanja matematičkih problema |
| | | 19. | Vježba |

| | | | |
|----------|---|--------------------------------------|--|
| | | 20. | Matematičke funkcije |
| | | 21. | Vježba |
| | | 22. | Ponavljjanje gradiva i praktična provjera usvojenosti nastavnih sadržaja |
| | | 23. | Znakovni tipovi (char i string sa vrlo jednostavnim primjerima upotrebe) |
| | | 24. | Vježba |
| | | 25. | Ostali operatori, prioritet operatora, tip rezultata operacije |
| DECEMBAR | 8 | 26. | Vježba |
| | | 27. | Ponavljjanje gradiva i praktična provjera usvojenosti nastavnih sadržaja |
| | | Programske strukture grananja | |
| | | 28. | Tipovi grananja i implementacija u programskom jeziku |
| | | 29. | Operatori poređenja |
| | | 30. | Vježba |
| | | 31. | Logički operatori, osnove matematičke logike |
| | | 32. | Vježba |
| | | 33. | Ponavljjanje gradiva i praktična provjera usvojenosti nastavnih sadržaja |

DRUGO POLUGODIŠTE

| | | | |
|---------|---|---|---|
| FEBRUAR | 8 | Programske strukture grananja (nastavak) | |
| | | 34. | Višestruko grananje naredbom switch-case, naredba goto |
| | | 35. | Vježba |
| | | 36. | Primjeri programskih zadataka sa grananjem, prevođenje realnih problema u programski kod |
| | | 37. | Vježba |
| | | 38. | Ponavljjanje gradiva i praktična provjera usvojenosti nastavnih sadržaja |
| | | Programske strukture ponavljanja | |
| | | 39. | Ponavljjanje naredbom for |
| | | 40. | Vježba |
| | | 41. | Različita dužina koraka, opadajuće petlje, naredbe break i continue , česte greške pri radu sa petljama |
| MART | 9 | 42. | Vježba |
| | | 43. | Neki karakteristični algoritmi sa petljama (stepenovanje, provjera da li je broj prost, rastavljanje broja na cifre) |
| | | 44. | Vježbe |
| | | 45. | Rješavanje problema iz drugih naučnih oblasti programiranjem: računanje sume reda, određivanje nula funkcije metodom polovljenja intervala, kosi hitac... |
| | | 46. | Vježba |
| | | 47. | Ponavljjanje gradiva i praktična provjera usvojenosti nastavnih sadržaja |
| | | 48. | Ponavljjanje naredbama while i do-while |
| | | 49. | Vježba |
| | | 50. | Višestruke (ugniježdene) petlje |
| APRIL | 9 | 51. | Vježba |
| | | 52. | Primjeri programskih zadataka sa višestrukim (ugniježenim) petljama (iscrtavanje oblika na ekranu itd.), česte greške |
| | | 53. | Vježba |
| | | 54. | Ponavljjanje gradiva |
| | | 55. | Praktična provjera usvojenosti nastavnih sadržaja |

| | | | |
|------|---|-------------------------------|---|
| | | Polja (nizovi, arrays) | |
| | | 56. | Nizovi cijelih i realnih brojeva |
| | | 57. | Vježba |
| | | 58. | Neki karakteristični algoritmi nad nizovima (maksimum/minimum, provjera da li članovi niza zadovoljavaju uslov, izbacivanje člana iz niza itd.) |
| | | 59. | Vježba |
| MAJ | 8 | 60. | Ponavljanje gradiva i praktična provjera usvojenosti nastavnih sadržaja |
| | | 61. | Niz znakova (string) kao vrsta niza, specifične operacije za rad sa stringovima (prolazak kroz string petljom, neke korisne metode string klase) |
| | | 62. | Vježba |
| | | 63. | Primjeri programskih zadataka sa stringovima (razdvajanje rečenice na riječi, prebrojavanje slova, zamjena velikih i malih slova...) |
| | | 64. | Vježba |
| | | 65. | Višedimenzionalna polja (matrice) |
| | | 66. | Vježba |
| | | 67. | Primjeri zadataka sa matricama (determinanta, glavna/sporedna dijagonala...), primjeri problema iz realnog svijeta koji se rješavaju pomoću matrica |
| JUNI | 3 | 68. | Vježba |
| | | 69. | Ponavljanje gradiva |
| | | 70. | Praktična provjera usvojenosti nastavnih sadržaja |

Prijedlog forme obrazaca za razvijanje nastavnog plana i programa

nastavnih predmeta modula IT za gimnaziju izbornog područja Informacionih tehnologija

Škola: GIMNAZIJA

Izorno područje/zanimanje: INFORMACIONE TEHNOLOGIJE

Nastavni predmet: PROGRAMIRANJE 1

Razred: I (prvi)

Broj sati sedmično: 2

Broj sati za školsku godinu: 70

CILJI I ZADACI:

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

Zadaci

Omogućiti učeniku da:

- definiše pojmove u programiranju
- nabraja i objašnjava faze programiranja
- objašnjava razvoj programskih jezika

- definiše pojam algoritma
- izrađuje algoritme
- rješavanja probleme dijagramom toka i pseudokodom
- definiše pojma i osnovne strukture programa
- nabrāja i objašnjava osnovne tipove podataka
- objašnjava i primjenjuje naredbu unos i ispis podataka
- objašnjava i primjenjuje osnovne aritmetičke i logičke operacije
- objašnjava i primjenjuje naredbe pridruživanja
- definiše pojma grananja
- objašnjava razlike između jednostrukog i višestrukog grananja
- primjenjuje grananje u rješavanju programskih zadataka
- objašnjava i primjenjuje petlje
- objašnjava i primjenjuje višestruke petlje
- upotrebljava složene naredbe ponavljanja i primjenjuje ih u jednostavnim programskim zadacima
- primjenjuje grananje i složene naredbe ponavljanja na kompleksnim proceduralnim programskim zadacima
- definiše pojma niz
- primjenjuje nizove, (različiti tipova, cijelo brojne, realne i znakovne) u programskim zadacima
- primjena jednostavnih dvodimenzionalnih nizova
- prihvaća znanje o raznim složenim algoritimima
- primjenjuje grananje, složene naredbe ponavljanja i nizove na kompleksnim programskim zadacima
- upoznaje se sa objektno orijentisanim stilom programiranja
- upoznaje se sa osnovnim pojmovima objektno orijentisane paradigme (objekta, klasa, nasljeđivanje)
- stječe znanja o elementima objektno orijentisane analize i objektno orijentisanom projektovanju programa

NASTAVNI PROGRAM

| Nastavne oblasti / Nastavne cjeline | Sati | Nastavni sadržaj / Nastavne jedinice | Po završetku nastavne cjeline učenik će imati sljedeća znanja, vještine i kompetencije: |
|---|------|---|---|
| Uvod u programiranje | 7 | <ul style="list-style-type: none"> • Uvod u predmet, literatura i pribor • Šta je to program? Šta znači znati programirati? • Princip rada računara (ponavljanje): način rada procesora, interna i eksterna memorija, mrežna komunikacija • Zašto učimo programirati? Zanimanje "programer" • Programski jezici • Pojam algoritma, pseudo-kod, dijagram toka | <ul style="list-style-type: none"> ✓ Razlikuje pojmove program i programiranje. ✓ Razumije princip rada procesora, mrežne komunikacije i interne i eksterne memorije. ✓ Razumije zašto učimo programiranje, šta znači biti programer i da postoje različiti programski jezici. ✓ Razumije pojam algoritma i biti u mogućnosti primijeniti pseudo-kod i dijagram toka. ✓ Demonstrirati korištenje pseudo-koda i dijagrama toka. ✓ Procjenjuje značaj i primjenu algoritama u rješavanju problema. |
| Programsko okruženje i jednostavan program | 7 | <ul style="list-style-type: none"> • Upoznavanje sa programskim okruženjem • Prvi program: struktura "Hello world" programa, unos i pokretanje, naredba za izlaz teksta na ekran, sekvencijalno izvršenje • Šta se dešava kada pokrenemo program? Pojmovi: kompajler, linker, interpreter, debugger • Šta se dešava kada imamo sintaksnu grešku u programu i kako je otkloniti? • Drugi program: deklaracija, ulaz i izlaz cjelobrojnih promjenljivih, matematička operacija i dodjela, česte greške | <ul style="list-style-type: none"> ✓ Upoznaje se sa načinom upravljanja okruženja i demonstrira kroz izradu programa „Hello world“ upotrebu programskog jezika. ✓ Razumije šta je to linker, interpreter, kompajler i debugger. ✓ Uz pomoć debugera biti će u mogućnosti da indentificira greške (ako postoje) u korištenom programskom jeziku visoke razine, a uz pomoć kompajlera napraviti će najjednostavnije programe, čiji će rad demonstrirati na računaru. ✓ Koristi jezike visoke razine, uključujući korištenje standardnih biblioteka kod programiranja. |
| Osnovni pojmovi programiranja, ulaz i izlaz | 13 | <ul style="list-style-type: none"> • Tipovi podataka, deklaracija promjenljivih • Pojam operatora i operanda, operator dodjele, najčešći matematički operatori • Konstante, komentari, ulaz i izlaz | <ul style="list-style-type: none"> ✓ Zna razlikovati i primijeniti različite tipove podataka. ✓ Zna deklarirati promjenjive, koristiti konstante, komentare i koristiti ih u izradi najjednostavnih programa koji sadrže neke od matematičkih |

| | | |
|---|--|--|
| | <ul style="list-style-type: none"> • numeričkih vrijednosti • Opšti princip rješavanja programskih zadataka na primjeru rješavanja matematičkih problema • Matematičke funkcije • Znakovni tipovi (char i string sa vrlo jednostavnim primjerima upotrebe) • Ostali operatori, prioritet operatora, tip rezultata operacije | <p>operacija ili funkcija.</p> <ul style="list-style-type: none"> ✓ Zna da kreira najosnovnije programe koji će biti korišteni od krajnjeg korisnika a sadržat će obradu najjednostavnih matematičkih operacija ili funkcija. |
| <p>Programske strukture grananja</p> | <p>11</p> <ul style="list-style-type: none"> • Tipovi grananja i implementacija u programskom jeziku • Operatori poredjenja • Logički operatori, osnove matematičke logike • Višestruko grananje naredbom switch-case, naredba goto • Primjeri programskih zadataka sa grananjem, prevođenje realnih problema u programski kod | <ul style="list-style-type: none"> ✓ Razumije i razlikuje različite tipove grananja, operatore poredjenja, logičke operatore, višestruke naredbe grananja kao što su switch-case i goto naredba. ✓ Zna da rješava realne probleme demonstrirajući svoje vještine na računaru, kreiranjem programa u programskom jeziku visoke razine. ✓ Povezuje odgovarajuće vrste podataka sa njihovom namjenom u programu. |
| <p>Programske strukture ponavljanja</p> | <p>17</p> <ul style="list-style-type: none"> • Ponavljanje naredbom for • Različita dužina koraka, opadajuće petlje, naredbe break i continue, česte greške pri radu sa petljama • Neki karakteristični algoritmi sa petljama (stepenovanje, provjera da li je broj prost, rastavljanje broja na cifre) • Rješavanje problema iz drugih naučnih oblasti programiranjem: računanje sume reda, određivanje nula funkcije metodom polovljenja intervala, kosi hitac... • Ponavljanje naredbama while i do-while • Višestruke (ugnuježdene) petlje | <ul style="list-style-type: none"> ✓ Razumije i razlikuje strukture (naredbe break i continue) ponavljanja kao i vrste (for, while i do-while) ponavljanja. ✓ Odabire algoritam koji podržava odgovarajuću strukturu podataka. ✓ Zna kako konstruisati na papiru naredbe break i continue, jednostruke i više struke petlje, a zatim odabira i primjenjuje na računaru neku od njih kako bih riješio realan problem (kroz primjenu jednostavnih algoritama u programskom jeziku visoke razine) kao što su stepenovanje, provjera da li je broj prost, rastavljanje broja na cifre, računanje sume reda, određivanje nula funkcije metodom |

| | | |
|-------------------------------|---|--|
| | <ul style="list-style-type: none"> • Primjeri programskih zadataka sa višestrukim (ugniježenim) petljama (iscrtavanje oblika na ekranu itd.), česte greške | <ul style="list-style-type: none"> ✓ polovljenja intervala, kosi hitac itd. ✓ Dizajnira, piše i otklanja greške (debuguje u programima). ✓ Predstavlja, diskutuje i upoređuje svoje algoritme sa algoritmima drugih učenika iz razreda. ✓ Procjenjuje značaj algoritama u rješavanju problema. ✓ Objašnjava kako program radi i kako se dokumentuje. |
| <p>Polja (nizovi, arrays)</p> | <p>15</p> <ul style="list-style-type: none"> • Nizovi cijelih i realnih brojeva • Neki karakteristični algoritmi nad nizovima (maksimum/minimum, provjera da li članovi niza zadovoljavaju uslov, izbacivanje člana iz niza itd.) • Višedimenzionalna polja (matrice) • Primjeri zadataka sa matricama (determinanta, glavna/sporedna dijagonala...), primjeri problema iz realnog svijeta koji se rješavaju pomoću matrica • Niz znakova (string) kao vrsta niza, specifične operacije za rad sa stringovima (prolazak kroz string petljom, neke korisne metode string klase) • Primjeri programskih zadataka sa stringovima (razdvajanje rečenice na riječi, prebrojavanje slova, zamjena velikih i malih slova...) | <ul style="list-style-type: none"> ✓ Razumije šta su to nizovi, različite tipovi nizova (nizovi cijelih brojeva, realnih brojeva i znakova (string)) i njihovu strukturu (jednodimenzionalni i višedimenzionalni). ✓ Analizira strukture podataka u algoritamskom pristupu rješavanja problema. ✓ Demonstrira svoje znanje o nizovima na izradi složenijih algoritama kao što su maksimum/minimum, provjera da li članovi niza zadovoljavaju uslove, izbacivanje člana iz niza, determinanta, glavna/sporedna dijagonala, razdvajanje rečenice na riječi, prebrojavanje slova, zamjena velikih i malih slova, itd. ✓ Povezuje strukturu višedimenzionalnog nizova (matrica) s algoritmima. ✓ Odabire i upotrabljiva algoritme i algoritamske strukture za rješavanje problema. ✓ Pomaže drugima, predstavlja, diskutuje i upoređuje svoje složene algoritme sa algoritmima drugih učenika iz razreda. ✓ Procjenjuje prednosti i ograničenja algoritamskog pristupa u rješavanju problema. ✓ Pojašnjava principe OOP u rješavanju problema. |

NAČINI OSTVARIVANJA PROGRAMA¹

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

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

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

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

MEĐUPREDMETNA KORELACIJA

- Bosanski, hrvatski, srpski jezik i književnost – komunikacione vještine, obrada teksta
- Engleski jezik – komunikacione vještine, obrada teksta, korištenje literature i interneta
- Matematika – numerička obrada podataka
- Fizika – kreativno računarstvo
- Ostale prirodne nauke – kreativno računarstvo
- Likovna kultura i medijska kultura – grafička obrada podataka
- Psihologija – kreativno računarstvo, grafička obrada podataka, prezentacijske vještine
- Sociologija – kreativno računarstvo, grafička obrada podataka, prezentacijske vještine
- Svi ostali predmeti – napredno pretraživanje interneta, prezentacijske vještine, grafička obrada podataka

¹ NPP Informatika za gimnaziju, MONKS avgust 2016. godine

PRAĆENJE, VREDNOVANJE I OCJENJIVANJE

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

Preporuke:

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

PROFIL I STRUČNA SPREMA NASTAVNIKA²

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

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.

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

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

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

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

PREPORUČENI IZVOR INFORMACIJA ZA UČENJE

Pokritveno je postojećim udžbenikom za Informatiku (drugi razred) uz povećanje broja časova.

³ NPP Informatika za gimnaziju, MONKS avgust 2016. godine