



PR5 – REPUBLIKA HRVATSKA

SADRŽAJ:

1.	OPIS PROCESA AKREDITACIJE NOVOG NASTAVNOG PROGRAMA U HRVATSKOJ.....	2
2.	MOTIVACIJA UČENIKA	8
3.	NASTAVNI PLAN I PROGRAM	9
3.1.	OOP KROZ RAZVOJ IGARA	10

Odricanje od odgovornosti:

Financirano sredstvima Europske unije. Izneseni stavovi i mišljenja su stavovi i mišljenja autora i ne moraju se podudarati sa stavovima i mišljenjima Europske unije ili Slovačka akademska udruga za međunarodnu suradnju (SAAIC). Ni Europska unija ni SAAIC ne mogu se smatrati odgovornima za njih.



1. Opis procesa akreditacije novog nastavnog programa u Hrvatskoj

Djelatnost srednjeg obrazovanja provode srednje škole i druge javne ustanove, a obuhvaća različite vrste i oblike obrazovanja, osposobljavanja i usavršavanja koji se ostvaruju u skladu s odredbama Zakona o odgoju i obrazovanju u osnovnoj i srednjoj školi (Narodne Novine, br. 87/2008, 86/2009, 92/2010, 105/2010-ispr., 90/2011, 16/2012, 86/2012, 94/2013, 152/2014, 7/2017, [68/2018](#), [98/2019](#), [64/2020](#), [151/2022](#) i [156/2023](#)).

Programi srednjeg obrazovanja su:

- programi za stjecanje niže razine srednjeg obrazovanja;
- programi za stjecanje srednjeg obrazovanja;
- programi osposobljavanja i usavršavanja.

Srednje škole, ovisno o vrsti obrazovnog programa, su:

- gimnazije;
- strukovne škole;
- umjetničke škole.

Gimnazije su četverogodišnje općeobrazovne škole koje učenici završavaju polaganjem državne mature. U gimnazijskim obrazovnim programima učenici stječu kompetencije (znanja i vještine) iz općeg znanja, što predstavlja kvalitetnu osnovu za nastavak obrazovanja na visokoškolskim ustanovama.

Gimnazijski obrazovni program obogaćen je izbornom i fakultativnom nastavom te izvannastavnim aktivnostima koje škole nude sukladno svojim materijalnim i kadrovskim mogućnostima.

Postoji pet vrsta gimnazijskih obrazovnih programa koji se razlikuju u povećanom broju sati onih predmeta koji su karakteristični za pojedini program, a to su:

- opća gimnazija;
- jezična gimnazija;
- klasična gimnazija;
- prirodoslovno-matematička gimnazija;
- prirodoslovna gimnazija.

U gimnazijama (svim vrstama) jedini predmet vezan uz programiranje i općenito informatiku je predmet Informatika. Ovisno o vrsti gimnazije, ovaj predmet se poučava od 1 do 4 godine, bilo kao obvezni ili kao izborni predmet.

Kurikulume i nastavne planove i programe obveznih i izbornih predmeta te međupredmetnih sadržaja i/ili modula donosi ministar odlukom, a fakultativni dio donosi škola svojim godišnjim nastavnim planom i programom. Kurikulum predmeta Informatika donosi Ministarstvo znanosti, obrazovanja i mladih, a odluka o donošenju kurikuluma dostupna je na sljedećoj poveznici (Narodne Novine, br. 22/2018):

https://narodne-novine.nn.hr/clanci/sluzbeni/2018_03_22_436.html

Kurikulum je dostupan i na stranicama Ministarstva znanosti, obrazovanja i mladih:

<https://mzom.gov.hr/UserDocsImages/dokumenti/Publikacije/Predmetni/Kurikulum%20nastavnog%20predmeta%20Informatika%20za%20osnovne%20skole%20i%20gimnazije.pdf>

Opća gimnazija daleko je najzastupljenija od svih ostalih tipova gimnazija, a više od polovice učenika koji upišu gimnazijsko obrazovanje upisuje opću gimnaziju. Detaljno istraživanje dostupnosti srednjoškolskih programa u Hrvatskoj od 2013. do 2021. godine proveo je Institut za društvena istraživanja u Zagrebu, a može se vidjeti na sljedećoj poveznici:

<https://mzom.gov.hr/UserDocsImages/dokumenti/Obrazovanje/NacionalniPlan2021-2027/Studija-srednje-obrazovanje.pdf>

Postoci upisanih učenika po vrstama gimnazijskih programa u školskoj godini 2021./2022. može se vidjeti u ovoj tablici:

Vrsta gimnazije	Postotak upisanih učenika
Opća gimnazija	57.7%
Jezična gimnazija	14.5%
Klasična gimnazija	5.7%
Prirodoslovno-matematička gimnazija	17.3%
Prirodoslovna gimnazija	3.7%
Ostale vrste gimnazija (eksperimentalni programi)	1.1%

U općoj gimnaziji, koja je najzastupljenija od svih gimnazija, predmet Informatika izvodi se u 2 oblika: kao obvezni predmet u 1. razredu i kao izborni predmet u preostala 3 razreda.

Kao što je vidljivo u kurikulumu predmeta Informatika za opće gimnazije, ishodi učenja koji se usvajaju u područjima programiranja, računalnog mišljenja i rješavanja problema su sljedeći:

Godina/razred (vrsta predmeta)	Ishodi učenja	Razrada ishoda učenja
1. (obavezan)	Učenik analizira problem, definira ulazne i izlazne vrijednosti te uočava	Učenik: <ul style="list-style-type: none"> • analizira problem, određuje vrstu i opseg ulaznih podataka, razmatra načine rješavanja problema

	<p>korake za rješavanje problema.</p>	<ul style="list-style-type: none"> • uočava zasebne cjeline i rastavlja problem na manje dijelove • prikazuje postupak rješavanja problema u koracima • prepoznaje u svojem algoritmu osnovne algoritamske strukture: slijed, grananje i ponavljanje • diskutira ispravnost algoritma te ga po potrebi mijenja • prepoznaje ograničenja algoritma
	<p>Učenik primjenjuje jednostavne tipove podataka te argumentira njihov odabir, primjenjuje različite vrste izraza, operacija, relacija i standardnih funkcija za modeliranje jednostavnoga problema u odabranome programskom jeziku.</p>	<p>Učenik:</p> <ul style="list-style-type: none"> • odabire tip podatka prikladan za rješavanje zadanoga problema • opisuje djelovanje pojedinih matematičkih i logičkih operacija • primjenjuje standardne operacije i funkcije nad jednostavnim tipovima podataka pri rješavanju zadanoga problema te funkcije unosa i ispisa • određuje rezultat matematičkih i logičkih izraza • analizira izraze • modelira jednostavni problem odgovarajućim izrazima • argumentira prednosti korištenja odabranim tipom podataka s obzirom na druge tipove podataka
	<p>Učenik razvija algoritam i stvara program u odabranome programskom jeziku rješavajući problem uporabom strukture grananja i ponavljanja.</p>	<p>Učenik:</p> <ul style="list-style-type: none"> • razvija algoritam primjenjujući strukture ponavljanja (s unaprijed određenim brojem ponavljanja te uvjetno) i grananja • prati izvođenje algoritma, implementira ga u odabranom programskom jeziku, testira i vrednuje u suradnji s drugima, koristi se različitim metodama ispravljanja pogrešaka • vrednuje različita rješenja istoga problema
2. (izborni)	<p>Učenik analizira osnovne algoritme s jednostavnim tipovima podataka i osnovnim programskim strukturama i primjenjuje ih pri rješavanju novih problema.</p>	<p>Učenik:</p> <ul style="list-style-type: none"> • objašnjava osnovnu ideju nekoliko ključnih algoritma (zbrajanje/množenje prirodnih brojeva unutar određenoga intervala, unosi i zbraja/množi n brojeva, provjera je li broj prost, najveću unesenu vrijednost sl.) • razmatra druge načine rješavanja istih problema. • analizira efikasnost algoritma ovisno o količini i vrsti ulaznih vrijednosti • primjenjuje poznate algoritme pri rješavanju novih problema
	<p>Učenik u zadanome problemu uočava manje cjeline, rješava ih te ih potom integrira u jedinstveno rješenje problema.</p>	<p>Učenik:</p> <ul style="list-style-type: none"> • raščlanjuje zadani problem na manje funkcionalne cjeline koje opisuje • određuje ulazne i izlazne parametre funkcionalnih cjelina • razlikuje globalne, lokalne i formalne varijable • funkcionalne cjeline rješava u konkretnome programskom jeziku ispravno koristeći se programskim funkcijama i integrira ih u cjelovito rješenje problema • surađuje u timskome rješavanju problema
	<p>Učenik rješava problem primjenjujući</p>	<p>Učenik:</p>

	<p>jednodimenzionalnu strukturu podataka.</p>	<ul style="list-style-type: none"> • opisuje jednodimenzionalnu strukturu podataka zadanoga programskog jezika • objašnjava ulogu indeksa • uočava mogućnost korištenja jednodimenzionalnom strukturom podataka pri rješavanju zadanoga problema • opisuje i koristi se standardnim funkcijama i metodama za rad s odabranom jednodimenzionalnom strukturom podataka • argumentira odabir jednodimenzionalne strukturu podataka za rješavanje zadanoga problema
	<p>Učenik u suradnji s drugima osmišljava algoritam, implementira ga u odabranome programskom jeziku, testira program, dokumentira i predstavlja drugima mogućnosti i ograničenja programa.</p>	<p>Učenik:</p> <ul style="list-style-type: none"> • u suradnji s drugima razmatra problem, osmišljava algoritam i razvija idejno rješenje • testira i dokumentira idejno rješenje • na temelju idejnoga rješenja razvija, testira i dokumentira programsko rješenje • predstavlja drugima mogućnosti i ograničenja programskoga rješenja • procjenjuje uspješnost programskoga rješenja
3. (izborni)	<p>Učenik primjenjuje standardne algoritme definirane nad cijelim brojevima.</p>	<p>Učenik:</p> <ul style="list-style-type: none"> • primjenjuje algoritam za traženje najvećeg i najmanjeg broja, algoritam za zamjenu vrijednosti dviju varijabli, algoritam za rastav broja na znamenke, algoritam za provjeru složenosti broja te Euklidov algoritam
	<p>Učenik analizira sortiranje podataka kao važan koncept za rješavanje različitih problema.</p>	<p>Učenik:</p> <ul style="list-style-type: none"> • opisuje i primjenjuje koji od jednostavnih algoritama za sortiranje i pretraživanje podataka • primjenjuje sortiranje kao dio strategije za rješavanje problema
	<p>Učenik koristeći neki grafički modul vizualizira i grafički prikazuje neki problem iz svoje okoline.</p>	<p>Učenik:</p> <ul style="list-style-type: none"> • primjenjuje osnovne funkcije odabranoga grafičkog modula pri crtanju kompozicija oblika različite složenosti te za prikaz matematičkih funkcija • vizualizira i grafički prikazuje koji realan problem
	<p>Učenik rješava problem primjenjujući složene tipove podataka definirane zadanim programskim jezikom.</p>	<p>Učenik:</p> <ul style="list-style-type: none"> • opisuje složene tipove podataka. Opisuje osnovne metode i funkcije složenoga tipa podataka. Uočava mogućnost korištenja složenim tipovima podataka u zadanome problemu. Rješava zadani problem primjenjujući metode i funkcije složenoga tipa podataka.
	<p>Učenik definira problem iz stvarnoga života i stvara programsko rješenje prolazeći sve faze</p>	<p>Učenik:</p> <ul style="list-style-type: none"> • definira problem iz stvarnoga života • analizira problem i razlaže ga na manje dijelove

	programiranja. Predstavlja programsko rješenje i vrednuje ga.	<ul style="list-style-type: none"> • koristi se mogućnostima programskoga jezika za rješavanje konkretnoga problema • dokumentira programsko rješenje i predstavlja ga ostalima • zajedno s ostalima vrednuje uspješnost programskoga rješenja
4. (izborni)	Učenik rješava problem primjenjujući rekurzivnu funkciju.	Učenik: <ul style="list-style-type: none"> • opisuje osnovne elemente rekurzivnoga postupka • zapisuje matematički opisanu rekurzivnu funkciju u programskome jeziku • uočava rekurzivnost u danome problemu, određuje rekurzivnu relaciju i uvjet prekida te realizira rekurzivnu funkciju u programskome jeziku • procjenjuje efikasnost rekurzivnoga rješenja • ovisno o problemu odabire rekurzivno odnosno induktivno rješenje
	Učenik uspoređuje različite algoritme sortiranja i pretraživanja podataka.	Učenik: <ul style="list-style-type: none"> • opisuje i primjenjuje standardne algoritme sortiranja i pretraživanja podataka • primjenjuje sortiranje kao dio strategije za rješavanje problema • argumentira upotrebu bržih algoritama sortiranja i pretraživanja navodeći primjere
	Učenik osmišljava objektni model s pripadnim složenim strukturama podataka i implementira ga u zadanome programskom jeziku.	Učenik: <ul style="list-style-type: none"> • opisuje osnovne pojmove povezane s objektnim usmjerenim programiranjem (klasa, objekt, svojstvo, metoda ...) • unutar zadanoga problema uočava osnovna svojstva i metode te oblikuje pripadnu klasu u konkretnome programskom jeziku • objašnjava pojam nasljeđivanja klasa • objektno modelira složeniji problem i implementira rješenje u programskome jeziku
	Učenik definira problem iz stvarnoga života i stvara programsko rješenje prolazeći sve faze programiranja. Predstavlja programsko rješenje i vrednuje ga.	Učenik: <ul style="list-style-type: none"> • definira problem iz stvarnoga života • analizira problem i razlaže ga na manje dijelove • koristi se mogućnostima programskoga jezika za rješavanje konkretnoga problema • dokumentira programsko rješenje i predstavlja ga ostalima • zajedno s ostalima vrednuje uspješnost programskoga rješenja

Očito je (kao što je propisano predmetnim kurikulumom za opće gimnazije) da je objektno orijentirano programiranje zastupljeno u vrlo maloj mjeri i to samo u 4. razredu kao izborni predmet, kao jedan ishod učenja.

Situacija u Srednjoj školi Ivanec je takva da je Informatika obavezna u 1. razredu, a kao izborni predmet se nudi u 2. i 3. razredu (u 4. razredu nema predmeta Informatika)¹.

To znači da se učenici Srednje škole Ivanec ni u kojoj mjeri ne susreću sa sadržajima objektno orijentiranog programiranja. Iz tog razloga jedino moguće rješenje je **izrada novog kurikuluma fakultativnog predmeta**. Na ovaj način će se naši učenici moći upoznati s osnovnim OOP konceptima, a učenici drugih škola koji i imaju Informatiku u 4. razredu i susreću se s osnovama objektno orijentiranog programiranja kroz izborni predmet, moći će steći i proširiti svoje poznavanje OOP koncepata na zabavniji i jednostavniji način.

Dakle, Srednja škola Ivanec planira izraditi kurikulum za **novi fakultativni predmet Objektno orijentirano programiranje kroz razvoj igara** u trajanju od 32 sata godišnje (1 školski sat tjedno) na temelju nastavnog plana i programa izrađenog u okviru projektnog rezultata PR3 ovog projekta i udžbenika za nastavnike izrađenog u okviru rezultata PR4. Predmet je namijenjen učenicima 4. razreda srednje škole (učenici opće gimnazije). Poznavanjem osnovnih informatičkih pojmova kao što su programiranje, algoritmi ili strukture podataka učenici će lakše razumjeti sadržaje obuhvaćene nastavnim planom i programom ovog predmeta, zbog čega se nastavni plan i program provodi u 4. razredu, nakon što učenici usvoje temeljne pojmove programiranja i ICT općenito.

Budući da za izvođenje pojedinog fakultativnog predmeta **ne postoji obveza pribavljanja suglasnosti Ministarstva**, srednja škola samostalno donosi nastavni plan i program/kurikulum fakultativnog predmeta i pripadajuće literature.

Kurikulum sadrži:

- uvodni dio,
- metodologiju izrade kurikuluma,
- svrhu i opis kurikuluma,
- obrazovne ciljeve izvođenja fakultativnog predmeta,
- ključne domene kurikuluma,
- ishode učenja, ključne sadržaje i razine usvojenosti,
- organizaciju učenja i poučavanja,
- materijale i sadržaje za učenje,
- evaluaciju usvojenosti ishoda učenja,
- popis potrebnih kvalifikacija nastavnika za realizaciju kurikuluma,
- literaturu.

Skup ishoda učenja sadrži sljedeće elemente:

- oznaku skupa ishoda učenja,
- naziv skupa ishoda učenja,
- popis ishoda učenja,
- opis ishoda učenja,
- ključne sadržaje za usvajanje odgojno-obrazovnih ishoda,

¹ Kao što je ranije navedeno u tekstu, škole mogu same odlučiti o zastupljenosti izbornih predmeta, ovisno o materijalnim i kadrovskim mogućnostima.

- preporuke za usvajanje odgojno-obrazovnih ishoda.

Glavni izvor za učenje i poučavanje bit će udžbenik za nastavnike i online platforma izrađena u okviru rezultata PR4, a kao dodatna literatura može se koristiti (odobreno od Ministarstva znanosti i obrazovanja):

- Budin, L., Brođanac, P., Markučić, Z., Perić, S., Wendling, E., INFORMATIKA 3, udžbenik za 3. razred srednjih škola, Element, 2020.
- Volarić, T., Toić Dlačić, K., Ivošević, I., Draganjac, M., Think IT, udžbenik za 4. razred srednjih škola, Alfa, 2021.

Nema dodatnih zahtjeva u postupku odobravanja literature.

Za izvođenje novog fakultativnog predmeta potrebno je izrađeni kurikulum prezentirati Nastavničkom vijeću škole, a suglasnost za njegovo izvođenje daje Školski odbor.

Preduvjet za izvođenje predmeta definiran je Državnim pedagoškim standardom² (broj polaznika u obrazovnoj skupini mora biti najmanje 10).

2. Motivacija učenika

Na kraju 3. razreda učenike ćemo motivirati da odaberu spomenuti fakultativni predmet kroz demonstraciju igara koje će izraditi i te dodatne informacije zašto će im biti korisno odabrati taj predmet. Tekst motivacije može izgledati ovako:

Dragi učenici,

imate jedinstvenu priliku da upišete ovaj fakultativni predmet koji vam može otvoriti vrata u svijet programiranja i razvoja igara! Naziv ovog predmeta je **Objektno orijentirano programiranje kroz razvoj igara** i kao što mu i sam naziv kaže, fokus je na stjecanju znanja i vještina iz osnova objektno orijentiranog programiranja, ali na interaktivan i zabavan način, tj. kroz razvoj igara. Evo nekoliko razloga zašto biste trebali odabrati ovaj predmet i koje prednosti on nudi:

1. Vrijedne vještine za budućnost:

- Programiranje je vrlo tražena vještina na današnjem tržištu rada. Učenjem osnova objektno orijentiranog programiranja dobit ćete solidne temelje za daljnji razvoj u IT području. Ove vještine značajno će vam olakšati upis na srodne sveučilišne programe, kao što su informatika, elektrotehnika i srodne discipline. Osim toga, znanje koje steknete može vam pomoći i u raznim drugim područjima koja koriste tehnologiju, poput znanosti, inženjerstva i poslovanja.

2. Razvoj igara kao zabavna metoda učenja:

- Programiranje može biti prilično izazovno, posebice objektno orijentirano programiranje, s kojim se možda dosad niste susreli. Učenje kroz razvoj igara uvelike će pojednostaviti i pomoći vam da shvatite temeljne

² https://narodne-novine.nn.hr/clanci/sluzbeni/2008_06_63_2130.html

https://narodne-novine.nn.hr/clanci/sluzbeni/2010_07_90_2539.html

koncepte na zabavan način. Stvaranjem vlastitih igara naučit ćete ključne koncepte programiranja kao što su klase, objekti, nasljeđivanje i asocijacije, a sve to uz zabavu. Ovaj pristup čini učenje manje stresnim i mnogo zanimljivijim.

3. Timski rad:

- Nastava će se izvoditi u timovima, što znači da ćete imati priliku raditi zajedno sa svojim vršnjacima, razmjenjivati ideje i razvijati vještine timskog rada. Rad u timovima simulira stvarna radna okruženja u IT industriji, gdje je suradnja ključ uspjeha. Naučit ćete kako učinkovito komunicirati, zajedno rješavati probleme i podijeliti zadatke na način koji najbolje koristi snage svakog člana tima.

4. Minimalna vremenska obveza:

- Predmet će se održavati samo jedan sat tjedno tako da vam neće oduzeti previše vremena, ali će vam pružiti značajna znanja i vještine. To znači da ćete moći lako uskladiti ovaj predmet s ostalim školskim obvezama. Jedan sat tjedno dovoljan je za stjecanje osnovnih vještina i znanja o objektu orijentiranom programiranju, a opet dovoljno kratak da vam ne predstavlja veliki teret.

5. Jednakost i inkluzivnost:

- Predmet je osmišljen tako da bude jednako koristan i za učenice i za učenike. Vjerujemo da svatko ima potencijal i mogućnost postati uspješan programer, bez obzira na spol. Programiranje i informatika su područja u kojima je raznolikost izuzetno važna, a mi želimo da svi učenici imaju jednaku priliku za razvoj i napredovanje.

Odabirom ovog predmeta zakoračit ćete u svijet programiranja i tehnologije, polja koja kroje budućnost. Iskoristite priliku da se zabavite, naučite nešto novo i pripremite se za buduće izazove. Prijavite se i postanite dio ove uzbudljive avanture!

Osim toga, za učenike se može organizirati pokazna radionica na kojoj će im se prezentirati koje će se igre razvijati, s mogućnošću da učenici sami implementiraju neku jednostavniju funkcionalnost kako bi ih dodatno motivirali za odabir predmeta.

3. Nastavni plan i program

Cilj nastavnog plana i programa je razvoj digitalnih kompetencija i osposobljavanje učenika srednjih škola u području objektu orijentiranog programiranja, s naglaskom na stjecanje iskustva u timskom radu. Ove vještine značajno će olakšati upis učenika na srodne sveučilišne programe, kao što su informatika, elektrotehnika i srodne discipline. Osim toga, znanje koje učenici steknu može im pomoći i u raznim drugim područjima koja koriste tehnologiju, poput znanosti, inženjerstva i poslovanja. S obzirom na to da programiranje može biti prilično izazovno, posebice objektu orijentirano programiranje kurikulum je osmišljen na način da učenici uče kroz razvoj igara što će im uvelike pojednostaviti i pomoći da shvate temeljne koncepte, ali na jedan zabavan način. Stvaranjem vlastitih igara naučit će ključne koncepte

programiranja kao što su klase, objekti, nasljeđivanje i asocijacije. Ovaj pristup čini učenje manje stresnim i mnogo zanimljivijim za učenike.

Ciljanu skupinu čine učenici srednjih škola, naročito učenici općih, jezičnih i klasičnih gimnazija koji u svojem redovnom obrazovanju ne stječu (ili stječu u vrlo maloj mjeri) osnovne koncepte o objektno orijentiranom programiranju.

Nastavni plan i program je prvenstveno namijenjen nastavnicima Informatike u srednjim školama koji u svoju nastavu žele uključiti poučavanje o objektno orijentiranom programiranju na zabavan i inovativan način i koji žele svoje učenike dodatno pripremiti za nastavak školovanja na fakultetima srodnog područja.

3.1. OOP kroz razvoj igara

Broj teme	Igra	Naziv teme	Sati	Softver
1.		Uvod u predmet, programski zahtjevi, metode i načini rada, obveze studenata	1	
2.		Ponavljjanje osnovnih pojmova programiranja (instrukcije, funkcije, grananja, petlje)	2	
3.	Uhvati muhu	Motivacijska igra: Uhvati muhu	2	Greenfoot
4.		Git	2	GitHub, SourceTree
5.	Igra obrane tornja	Greenfoot i jezik Java - uvod	3	Greenfoot
6.		Algoritam, kontrola aplikacije, kreiranje metode	3	Greenfoot
7.		Grananje i kontrola neprijatelja	3	Greenfoot
8.		Varijable i izrazi	3	Greenfoot
9.		Asocijacija	4	Greenfoot
10.		Nasljeđivanje	4	Greenfoot
11.		Enkapsulacija	3	Greenfoot
12.	Igre učenika	Prezentacije samostalnih radova učenika (izrada vlastite igre - cjelogodišnji projekt)	2	Greenfoot