



KURIKULUM ZA UČENJE PROGRAMIRANJA U PROGRAMSKOM JEZIKU PYTHON

Autori:

Saida Deljac, V. gimnazija Zagreb
Predrag Brođanac, V. gimnazija Zagreb
Marinko Zubac, Gimnazija Metković
Sanja Pavlović Šijanović, Gimnazija Vukovar

Recenzent:

mr. sc. Ivana Turčić Prstačić

*V. gimnazija, Klaićeva 1,
Zagreb 2015.*

Sadržaj

<i>Uvod</i>	3
<i>Modul I</i>	5
<i>Razrada jedinica ishoda učenja za Modul I</i>	9
<i>Modul II</i>	19
<i>Razrada jedinca ishoda učenja za Modul II</i>	23
<i>Modul III</i>	35
<i>Razrada jedinica ishoda učenja za Modul III</i>	38
<i>Dodatni sadržaji s ishodima učenja</i>	47
<i>Razrada jedinica ishoda učenja za dodatne sadržaje</i>	49
<i>Literatura</i>	52

Uvod

Kurikulum za učenje programiranja u programskom jeziku Python nastao je iz potrebe uvođenja novih koncepata učenja programiranja koji se ne mogu realizirati tradicionalnim programskim jezicima.

Kurikulum se sastoji od 3 modula. Svaki modul namijenjen je jednoj godini učenja programiranja u programskom jeziku Python. Za svaki modul (s pripadnim ishodima učenja) provedena je razrada jedinica ishoda učenja prema uputama za razvoj obrazovnih programa temeljem Priručnika za razvoj strukovnih standarda, zanimanja kvalifikacija i kurikuluma (studeni 2011.) i Metodologije izrade strukovnih standarda, zanimanja, kvalifikacija i kurikuluma (prosinac 2011.) proizašlih iz *IPA projekta 2007-2009 Jačanje institucionalnog okvira za razvoj strukovnih standarda, kvalifikacija i kurikuluma EuropeAid127472/d/ser/HR* i *Hrvatskog kvalifikacijskog okvira (2009)*.

Svaki od tri modula sadrži: naziv predmeta, ishode učenja koje će učenik steći u određenoj godini učenja, nastavne cjeline i razrađene nastavne teme, okruženje za učenje, metode i oblike rada, elemente i oblike praćenja i vrednovanja polaznika te literaturu za učenike. Okruženje i uvjeti za organizirano učenje sadrže uvijete koji su neophodni za ostvarivanje ishoda učenja korištenjem računalne i programske opreme.

Razrada jedinica ishoda učenja je dana u Kurikulumu za svaki modul i sve jedinice (nastavne teme) modula. Pojedina jedinica sadrži: naziv, razinu, obujam, cilj/svrhu, ishode učenja (5-10), uvjete u kojima se stječu kompetencije, provjeru i vrednovanje te primjer provjere i vrednovanja u skladu s preporučenim i propisanim obrascem za standard kvalifikacije.

Razina ishoda učenja je 4 i 5 što prema Hrvatskom kvalifikacijskom okviru odgovara razini srednjoškolskog obrazovanja.

Obujam ishoda učenja naveden je na početku svakog modula i predstavlja ukupno vrijeme koje je potrebno učeniku kako bi usvojio ishode učenja kako pojedinog modula tako i pojedine nastavne teme unutar modula. Jedinica od 1 obujma odgovara 25 sati učenja. U obujam je uključeno vrijeme povedeno na nastavi kao i vrijeme potrebno za izradu domaće zadaće, projektnih zadataka i drugih oblika organiziranog i samostalnog učenja.

Broj ishoda učenja jasno odražava raspon sadržaja učenja, a svaki ishodi započinje jednim aktivnim glagolom povezan sa sadržajem na koji se odnosi. Primjeri provjere i vrednovanja predstavljaju provjeru usvojenog sadržaja i primjene znanja u okviru minimalnih učeničkih postignuća.

Ovisno o broju godina učenja programiranja, ovaj kurikulum se može primjenjivati u nastavi informatike u općim, jezičnim i prirodoslovno-matematičkim gimnazijama kao i u strukovnim školama koje u svojim programima imaju temu programiranja.

Naziv predmeta	INFORMATIKA
Cilj predmeta	<p><i>Povezivanje s ciljevima jedinica ishoda učenja iz standarda kvalifikacije postići sljedeće:</i></p> <ul style="list-style-type: none"> • <i>razviti sposobnost algoritamskog pristupa rješavanju problema</i> • <i>steći znanja iz područja programiranja u programskom jeziku Python</i> • <i>primijeniti jednostavne i složene algoritamske strukture u osmišljavanju i stvaranju programa u programskom jeziku Python</i> • <i>razviti sposobnost vrednovanja pojedinih algoritama</i> • <i>razviti sposobnost donošenja odluka i zaključivanja</i> • <i>razviti logičke misaone procese u postupku rješavanja problema</i>
Opis predmeta	<p><i>Današnje društvo temelji se na informatičkim znanjima. Gotovo da i ne postoji područje ljudskog djelovanja u kojem se ne koriste različite aplikacije koje se izvode na računalu. Područje razvoja aplikacija je široko i perspektivno te predstavlja područje u koje se isplati ulagati i razvijati. Učenje i Razvoj sposobnosti programiranja potrebno je provesti na kreativan i učenicima pristupačan način koristeći jednostavan, a opet cjeloviti programski jezik s kojim se mogu zadovoljiti sve potrebe današnjih koncepata i načina programiranja.</i></p> <p><i>Python je edukativan i metodičko- didaktički koncipiran programski jezik koji se može prilagoditi različitim uzrastima učenika. Omogućuje stjecanje znanja iz područja različitih (jednostavnih i složenih) programskih struktura tako da čini postupak učenja jasnijim i jednostavnijim.</i></p>
Opis modula	<p><i>Cjelokupan kurikulum je podijeljen u 3 modula + dodatni modul. Moduli su raspodijeljeni prema godinama učenja i trenutnoj satnici informatike u srednjim školama. Može se primjenjivati u gimnazijama i strukovnim školama koje u svojim programima imaju podučavanje programiranja. U školama koje prema nastavnom planu i programu nemaju višegodišnje učenje programiranja može se primjenjivati u dodatnoj, izbornoj ili fakultativnoj nastavi. Dodatni sadržaji su namijenjeni posebnim programima u prirodoslovno- matematičkim gimnazijama koji imaju proširenu satnicu nastave informatike.</i></p>

Modul I

Naziv predmeta u prvoj godini učenja	INFORMATIKA
Kroz ovaj predmet u prvoj godini učenja učenik će steći sljedeće ishode učenja:	Programiranje i programski jezik Python <ol style="list-style-type: none">1. Opisati korake u rješavanju problema programiranjem2. Objasniti pojam i ulogu algoritma u postupku programiranja3. Usporediti proceduralne i objektno usmjerene programske jezike4. Argumentirati prednosti programskog jezika Python5. Pripremiti programsko okruženje za pisanje i izvođenje programa na računalu u programskom jeziku Python Aritmetički operatori i ugrađene funkcije <ol style="list-style-type: none">1. Razlikovati tipove podataka u Pythonu2. Koristiti aritmetičke operatore u Pythonu3. Objasniti pojam varijable4. Objasniti naredbu za pridruživanje5. Objasniti operatore proširenog pridruživanja6. Objasniti razliku između proceduralnog programskog jezika i Python-a u postupku pohranjivanja vrijednosti varijable u memoriji7. Opisati djelovanja osnovnih funkcija u Pythonu8. Upotrebljavati aritmetičke operatore, naredbe za pridruživanje i funkcije u osmišljavanju programa u Pythonu9. Upisati, spremiti i izvesti program u programskom okruženju Python-a Logički operatori i uvjetne naredbe <ol style="list-style-type: none">1. Objasniti djelovanje relacijskih i logičkih operatora2. Riješiti složeni izraz3. Opisati uvjetne naredbe oblika if i if-else4. Opisati uvjetnu naredbu oblika if-elif-else5. Osmisliti rješenje za zadatak s uvjetnim naredbama6. Opisati ulogu modula (zbirke funkcija) Algoritmi s naredbama za ponavljanje <ol style="list-style-type: none">1. Opisati naredbu sa unaprijed poznatim brojem ponavljanja (for)

2. Objasniti korake izvođenja programa s for petljom
3. Upotrijebiti algoritam za provjeru je li broj prost
4. Koristiti algoritam za zbrajanje ili množenje niza brojeva
5. Objasniti izvođenje ugniježdene for petlje
6. Opisati naredbu za uvjetno ponavljanje bloka naredbi (while)
7. Objasniti korake izvođenja programa s while petljom
8. Upotrijebiti algoritam za traženje najvećeg zajedničkog djelitelja
9. Analizirati algoritam za raščlambu prirodnog broja na znamenke
10. Upotrijebiti algoritme s naredbama za ponavljanje u rješavanju zadataka

Pisanje vlastitih funkcija

1. Opisati opći oblik pisanja vlastite funkcije
2. Dati primjer vlastite funkcije
3. Objasniti funkciju main()
4. Objasniti postupak stvaranja vlastitog modula
5. Primijeniti postupak stvaranja vlastite funkcije i modula u izradi programa

Niz znakova (String)

1. Objasniti osobine niza znakova kao zbirke u Pythonu
2. Objasniti operatore za rad s nizom znakova
3. Opisati ugrađene funkcije za rad s nizom znakova
4. Primijeniti postupak dohvaćanje jednog ili više znakova na problemu provjere je li riječ palindrom
5. Objasniti pravila rada s isječcima u znakovnom nizu
6. Opisati metode za rad sa stringovima
7. Primijeniti algoritam za traženje broja riječi u zadanoj rečenici
8. Primijeniti operatore, funkcije i isječke u stvaranju vlastitih algoritama

Liste u Pythonu

1. Objasniti osobine liste kao zbirke podataka u Pythonu
2. Stvarati različite vrste listi u programu
3. Opisati operatore i funkcije za rad s listama
4. Opisati metode za rad s listama
5. Opisati pretvorbu stringa u listu

	<p>6. Opisati pretvorbu liste stringova u jedan string</p> <p>7. Primijeniti liste kao zbirke u stvaranju programa u Pythonu</p>
Razrada	
<i>Nastavne cjeline</i>	<i>Nastavne teme</i>
Programiranje i programski jezik Python	<p>Proces programiranja</p> <p>Pojam i uloga algoritma</p> <p>Proceduralni i objektno usmjereni jezici</p> <p>Programsko okruženje Pythona</p>
Aritmetički operatori i ugrađene funkcije	<p>Tipovi podataka i varijable</p> <p>Znakovi pridruživanja</p> <p>Osnovne funkcije</p>
Logički operatori i uvjetne naredbe	<p>Relacijski i logički operatori</p> <p>Uvjetna naredba (if, if-else, if-elif-else)</p>
Algoritmi s naredbama za ponavljanje	<p>Naredba sa unaprijed poznatim brojem ponavljanja (for petlja)</p> <p>Algoritam za ispitivanje prostih brojeva</p> <p>Ugniježđena for petlja</p> <p>Naredba za uvjetno ponavljanje bloka naredbi (while petlja)</p> <p>Algoritam za traženje najvećeg zajedničkog djelitelja</p> <p>Algoritam za raščlambu prirodnog broja na znamenke</p>
Pisanje vlastitih funkcija	<p>Definiranje vlastitih funkcija</p> <p>Oblici funkcija</p> <p>Pojam modula i postupak izrade modula</p>
Niz znakova (String)	<p>String kao zbirka podataka u Pythonu</p> <p>Operacije nad stringovima</p> <p>Ugrađene funkcije za rad sa stringovima</p> <p>Isječci stringa</p> <p>Metode za rad sa stringom</p>
Liste u Pythonu	<p>Lista kao zbirka podataka u Pythonu</p> <p>Stvaranje listi</p> <p>Operatori za rad s listama</p> <p>Ugrađene funkcije za rad s listama</p> <p>Metode za rad s listama</p> <p>Pretvorba stringa u listu i liste u string</p>
Okruženje i uvjeti za organizirano učenje	<p>U cilju ostvarivanja navedenih ishoda učenja nužno je tijekom cijelog nastavnog procesa polazniku omogućiti korištenje računala u specijaliziranoj informatičkoj učionici s najviše 16 računala i grupom od najviše 16 polaznika, po principu za jednim računalom jedan polaznik.</p> <p>Razredni odjel koje ima više od 16 polaznika mora se dijeliti na grupe kako bi se zadovoljio gornji kriterij.</p>

	U slučaju da nije moguće cjelokupnu nastavu izvoditi u specijaliziranoj informatičkoj učionici u grupi po 16 polaznika, moguće je 50% nastave izvoditi u učionici opremljenoj s jednim računalom s projektorom i pristupom na internet te 50% u specijaliziranoj informatičkoj učionici s najviše 16 računala i grupom od najviše 16 računala, po principu za jednim računalom jedan polaznik.
Ostalo	
Metode i oblici rada:	<p>Metode: predavačka metoda, metoda dijaloga, heuristička metoda, problemska metoda, programirana metoda, demonstracijska metoda i istraživačka metoda.</p> <p>Oblici: frontalni oblik nastave, diferencirana nastava, individualizirana nastava, problemska nastava, programirana nastava, egzemplarna nastava, mentorska nastava, demonstracijska nastava.</p> <p>Napomena: izbor metoda i oblika rada za svaki nastavni sat određuje nastavnik, sukladno nastavnim sadržajima, osobitosti polaznika te materijalnim i drugim uvjetima.</p>
Elementi i oblici praćenja i vrjednovanja polaznika	<p>Elementi: usvojenost sadržaja (usmenim i pisanim načinom), primjena znanja (usmenim, pisanim načinom, provjera znanja i usvojenosti vještina na računalu, projektni zadatak).</p> <p>Oblici: samostalan rad (domaće zadaće, istraživanje, suradnja u nastavi i dr.).</p>
Literatura za učenike:	Prema Katalogu odobrenih udžbenika i drugih pomoćnih nastavnih sredstava Ministarstva znanosti, obrazovanja i sporta.

Razrada jedinica ishoda učenja za Modul I Obujam (3.4)

Naziv	<i>Programiranje i programski jezik Python</i>
Razina	4
Obujam	0.1
Cilj/svrha jedinice ishoda učenja	Objasniti korake u rješavanju problema programiranjem te se pripremiti za pisanje i izvođenje programa u programskom jeziku Python.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Opisati korake u rješavanju problema programiranjem 2. Objasniti pojam i ulogu algoritma u postupku programiranja 3. Usporediti proceduralne i objektno usmjerene programske jezike 4. Argumentirati prednosti programskog jezika Python 5. Pripremiti programsko okruženje za pisanje i izvođenje programa na računalu u programskom jeziku Python
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<ol style="list-style-type: none"> 1. Opisati korake u rješavanju problema programiranjem Navesti korake u rješavanju problema programiranjem. 2. Objasniti pojam i ulogu algoritma u postupku programiranja Izreći definiciju algoritma i dati primjer algoritma. 3. Usporediti proceduralne i objektno usmjerene programske jezike Navesti jednu prednost i jedan nedostatak proceduralnih i objektno usmjerenih programskih jezika. 4. Argumentirati prednosti programskog jezika Python Opisati 2 prednosti programskog jezika Python koje ga razlikuju od ostalih programskih jezika. 5. Pripremiti programsko okruženje za pisanje i izvođenje programa na računalu u programskom jeziku Python Otvoriti interaktivno programsko okruženje - Python Shell

Naziv	Aritmetički operatori i ugrađene funkcije
Razina	4
Obujam	0.2
Cilj/svrha jedinice ishoda učenja	Koristiti varijable , aritmetičke operatore, naredbu za pridruživanje i ugrađene funkcije u izradi jednostavnih programa u programskom jeziku Python.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Razlikovati tipove podataka u Pythonu 2. Koristiti aritmetičke operatore u Pythonu 3. Objasniti pojam varijable 4. Objasniti naredbu za pridruživanje 5. Objasniti operatore proširenog pridruživanja 6. Objasniti razliku između proceduralnog programskog jezika i Python-a u postupku pohranjivanja vrijednosti varijable u memoriji 7. Opisati djelovanja osnovnih funkcija u Pythonu 8. Upotrebljavati aritmetičke operatore, naredbe za pridruživanje i funkcije u osmišljavanju programa u Pythonu 9. Upisati, spremi i izvesti program u programskom okruženju Python-a
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrjednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Razlikovati tipove podataka u Pythonu Dati primjere tipova podatka u Pythonu: cijeli broj, broj s pomičnom točkom, logički tip, znakovni niz ili string.</p> <p>2. Koristiti aritmetičke operatore u Pythonu Ispisati aritmetičke operatore za zbrajanje, oduzimanje, množenje, dijeljenje, cjelobrojno dijeljenje i ostatak cjelobrojnog dijeljenja.</p> <p>3. Objasniti pojam varijable Objasniti čemu služe varijable u programu.</p> <p>4. Objasniti naredbu za pridruživanje Napisati i objasniti jedan primjer jednostrukog pridruživanja vrijednosti varijabli te jedan primjer višestrukog pridruživanja vrijednosti varijablama.</p> <p>5. Objasniti operatore proširenog pridruživanja Dati primjer za operatore proširenog pridruživanja (+=, -=, *=, /=)</p> <p>6. Objasniti razliku između proceduralnog programskog jezika i Pythona u postupku pohranjivanja vrijednosti varijable u</p>

	<p>memoriji Objasniti način pohranjivanja vrijednosti varijable u memoriju u programskom jeziku Python</p> <p>7. Opisati djelovanje ugrađenih funkcija u Pythonu Opisati što rade funkcije: int(), round(), float(), bin(), oct(), hex(), sqrt(), random(), ord(), chr(), input(), print()</p> <p>8. Upotrebljavati aritmetičke operatore, naredbe za pridruživanje i ugrađene funkcije u osmišljavanju programa u Python-u Napisati programa kojim će unijeti dva cijela broja te ispisati njihov zbroj, razlika, umnožak i kvocijent.</p> <p>9. Upisati, spremiti i izvesti program u programskom okruženju Pythona Unijeti, izvesti i spremiti program u programskom jeziku Python.</p>
--	--

Naziv	Logički operatori i uvjetne naredbe
Razina	4
Obujam	0.4
Cilj/svrha jedinice ishoda učenja	Primjenjivati aritmetičke, logičke i relacijske operatore u izradi programa u kojima se koriste uvjetne naredbe.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti djelovanje relacijskih i logičkih operatora 2. Riješiti složeni izraz 3. Opisati uvjetne naredbe oblika if i if-else 4. Opisati uvjetnu naredbu oblika if-elif-else 5. Osmisliti rješenje za zadatak s uvjetnim naredbama 6. Opisati ulogu modula (zbirke funkcija)
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Objasniti djelovanje relacijskih i logičkih operatora Objasniti primjenu relacijskih operatora ($>$, $<$, $>=$, $<=$, $==$, $!=$) i logičkih operatora (and or, not) u programu u Pythonu.</p> <p>2. Riješiti složeni izraz Riješiti složeni izraz koji se sastoji od relacijskih, aritmetičkih i logičkih operacija poštujući prioritete u rješavanju (1. aritmetički, 2. relacijski, 3. logički)</p> <p>3. Opisati uvjetnu naredbu oblika if-else Objasniti kako radi uvjetna naredba oblika if, if-else napisana u pseudojeziku i programskom jeziku Python.</p> <p>4. Opisati uvjetnu naredbu oblika if-elif-else Objasniti kako radi uvjetna naredba oblika if-elif-else napisana u pseudojeziku i programskom jeziku Python.</p> <p>5. Osmisliti rješenje za zadatak s uvjetnim naredbama Napisati primjer programa s uvjetnom naredbom (uspoređivanje brojeva, ispitivanje djeljivosti broja, ispitivanje parnosti broja ili ispitivanje pozitivnosti broja)</p> <p>6. Opisati ulogu modula (zbirke funkcija) Objasniti namjenu modula za rad s matematičkim funkcijama (math) i modula za rad sa slučajnim brojevima (random).</p>

Naziv	Algoritmi s naredbama za ponavljanje
Razina	5
Obujam	1.1
Cilj/svrha jedinice ishoda učenja	Osmišljavati i primjenjivati algoritme s naredbama za ponavljanje u pisanju programa u Pythonu.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Opisati naredbu sa unaprijed poznatim brojem ponavljanja (<i>for</i>) 2. Objasniti korake izvođenja programa s <i>for</i> petljom 3. Upotrijebiti algoritam za provjeru je li broj prost 4. Koristiti algoritam za zbrajanje ili množenje niza brojeva 5. Objasniti izvođenje ugniježdene <i>for</i> petlje 6. Opisati naredbu za uvjetno ponavljanje bloka naredbi (<i>while</i>) 7. Objasniti korake izvođenja programa s <i>while</i> petljom 8. Upotrijebiti algoritam za traženje najvećeg zajedničkog djelitelja 9. Analizirati algoritam za raščlambu prirodnog broja na znamenke 10. Upotrijebiti algoritme s naredbama za ponavljanje u rješavanju zadataka
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Opisati naredbu sa unaprijed poznatim brojem ponavljanja (<i>for</i>) Objasniti kako radi naredba za ponavljanje <i>for</i> napisana u pseudojeziku i programskom jeziku Python.</p> <p>2. Objasniti korake izvođenja programa s <i>for</i> petljom Izračunati vrijednosti varijabli u zadanom programskom kôdu koji sadrži <i>for</i> petlju.</p> <p>3. Upotrijebiti algoritam za provjeru je li broj prost Objasniti algoritam i program u Python-u za zadatak ispitivanja prostog broja korištenjem naredbe <i>for</i>.</p> <p>4. Koristiti algoritam za zbrajanje ili množenje niza brojeva Objasniti program u Programskom jeziku Python koji će zbrojiti sve parne prirodne brojeve do zadanog prirodnog broja n.</p> <p>5. Objasniti izvođenje ugniježdene <i>for</i> petlje Objasniti izvođenje ugniježdene <i>for</i> petlje na primjeru algoritma ispitivanja prostih brojeva za prvih n prirodnih brojeva.</p> <p>6. Opisati naredbu za uvjetno ponavljanje bloka naredbi</p>

	<p>(while) Objasniti kako radi naredba za ponavljanje <i>while</i> napisana u pseudojeziku i programskom jeziku Python.</p> <p>7. Objasniti korake izvođenja programa s while petljom Izračunati vrijednosti varijabli u zadanom programskom kôdu koji sadrži <i>while</i> petlju.</p> <p>8. Upotrijebiti algoritam za traženje najvećeg zajedničkog djelitelja Koristeći se <i>for</i> petljom napisati algoritam i program u Pythonu koji će dati za rješenje najveći zajednički djelitelj za dva unesena prirodna broja.</p> <p>9. Analizirati algoritam za raščlambu prirodnog broja na znamenke Koristeći se <i>for</i> petljom ispisati algoritam i program u Pythonu koji će za uneseni broj ispitati je li Armstrongov broj.</p> <p>10. Upotrijebiti algoritme s naredbama za ponavljanje u rješavanju zadataka Napisati primjer programa s naredbom za ponavljanje (računanje prosjeka brojeva ili traženje najmanjeg i najvećeg u nizu brojeva)</p>
--	---

Naziv	<i>Pisanje vlastitih funkcija</i>
Razina	5
Obujam	0.2
Cilj/svrha jedinice ishoda učenja	Osmišljavati algoritme za programe koji će za dijelove svojih rješenja koristiti vlastite funkcije.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Opisati opći oblik pisanja vlastite funkcije 2. Dati primjer vlastite funkcije 3. Objasniti funkciju main() 4. Objasniti postupak stvaranja vlastitog modula 5. Primijeniti postupak stvaranja vlastite funkcije i modula u izradi programa
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Objasniti opći oblik pisanja vlastite funkcije Ispisati opći oblik funkcije tako da su u ispis uključene ključne riječi (<i>def</i> i <i>return</i>), naziv funkcije, popis parametara i blok naredbi koje će se izvršiti unutar funkcije.</p> <p>2. Dati primjer vlastite funkcije Napisati program u Pythonu koji sadrži vlastitu funkciju.</p> <p>3. Objasniti funkciju main() Napisati program u Pythonu koji će sadržavati funkciju main()</p> <p>4. Objasniti postupak stvaranja vlastitog modula Izvesti na jednom primjeru postupak stvaranja vlastitog modula.</p> <p>5. Primijeniti postupak stvaranja vlastite funkcije i modula u izradi programa Napisati primjer funkcije (računanje prostog broja ili računanje Armstrongovog broja) koja će se pozivati unutar glavnog programa (main()).</p>

Naziv	Niz znakova (String)
Razina	5
Obujam	0.6
Cilj/svrha jedinice ishoda učenja	Primjenjivati operatore, funkcije i metode za rad sa nizovima znakova u izradi programa u programskom jeziku Python.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti osobine niza znakova kao zbirke u Pythonu 2. Objasniti operatore za rad s nizom znakova 3. Opisati ugrađene funkcije za rad s nizom znakova 4. Primijeniti postupak dohvaćanje jednog ili više znakova na problemu provjere je li riječ palindrom 5. Objasniti pravila rada s isječcima u znakovnom nizu 6. Opisati metode za rad sa stringovima 7. Primijeniti algoritam za traženje broja riječi u zadanoj rečenici 8. Primijeniti operatore, funkcije i isječke u stvaranju vlastitih algoritama
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<ol style="list-style-type: none"> 1. Objasniti osobine niza znakova kao zbirke u Pythonu Opisati osobine zbirke sa slijednim smještajem elementa. 2. Objasniti operatore za rad s nizom znakova Opisati operatore (+, *, in, not in) za rad s nizom znakova (stringom) u programskom jeziku Python. 3. Opisati ugrađene funkcije za rad s nizom znakova Opisati funkcije len(), min(), max(), str(). 4. Primijeniti postupak dohvaćanje jednog ili više znakova na problemu provjere je li riječ palindrom Napisati algoritam i program u Pythonu za ispitivanje je li zadana riječ palindrom. 5. Objasniti pravila rada s isječcima u znakovnom nizu. Dati 3 primjera isječaka s različitim intervalima isijecanja i različitim korakom isijecanja. 6. Opisati metode za rad sa stringovima. Opisati metode indeks(), replace(), capitalize(), upper(), lower(), center(). 7. Primijeniti algoritam za traženja broja riječi u zadanoj

	<p>rečenici.</p> <p>Napisati algoritam i program u Pythonu za traženje broja riječi u zadanoj rečenici.</p> <p>8. Primijeniti operatore, funkcije i isječke u stvaranju vlastitih programa.</p> <p>Napisati primjer programa za rad sa stringovima (prebrojavanje zadanog slova u nizu znakova ili traženje podniza u zadanom nizu)</p>
--	---

Naziv	Liste u Pythonu
Razina	5
Obujam	0.8
Cilj/svrha jedinice ishoda učenja	Osmišljavati algoritme za programe u Pythonu koji će u svojim rješenjima koristiti liste kao zbirke podataka.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti osobine liste kao zbirke podataka u Pythonu 2. Stvarati različite vrste listi u programu 3. Opisati operatore i funkcije za rad s listama 4. Opisati metode za rad s listama 5. Opisati pretvorbu stringa u listu 6. Opisati pretvorbu liste stringova u jedan string 7. Primijeniti liste kao zbirke u stvaranju programa u Pythonu
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Objasniti osobine liste kao zbirke podataka u Pythonu Objasniti razmještaj i srodnost elemenata u listi te način dohvaćanja elemenata liste.</p> <p>Stvarati različite vrste listi u programu Konstruirati listu na načine: izrazom i for petljom; kao iterabilni objekt.</p> <p>2. Opisati operatore i funkcije za rad s listama Dati primjere u programu za operatore +, -, in i not in i funkcije min(), max().</p> <p>3. Opisati metode za rad s listama Dati primjere u programu za metode: append(), extend(), insert(), remove(), reverse(), sort(), split(sep).</p> <p>4. Opisati pretvorbu stringa u listu Objasniti kako radi funkcija list().</p> <p>5. Opisati pretvorbu liste stringova u jedan string Objasniti kako radi funkcija sep.join(lista).</p> <p>6. Primijeniti liste kao zbirke u stvaranju programa u Pythonu Napisati primjer programa za unos 10 članova liste te ispis zbroja svih elemente liste koji su djeljivi sa 3. Ispisati i sortiranu ulaznu listu.</p>

Modul II

<p>Naziv predmeta u drugoj godini učenja</p>	<p>INFORMATIKA</p>
<p>Kroz ovaj predmet u drugoj godini učenja učenik će steći sljedeće ishode učenja:</p>	<p>Zbirke s raspršenim smještajem elemenata (skup i rječnik)</p> <ol style="list-style-type: none"> 1. Objasniti osobine skupa i rječnika kao zbirke podataka u Python-u 2. Opisati operatore i funkcije za rad sa skupovima 3. Objasniti algoritam Eratostenovog sita 4. Upotrijebiti skupove u osmišljavanju rješenja programa u Python-u 5. Napraviti rječnik 6. Ispisati elemente rječnika 7. Opisati funkcije i metode za rad s rječnikom 8. Razlikovati načine sortiranja elemenata rječnika 9. Primijeniti rječnik u rješavanju programa u programskom jeziku Python <p>Datoteke</p> <ol style="list-style-type: none"> 1. Objasniti ulogu datoteka u programu 2. Opisati korake za otvaranje datoteke u Python-u 3. Objasniti razlike u metodama za čitanje podatka iz datoteke u Python-u 4. Objasniti razlike u metodama za ispis podataka u datoteku u Python-u 5. Primijeniti metode za rad s datotekama u pisanju programa u programskom jeziku Python. <p>Kornjačina grafika</p> <ol style="list-style-type: none"> 1. Objasniti izgled grafičkog prozora 2. Opisati funkcije za gibanje pera 3. Primijeniti algoritam za crtanje pravilnog n-terokuta 4. Opisati metodu za crtanje kruga i kružnog isječka 5. Objasniti metode za postavljanje boje pera i ispune 6. Objasniti pojam linearne interpolacije boja 7. Navesti metode za upis i ispis teksta u grafičkom prozoru 8. Primjenjivati grafičke funkcije u izradi programa u Pythonu <p>Objektno usmjereno programiranje</p> <ol style="list-style-type: none"> 1. Objasniti strukturu objektno usmjerenog programa 2. Povezati ugrađene zbirke te funkcije i metode nad tim zbirkama u Python-u s objektno usmjerenim programiranjem 3. Objasniti konstruktor klase u Python-u 4. Dati primjer metode s dodatnim parametrima 5. Opisati specijalne metode za rad s objektima 6. Objasniti pojam <i>nasljeđivanja klase</i> 7. Primijeniti objektno usmjereni način programiranja u stvaranju algoritma i programa u programskom jeziku Python.

	<p>Algoritmi s rekurzijom</p> <ol style="list-style-type: none"> 1. Objasniti induktivni pristup rješavanju problema 2. Objasniti pojam rekurzije 3. Izvesti rekurzivnu relaciju i uvjet prekida za zadani problem 4. Objasniti pojam <i>memoizacija</i> 5. Primjenjivati algoritme s rekurzijama u rješavanju programa u Python-u <p>Ocjena složenosti algoritma</p> <ol style="list-style-type: none"> 1. Objasniti pojam složenosti algoritma 2. Kategorizirati složenost algoritama prema notaciji <i>veliko O</i> 3. Analizirati algoritam slijednog traženja u listi s nepoređanim elementima 4. Analizirati algoritam traženja elementa u poredanoj (sortiranoj) listi 5. Usporediti vremena traženja elementa u nepoređanoj i poredanoj listi <p>Apstraktne strukture podataka</p> <ol style="list-style-type: none"> 1. Objasniti strukturu stoga (stack) 2. Primijeniti osnovne operacije nad stogom 3. Izračunati vrijednost aritmetičkog izraza zadano u infix, postfix ili prefix notaciji 4. Objasniti strukturu reda (queue) 5. Primijeniti osnovne operacije nad redom 6. Opisati pojmove vezane uz strukturu - stablo (tree) 7. Primijeniti algoritme obilaska stabla (preorder, inorder, postorder) na konkretnom primjerku stabla 8. Rekonstruirati binarno stablo traženja na osnovu zadane riječi 9. Rekonstruirati binarnu hrpu na osnovnu zadane riječi <p>Algoritmi sortiranja</p> <ol style="list-style-type: none"> 1. Opisati algoritam sortiranje zamjenom elemenata (exchange sort) 2. Opisati algoritam izbora najmanjeg elementa (selection sort) 3. Opisati algoritam zamjene susjednih elemenata (bubble sort) 4. Opisati algoritam umetanjem (insertion sort) 5. Opisati algoritam sjedinjavanja (merge sort) 6. Opisati algoritam brzog sortiranja (quick sort) 7. Opisati algoritam razvrstavanjem (bucket sort) 8. Usporediti algoritme sortiranja
Razrada	
<i>Nastavne cjeline</i>	<i>Nastavne teme</i>
Zbirke s raspršenim smještajem elemenata (skup i rječnik)	Skup zbirka podataka u Pythonu Operacije i funkcije za rad sa skupom Algoritam generiranja prostih brojeva do nekog zadanog broja Stvaranje rječnika i dohvaćanje elementa rječnika Metode i funkcije za rad s rječnikom

	Sortiranje rječnika
Datoteke	Uloga datoteke u programu Metode za upis podataka u datoteku Metode za čitanje podataka iz datoteke
Kornjačina grafika	Grafički prozor Funkcije za pomicanje kornjače Crtanje pravilnog n-terokuta i kružnice Metode za postavljanje boje pera i ispune Interpolacija boja Metode za rad s tekстом unutar grafičkog prozora
Objektno usmjereno programiranje	Osnovni pojmovi u objektno usmjerenom programiranju (klasa, svojstvo, metoda, objekt) Povezivanje poznatih funkcija s objektno usmjerenim programiranjem Pristup atributima putem posrednika Metode s dodatnim i objektnim parametrima Specijalne metode Nasljeđivanje i kolekcije klasa
Algoritmi s rekurzijom	Pojam rekurzije Rekurzivni postupci
Ocjena složenosti algoritma	Složenost algoritma Ocjena složenosti algoritma
Apstraktne strukture podataka	Struktura podataka -Stog Matematički izrazi u infix, postfix ili prefix notaciji Struktura podataka - Red Struktura podataka - Stablo Obilasci stabla Rekonstrukcija stabla Binarno stablo traženja Binarna hrpa
Algoritmi sortiranja	Sortiranje zamjenom elemenata (exchange sort) Sortiranje izborom najmanjeg elementa (selection sort) Sortiranje zamjenom susjednih elemenata (bubble sort) Sortiranje umetanjem (insertion sort) Sortiranje sjedinjavanjem (merge sort) Brzo sortiranje (quick sort) Sortiranje razvrstavanjem (bucket sort)
Okruženje i uvjeti za organizirano učenje	U cilju ostvarivanja navedenih ishoda učenja nužno je tijekom cijelog nastavnog procesa polazniku omogućiti korištenje računala u specijaliziranoj informatičkoj učionici s najviše 16 računala i grupom od najviše 16 polaznika, po principu za jednim računalom jedan polaznik. Razredni odjel koje ima više od 16 polaznika mora se dijeliti na grupe kako bi se zadovoljio gornji kriterij.

	U slučaju da nije moguće cjelokupnu nastavu izvoditi u specijaliziranoj informatičkoj učionici u grupi po 16 polaznika, moguće je 50% nastave izvoditi u učionici opremljenoj s jednim računalom s projektorom i pristupom na internet te 50% u specijaliziranoj informatičkoj učionici s najviše 16 računala i grupom od najviše 16 računala, po principu za jednim računalom jedan polaznik.
Ostalo	
Metode i oblici rada:	<p>Metode: predavačka metoda, metoda dijaloga, heuristička metoda, problemska metoda, programirana metoda, demonstracijska metoda i istraživačka metoda.</p> <p>Oblici: frontalni oblik nastave, diferencirana nastava, individualizirana nastava, problemska nastava, programirana nastava, egzemplarna nastava, mentorska nastava, demonstracijska nastava.</p> <p>Napomena: izbor metoda i oblika rada za svaki nastavni sat određuje nastavnik, sukladno nastavnim sadržajima, osobitosti polaznika te materijalnim i drugim uvjetima.</p>
Elementi i oblici praćenja i vrjednovanja polaznika	<p>Elementi: usvojenost sadržaja (usmenim i pisanim načinom), primjena znanja (usmenim, pisanim načinom, provjera znanja i usvojenosti vještina na računalu, projektni zadatak).</p> <p>Oblici: samostalan rad (domaće zadaće, istraživanje, suradnja u nastavi i dr.).</p>
Literatura za učenike:	Prema Katalogu odobrenih udžbenika i drugih pomoćnih nastavnih sredstava Ministarstva znanosti, obrazovanja i sporta.

**Razrada jedinica ishoda učenja za Modul II
OBUJAM(3.3)**

Naziv	Zbirke s raspršenim smještajem elemenata (skup i rječnik)
Razina	4
Obujam	0.4
Cilj/svrha jedinice ishoda učenja	Primjenjivati skupove i rječnike u osmišljavanju algoritama i programa u programskom jeziku Python.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti osobine skupa i rječnika kao zbirke podataka u Python-u 2. Opisati operatore i funkcije za rad sa skupovima 3. Algoritam generiranja prostih brojeva do nekog zadanog broja 4. Upotrijebiti skupove u osmišljavanju rješenja programa u Python-u 5. Napraviti rječnik 6. Ispisati elemente rječnika 7. Opisati funkcije i metode za rad s rječnikom 8. Razlikovati načine sortiranja elemenata rječnika 9. Primijeniti rječnik u rješavanju programa u programskom jeziku Python
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Objasniti osobine skupa i rječnika kao zbirke podataka u Python-u Opisati razmjestaj, srodnost i način dohvaćanja elemenata u <i>skupu</i> i <i>rječniku</i>.</p> <p>2. Opisati operatore za rad sa skupovima Opisati operatore za rad sa skupovima ((presjek), & (unija), - (razlika), ^ (simetrična razlika), in (element skupa), not in (nije element), > (nadskup), < (podskup)) i funkciju set().</p> <p>3. Algoritam generiranja prostih brojeva do nekog zadanog broja Objasniti algoritam Eratostenovog sita primjenom skupa.</p> <p>4. Upotrijebiti skupove u osmišljavanju rješenja programa u Python-u Napisati program u programskom jeziku Python koji će ispisati koliko je ukupno različitih znakova u dvije unesene riječi.</p> <p>5. Napraviti rječnik</p>

	<p>U interaktivnom programskom okruženju Python Shell napraviti rječnik koji za ključ ima niz znakova, a za vrijednost cijeli broj.</p> <p>6. Ispisati elemente rječnika U interaktivnom programskom okruženju Python Shell za zadani rječnik ispisati sve elemente (vrijednosti) rječnika ili pojedinačne (prema zadanom ključu).</p> <p>7. Opisati funkcije i metode za rad s rječnikom Opisati funkcije <code>del ()</code> i <code>dict()</code> te metode <code>keys()</code>, <code>values()</code>, <code>items()</code>, <code>update()</code>, <code>clear()</code>.</p> <p>8. Razlikovati načine sortiranje elemenata rječnika Napisati funkciju za sortiranje (<code>sorted()</code>) koja će sortirati rječnik: a) po ključu; b) po vrijednosti.</p> <p>9. Primijeniti rječnik u rješavanju programa u programskom jeziku Python Napisati primjer programa koji će unijeti zadanu riječ te ispisati ukupan broj pojavljivanja različitih znakova u zadanoj riječi.</p>
--	--

Naziv	Datoteke
Razina	4
Obujam	0.2
Cilj/svrha jedinice ishoda učenja	Primjenjivati algoritme za izradu programa u Python-u koji će u svojim rješenjima za ulaz i izlaz podataka koristiti datoteku.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti ulogu datoteka u programu 2. Opisati korake za otvaranje datoteke u Python-u 3. Objasniti razlike u metodama za čitanje podatka iz datoteke u Python-u 4. Objasniti razlike u metodama za ispis podataka u datoteku u Python-u 5. Primijeniti metode za rad s datotekama u pisanju programa u programskom jeziku Python.
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Objasniti ulogu datoteka u programu Navesti ulogu trajnog pohranjivanja podataka na vanjsku memoriju računala i razmjenu informacija između programa.</p> <p>2. Objasniti metodu za otvaranje datoteke u Python-u Objasniti parametre funkcije koju koristimo za pristup datoteci .</p> <p>3. Objasniti razlike u metodama za čitanje podatka iz datoteke u Python-u Opisati metode read(), readline() i readlines().</p> <p>4. Objasniti razlike u metodama za ispis podataka u datoteku u Python-u Opisati metode write() i writelines().</p> <p>5. Primijeniti metode za rad s datotekama u pisanju programa u programskom jeziku Python. Napisati primjer programa koji podatke za obradu čita iz ulazne datoteke (<i>ulaz.txt</i>) te rezultate obrade (sve riječi koje počinju zadanim slovom) sprema u odgovarajuće retke izlazne datoteke (<i>izlaz.txt</i>).</p>

Naziv	Kornjačina grafika
Razina	4
Obujam	0.7
Cilj/svrha jedinice ishoda učenja	Primjenjivati grafičke funkcije i metode u oblikovanju rješenja u programskom jeziku Python.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti izgled grafičkog prozora 2. Opisati funkcije za gibanje pera 3. Primijeniti algoritam za crtanje pravilnog n-terokuta 4. Opisati metodu za crtanje kruga i kružnog isječka 5. Objasniti metode za postavljanje boje pera i ispune 6. Objasniti pojam linearne interpolacije boja 7. Navesti metode za upis i ispis teksta u grafičkom prozoru 8. Primijeniti grafičke funkcije u izradi programa u Pythonu
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<ol style="list-style-type: none"> 1. Objasniti izgled grafičkog prozora Navesti veličinu grafičkog prozora, označiti mjesto ishodišta unutar grafičkog prozora te imenovati modul u kojem se nalaze grafičke funkcije za crtanje. 2. Opisati funkcije za gibanje pera Opisati funkcije <code>pu()</code>, <code>pd()</code>, <code>rt()</code>, <code>lt()</code>, <code>fd()</code>, <code>bk()</code>, <code>goto(x,y)</code>, <code>setx(x)</code>, <code>sety(y)</code>. 3. Primijeniti algoritam za crtanje pravilnog n-terokuta Napisati program za crtanja n-terokuta duljine stranice <i>a</i>. 4. Opisati metodu za crtanje kruga i kružnog isječka Opisati parametre metode <code>circle()</code>. 5. Objasniti metode za postavljanje boje pera i ispune Opisati metode za definiranje boja: <code>colormode(r,g,b)</code>, <code>color(r,g,b)</code>, <code>pencolor(string)</code> i ispunu: <code>fillcolor(string)</code>, <code>begin_fill()</code>, <code>end_fill()</code>. 6. Objasniti pojam linearne interpolacije boja Grafički ilustrirati linearnu interpolaciju boja. 7. Navesti metode za upis i ispis teksta u grafičkom prozoru Navesti funkcije <code>textinput()</code> i <code>write()</code>. 8. Primjenjivati grafičke funkcije u izradi programa u Pythonu Napiši program koji će unositi polumjer kružnice <i>r</i> i crtati olimpijske kružnice.

Naziv	Objektno usmjereno programiranje
Razina	5
Obujam	0.4
Cilj/svrha jedinice ishoda učenja	Primijeniti objektno orijentirani pristup u sastavljanju rješenja programa programskom jeziku Python.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti strukturu objektno usmjerenog programa 2. Povezati ugrađene zbirke te funkcije i metode nad tim zbirkama u Python-u s objektno usmjerenim programiranjem 3. Objasniti konstruktor klase u Python-u 4. Dati primjer metode s dodatnim parametrima 5. Opisati specijalne metode za rad s objektima 6. Objasniti pojam <i>nasljeđivanja klase</i> 7. Primijeniti objektno usmjereni način programiranja u stvaranju algoritma i programa u programskom jeziku Python.
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Objasniti strukturu objektno usmjerenog programa Objasniti osnovne pojmove vezane uz objektno usmjereno programiranje (objekt, svojstvo, metoda, klasa).</p> <p>2. Povezati ugrađene zbirke te funkcije i metode nad tim zbirkama u Python-u s objektno usmjerenim programiranjem Objasniti povezanost zbirke <i>string</i> te njezinih funkcija i metoda sa objektno usmjerenim programiranjem.</p> <p>3. Objasniti konstruktor klase u Python-u Objasniti kako se u programu uz pomoć parametra <i>self</i> postavljaju svojstva klase na neku vrijednost.</p> <p>4. Dati primjer metode s dodatnim parametrima Napisati metodu klase koja će sadržavati jedan dodatni parametar.</p> <p>5. Opisati specijalne metode za rad s objektima Objasniti primjenu specijalnih metoda <code>__repr__()</code>, <code>__str__()</code>, <code>__add__()</code>, <code>__lt__()</code>, <code>__gt__()</code> u objektno usmjerenom programu u Python-u.</p> <p>6. Objasniti pojam <i>nasljeđivanja klase</i> Dati primjer definicije nove klase u Python-u koja nasljeđuje svojstvo i metodu iz postojeće klase.</p>

7. Primijeniti objektno usmjereni način programiranja i u stvaranju algoritma i programa u programskom jeziku Python.

Napisati primjer program sa zadanom klasom, pripadajućim svojstvima, zadanim metodama i objektima (računanje opsega/površine te crtanje i/ili ispunjavanje četverokuta/trokuta zadanom bojom).

Naziv	Algoritmi s rekurzijom
Razina	5
Obujam	0.3
Cilj/svrha jedinice ishoda učenja	Primjenjivati algoritme s rekurzivnim funkcijama u sastavljanju rješenja programa u programskom jeziku Python.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti induktivni pristup rješavanju problema 2. Objasniti pojam rekurzije 3. Izvesti rekurzivnu relaciju i uvjet prekida za zadani problem 4. Objasniti pojam <i>memoizacija</i> 5. Primjenjivati algoritme s rekurzijama u rješavanju programa u Python-u
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Objasniti induktivni pristup rješavanju problema Opisati način rješavanje problema od objekta manjih dimenzija prema objektu većih dimenzija na primjeru programa zbrajanja prvih n prirodnih brojeva.</p> <p>2. Objasniti pojam rekurzije Opisati rekurziju kao način rješavanja problema od objekata većih dimenzija prema objektima manjih dimenzija, odnosno postupak kod kojeg se za rješenje jednog stanja koriste rješenja drugih stanja.</p> <p>3. Izvesti rekurzivnu relaciju i uvjet prekida za zadanu funkciju Napisati korake računanja rekurzivne funkcije za problem množenja prvih n prirodnih brojeva.</p> <p>4. Objasniti pojam <i>memoizacija</i> Na primjeru Fibonaccijeva niza objasniti pamćenje višestrukih rekurzivnih postupaka.</p> <p>5. Primjenjivati algoritme s rekurzijama u rješavanju programa u Python-u Napisati rekurzivnu funkciju koja za neki zadani red računa (zbroj i/ili umnožak) prvih n elemenata.</p>

Naziv	Ocjena složenosti algoritma
Razina	5
Obujam	0.2
Cilj/svrha jedinice ishoda učenja	Vrednovati algoritme pretraživanja ocjenom složenosti.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti pojam složenosti algoritma 2. Kategorizirati složenost algoritama prema notaciji <i>veliko O</i> 3. Objasniti algoritam slijednog traženja u listi s nepoređanim elementima 4. Objasniti algoritam traženja elementa u poredanoj (sortiranoj) listi 5. Usporediti vremena traženja elementa u nepoređanoj i poredanoj listi
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Objasniti pojam složenosti algoritma Objasniti utjecaj broja algoritamskih operacija na složenost algoritma.</p> <p>2. Kategorizirati složenost algoritama prema notaciji <i>veliko O</i> Navedi kategorije složenosti algoritma: konstantna složenost, linearna složenost, kvadratna složenost, kubna složenost, polinom sa složenošću četvrtog reda.</p> <p>3. Objasniti algoritam slijednog traženja u listi s nepoređanim elementima Opisati algoritam slijednog traženja elementa u nepoređanoj listi.</p> <p>4. Objasniti algoritam traženja elementa u poredanoj (sortiranoj) listi Ilustrirati korake binarnog traženja za niz od 10 prirodnih brojeva .</p> <p>5. Usporediti vremena traženja elementa u nepoređanoj i poredanoj listi Valorizirati pretraživanje liste na temelju mjerenja vremena izvođenja programa koji koristi algoritam slijednog traženja elementa u nesortiranoj listi i programa koji koristi algoritam binarnog traženja u sortiranoj listi.</p>

Naziv	Apstraktne strukture podataka
Razina	5
Obujam	0.7
Cilj/svrha jedinice ishoda učenja	Upoznati osnovne apstraktne strukture podataka, osnovne algoritme nad tim strukturama te primijeniti strukturu za rješenje konkretnog problema.
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti strukturu stoga (stack) 2. Primijeniti osnovne operacije nad stogom 3. Izračunati vrijednost aritmetičkog izraza zadano u infix, postfix ili prefix notaciji 4. Objasniti strukturu reda (queue) 5. Primijeniti osnovne operacije nad redom 6. Opisati pojmove vezane uz strukturu - stablo (tree) 7. Primijeniti algoritme obilaska stabla (preorder, inorder, postorder) na konkretnom primjerku stabla 8. Rekonstruirati binarno stablo traženja na osnovu zadane riječi 9. Rekonstruirati binarnu hrpu na osnovu zadane riječi
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<ol style="list-style-type: none"> 1. Objasniti strukturu stoga (stack) Objasniti razlog zbog kojeg se struktura stoga zove još i LIFO. 2. Primijeniti osnovne operacije nad stogom Za zadano stanje na stogu te niz operacija nad stogom (push i pop) odrediti koji će elementi ostati na stogu. 3. Izračunati vrijednost aritmetičkog izraza zadano u infix, postfix ili prefix notaciji Za izraz zadan u prefix notaciji odrediti pripadni izraz u infix notaciji. 4. Objasniti strukturu reda (queue) Navesti primjer reda iz svakodnevnog života te obrazložiti vezu sa strukturom reda. 5. Primijeniti osnovne operacije nad redom Za zadano stanje na redu te niz operacija nad redom (enqueue i dequeue) odrediti koji će vrijednosti biti skinute s reda. 6. Opisati pojmove vezane uz strukturu- stablo (tree) Za zadano stablo odrediti broj unutarnjih čvorova. 7. Primijeniti algoritme obilaska stabla (preorder, inorder,

	<p>postorder) na konkretnom primjerku stabla Za zadano stablo ispisati čvorove onim redoslijedom kako će biti obiđeni metodom preorder.</p> <p>8. Rekonstruirati binarno stablo traženja na osnovu zadane riječi Za zadanu riječ kreirati binarno stablo traženja.</p> <p>9. Rekonstruirati binarnu hrpu na osnovu zadane riječi Na zadanu hrpu dodati slova neke riječi tako da se zadrži struktura hrpe.</p>
--	--

Naziv	Algoritmi sortiranja
Razina	5
Obujam	0.4
Cilj/svrha jedinice ishoda učenja	Valorizirati algoritme sortiranja .
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Opisati algoritam sortiranja zamjenom elemenata (exchange sort) 2. Opisati algoritam izbora najmanjeg elementa (selection sort) 3. Opisati algoritam zamjene susjednih elemenata (bubble sort) 4. Opisati algoritam umetanjem (insertion sort) 5. Opisati algoritam sjedinjavanja (merge sort) 6. Opisati algoritam brzog sortiranja (quick sort) 7. Opisati algoritam razvrstavanjem (bucket sort) 8. Usporediti algoritme sortiranja
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<ol style="list-style-type: none"> 1. Opisati algoritam sortiranja zamjenom elemenata (exchange sort) Ilustrirati korake sortiranja za algoritam zamjene na primjeru liste od 10 elemenata. 2. Opisati algoritam izbora najmanjeg elementa (selection sort) Usporediti algoritam izbor najmanjeg elementima s algoritmom zamjene elemenata. 3. Opisati algoritam zamjene susjednih elemenata (bubble sort) Ilustrirati korake sortiranja za algoritam zamjene susjednih elemenata na primjeru liste od 10 elemenata. 4. Opisati algoritam umetanjem (insertion sort) Ilustrirati korake sortiranja za algoritma umetanjem na primjeru listu od 10 elemenata. 5. Opisati algoritam sjedinjavanja (merge sort) Ilustrirati korake sortiranja za algoritam sjedinjavanja na primjeru liste od 10 elemenata. 6. Opisati algoritam brzog sortiranja (quick sort) Ilustrirati korake sortiranja za algoritam brzog sortiranja po Hoareu na

	<p>primjeru liste od 10 elemenata.</p> <p>7. Opisati algoritam razvrstavanjem (bucket sort) Opisati korake sortiranja za algoritam razvrstavanjem koristeći listu i rječnik kao zbirke podataka.</p> <p>8. Usporediti algoritme sortiranja Valorizirati algoritme sortiranja ocjenom složenosti.</p>
--	--

Modul III

Naziv predmeta u trećoj godini učenja	INFORMATIKA
Kroz ovaj predmet u trećoj godini učenja učenik će steći sljedeće ishode učenja:	<p>Kriptografija</p> <ol style="list-style-type: none"> 1. Objasniti osnovne pojmove vezane uz kriptografiju (otvoreni tekst, kriptiranje, kriptirani tekst,...) 2. Primijeniti Cezarovo kriptiranje 3. Primijeniti kriptiranje s pomakom 4. Primijeniti osnovne operacije u grupama Z_n 5. Primijeniti Vigenéovo kriptiranje 6. Primijeniti transpozicijsko kriptiranje 7. Opisati postupak kriptiranja s javnim ključem 8. Primijeniti RSA algoritam kriptiranja 9. Opisati digitalni potpis <p>Programi s grafičkim korisničkim sučeljima</p> <ol style="list-style-type: none"> 1. Opisati klase za osnovne grafičkog korisničkog sučelja 2. Opisati postupak dodavanja događaja na gumbu na grafičkom korisničkom sučelju 3. Objasniti postupak uzimanja unesenih podataka iz polja za unos na grafičkom korisničkom sučelju 4. Napraviti aplikaciju s nekoliko osnovnih elemenata grafičkog korisničkog sučelja te nekoliko događaja 5. Primijeniti klase za izradu okruglog i kvadratnog gumba na grafičkom korisničkom sučelju 6. Primijeniti klasu za izradu izbornika na grafičkom korisničkom sučelju 7. Primijeniti klasu za kreiranja liste elemenata na grafičkom korisničkom sučelju 8. Primijeniti svojstva i metoda klase Text za oblikovanje teksta unutar područja za tekst 9. Primijeniti klasu za stvaranje klizne trake na grafičkom korisničkom sučelju 10. Primijeniti metode klase Canvas za crtanje jednostavnih oblika na crtačem platnu <p>Baze podataka</p> <ol style="list-style-type: none"> 1. Objasniti osnovne pojmove vezane uz baze podataka (entitet, atribut, model, primarni i strani ključ,...) 2. Objasniti tipove relacija među entitetima 3. Napisati tablicu sa zadanim atributima, tipovima atributa te primarnim ključem 4. Napisati SQL upit za dodavanje podataka u zadanu tablicu 5. Napisati SQL upit za izmjenu određenih podataka unutar tablice

	<p>6. Napisati SQL upit za brisanje podataka iz tablice prema zadanom kriteriju</p> <p>7. Napisati SQL upit za selektiranje podataka iz jedne ili više tablice prema zadanom kriteriju</p> <p>8. Osmisliti bazu podataka s potrebnim entitetima, atributima, primarnim i stranim ključevima,... za zadani problem</p> <p>9. Integrirati bazu podataka s programskim jezikom Python</p> <p>Mrežno programiranje</p> <p>1. Objasniti osnovne pojmove vezane uz mrežnu komunikaciju (osnovni protokoli, elementi paketa, portovi,...)</p> <p>2. Objasniti pojam socketa</p> <p>3. Nabrojiti osnove naredbe Pythona za mrežno programiranje</p> <p>4. Upotrijebiti osnovne metode klase Socket</p> <p>5. Napisati jednostavnu klijent poslužitelj aplikaciju u Pythonu</p> <p>Višedretveno programiranje</p> <p>1. Identificirati potrebu za paralelnim izvođenjem dvaju ili više dijelova programa istovremeno</p> <p>2. Objasniti pojam dretve (threada)</p> <p>3. Objasniti osnovna svojstva i metode klase Thread</p> <p>4. Napraviti program u kojem se uz glavni program pokreće još jedna ili više dretvi</p> <p>5. Osmisliti višeprosorsko rješenje jednostavnog problema</p> <p>6. Ustanoviti vremensku razliku u izvođenju zadanog algoritma realiziranog u obliku klasičnog i višedretvenog programa</p>
Razrada	
<i>Nastavne cjeline</i>	<i>Nastavne teme</i>
Kriptografija	<p>Osnove kriptiranja</p> <p>Kriptiranje s pomakom</p> <p>Vigenéovo kriptiranje</p> <p>Transpozicijsko kriptiranje</p> <p>RSA algoritam</p> <p>Digitalni potpis</p>
Programi s grafičkim korisničkim sučeljima	<p>Osnovni elementi grafičkog sučelja</p> <ul style="list-style-type: none"> - prozor (Tk()) - ispis teksta (Label()) - okvir (Frame()) - gumb (Button()) <p>Metode place(), grid() i pack()</p> <p>Klasa Entry() i metoda get()</p> <p>Događaji na gumbu</p> <p>Složeni elementi grafičkog sučelja</p> <ul style="list-style-type: none"> - Okrugli gumb (Radiobutton ()) - Kvadratni gumb (Checkbutton()) - Izbornici (Menu())-

	<ul style="list-style-type: none"> - Lista (Listbox()) - Tekstni okvir (Tekstbox()) - Klizna traka (Scrollbar()) - Crtanje oblika (Canvas())
Baze podataka	<p>Uvod u baze podataka Relacije među entitetima u bazi Osnovni upiti u bazi Složeni upiti</p>
Mrežno programiranje	<p>Osnovni pojmovi mrežne komunikacije Naredbe za mrežnu komunikaciju</p>
Višedretveno programiranje	<p>Paralelno programiranje Pojam dretve Pokretanje više dretvi</p>
Okruženje i uvjeti za organizirano učenje	<p>U cilju ostvarivanja navedenih ishoda učenja nužno je tijekom cijelog nastavnog procesa polazniku omogućiti korištenje računala u specijaliziranoj informatičkoj učionici s najviše 16 računala i grupom od najviše 16 polaznika, po principu za jednim računalom jedan polaznik.</p> <p>Razredni odjel koje ima više od 16 polaznika mora se dijeliti na grupe kako bi se zadovoljio gornji kriterij.</p> <p>U slučaju da nije moguće cjelokupnu nastavu izvoditi u specijaliziranoj informatičkoj učionici u grupi po 16 polaznika, moguće je 50% nastave izvoditi u učionici opremljenoj s jednim računalom s projektorom i pristupom na internet te 50% u specijaliziranoj informatičkoj učionici s najviše 16 računala i grupom od najviše 16 računala, po principu za jednim računalom jedan polaznik.</p>
Ostalo	
Metode i oblici rada:	<p>Metode: predavačka metoda, metoda dijaloga, heuristička metoda, problemska metoda, programirana metoda, demonstracijska metoda i istraživačka metoda.</p> <p>Oblici: frontalni oblik nastave, diferencirana nastava, individualizirana nastava, problemska nastava, programirana nastava, egzemplarna nastava, mentorska nastava, demonstracijska nastava.</p> <p>Napomena: izbor metoda i oblika rada za svaki nastavni sat određuje nastavnik, sukladno nastavnim sadržajima, osobitosti polaznika te materijalnim i drugim uvjetima.</p>
Elementi i oblici praćenja i vrjednovanja polaznika	<p>Elementi: usvojenost sadržaja (usmenim i pisanim načinom), primjena znanja (usmenim, pisanim načinom, provjera znanja i usvojenosti vještina na računalu, projektni zadatak).</p> <p>Oblici: samostalan rad (domaće zadaće, istraživanje, suradnja u nastavi i dr.).</p>

Literatura za učenike:	Prema Katalogu odobrenih udžbenika i drugih pomoćnih nastavnih sredstava Ministarstva znanosti, obrazovanja i sporta.
------------------------	---

**Razrada jedinica ishoda učenja za Modul III
OBUJAM(2.5)**

Naziv	<i>Kriptografija</i>
Razina	5
Obujam	0.3
Cilj/svrha jedinice ishoda učenja	Primijeniti osnovne kriptografske algoritme
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti osnovne pojmove vezane uz kriptografiju (otvoreni tekst, kriptiranje, kriptirani tekst,...) 2. Primijeniti Cezarovo kriptiranje 3. Primijeniti kriptiranje s pomakom 4. Primijeniti osnovne operacije u grupama Z_n 5. Primijeniti Vigenéovo kriptiranje 6. Primijeniti transpozicijsko kriptiranje 7. Opisati postupak kriptiranja s javnim ključem 8. Primijeniti RSA algoritam kriptiranja 9. Opisati digitalni potpis
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Objasniti osnovne pojmove vezane uz kriptografiju (otvoreni tekst, kriptiranje, kriptirani tekst,...) Objasniti pojam kriptanalize.</p> <p>2. Primijeniti Cezarovo kriptiranje Za zadani otvoreni tekst ispisati kriptirani primjenjujući Cezarovo kriptiranje.</p> <p>3. Primijeniti kriptiranje s pomakom Za zadani otvoreni tekst s te prirodan broj n odrediti tekst koji se dobije kriptiranjem teksta s ključem n, pri čemu se tekst kriptira kriptiranjem s pomakom.</p> <p>4. Primijeniti osnovne operacije u grupama Z_n Za zadani element k grupe Z_n odrediti njegov multiplikativni inverz.</p> <p>5. Primijeniti Vigenéovo kriptiranje Odrediti tekst koji se dobije kriptiranjem zadanog teksta (s)</p>

	<p>Vigenéovim kriptiranjem, pri čemu je ključ kriptiranja k.</p> <p>6. Primijeniti transpozicijsko kriptiranje Odrediti tekst koji će se dobiti kriptiranjem otvorenog teksta (s) transpozicijskim kriptiranjem pri čemu je ključ kriptiranja lista od n elemenata.</p> <p>7. Opisati postupak kriptiranja s javnim ključem Opisati postupak kriptiranja poruke javnim ključem primatelja te dekriptiranja tako kriptirane poruke.</p> <p>8. Primijeniti RSA algoritam kriptiranja Za zadani otvoreni tekst (broj) odrediti pripadni kriptirani tekst ako se za kriptiranje koristi RSA čiji su parametri zadani.</p> <p>9. Opisati digitalni potpis Opisati postupak kreiranja digitalno potpisane poruke te provjere autentičnosti.</p>
--	--

Naziv	Programi s grafičkim korisničkim sučeljima
Razina	5
Obujam	1
Cilj/svrha jedinice ishoda učenja	Primijeniti koncepte izrade potpuno funkcionalnih programa s grafičkim korisničkim sučeljem
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Opisati klase za osnovne grafičkog korisničkog sučelja 2. Opisati postupak dodavanja događaja na gumbu na grafičkom korisničkom sučelju 3. Objasniti postupak uzimanja unesenih podataka iz polja za unos na grafičkom korisničkom sučelju 4. Napraviti aplikaciju s nekoliko osnovnih elemenata grafičkog korisničkog sučelja te nekoliko događaja 5. Primijeniti klase za izradu okruglog i kvadratnog gumba na grafičkom korisničkom sučelju 6. Primijeniti klasu za izradu izbornika na grafičkom korisničkom sučelju 7. Primijeniti klasu za kreiranja liste elemenata na grafičkom korisničkom sučelju 8. Primijeniti svojstva i metoda klase Text za oblikovanje teksta unutar područja za tekst 9. Primijeniti klasu za stvaranje klizne trake na grafičkom korisničkom sučelju 10. Primijeniti metode klase Canvas za crtanje jednostavnih oblika na crtaćem platnu
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<ol style="list-style-type: none"> 1. Opisati klase za osnovne grafičkog korisničkog sučelja Opisati klase za osnovne elemente grafičkog korisničkog sučelja (Tk(), Frame(), Label(), Button()) 2. Opisati postupak dodavanja događaja na gumbu na grafičkom korisničkom sučelju Napraviti gumb <i>b</i> koji će imati svojstvo da će se u trenutku kada korisnik na njega klikne mišem pozvati kreirana funkcija <i>f()</i>. 3. Objasniti postupak uzimanja unesenih podataka iz polja za unos na grafičkom korisničkom sučelju

	<p>Objasniti vezu klase Entry() s metodom get()</p> <p>4. Napraviti aplikaciju s nekoliko osnovnih elemenata grafičkog korisničkog sučelja te nekoliko događaja Napraviti program s korisničkim grafičkim sučeljem koji će sadržavati 2 polja za unos teksta, jedan gumb te okvir za ispis teksta (labelu). Nakon što korisnik unese u polja za tekst brojeve te klikne na gumb u okviru za ispis teksta (labeli) se treba pojaviti zbroj unesenih brojeva.</p> <p>5. Primijeniti klase za izradu okruglog i kvadratnog gumba na grafičkom korisničkom sučelju Napraviti program (kviz sa tri pitanja) u kojem će se za kreiranje pitanja primijeniti klase Radiobutton() i Checkbutton().</p> <p>6. Primijeniti klasu za izradu izbornika na grafičkom korisničkom sučelju Napraviti program koji će sadržavati 2 izbornika s 2 podizbornika u svakom.</p> <p>7. Primijeniti klasu za kreiranja liste elemenata na grafičkom korisničkom sučelju Napraviti program s listom (listbox) <i>b</i>, koja će između ostaloga imati funkcionalnost da korisnik kada promijeni selektiranu stavku da se pozove kreirana funkcija <i>f()</i>.</p> <p>8. Primijeniti svojstva i metoda klase Text za oblikovanje teksta unutar područja za tekst Napraviti program koji će korisniku omogućiti oblikovanje odabranog teksta unutar područja za tekst (promjena boje teksta, oblika, veličine i stila znakova)</p> <p>9. Primijeniti klasu za stvaranje klizne trake na grafičkom korisničkom sučelju Napraviti program koji će unutar textboxa imati okomitu i/ili vodoravnu kliznu traku (scrollbar).</p> <p>10. Primijeniti metode klase Canvas za crtanje jednostavnih oblika na crtaćem platnu Napraviti program unutar kojeg će korisnik na crtaćem platnu koristeći funkcije miša (click i drag and drop) nacrtati kvadrat.</p>
--	---

Naziv	Baze podataka
Razina	5
Obujam	0.5
Cilj/svrha jedinice ishoda učenja	Primijeniti osnove rada s bazama podataka
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti osnovne pojmove vezane uz baze podataka (entitet, atribut, model, primarni i strani ključ,...) 2. Objasniti tipove relacija među entitetima 3. Napisati tablicu sa zadanim atributima, tipovima atributa te primarnim ključem 4. Napisati SQL upit za dodavanje podataka u zadanu tablicu 5. Napisati SQL upit za izmjenu određenih podataka unutar tablice 6. Napisati SQL upit za brisanje podataka iz tablice prema zadanom kriteriju 7. Napisati SQL upit za selektiranje podataka iz jedne ili više tablice prema zadanom kriteriju 8. Osmisliti bazu podataka s potrebnim entitetima, atributima, primarnim i stranim ključevima,... za zadani problem 9. Integrirati bazu podataka s programskim jezikom Python
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Objasniti osnovne pojmove vezane uz baze podataka (entitet, atribut, model, primarni i strani ključ,...) Navedi neke od atributa za zadani entitet (primjeri entiteta mogu biti učenik, razredno odjeljenje,...).</p> <p>2. Objasniti tipove relacija među entitetima Objasniti relaciju 1 : n.</p> <p>3. Napisati tablicu sa zadanim atributima, tipovima atributa te primarnim ključem Napisati SQL upit kojim će se kreirati <i>Ucenici</i> tablicu koja će imati sljedeće stupce:</p> <ul style="list-style-type: none"> • <i>ID</i> – autonumber • <i>ime</i> – text(20) • <i>prezime</i> – text(20) <p>pri čemu će ID biti primarni ključ.</p>

4. Napisati SQL upit za dodavanje podataka u zadanu tablicu

Napisati SQL upit kojim će se u tablicu *Ucenici* čiji su stupci: *ID*, *ime* i *prezime*, pri čemu je *ID* autonumber dodati učenica *Marija Kovačević*.

5. Napisati SQL upit za izmjenu određenih podataka unutar tablice

Napisati SQL upit kojim će se u tablici *Ucenici* čiji su stupci: *ID*, *ime* i *prezime*, svim učenicima čije je ime *Stjepan* promijeniti prezime u *Perić*.

6. Kreirati SQL upit za brisanje podataka iz tablice prema zadanom kriteriju

Napisati SQL upit kojim će se u tablici *Ucenici* čiji su stupci: *ID*, *ime* i *prezime*, obrisati svi učenike čije ime započinje slovom *P*.

7. Napraviti SQL upit za selektiranje podataka iz jedne ili više tablice prema zadanom kriteriju

Zadane su tablice:

- *Ucenici*:
 - *ID* – autonumber
 - *ime* – text(20)
 - *prezime* – text(20)
 - *IDRazreda* – number
- *Razredi*:
 - *IDRazreda* – autonumber
 - *oznaka* – text(2)

Pri čemu je *IDRazreda* u tablici *Ucenici* stvarni ključ čiji je pripadni primarni ključ stupac *IDRazreda* u tablici *Razredi*. Napiši upit kojim ćeš ispisati imena i prezimena svih učenika 2C razreda.

8. Osmisliti bazu podataka s potrebnim entitetima, atributima, primarnim i stranim ključevima,... za zadani problem

Osmisliti bazu podataka u kojoj bi se pohranjivali podaci o učenicima i razrednim odjeljenjima koje pohađaju. Baza se treba sastojati od najmanje dvije tablice.

9. Integrirati bazu podataka s programskim jezikom Python

Napraviti program u Pythonu koji će se spojiti na bazu unutar koje se između ostalih nalazi i tablica *Ucenici* čija je struktura:

- *ID* – autonumber
- *ime* – text(20)
- *prezime* – text(20)

Nakon što se spoji na bazu program treba od korisnika tražiti unos imena i prezimena nekog učenika te tog učenika dodati u bazu podataka.

Naziv	Mrežno programiranje
Razina	5
Obujam	0.3
Cilj/svrha jedinice ishoda učenja	Objasniti osnovne pojmove mrežnog programiranja te izrađivati osnovne mrežne aplikacije
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti osnovne pojmove vezane uz mrežnu komunikaciju (osnovni protokoli, elementi paketa, portovi,...) 2. Objasniti pojam socketa 3. Nabrojiti osnove naredbe Pythona za mrežno programiranje 4. Upotrijebiti osnovne metode klase Socket 5. Napisati jednostavnu klijent poslužitelj aplikaciju u Pythonu
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<ol style="list-style-type: none"> 1. Objasniti osnovne pojmove vezane uz mrežnu komunikaciju (osnovni protokoli, elementi paketa, portovi,...) Imenovati broj portova na računalu. 2. Objasniti pojam socketa Objasniti što predstavlja socket u mrežnoj komunikaciji. 3. Nabrojiti osnove funkcije Pythona za mrežno programiranje Objasni naredbu <code>gethostbyname(name)</code>. 4. Upotrijebiti osnovne metode klase Socket Za zadani objekt s tipa Socket napisati naredbu kojom će se na socket s upisati poruka <i>Kako si</i>. 5. Napisati jednostavnu klijent poslužitelj aplikaciju u Pythonu Napisati serversku aplikaciju koja će čekati klijenta na zadanom portu. Nakon što se klijent spoji, klijent traži unos jedne riječi (ime). Klijent šalje poslužitelju ime, nakon čega poslužitelj klijentu šalje ukupni broj slova te riječi. Na kraju klijent ispisuje broj slova na ekran.

Naziv	Višedretveno programiranje
Razina	5
Obujam	0.4
Cilj/svrha jedinice ishoda učenja	Objasniti osnovne pojmove višedretvenog programiranja te izrađivati jednostavnije višedretvene aplikacije
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Identificirati potrebu za paralelnim izvođenjem dvaju ili više dijelova programa istovremeno 2. Objasniti pojam dretve (threada) 3. Objasniti osnovna svojstva i metode klase Thread 4. Napraviti program u kojem se uz glavni program pokreće još jedna ili više dretvi 5. Osmisliti višeprosorsko rješenje jednostavnog problema 6. Ustanoviti vremensku razliku u izvođenju zadanog algoritma realiziranog u obliku klasičnog i višedretvenog programa
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<ol style="list-style-type: none"> 1. Identificirati potrebu za paralelnim izvođenjem dvaju ili više dijelova programa istovremeno Objasniti problem na koji nailazimo kada želimo implementirati poslužiteljsku aplikaciju koja će istovremeno moći prihvaćati nove klijente i komunicirati s postojećim klijentima. 2. Objasniti pojam dretve (threada) Objasniti pojam dretve u programu 3. Objasniti osnovna svojstva i metode klase Thread Objasniti metodu start() klase Thread 4. Napraviti program u kojem se uz glavni program pokreće još jedna ili više dretvi Napisati program koji će pokrenuti dvije dretve. Prva dretva će izvoditi definiranu funkciju f(), dok će druga izvoditi definiranu funkciju g()). 5. Osmisliti višeprosorsko rješenje jednostavnog problema Implementirati višeprosorsko rješenje za generiranje svih prostih brojeva do zadanog broja n. 6. Ustanoviti vremensku razliku u izvođenju zadanog algoritma realiziranog u obliku klasičnog i višedretvenog programa

	Na primjeru programa za prebrojavanje prostih brojeva ustanoviti razliku u trajanju standardne i višedretvene implementacije.
--	---

Dodatne teme s ishodima učenja

Naziv predmeta za dodatne teme	INFORMATIKA
Kroz ove dodatne teme učenja učenik će steći sljedeće ishode učenja:	<p>Kombinatorni algoritmi</p> <ol style="list-style-type: none"> 1. Obrazložiti razliku između standardne i generatorske funkcije 2. Primijeniti postupak generiranja svih interpretacija za n logičkih varijabli 3. Primijeniti Grayev kod za generiranje sljedeće interpretacije logičke funkcije 4. Analizirati rekurzivni algoritam generiranja svih permutacija skupa od n elemenata 5. Primijeniti postupak generiranja permutacija u leksikografskom poretku 6. Opisati postupak generiranja svih podskupova zadanog skupa od n elemenata 7. Analizirati generatorsku funkciju koja za zadani skup od n elemenata generira sve kombinacije duljine k 8. Opisati postupak generiranja svih particija zadanog broja n 9. Usporediti brzine izvođenja polinomske i NP složenosti <p>Grafovi</p> <ol style="list-style-type: none"> 1. Objasniti osnovne pojmove vezane uz graf 2. Napraviti matricu i listu susjedstva za dani graf 3. Opisati osnovne operacije za rad s matricama 4. Primijeniti obilasci grafa 5. Demonstrirati postupak kreiranja topološki sortiranog grafa za zadani graf 6. Primjenjivati Dijkstrin algoritam
Razrada	
<i>Nastavne cjeline</i>	<i>Nastavne teme</i>
Kombinatorni algoritmi	Standardne i generatorske funkcije Grayev kod Permutacije
Grafovi	Osnovni pojmovi Operacije nad matricama Obilasci grafa Dijkstra algoritam
Okruženje i uvjeti za organizirano učenje	U cilju ostvarivanja navedenih ishoda učenja nužno je tijekom cijelog nastavnog procesa polazniku omogućiti korištenje računala u specijaliziranoj informatičkoj učionici s najviše 16 računala i grupom od najviše 16 polaznika, po principu za jednim računalom jedan polaznik.

	<p>Razredni odjel koje ima više od 16 polaznika mora se dijeliti na grupe kako bi se zadovoljio gornji kriterij.</p> <p>U slučaju da nije moguće cjelokupnu nastavu izvoditi u specijaliziranoj informatičkoj učionici u grupi po 16 polaznika, moguće je 50% nastave izvoditi u učionici opremljenoj s jednim računalom s projektorom i pristupom na internet te 50% u specijaliziranoj informatičkoj učionici s najviše 16 računala i grupom od najviše 16 računala, po principu za jednim računalom jedan polaznik.</p>
Ostalo	
Metode i oblici rada:	<p>Metode: predavačka metoda, metoda dijaloga, heuristička metoda, problemska metoda, programirana metoda, demonstracijska metoda i istraživačka metoda.</p> <p>Oblici: frontalni oblik nastave, diferencirana nastava, individualizirana nastava, problemska nastava, programirana nastava, egzemplarna nastava, mentorska nastava, demonstracijska nastava.</p> <p>Napomena: izbor metoda i oblika rada za svaki nastavni sat određuje nastavnik, sukladno nastavnim sadržajima, osobitosti polaznika te materijalnim i drugim uvjetima.</p>
Elementi i oblici praćenja i vrjednovanja polaznika	<p>Elementi: usvojenost sadržaja (usmenim i pisanim načinom), primjena znanja (usmenim, pisanim načinom, provjera znanja i usvojenosti vještina na računalu, projektni zadatak).</p> <p>Oblici: samostalan rad (domaće zadaće, istraživanje, suradnja u nastavi i dr.).</p>
Literatura za učenike:	Prema Katalogu odobrenih udžbenika i drugih pomoćnih nastavnih sredstava Ministarstva znanosti, obrazovanja i sporta.

Razrada jedinica ishoda učenja (Dodatne teme)
OBUJAM(0.9)

Kôd	
Naziv	<i>Kombinatorni algoritmi</i>
Razina	5
Obujam	0.4
Cilj/svrha jedinice ishoda učenja	Usvojiti osnovne kombinatorne strukturama te način njihova generiranja pomoću računala
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Obrazložiti razliku između standardne i generatorske funkcije 2. Primijeniti postupak generiranja svih interpretacija za n logičkih varijabli 3. Primijeniti Grayev kod za generiranje sljedeće interpretacije logičke funkcije 4. Analizirati rekurzivni algoritam generiranja svih permutacija skupa od n elemenata 5. Primijeniti postupak generiranja permutacija u leksikografskom poretku 6. Opisati postupak generiranja svih podskupova zadanog skupa od n elemenata 7. Analizirati generatorsku funkciju koja za zadani skup od n elemenata generira sve kombinacije duljine k 8. Opisati postupak generiranja svih particija zadanog broja n 9. Usporediti brzine izvođenja polinomske i NP složenosti
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrjednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<p>1. Obrazložiti razliku između generatora i funkcije Obrazložiti razliku između funkcije koja vraća listu svih parnih brojeva do n te generatorske funkcije koja radi isto.</p> <p>2. Primijeniti postupak generiranja svih interpretacija za n logičkih varijabli Odrediti sve interpretacije logičke funkcije s n logičkih varijabli.</p> <p>3. Primijeniti Grayev kod za generiranje sljedeće interpretacije</p>

	<p>logičke funkcije Za zadanu interpretaciju odrediti interpretaciju koja dolazi nakon nje prema Grayevom kôdu.</p> <p>4. Analizirati rekurzivni algoritam generiranja svih permutacija skupa od n elemenata Na konkretnom primjeru opisati rekurzivni algoritam za generiranje svih permutacija.</p> <p>5. Primijeniti postupak generiranja permutacija u leksikografskom poretku Za zadanu permutaciju odrediti permutaciju koja leksikografski dolazi neposredno nakon nje.</p> <p>6. Opisati postupak generiranja svih podskupova zadanog skupa od n elemenata Za zadani skup od n elemenata odrediti sve njegove podskupove prema njihovim rednim brojevima.</p> <p>7. Analizirati generatorsku funkciju koja za zadani skup od n elemenata generira sve kombinacije duljine k Za n-teročlani skup odrediti sve njegove podskupove duljine k.</p> <p>8. Opisati postupak generiranja svih particija zadanog broja n Za zadani prirodan broj n pronaći sve njegove particije.</p> <p>9. Usporediti brzine izvođenja polinomske i NP složenosti Obrazložiti prosječno vrijeme izvođenja algoritma sortiranja i algoritma generiranja svih permutacija liste čiji su elementi n različitih prirodnih brojeva.</p>
--	---

Naziv	Grafovi
Razina	5
Obujam	0.5
Cilj/svrha jedinice ishoda učenja	Usvojiti osnovne pojmove vezane uz graf te standardne algoritme za rad s grafovima
Ishodi učenja (5-10)	<ol style="list-style-type: none"> 1. Objasniti osnovne pojmove vezane uz graf 2. Napraviti matricu i listu susjedstva za dani graf 3. Opisati osnovne operacije za rad s matricama 4. Primijeniti obilasku grafa 5. Demonstrirati postupak kreiranja topološki sortiranog grafa za zadani graf 6. Primjenjivati Dijkstrin algoritam
Uvjeti u kojima se stječu kompetencije	<i>Uvjeti u kojima se stječu kompetencije propisani su Državnim pedagoškom standardom srednjoškolskog sustava odgoja i obrazovanja (NN 63/08 i 90/10)</i>
Provjera i vrednovanje	<i>Provjera i vrjednovanje propisani su Zakonom o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12) i Pravilnikom o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10).</i>
Primjer provjere i vrednovanja	<ol style="list-style-type: none"> 1. Objasniti osnovne pojmove vezane uz graf Za zadani graf odrediti stupnjeve svih čvorova. 2. Napraviti matricu i listu susjedstva za dani graf Za zadani graf napraviti matricu i listu susjedstva. 3. Opisati osnovne operacije za rad s matricama Za dvije zadane matrice odrediti njihov umnožak. 4. Primijeniti obilasku grafa Za zadani graf te neki vrh tog grafa odrediti redoslijed posjećivanja vrhova širinskim obilaskom grafa. 5. Demonstrirati postupak kreiranja topološki sortiranog grafa za zadani graf Za zadani topološki nesortirani graf odrediti pripadni topološki sortirani graf. 6. Primijeniti Dijkstrin algoritam Za zadani težinski graf te jedan njegov čvor odrediti udaljenost od tog čvora do svih ostalih čvorova.

Dokumenti:

- 1) *Metodologija za razvoj strukovnih standarda zanimanja, kvalifikacija i kurikuluma; Agencija za strukovno obrazovanje i obrazovanje odraslih, Zagreb, prosinac 2011.*
- 2) *Priručnik za razvoj strukovnih standarda zanimanja, kvalifikacija i kurikuluma; Agencija za strukovno obrazovanje i obrazovanje odraslih, Zagreb, prosinac 2011.*
- 3) *Nacionalni okvirni kurikulum za predškolski odgoj i obrazovanje te opće obvezno i srednjoškolsko obrazovanje, Ministarstvo znanosti, obrazovanja i športa, Zagreb, srpanj 2010.*
- 4) *HRVATSKI KVALIFIKACIJSKI OKVIR Uvod u kvalifikacije; Vlada Republike Hrvatske Ministarstvo znanosti, obrazovanja i športa, Zagreb, kolovoz 2009.*
- 5) *Pravilnik o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi (NN 112/10)*
- 6) *Zakon o odgoju i obrazovanju u osnovnoj i srednjoj školi (NN 87/08, 86/09, 92/10, 105/10-isp., 90/11, 16/12 i 86/12)*