

Alati namijenjeni podučavanju programiranja

Okić, Amel

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:695732>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-05**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

AMEL OKIĆ

ALATI NAMIJENJENI PODUČAVANJU PROGRAMIRANJA

Završni rad

Pula, rujan 2017.

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

AMEL OKIĆ

ALATI NAMIJENJENI PODUČAVANJU PROGRAMIRANJA

Završni rad

JMBAG: 3399-E

Studijski smjer: Informatika

Predmet:

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informatičke i komunikacijske znanosti

Znanstvena grana: Informatički sustavi i informatologija

Mentor: Doc. dr. sc. Tihomir Orehovački

Pula, rujan 2017.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za sveučilišnog prvostupnika informatike, ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, _____ godine

IZJAVA
o korištenju autorskog djela

Ja, _____ dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom _____ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ godine

Potpis

SADRŽAJ:

1. Uvod	1
2. O programiranju	3
2.1. Definicija programiranja	3
2.2. Osnove programiranja	4
2.3. Proces programiranja	6
2.4. Komparativni prikaz programskih alata i okruženja	10
3. Programski jezici	13
3.1. Niži i viši programski jezici	14
3.2. Proceduralni programski jezici i programski jezici neovisni o računalu i operacijskom sustavu	15
3.3. Strukturno i objektno orijentirano programiranje	15
3.4. Odabrani programski jezici	16
4. E-učenje i igrice za učenje kao vrste alata za programiranje	20
4.1. O e-učenju	20
4.2. Uloga igrice u učenju programiranja	22
5. Interaktivne web stranice	25
5.1. CodeAcademy	25
5.2. Coursera	27
5.3. Udemy	28
5.4. W3Schools	29
5.5. Free Code Camp	30
5.6. GitHub	31
5.7. Net Beans	32
5.8. Android Studio i Eclipse	33
6. Zaključak	37
Literatura	39
Popis slika	42
Popis tablica	43
Sažetak	44
Summary	45

1. Uvod

Popularizacija programiranja započela je oko 70-ih godina prošloga stoljeća i od tada do danas sve se više intenzivira. U začetima razvoja ovoga procesa na njega se gledalo isključivo kao na mehaničko nizanje kodova u programskom editoru, a s njegovim razvojem ta se percepcija značajno mijenja.

Danas programiranje predstavlja način logičkog razmišljanja koje za cilj ima reduciranje i otklanjanje konkretnih problema ili pojednostavljenje određenih procesa u informatici, ali i šire. Vjeruje se kako će broj alata koji se koriste za usvajanje znanja i razvoj vještina u programiranju nastaviti svoj rast u budućnosti, kao i broj pojedinih vrsta unutar ovih skupina.

Osim tradicionalne izobrazbe za ove potrebe, danas se sve više koriste alternativna rješenja ili kanali koji upotpunjuju formalno obrazovanje, a neki od oblika su e-učenje ili online učenje, računalne igrice i slično. Smatra se da ovi alati doprinose motivaciji subjekata, minimizaciji troškova, skraćanju vremena, ali i praćenju međunarodnih praksi.

Cilj rada je istražiti pojmove poput programskih jezika, programiranja i alata za učenje programiranja. Pored toga, cilj je ukazati na važnost i značaj procesa programiranja u današnjem informatiziranom, digitaliziranom i suvremenom svijetu.

Svrha rada je provesti analizu alata za učenje programiranja. Poseban naglasak pri tome se postavlja na e-učenje, igrice i interaktivne web stranice.

Struktura rada sastoji se od četiri poglavlja, uvoda i zaključka. Prvo poglavlje posvećeno je procesu programiranja, a osim same definicije i specifičnosti ovoga pojma, daje pregled i njegova tijeka, odnosno fazne izvedbe. Sljedeće poglavlje istražuje značenje i vrste programskih jezika. Poglavlje što slijedi obrađuje ulogu i važnost e-učenja i računalnih igrica koje služe za usvajanje znanja i razvoj vještina u programiranju, dok posljednje poglavlje rada na isti način obrađuje interaktivne web stranice.

Metodologija istraživanja odnosi se na primjenu nekih osnovnih metoda. Pri tome se misli na metodu analize i sinteze, induktivnu i deduktivnu metodu, metodu apstrakcije, metodu komparacije i metodu deskripcije.

2. O programiranju

U literaturi i praksi postoji nekoliko definicija programiranja. Ono što je moguće istaknuti na samom početku jest činjenica da ovaj pojam označava kompleksan proces u informatici, kao i jedan od ključnih termina. Kako bi bilo moguće pojmiti značenje ovoga termina, važno je razlučiti neke osnovne pojmove i činjenice u svezi njega.

Računalo, kao jedan od vodećih resursa i alata u suvremenom poslovnom i svakidašnjem životu, biva sastavljeno od dvije komponente, hardvera ili „tvrdog dijela“ i softvera ili „mekog dijela“. Kao što i sam naziv govori, hardver predstavlja opipljivi, mehanički dio, odnosno ono od čega je građeno računalo, dok softver označava računalne programe koje računalo koristi za izvršenje naredbi ili zadataka.

U okviru ove problematike, središte zanimanja postavlja se na softver, odnosno na program i proces njegova stvaranja ili programiranje. Detaljnija analiza njegova pojmovnog određenja, tijeka i svih specifičnosti obrađuje se u ovome poglavlju.

2.1. Definicija programiranja

Osnovni pristup definiranju programiranja je onaj koji identificira užu i širu poimanja. U širem smislu, programiranje je procedura ili proces, odnosno postupak rješavanja nekog računalnog problema (engl. *problem solving procedure*). „Rezultat provedbe ovoga procesa je definiranje, osmišljavanje i stvaranje informatičkog ili računalnog programa (Zovko, 2015, str. 7).“

U užem smislu, programiranje je pojam užeg značenja od softvera, s kojim se često izjednačava u praksi. Kako bi njegova izvedba bila moguća, važno je koristiti neki od programskih jezika. Osim ovog preduvjeta, za provedbu predmetnog procesa važno je neki problem raščlaniti na manje, nedvosmislene cjeline ili korake, koje računalo može procesuirati i izvršiti.

Za cjelovito razumijevanje značenja programiranja kao procesa, bilo da je riječ o

njegovu užem ili širem poimanju, važno je definirati program i softver. Pri tome, program predstavlja niz naredbi ili naloga računalu koja se izvršavaju nekim preciznim slijedom. Njima se doprinosi realizaciji nekog zadatka, a pri čemu se podaci transformiraju u informacije (Zovko, 2015).

Iako mnogi znanstvenici, stručnjaci i ostali dionici izjednačavaju značenje pojma program i softver, to nije sasvim točno. Naime, softver predstavlja programe, postupke, dokumentaciju i sve potrebite podatke koji se povezuju s radom računalnog sustava. Prema tome, program je tek dio kompleksnog softvera.

„Pisanje računalnog programa je oko 10-20% vremena i napora za razvoj softvera, što proporcionalno ukazuje na omjer ili odnos ovih pojmova. Financijski trošak programa predstavlja oko 10% ukupnih troškova za razvoj softvera, što je vrlo značajan podatak u informatičkom ili digitaliziranom svijetu današnjice (Zovko, 2015, str. 9).“

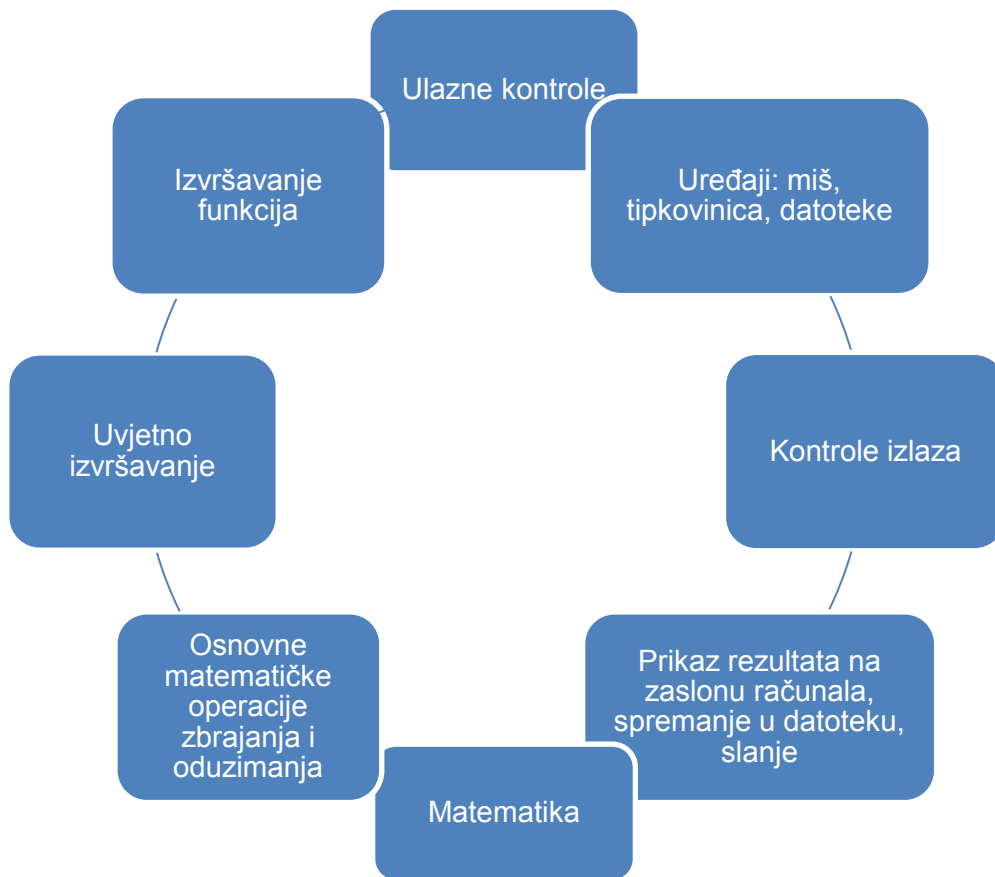
Slijedom navedenog, programiranje se piše u programskom jeziku, a pri tome program predstavlja uređen i konačan niz instrukcija, a pri njegovu izvršavanju računalo se navodi na postupak koji je prethodno uređen (Geodetski fakultet u Zagrebu, n.d.).

Nakon sušte definicije programiranja, važno je osvrnuti se na neke njegove osnovne značajke, a misli se na osnove programiranja.

2.2. Osnove programiranja

Redoslijed instrukcija koje sačinjavaju programiranje kao proces definiran je konkretnom zadaćom ili željenim rezultatom koji se želi dostići. Iako postoji veliki broj programa, odnosno nekoliko vrsta istih, može se generalizirati da su svi oni sačinjeni od nekoliko unificiranih dijelova (Slika 1.).

Slika 1. Dijelovi računalnih programa



Izvor: Izrada autora prema: Miler, M.; Odošić, D., 2010, str. 108.-110.

Programiranje se može odrediti i kao raščlanjivanje problema na manje cjeline, kako bi rješenje istoga bilo jednostavnije. O tome je već dijelom bilo riječi. Takvo poimanje ovoga procesa nalaže stvaranje relacije spram problema i grešaka. U informatici se razlikuju tri skupa grešaka, a njihovo rješavanje naziva se *debugging*. Slijedom navedenog, greške se dijele na (Miller, Odošić, 2010):

- Sintaksne – najbrže se otkrivaju, a neki od primjera mogu biti krivo napisane instrukcije;
- Izvršne – relativno brzo se otklanjaju, a otkrivaju se pri izvršenju programa. Neki od primjera su nemogućnost otvaranja datoteke, pokušaj programa da dijeli s nulom i slično;
- Semantičke – najteže se otkrivaju i otklanjaju jer ih program uspijeva uspješno izvršiti, no ne dobiva se željeni rezultat.

Jasno je da se greške klasificiraju s obzirom na jednostavnost njihova uviđanja,

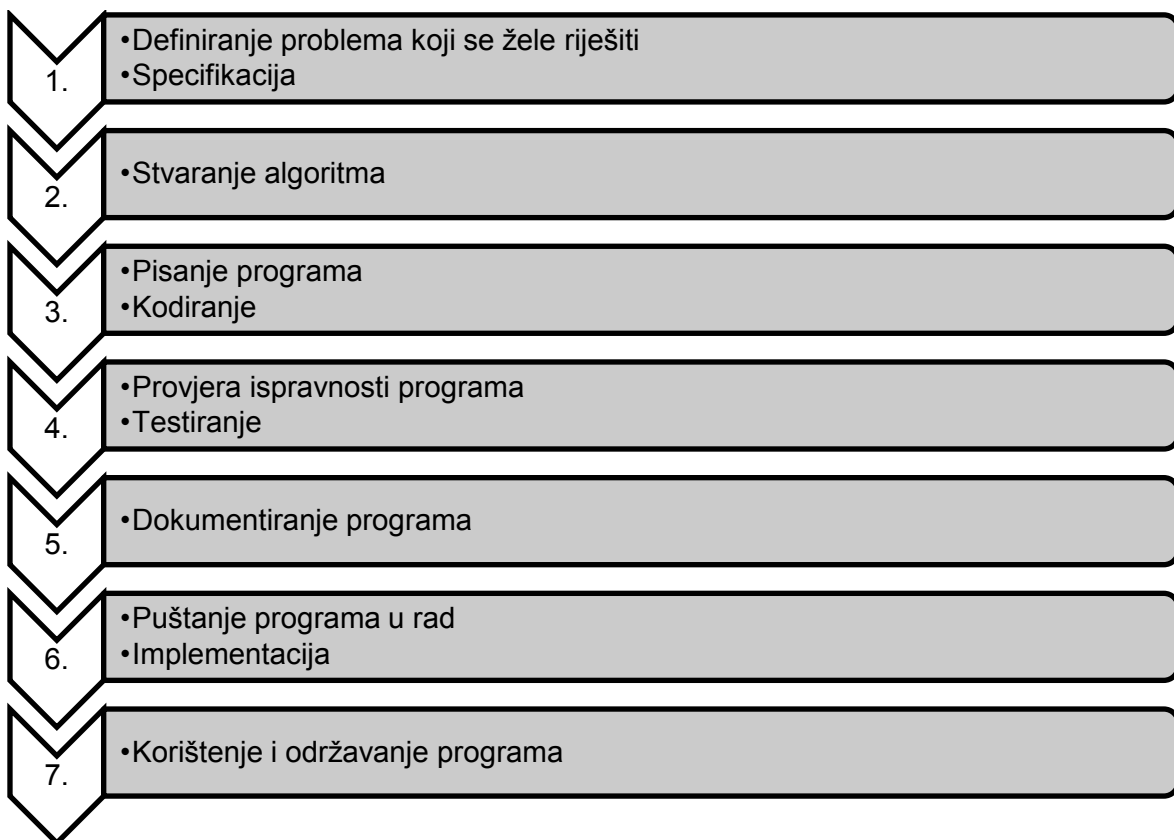
odnosno identifikacije i rješavanje. Kako bi problematika programiranja ili rješavanja ovih pogrešaka bila jasnija, u nastavku poglavlja daje se prikaz tijeka procesa programiranja, uz detaljnu razradu svake od pripadajućih faza.

2.3. Proces programiranja

Iz najjednostavnije definicije programiranja moguće je izdvojiti dvije osnovne zadaće ili funkcije ovoga procesa. Kao takav, predstavlja fazni postupak ili evolucijski tijek stvaranja računalnog programa, a u konačnici i samog softvera. To su ujedno i njegove osnovne funkcije pa i glavni ciljevi. Svrha istoga bila bi omogućiti računalu izvođenje određenih naredbi, zadataka i procesa, odnosno osigurati korisno funkcioniranje ovog suvremenog alata.

Proces programiranja može se prikazati na sljedeći način (Slika 2.).

Slika 2. Tijek procesa programiranja



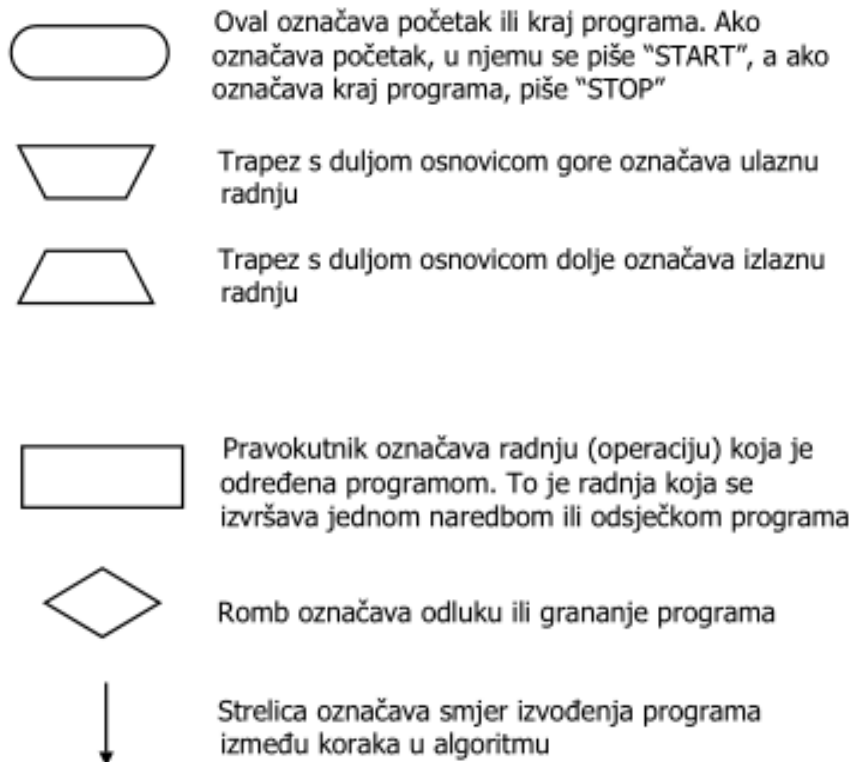
Izvor: Izrada autora prema: Geodetski fakultet u Zagrebu, n.d.

Evidentno je da je proces programiranja vrlo složen, a razvojem informatike i računarstva, te s pojavom suvremenoga doba, isti biva sve kompleksniji i dinamičniji. U takvim poslovnim i životnim uvjetima on zahtijeva uključivanje većeg broja stručnjaka, odnosno angažman tima ljudi (Nikola Tesla, 2017).

Prva od navedenih faza je definiranje ili identificiranje problema. Ona se naziva i fazom specifikacije. Kao takva podrazumijeva detaljan popis i opis zadataka, kao i željenih rezultata. Pri tome ona ne označava način rješavanja problema, već samo daje točan opis onoga što se želi realizirati. Konkretno, ova faza specificira željeni rezultat programiranja.

Sljedeća među njima je faza stvaranja algoritma. Riječ je o stručnom nazivu koji označava slijed radnji ili zadataka koji vode do rješenja definiranog problema. On je zapravo postupak rješavanja istoga putem njegova predočavanja u onom obliku koji je lako pretvoriti u računalni program. Za njegov prikaz koristi se dijagram toga i pseudokoda. U nastavku se daju glavni simboli dijagrama toka, kao i njihova pojašnjenja (Slika 3.).

Slika 3. Glavni simboli dijagrama toga



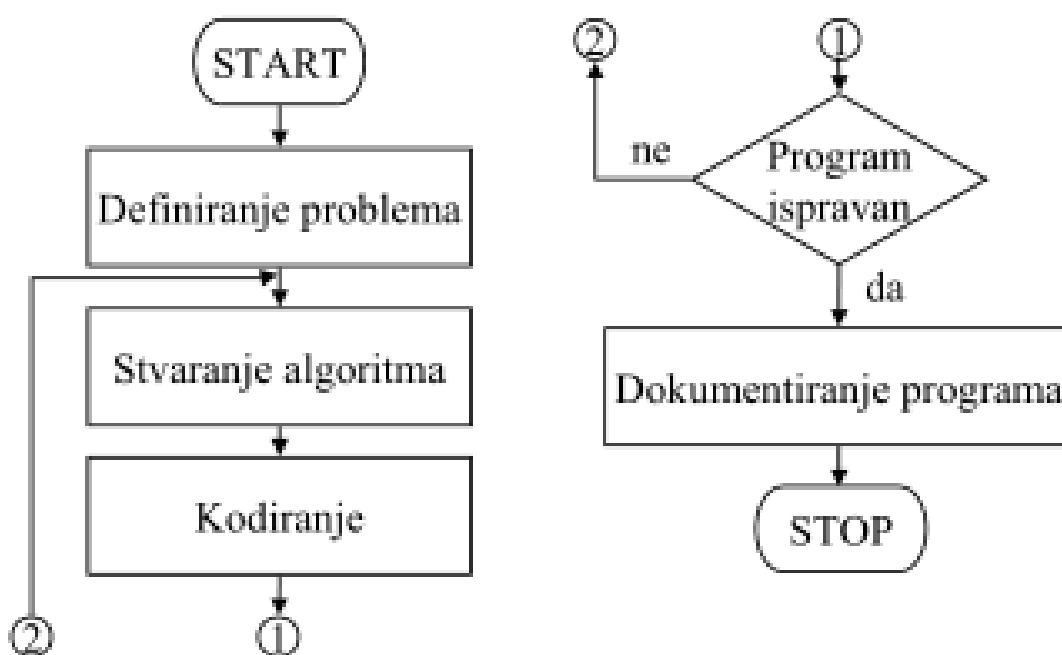
Izvor: Geodetski fakultet u Zagrebu, n.d.

Pseudokod je sljedeći pojam u kontekstu analize i korištenja algoritma. On ga prikazuje na način da se algoritam jednostavno pretvara u naredbe programa koji je iskazan u nekom programskom jeziku. On se sastoji od kratkih izraza na govornom jeziku koji opisuju i ukratko objašnjavaju pojedine zadatke istoga (Geodetski fakultet u Zagrebu, n.d.). Isti se piše tako da se na osnovu njega daje ispisati program u bilo kojem programskom jeziku, a uputno je da bude toliko detaljan da je moguće pojmiti što se očekuje, što program treba raditi i što se želi postići.

Kodiranje je sljedeći postupak u kontekstu ovoga procesa. On se odnosi na popis niza instrukcija ili naredbi koje tvore računalni program, a nerijetko je povezano i sa softverom, odnosno programskim jezikom programa, ali i s hardverom. Kako bi ono bilo uspješno provedivo, presudan značaj ima poznavanje značenja i svrhe svake od instrukcija, ali i sintaksi, odnosno pravila programskog jezika o načinu korištenja i slaganja tih instrukcija.

Od sljedećih faza primarno treba spomenuti provjeru logične ispravnosti programa. Međutim, uloga u toj fazi potpuno pripada programeru. Naime, računalo mu u izvedbi ove faze ne može pomoći jer samo on zna i ima spoznaju o zadatku koji treba provesti i cilju koji treba računalo dostići. Testiranje programa je provjera navedenoga, odnosno logične ispravnosti, ali i izvršenje ispravaka na opetovani način. Sam tijek ove faze prikazuje se na sljedeći način (Slika 4.).

Slika 4. Testiranje programa



Izvor: Geodetski fakultet u Zagrebu, n.d.

Vidljivo je da, ovisno o tome da li je program ispravan ili ne, postoje dvije opcije u kontekstu nastavka procesa programiranja. Ukoliko on nije ispravan, provode se korekcija i ponovna provjera, odnosno testiranje. Ukoliko je program ispravan, nastupiti će njegovo dokumentiranje. Zapravo se misli na izradu onih uputa koje će poslužiti instalaciji programa, izradi priručnika za korisnike, tehničkog opisa i ostalih specifičnosti. Ova faza, osim što predstavlja okončanje procesa, biva iznimno značajna za daljnje održavanje programa, koje je pravilo, a ne iznimka.

Održavanje programa može se definirati kao postupak mijenjanja programa tijekom

njegovog životnog vijeka i poduka korisnika o njegovoj upotrebi (Miler, Odošaić, 2010). Eventualni razlozi izmjene programa su brojni, a neki od njih mogu biti izmjene zakona i načina rada, promjena hardvera na računalu, identificiranje nedostataka ili pogreški te slično.

Nakon analize osnovnih podataka u svezi programiranja, faze ovoga procesa i važnosti njegove provedbe u praksi, smatra se potrebnim detaljnije pristupiti analizi temeljnog elementa u okviru navedene problematike. Pri tome se misli na programske jezike.

2.4. Komparativni prikaz programskih alata i okruženja

U kontekstu razrade programskih alata i okruženja za učenje programiranja, moguće je govoriti o nekoliko osnovnih, koji se grupiraju prema njihovoj namjeni. To su (Sinožić, 2017):

- Alati za učenje programiranja u okruženju Web 2.0;
- Alati za učenje programiranja putem video igara;
- Vizualni alati za učenje programiranja;
- Skupina viših jezika za programiranje zajedno s alatima za pomoć u učenju programiranja.

WEB servis je mjesto koje omogućuje stjecanje predznanja i vještina koje osiguravaju shvaćanje i savladavanje osnova o programiranju. Može se istaknuti kako je to temelj stjecanja znanja ove prirode. U interakciji s ostalim alatima omogućuje aktivnosti unutar servisa. Najpogodniji alati i aplikacije u ovoj skupini su Cloud9 IDE, Ideone i JDoodle (Sinožić, 2017).

Učenje programiranja kroz igrice kreativni je pristup učenja programiranja i smatra se nadopunom ostalih alata. Svrha ovoga alata je potaknuti i motivirati programera da se zainteresira i angažira u smjeru stjecanja znanja i vještina ove prirode. U ovoj skupini posebice se izdvajaju Tynker, Waterbear i RoboMind (Sinožić, 2017).

U naravi, vizualni se alati koriste za prelazno razdoblje, odnosno razdoblje između faze učenja programiranja kroz igrice i faze viših programskih jezika. Značajni su jer

je u praksi izravan prijelaz zapravo nemoguć. Smatra se kako su značajni jer usmjeravaju programera u bit ili srž programiranja. Neki od primjera su RAPTOR, Guido van Robot te VISa (Sinožić, 2017).

Viši alati za programiranje koriste se nakon usvajanja osnovnih znanja i vještina rada na ovim alatima. Oni su najnaprednija i najkompleksnija skupina alata za učenje ali i razvoj znanja o programiranju. Posebice se izdvajaju C++, Java, Python (Sinožić, 2017).

Usporedni ili komparativni prikaz ovih alata slijedi (Tablica 1.).

Tablica 1. Komparativni prikaz alata za učenje programiranja

Skupina alata	Vrsta alata	Ime alata	Kriterij		
			Motiviranost	Savladavanje alata	Smanjivanje odustajanja od studiranja
Web 2.0 okruženje	Najpogodniji alati u web 2.0 okruženju	Cloud9 IDE	***	****	****
		Ideone	***	****	****
		JDoodle	***	****	****
	Kolaborativni alatai	Blog	*****	****	*****
		Wiki	*****	****	*****
Video igre	Igre namjenjene djeci	Tynker	****	****	****
		Waterbear	****	****	****
		RoboMind	****	*****	*****
	Igre namjenjene studentima	Code Hero	*****	*****	*****
		Code Hunt	****	****	****
Vizualni alati	Dijagram	Raptor	*****	*****	****
	Mini jezici	Guido van Robot	****	****	****
	Vizualizacija	ViSA	*****	****	****
Viši jezici	Viši programski jezici	C++	*****	*****	*****
		Java	****	****	****
		Python	****	***	****

Ocjene: * loše, ** dovoljno, *** dobro, **** vrlo dobro, ***** odlično.

Izvor: Sinožić, 2017, str. 42.-43.

Daje se zaključiti kakao su svi programski alati ocijenjeni povoljnim ocjenama, prema pojedinim kriterijima relevantnim za učenje programiranja. Ocjene se kreću od dovoljne do odlične.

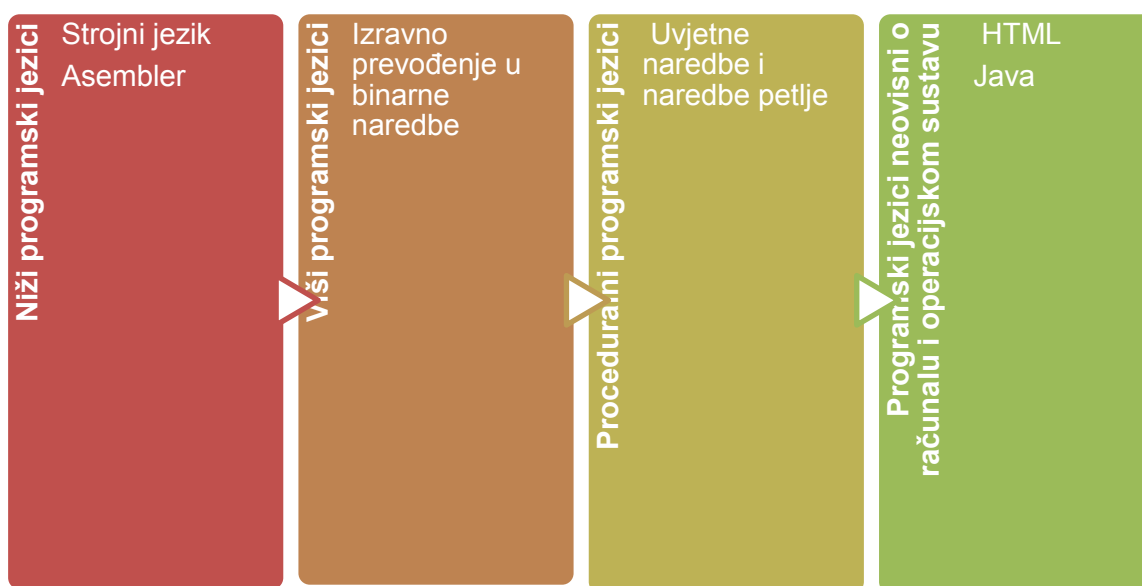
3. Programski jezici

Programski jezici u današnjici predstavljaju kompleksno područje u okviru informatike i računarstva. Ujedno je riječ o jednoj od najistraživanijih tema u krugu informatičara, ali i šire. O popularnosti iste svjedoče sve brojniji autori i radovi koji se njome bave, inicijative dionika za unaprjeđenje ovoga područja i slično.

Zapravo je nemoguće zamisliti funkcioniranje računala bez uspješnog programiranja. Može se zaključiti da ono predstavlja osnovu ili temelj informatičkog svijeta. Međutim, ono nije namijenjeno samo stručnjacima u informatici, već i ostalim subjektima pa se u praksi razlikuje nekoliko vrsta programskih jezika, koji se ujedno i koriste u različite svrhe.

Programski jezik može se definirati kao skup riječi i pravopisnih, odnosno sintaksnih pravila, koja se koriste za slanje naredbe računalu da izvrši neki zadatak (Geodetski fakultet u Zagrebu, n.d.). Oni se klasificiraju na sljedeći način (Slika 6.).

Slika 5. Vrste programskih jezika



Izvor: Izrada autora prema: Geodetski fakultet u Zagrebu, n.d.

O programskim jezicima već je bilo riječi, no u ovome poglavlju pristupa se njihovoj detaljnijoj analizi jer označavaju srž ovoga procesa, odnosno vodeći alat za provedbu istoga. Prikaz pojedinih skupina slijedi u nastavku poglavlja.

3.1. Niži i viši programski jezici

Nižim programskim jezicima pripadaju strojni jezik i Asembler. Strojni jezik je najniža razina programskog jezika¹ i onaj jezik koji je računalu razumljiv. To uvjetuje jednostavno izvršenje onog programa koji je pisan tim jezikom, bez dodatnih prevođenja. Ono što je važno istaknuti jest činjenica da program koji je pisan u strojnom jeziku za određenu vrstu računala ne može biti korišten na nekoj drugoj vrsti. Razlog tome je taj što je on izravno ovisan o središnjoj jedinici za obradu računala (engl. *Central Processing Unit* – CPU).

Povezano s time, onaj programer koji osmišljava i piše programski jezik ove vrste ima za obvezu cjelovito i kvalitetno poznavanje spomenutog dijela računala, odnosno njegove građe jer se pisani jezik odnosi na pojedine sastavne dijelove.

Nižim programskim jezicima pripada i Asembler, kao jezik koji umjesto binarnih kodova, kao što je primjer na prethodnoj vrsti programskog jezika u okviru ove kategorije, koristi simboličke kodove² za ispisivanje naredbi računalu.

Ovaj jezik se najjednostavnije prevodi u prethodnu vrstu strojnog jezika, a orijentiran je prema računalu, što određuje jednako pravilo kao i na prethodnom primjeru. Točnije, programski jezik napisan za neku vrstu računala nije primjenjiv na onoj drugoj vrsti.

Sljedeća kategorija programskih jezika su viši programski jezici, koji bivaju sačinjeni od naredbi koje se mogu izravno prevoditi u binarne naredbe strojnoga jezika. Neki od primjera su BASIC, FORTRAN, Pascal, C, C ++, Java i drugi. Za razliku od prethodne skupine jezika, oni nisu toliko orijentirani računalu, već korisniku i nekom

¹ Binarni 0 i 1.

² Skraćenice koje se sastoje od nekoliko slova. Na primjer MOV (za *move*), STO (za *store*) i slično.

problemu.

Programeri koji se koriste ovim programom ne moraju imati znanje o građi računala, što pogoduje svim ostalim subjektima koji nisu informatičari, stručnjaci iz ovoga područja. Točnije, njime se mogu koristiti svi dionici, a također on biva primjenjiv na svim vrstama računala. Upravo to determinira njihovu popularnost u informatičkom svijetu, ali i šire.

3.2. Proceduralni programski jezici i programski jezici neovisni o računalu i operacijskom sustavu

Sljedeća skupina ili kategorija programskih jezika, koji su prikazani i prethodnom slikom, su proceduralni programski jezici. Njih određuje sadržajnost uvjetnih naredbi³ i naredbi petlje⁴, a kojima oni upravljaju tijekom izvedbe programa koji su prethodno obrađeni. Ovi programi najčešće sadrže naredbe skoka i imaju sposobnost da izvršavaju samostalne procedure ili potprograme (Geodetski fakultet u Zagrebu, n.d.).

Programski jezici koji su neovisni o računalu i operacijskom sustavu primjenjuju se na svim računalima, a popularni postaju s pojavom Interneta koji je omogućio njihovu prenosivost. Najpoznatiji jezici iz ove skupine su HTML i Java.

HTML (engl. *Hypertext Markup Language*) koristi se za prikaz hiperteksta dokumenta, a omogućuje razmjenu podataka korištenjem Interneta. Opće je poznat i često primjenjivan, a kako bi naredbe na ovome jeziku i na svakome računalu imale isto značenje, važno je posjedovanje onog programa koji ih prevodi i izvršava, odnosno preglednika. Sljedeći među navedenima je Java programski jezik, koji je objektno orijentiran i ima opću namjenu. Njegovo korištenje nije uvjetovano Internetom.

3.3. Strukturno i objektno orijentirano programiranje

Kod analize programskih jezika, važno je predstaviti strukturno i objektno orijentirano programiranje. Pri tome, strukturno programiranje biva definirano kao skup

³ Na primjer *if, then, case* i slično.

⁴ Na primjer *for, when, net, do, while* i slično.

programskih metoda koje vode ka logičkoj organizaciji i čitljivosti programa. Ona pojednostavljuje pisanje, održavanje i ispravke programa, koji je podijeljen u nekoliko manjih cjelina, a one naknadno grade glavni program. To pojednostavljuje sam proces i reducira moguće pogreške u njegovoj izvedbi. Jezici koji se koriste u ove svrhe nose isti naziv, a neki od njih su Dbase, PASCAL i slični (Geodetski fakultet u Zagrebu, n.d.).

Objektno orijentirano programiranje je postupak izrade programa upotrebom skupa objekata ili pojedinačnih programa za izvršenje konkretnih zadataka, a koji međusobno razmjenjuju poruke. Najpoznatiji su C++, Visual BASIC, Delphi i ostali (Geodetski fakultet u Zagrebu, n.d.).

3.4. Odabrani programski jezici

Nakon analize skupina i vrsta programskih jezika, smatra se korisnim dati prikaz nekolicine spomenutih. Pri tome se misli na programski jezik BASIC, Pascal, Java, HTML i Delphi.

BASIC (engl. Beginner's All-purpose Symbolic Instruction Code) je razvijen 1963. godine na sveučilištu Dartmouth u Hanoveru (Wikibooks, 2017). Kao i većina programskih jezika, on se temelji na engleskom jeziku, a pri tome je važno poznavati tek osnovne naredbe koje se koriste. Na primjeru njega, također nije presudno ni visoko znanje u području matematike ili ostalih znanosti.

Iako ovaj programski jezik posjeduje svjetski standard (engl. *ANSI//ISO//IEC Standard for Full BASIC*) mnoge kompanije unose dio specifičnosti ili osobnosti pri njegovoj izradi. Nadalje, u pojedinim izvedbama ovaj programski jezik podržava spomenuto objektivno orijentirano programiranje (Visual Basic, Realbasic).

Često se karakterizira kao relativno zastario programski jezik, no programerima pruža mogućnost rapidnog razvoja programa, to jest validiranja algoritma. Noviji, odnosno suvremeniji programski jezici poput Java potisnuli su njegovu popularnost i primjenu, no još uvijek biva međunarodno priznat i primjenjivan programski jezik.

Strukturno, njega čine brojevi, slova i posebni znakovi. U okviru njega javljaju se i skupovi riječi kao ključne riječi koje bivaju definirane isključivo za programere. Neke od njih su *save*, *load*, *loadc*, *list*, *input*, *data*, *read*, *restore*, *print*, *for*, *next*, *goto* i slične (Wikibooks, 2017).

Pascal je viši programski jezik koji je izumio švicarski matematičar Nikaus Wirth, a njegova sređena verzija predstavljena je 1974. godine (Infosremac, 2017). Njegova primarna svrha bilo je učenje principa programiranja pa zaslužuje posebnu pažnju u informatici i analiziranju ove problematike.

Simboli Pascala su sljedeći (Infosremac, 2017):

- Slova – engleska abeceda;
- Brojevi (dekadni zapisi) od 0 do 9;
- Specifični znakovi - +, -, x, /, =, (,), :, ; i ostali;
- Riječi koje imaju uvijek isto značenje – *and*, *array*, *begin*, *case*, *div*, *do*, *else*, *end*, *file*, *for*, *while* i ostale;
- Imena i identifikatori – označuju osnovne objekte jezika: promjenjivih, konstanti, tipova, funkcija i pocedura. Mogu biti sastavljena samo od brojeva i slova, a prvi znak uvijek mora biti slovo;
- Razmak – ima ulogu pokazati kraj naredbe i početak sljedeće.

Važno je istanuti da ovaj program ima i nasljednika, a riječ je o objektno orijentiranom programskom jeziku Delphi.

Java je moderan ili suvremen programski jezik koji je međunarodno poznat, kako stručnjacima iz područja informatike, tako i mnogim drugim dionicima. Može se reći da je on jedan od najpopularnijih programskih jezika u svijetu, a baziran je na klasama, odnosno objektima pa se svrstava u objektno orijentirane programske jezike (Plavljanić, 2014).

Ideja na kojoj se zasnivala njegova izrada očitovala se kroz omogućenje razvoja aplikacija programerima, uz što manje dodataka. U okviru njega moguće je isprogramirati gotovo sve pa je zato i vrlo popularan. Mnogi jezici, poput Pythona ili

C, C++ i drugih brži su od ovoga, no istovremeno su ograničeni.

Java kao programski jezik nudi najviše, a koristi se, između ostaloga, i za izradu manjih aplikacija ili *appleta* koje se pokreću na računalo ili u web pregledniku. Ovaj se programski jezik može instalirati na svako računalo i koristi istim stvarima.

Cilj ovoga programa je prilagoditi se potrebama svih korisnika i postati najraširenija tehnologija u svijetu. Međutim, važno je ukazati i na konkretan nedostatak Jave, a to je spomenuta sporost. Uzmu li se u obzir koristi koje ona nudi, ovaj nedostatak gotovo je i zanemariv.

HTML je osnovni jezik koji se koristi za izradu web stranica i zato je od posebnog značaja u okviru ove problematike. U interakciji s web preglednicima daje podatke o sadržaju i strukturi učitane stranice, a koji od njih dalje oblikuje tu stranicu.

U literaturi se najjednostavnije ovaj jezik poima kao onaj za označavanje hipertekstualnih dokumenata (Nikola Tesla, 2017). Oni se od običnih dokumenata razlikuju po tome što sadrže hiperveze ili hiperlinkove preko kojih se povezuju s drugim dokumentima.

Ovaj programski jezik definirao je današnji ravnatelj World Wide Web konzorcija, Timothy Berners Lee, 1990. godine. Tijekom narednih godina nekoliko puta je nadograđivan, a konačna verzija istoga predstavljena je 1999. godine (Nikola Tesla, 2017).

Delphi je posljednji na ovome popisu programskih jezika. U tekstu je već spomenut pri prazradi BASIC-a. Delphi se predstavlja kao jedan od najboljih alata za razvijanje poslovnih aplikacija u okruženju Windows-a. On ima velike mogućnosti, a razmatraju se kroz vizualni identitet, odnosno (Murk, 2006):

- Boju – nudi više mogućnosti kroz Color i Font.Color., lako se mijenja boja kontrola i boja fonta;
- Veličinu i položaj – rezolucija i veličina kontrola;
- Font – veličina sova.

Pored ovih programskih jezika postoje i mnogi drugi. Međutim ne predstavljaju samo

oni alate za programiranje, već je potrebno osvrnuti se i na neke druge. U tom smislu misli se na pregršt opcija koje su na raspolaganju programerima, bilo da je riječ o stručnjacima ili početnicima, amaterima. Detaljnije o ovim alatima slijedi u narednim poglavljima rada.

4. E-učenje i igrice za učenje kao vrste alata za programiranje

O programiranju i programskim jezicima raspravljalo se u prethodnim poglavljima. Njihov logičan slijed je analiza nekih alternativnih ili suvremenih alata za programiranje, a među njima svakako treba izdvojiti e-učenje i igrice.

E-učenje je moguće razmatrati s gledišta suvremenosti, odnosno kao novitet koji se javlja u suvremeno doba i u informatiziranom te digitaliziranom globalnom društvu. Ono je doživjelo ekspanziju u relativno kratkom razdoblju, a danas predstavlja jednu sasvim uobičajenu praksu koja služi mnogočemu. Osim pukog učenja i usavršavanja, svakako je važno istaknuti i ulogu e-učenja u programiranju.

Pored navedenog alata, sve više se populariziraju igrice kao jedan od instrumenata za programiranje. O ovoj temi sve se više raspravlja u znanosti, ali i javnosti, a njezina analiza daje se i u ovome poglavlju.

4.1. O e-učenju

Učenje putem Interneta javlja se s pojavom suvremnoga doba, točnije 80-ih godina prošloga stoljeća. Ono nastupa prvenstveno u razvijenijim državama svijeta, a danas se primjenjuje u gotovo svakoj državi.

Ovaj način učenja definira se na nekoliko načina, a naziva se i e-učenjem ili online učenjem. Riječ je o usvajanju nastavnih sadržaja putem Interneta. Kako bi isto bilo provedivo u praksi, važno je osigurati kontinuitet u objavljivanju, ažuriranju i nadogradnji nastavnih sadržaja koji su dostupni korisnicima. Pored toga, presudno je da korisnici posjeduju računalo s priključkom na Internet.

Nastavni sadržaji se ovim putem prezentiraju, usvajaju i vježbaju posredstvom računala, interneta i ostale elektroničke opreme, kao što su kamere, fotoaparati i mobiteli (Markić, 2010).

E-učenje (engl. *E-learning*) omogućuje učenicima i studentima rješavanje raznih problema i pronalaženje informacija. Ono nije zamjena za nastavnike već nadopuna tradicionalnog sustava obrazovanja i usavršavanja polaznika. Prema tome, može se zaključiti da ovaj oblik učenja doprinosi nastavnicima, ali i polaznicima. Ono što je važno pri njegovoj provedbi je aktivan angažman objih strana u ovome procesu.

Vrlo često se ovaj način učenja opisuje sljedećim elementima (Srce, 2017):

- Iskustveno učenje (engl. *Experience Learning*);
- Učenje posvuda (engl. *Everywhere Learning*);
- Povećano ili bolje učenje (engl. *Enhanced Learning*);
- Prošireno učenje (engl. *Extended Learning*).

Slijedom navedenog, e-učenje predstavlja dinamičan proces obrazovanja, odnosno učenja i podučavanja, a koje se izvodi uz upotrebu nekog od oblika informacijske i komunikacijske tehnologije. Ono za cilj ima unaprjeđenje procesa obrazovanja i kvalitete ishoda obrazovanja. Ova definicija lako se daje primijeniti na primjer programiranja pa bi e-učenje kao alat za programiranje podrazumijevalo usvajanje znanja i vještina u svezi programiranja putem učenja posredstvom interneta te komunikacijske i informacijske tehnologije.

E učenje, bilo ono generalno ili za potrebe programiranja, može se klasificirati na sljedeći način (Srce, 2017):

- Klasična nastava – nastava u učionici (engl. *face to face – F2F*);
- Nastava uz pomoć ICT tehnologije u službi nadopune klasične nastave (engl. *ICT supported teaching and learning*);
- Hibridna ili mješovita nastava – kombinacija nastave u učionici i nastave uz pomoć tehnologije (engl. *hybrid, mixed mode; blended learning*);
- Online nastava – nastava koja je pomoću ICT tehnologije potpuno organizirana na daljinu (engl. *Fully online*).

U praksi se danas primjenjuju svi oblici koji su prethodno prikazani. Smatra se da je najpopularniji oblik kombinirane nastave te da će on u budućnosti nastaviti svoj razvoj brže od ostalih.

U kontekstu ovoga alata za programiranje, odnosno učenja i razvoja vještina za programiranje, vjeruje se da će se njegov značaj dodatno intenzivirati u budućnosti. Razlozi za to su brojni pa, osim generalnih trendova u međunarodnom pristupu obrazovanja i cjeloživotnog učenja, treba spomenuti dinamičnost ovoga područja, kompleksnost samoga procesa, zahtijevnost, značaj kontinuirane prakse (vježbanja), presudnu ulogu razmjene znanja i iskustva te slično.

4.2. Uloga igrica u učenju programiranja

Popularizacija programiranja započela je s pojavom suvremenoga doba, odnosno oko 70-ih godina prošloga stoljeća. U to vrijeme, programiranje je uglavnom predstavljalo sušto mehaničko ispisivanje naredbi, odnosno kodova nekom programskom editoru. Međutim, s razvojem ono poprima jednu sasvim novu dimenziju.

Danas ovaj proces biva namijenjen rješavanju pogrešaka i nedostataka te pojednostavljenju procesa i izvedbi namijenjenih različitim potrebama. Točnije, riječ je o pronalasku i razvoju kompleksnih strategija za rješavanje problema koji se mogu odnositi i na neprogramska područja (Hercigonja, 2017).

Kao takvo, u suvremeno se doba definira kao moderan način razmišljanja koji parira matematičkom razmišljanju i rješavanju problema. Prema tome, on nadilazi granice tradicionalnog programiranja i rutinskog postupanja te se sve više ističe inovativnošću, dinamičnošću i suvremenostću.

Informatika se kao znanost uvodi u škole diljem svijeta već u to vrijeme. Razlog takve prakse očitovao se u njezinu značaju s gledišta svakodnevnog života, ali i poslovanja. Kako bi obrazovanje u ovome području bilo što kvalitetnije, adekvatno suvremenim zahtjevima i načelima te privlačno svim segmentima učenika, kontinuirano se potražuju neki alternativni načini za usvajanje ovih znanja, a time i znanja u svezi programiranja.

Često se danas govori kako je suvremeno društvo digitalizirano i informatizirano.

Djeca kao njegovi članovi imaju izražene vizualne sposobnosti, što biva jedno od korisnih obilježja za usvajanje znanja o programiranju putem računalnih igara. Prema tome, ovaj se alat uzima kao suvremeno sredstvo za učenje i razvoj znanja u svezi programiranja, a namijenjen je prvenstveno djeci, ali i odraslima.

Igre predstavljaju prilagodljiv medij koji se lako prilagođava svakoj tehnologiji. Danas postoji veliki broj igara, u smislu njihovih vrsti, koje odgovaraju svakom profilu korisnika. Pri tome se misli na ratne, logičke, strateške, društvene i ostale igre (Kirriemuir, McFarlane, 2004). Definicija učenja programiranja putem računalnih igara podrazumijeva uvježbavanje vještina za rješavanje problemskih situacija. To zahtijeva kontinuiranu interakciju, pored primarne motivacije korisnika i interesa za učenjem programiranja.

Programiranje, kao proces o kojem je već bilo riječi, zahtijeva iznimnu razinu motivacije, a ona se dostiže upravo ovim putem. O njoj izravno ovisi angažman subjekata, ali i uspješnost usvajanja znanja i izvedbe. Sama ideja učenja programiranja putem igrica jest naučiti učenike da logično razmišljaju i to razmišljanje primijenjuju na računala, kako bi usvojene modele primijenili kasnije za pisanje linija koda, odnosno za programiranje.

U kontekstu ove problematike, vrlo je važno istražiti aplikacije za programiranje igara. One se koriste u svrhu stvaranja i razvoja različitih scenarija računalne igre u dvodimenzionalnoj ili trodimenzionalnoj grafici (Hercigonja, 2017). Ovaj pristup programiranja razlikuje se po tome što podrazumijeva grafički način ispisivanja i formiranja programa. Tehnika koja se primijenjuje naziva se „pritisni-povuci“, a podrazumijeva slaganje grafičkih ikona. Pri tome, potpuno su isključene sintaktičke i leksičke pogreške pa je time jednostavniji proces osmišljavanja rješenja i izrade programa.

U nastavku se daje prikaz dvaju aplikacija za programiranje računalnih igara, u svrhu detaljnije analize predmetne problematike (Tablica 2.).

Tablica 2. Aplikacije za programiranje računalnih igara – Game Maker – Studio 1.4 i Construct 2

Game Maker – Studio 1.4
<ul style="list-style-type: none"> • Komercijalna aplikacija za razvoj dvodimenzionalnih i trodimenzionalnih računalnih igara; • Podržana i za Windows te Mac Os operacijske sustave, ali i Android; • Besplatna verzija 30 dana – probna verzija; • Multiplatform development i Complete Studio package – licencirana verzija; • Postojanje svih programskih struktura – uvjetne naredbe, <i>while</i>, <i>repeat</i>, <i>do</i> i <i>for</i> petlje, funkcije i nizovi; • Programiranje: kreiranje objekata koji služe kao glavni likovi (engl. <i>Sprites</i>), a mogu se sami nacrtati ili preuzeti. Kreiranje sobe i odabir radnji (gibanje, skakanje, mjerenje vremena, rad s uvjetima, akcije bodovanja i slično); • Nakon navedenoga slijedi provjera ili testiranje naredbom <i>Run</i>.
Construct 2
<ul style="list-style-type: none"> • Jednostavna komercijalna aplikacija za kreiranje i dizajniranje dvodimenzionalnih računalnih igara; • Kodiranje putem grafičkih ikona koje se razvrstavaju tehnikom „pritisni-povuci“; • Besplatna verzija 30 dana; • Nekoliko komercijalnih inačica; • Rad u Windows i Mac Os operativnom sustavu; • Sučelje je intuitivno i jednostavno za korištenje; • Postoji nekoliko dijelova: <i>Properties</i> sa svim podacima o objektima i pozadini, <i>Event Sheet</i> za programiranje igre, <i>Run Layout</i> za testiranje; • Programiranje: odabir objekta, dizajn istih i dodjela akcija, određivanje redosljeda akcija i pokretanje testiranja.

Izvor: Izrada autora prema: Hercigonja, Z., 2017.

Daje se zaključiti kako je ovaj alat dostupan svima koji raspolažu računalom i Internetskom vezom. Besplatne inačice uglavnom su dostupne na neko kraće razdoblje, no moguće je kupiti i licencirane verzije koje cjenovno variraju. Potvrđuje se da je ovo jedan od jednostavnih, pristupačnih i alternativnih alata za usvajanje

osnovnih i naprednih znanja o programiranju.

5. Interaktivne web stranice

Za potrebe cjelovite analize predmetne problematike rada, važno je detaljnije istražiti i interaktivne web stranice kao jedan od alata za učenje i razvoj znanja te vještina u svezi programiranja. Danas postoji nekoliko ovakvih primjera, a njihov broj kontinuirano raste.

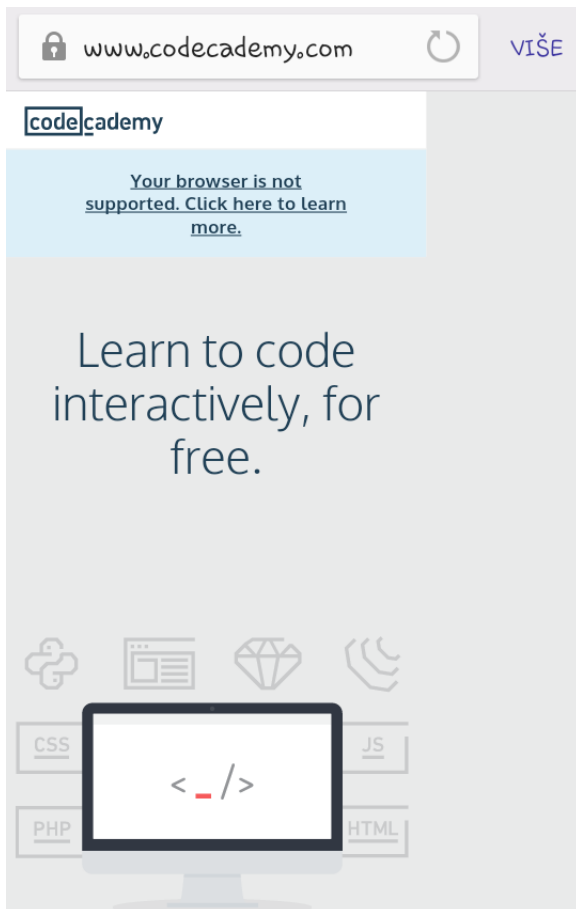
Konkretnije, misli se na interaktivne web stranice. One danas bivaju jedan od temelja u području ove znanosti, poslovanja i svakidašnjeg života pa se time potvrđuje i popularnost ovoga alata u svrhu učenja programiranja.

Interaktivna web stranica (engl. *Interactive Web Page*) je svaka web stranica koja na bilo koji način mijenja sadržaj u zavisnosti od djelovanja nekog korisnika, odnosno posjetitelja. Na njima korisnici imaju mogućnost proširiti izbornike, pokrenuti neke aktivnosti i odabrati željene sadržaje (Megabajt, 2017). Pored toga, ovaj se pojam odnosi i na one stranice čije je ponašanje određeno interakcijom posjetitelja. Učenje ovim putem često se naziva web programiranjem, a kako bi bilo jasnije o čemu je zapravo riječ, u nastavku se istražuju neke od njih.

5.1. CodeAcademy

Ova interaktivna web stranica jedna je od vodećih u kontekstu pokreta programiranja. Ona korisnicima nudi mnogo korisnih znanja u svezi programiranja, a kontinuirano se nadograđuje novima (Slika 7.).

Slika 6. Code Academy interaktivna web stranica



Izvor: Code Academy, 2017.

Zapravo je riječ o web stranici koja raspolaže s nekoliko tečajeva, a konkretno se misli na sljedeće (Code Academy, 2017):

- Osnove weba;
- PHP;
- JQuery;
- JavaScript;
- Python;
- Ruby;
- Projekti;
- API (engl. *application programming interface*).

Svaki od njih koncipiran je na način da nudi pregršt zadataka i projekata za usvajanje

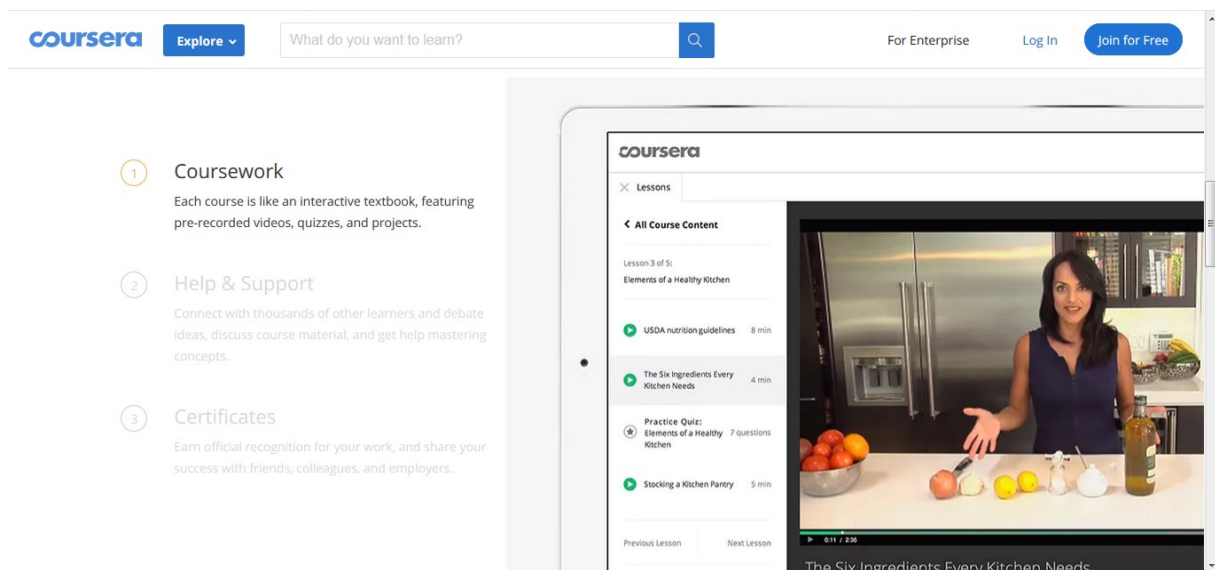
novih suvremenih znanja i vještina. Svaka od ovih cjelina podijeljena je u nekoliko dijelova. Kao primjer navodi se poglavlje Osnove weba koje čini šest dijelova. Prva tri posvećena su HTML-u kao temelju Interneta i jezika za izradu web stranica. svakog tečaja slijedi projekt za provjeru naučenoga. Jednako je i na primjeru ostalih poglavlja ili cjelina.

5.2. Coursera

Jedan od načina besplatnog online obrazovanja iz područja programiranja je i posjet interaktivnoj stranici Coursera. Ona omogućuje pohađanje preko 600 besplatnih programa, što je jedan od razloga njezine popularnosti na međunarodnoj razini. Na njoj se susreće i preko stotinu profesora raznih svjetskih sveučilišta (Srednja, 2017).

Prikaz Coursera interaktivne stranice slijedi (Slika 8.).

Slika 7. Coursera



Izvor: Coursera, 2019.

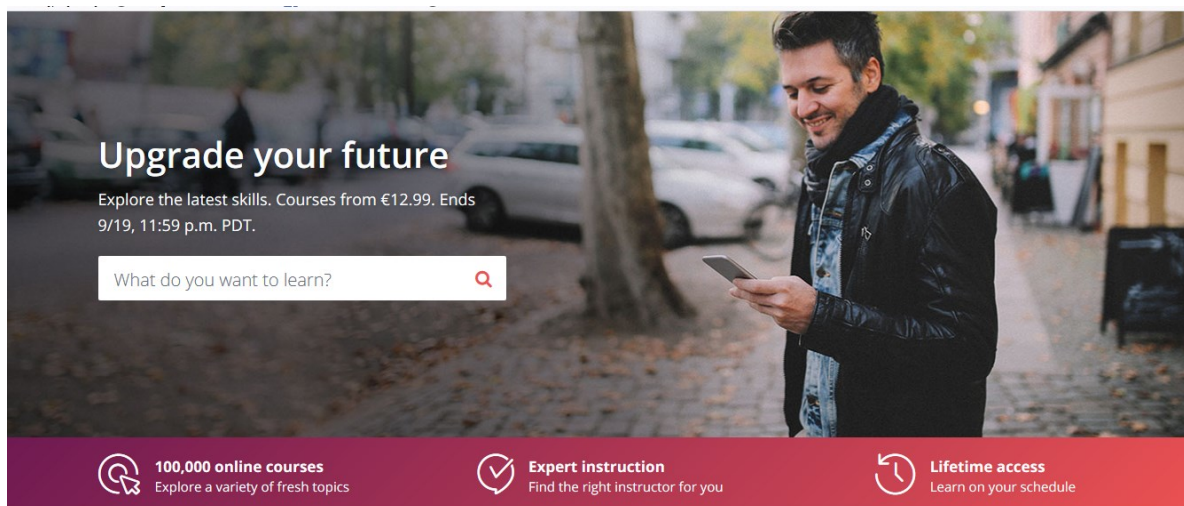
Istu čini 25 kategorija, a najveća među njima pripada humanističkim znanostima, zatim sociologiji i biologiji. Osim njih tu su i profesionalni razvoj učitelja, muzika, film, audio, inženjerstvo, informatika ili računarstvo i slično. Vidljivo je da ona biva namijenjena, ne samo programiranju i usvajanju ovih znanja, već pokriva sve znanosti.

Važno je istaknuti da njezini polaznici dobivaju priznanje o uspjehu. Tečajevi se polažu preko kolokvija, odnosno kvizova i izrade završnog rada. Komunikacija s profesorima odvija se isključivo online, a način vrednovanja uspjeha jednak je onome u tradicionalnim obrazovnim sustavima.

5.3. Udemy

Udemy je sljedeća interaktivna web stranica koja nudi pregršt praktičnih znanja o programiranju, ali i suvremenoj tehnologiji općenito. Njezin prikaz slijedi (Slika 9.).

Slika 8. Udemy



Izvor: Udemy, 2019.

Edukacija se provodi online putem osmišljenog tečaja za programere. U okviru njega uključena su sljedeća znanja i vještine (Udemy, 2017):

- Izrada web stranica korištenjem jezika HTML, CSS i JavaScript;
- Usvajanje znanja o tome zašto je programiranje potrebno i važno u suvremeno doba;
- Izrada aplikacija putem konkretnih programskih strategija;
- Identificiranje i spoznaja razlika između programera i softverskog inženjera;
- Učenje algoritama i baza podataka;
- Spoznaja o tome što kompanije traže pri zapošljavanju programera.

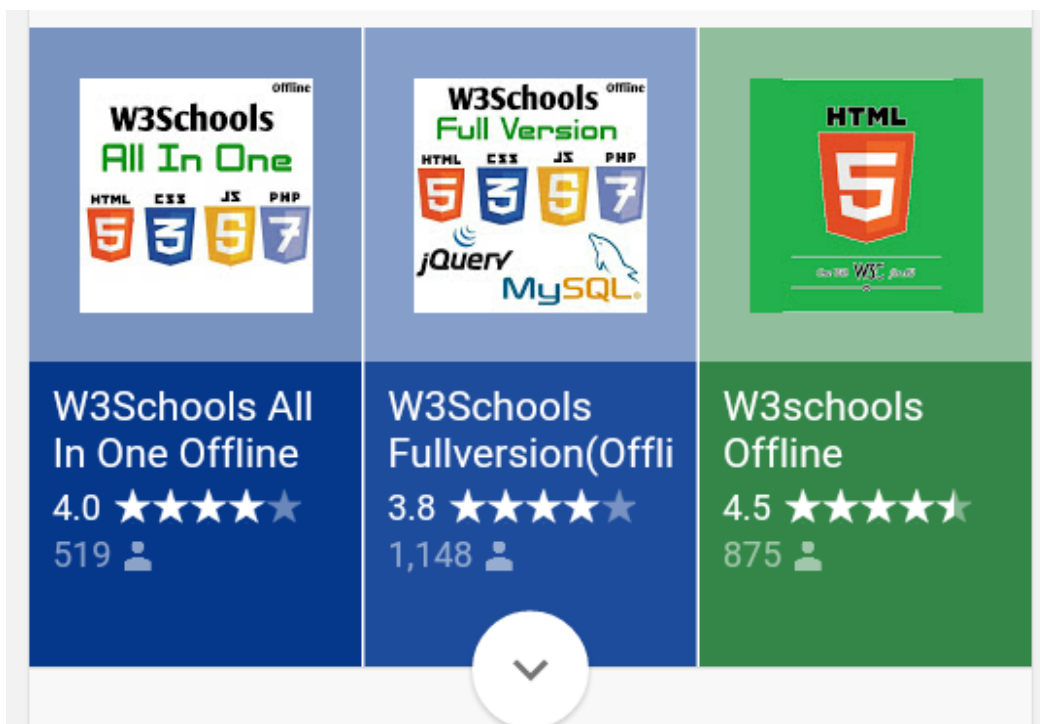
Za prijavu je dovoljno poslati životopis i eventualno dodatnu dokumentaciju, a pri kupovini ovoga tečaja dobiva se 6 satni video, neograničen pristup, pristup preko mobitela i TV-a, potvrda ili certifikat o polaganju istoga.

5.4. W3Schools

Ova interaktivna stranica priladna je za učenje, testiranje i unapređenje znanja iz područja informatika, s posebnim naglaskom na programske jezike i programiranje. Znanja koja nudi korespondentna su nekim osnovnim, ali i kompleksnim ili naprednim spoznajama o navedenome.

Ona je koncipirana na način da nudi video prikaze, preporuke ili osvrte i primjere iz nekoliko pojedinih cjelina. Vrlo se lako instalira, a prikladna je i za mobilne uređaje pa je na raspolaganju korisnicima u svako vrijeme (Slika 10.).

Slika 9. W3Schools



Izvor: W3Schools, 2017.

Strukturno ili sadržajno sastoji se od sljedećih cjelina (W3Schools, 2017):

- HTML – jezik za izradu web stranica;

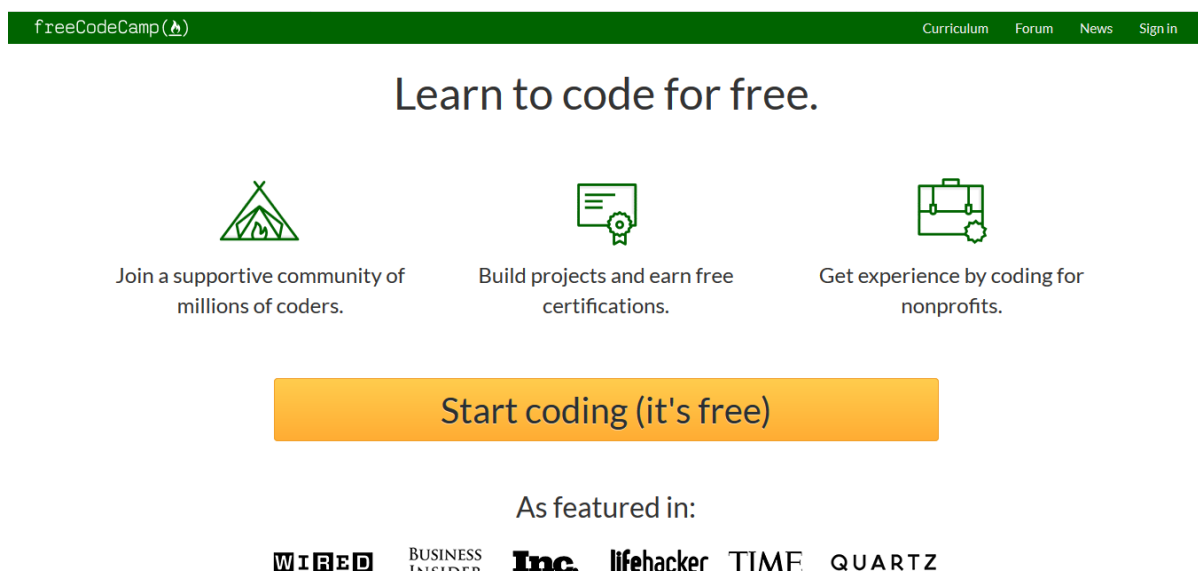
- CSS – jezik za uređenje web stranica;
- JavaScript – jezik za programiranje web stranica;
- SQL – jezik za pristup bazi podataka;
- PHP – jezik za programiranje web uslužnika;
- JQuery – razvoj web stranica;
- W3.CSS – moderan Css okvir za brže i bolje web stranice;
- Color Picker – CSS okvir za bolje dizajniranje web stranica.

Vidljivo je da je ova web stranica, kao i tečaj koji nudi, koncipiran na vrlo sličan način kao i ostali primjeri. Ono što je razliku je uključenost modernih alata za unapređenje dizajna i kvalitete web stranica.

5.5. Free Code Camp

Usluge besplatnog obrazovanja iz područja programiranja nudi i Free Code Camp. Ova stranica promiče podržavajuću društvenu zajednicu, suradnju, projektnu orijentaciju i razvoj znanja iz područja informatike, a posebice programiranja (Slika 11.).

Slika 10. Free Code Camp



Izvor: Free Code Camp, 2019.

Razvoj znanja i vještina nudi se u okviru pojedinih poglavlja ili cjelina, kao što je

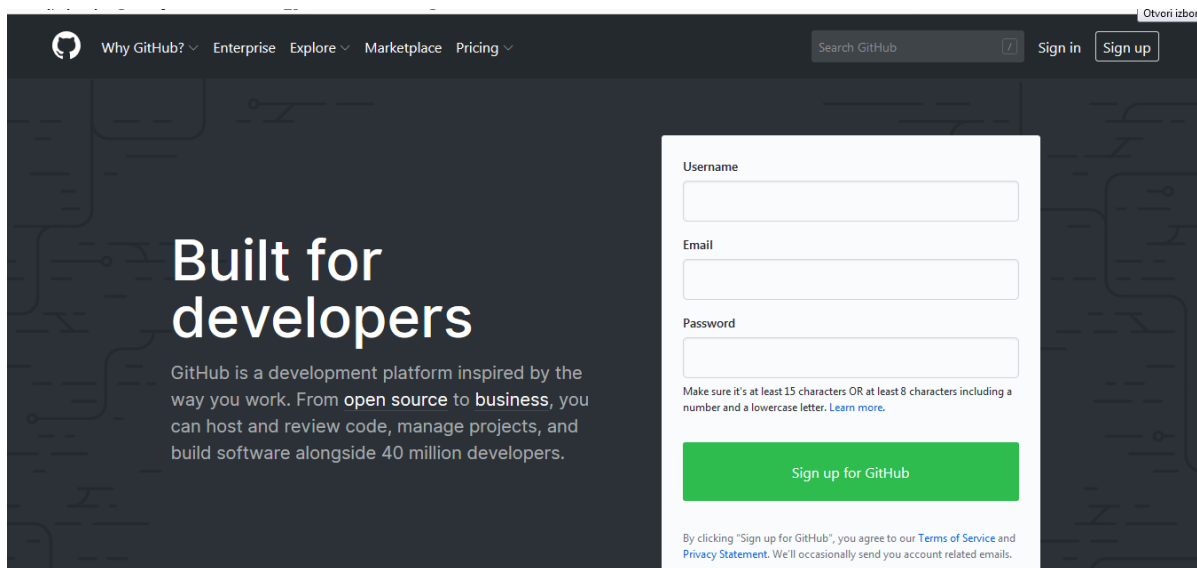
praksa i ostalih stranica. Pri tome se misli na HTML5, CSS3, JavaScript, Database, Git&GitHubNode, React, i D3 (Free Code Camp, 2017).

O uspješnosti ove stranice svjedoče javno objavljene izjave polaznika koji su usvojili kvalitetna znanja i razvili vještine u okviru ovih područja.

5.6. GitHub

GitHub je web stranica, odnosno servis za Git repozitorije koji omogućuju korištenje svih funkcionalnosti u kombinaciji sa setom ostalih mogućnosti. Točnije, ovo je mjesto na kojem svaki subjekt, polaznik može usvojiti korisna znanja iz područja programiranja, a svoj izvorni kod podijeliti s ostalim subjektima diljem svijeta (Slika 12.).

Slika 11. GitHub



Izvor: *GitHub*, 2019.

Ova platforma je vrlo cijenjena, a danas je koristi gotovo svaka tehnološka kompanija, među kojima su i Google, Twitter, Mozilla, Facebook, IBM, Dropbox ili PayPal.

GitHub služi mnogočemu. Prvenstveno, riječ je o optimalnom mjestu ili alatu za pohranjivanje koda, no i za izradu web stranica, menadžer projekte, suradnju između

programera i stvaranje zajednice ovih stručnjaka i slično.

U kontekstu GitHubovih web stranica, postoji nekoliko mogućnosti za izradu iste. Pri tome je moguće kupiti hosting i domenu te započeti njezino kreiranje od samoga početka. Osim toga, moguće je koristiti neki od postojećih servira za njezino brzo kreiranje ili pomoću ovih stranica izraditi vlastitu web stranicu koja će biti pod potpunom kontrolom korisnika (Linux za sve, 2017).

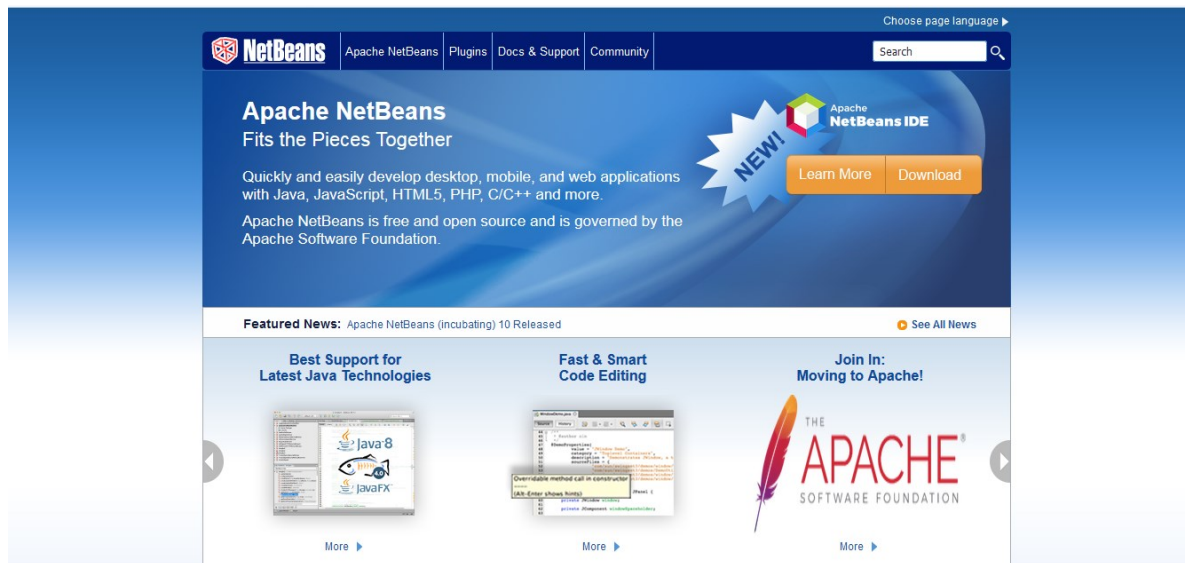
GitHub nudi i paket za studente programere i to je od osobitog značaja za ovu problematiku. Naime, preko njega, moguće je preuzeti oko 15-ak alata za programiranje koji su potpuno besplatni ili imaju veliki popust. Oni se nude od strane GitHuba i partnera, a potrebno je tek imati e-mail adresu s .edu domenom. Ono što je značajno jest da se obim putem dobiva i SSL certifikat, što znači da je stranica zaštićena i omogućen je jedino pristup preko https protokola (Linux za sve, 2017).

5.7. Net Beans

Net Beans pokriva Javu, PHP, C, C++, Ruby, HTML, CSS i JavaScript. On je potpuno besplatan na svim platformama od Windowsa do Linuxa, Solarisa i OSX-a, a njegov razvoj započeo je u Češkoj po uzoru na Delphi. Ono po čemu se ova web stranica razlikuje od ostalih je postojanje integriranog seta alata za izgradnju i razvoj grafičkih sučelja koja su bazirana na standardnim AWT i JFC/Swing komponentama (Novak, 2017).

Njegovo preuzimanje je vrlo jednostavno, a pri instalaciji moguće je izvršiti odabir između instalacijskih datoteka o kojima je bilo riječi ili je moguće izvršiti kompletnu instalaciju. Prikaz slijedi (Slika 13.).

Slika 12. Net Beans



Izvor: NetBeans, 2019.

Nova verzija Net Beansa je Net Beans 8, a razvijen je da podrži i implementira novitete uz pojavu Jave 8 i Java ME embedded 8 tehnologije, koja omogućuje rad sistema slabijih hardverskih performansi (Novak, 2017).

5.8. Android Studio i Eclipse

Android aplikacije sve intenzivnije obilježavaju suvremeno doba i suvremeno društvo. Android je poznat kao operativni sustav za mobilne uređaje koji je modificiran prema verziji Linuxa, a od 2005. godine je u vlasništvu Google-a. Međutim, on predstavlja puno više od toga. Riječ je o skupu *open source* softvera koji uključuju OS, Middleware i aplikacija, zajedno s API za pisanje mobilnih aplikacija koje određuju izgled i funkciju mobilnih uređaja.

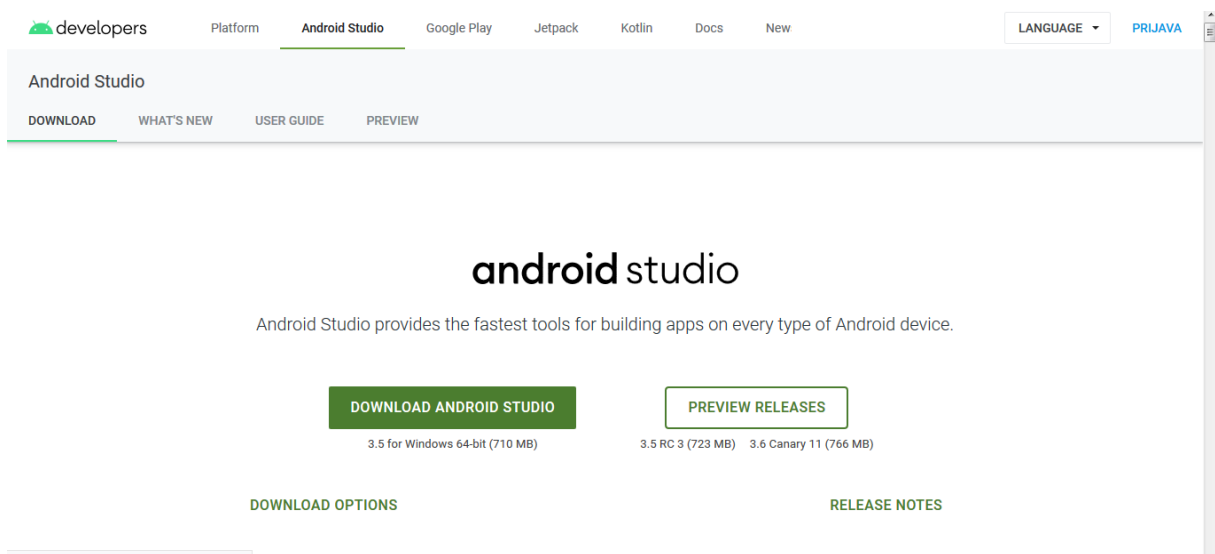
Kao takav, Android podržava sljedeće (Prirodoslovno matematički fakultet u Zagrebu, 2017):

- Bluetooth, SMS, MMS, Multitouch screen, Flash i ostalo;
- MP3, MP4, wav, midi, jpeg, png, gif i ostalo;
- WebKit preglednik;

- Fotoaparat, kameru, GPS, digitalni kompas;
- SQLite relacijsku bazi podataka.

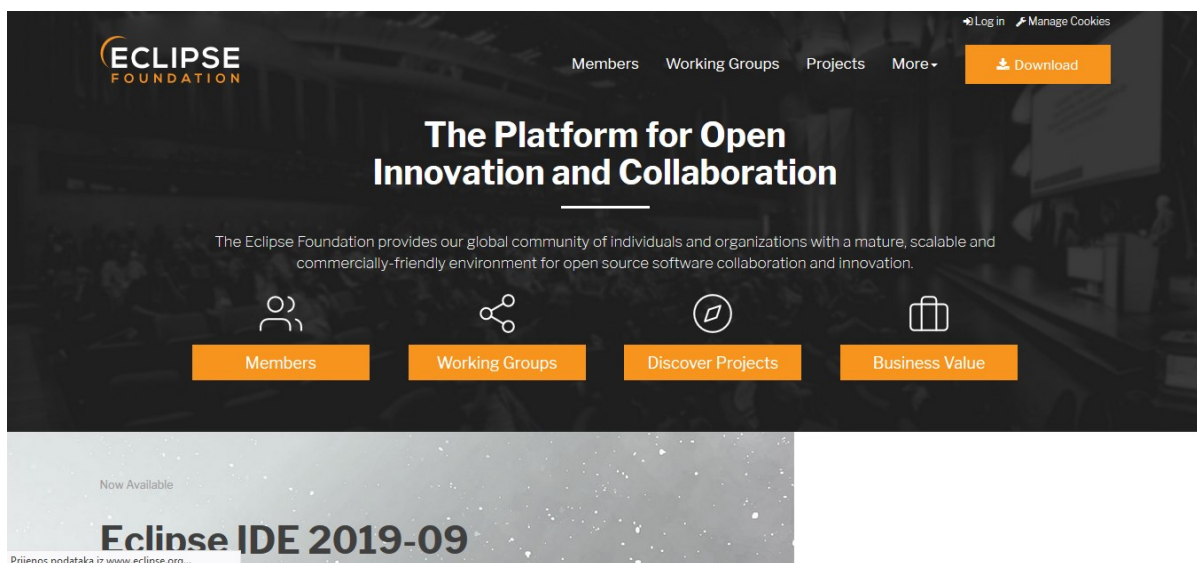
U kontekstu Androida, moguće je raspravljati još o mnogočemu poput njegove arhitekture, primjenjivosti i redom dalje. Međutim, za potrebe ovoga rada, analizira se integrirana okolina za razvoj ovih aplikacija, odnosno Android Studio (Slika 14.). On je potpuno besplatan, a često se istražuje kao opreka integrirane okoline naziva Eclipse (Slika 15.).

Slika 13. Android Studio



Izvor: Android Studio, 2019.

Slika 14. Eclipse



Izvor: Eclipse, 2019.

Pregled razlika između njih daje se u nastavku (Tablica 3.).

Tablica 3. Eclipse i Android Studio

Razlika	Specifikacija
Build alati	<ul style="list-style-type: none">• Android Studio koristi Gradle sustav buildanja;• Skriptiranje buildova i naknadna automatizacija u svrhu testiranja;• Eclipse koristi Apache Ant kao sonsonvi build sustav koji koristi robusni XML.
Napredno autozavršavanje koda i refaktorizacija	<ul style="list-style-type: none">• oba koriste standardno Java autozavršavanje koda, ali kod Android Studia postoje dodatne stavke u svezi autofaktorizacije.
Organizacija projekta	<ul style="list-style-type: none">• Eclipse projekte organizira u workspace koji se određuje kod pokretanja – ograničavajući način rada jer dopušta korištenje samo određenih projekata iz jednog workspace-a android Studio koristi module, a svaki može biti dio jednog projekta i svaki može imati svoje Grade build filove te zahtijevati pakete za ovisnost;
IDE performanse i stabilnost	<ul style="list-style-type: none">• Eclipse je razvijen u svrhu razvoja Java aplikacija i zato zahtijeva puno RAM-a i CPU-a kako bi radio bez greške;• Android Studio je još u beta verziji razvoja i nosi probleme sa stabilnosti.

Izvor: Prirodoslovno matematički fakultet u Zgarebu, 2017.

Android Studio može se besplatno preuzeti, ali zahtijeva posjedovanje Java

Development Kit-a. Upute za instalaciju javno su objavljene i specificirane, a daljnje podešavanje primarno podrazumijeva odabir Jave za korištenje. Nakon toga važno je instalirati podršku za sve verzije Androida.

6. Zaključak

U današnjici je nemoguće zamisliti život i poslovanje bez izravne podrške tehnologije i inovativnih tehnoloških rješenja kakva postoje danas. U javnosti i praksi često se današnje suvremeno društvo poima kao informatizirano i digitalizirano. Računala i informacijska tehnologija bivaju vodeći resurs ili input u svim poslovnim procesima, ali i odvijanju rutinskih, svakodnevnih životnih radnji.

Računala se sastoje od dva dijela, hardvera i softvera. Hardver čini njegovu građu, dok softver označava onaj neopipljivi dio i često se izjednačava s pojmom programa. Međutim to nije sasvim točno. Ističe se da je računalni program dio softvera, a često dobiva i prefiks najvažnijeg.

Računalni programi zapravo omogućuju rad računala i izvršenje konkretnih radnji, procesa i sličnoga. Za shvaćanje računalnih programa važno je pojmiti ulogu i važnost programskih jezika koji se klasificiraju na brojne načine. Riječ je skupu riječi i pravopisnih, odnosno sintaksnih pravila, koja se koriste za slanje naredbe računalu da izvrši neki zadatak. Upravo se u tome i očituje njihov značaj pa se i opravdava sve veći interes informatičara i ostalih dionika za njihovim istraživanjem, unapređenjem i daljnjim razvojem.

Programiranje je kompleksan proces koji se provodi u nekoliko unificiranih faza. Za razliku od nekadašnjih tradicionalnih metoda programiranja, koje su se temeljile na pukom mehaničkom ispisivanju kodova i naredbi, danas se sve više potražuju i primijenjuju alternativne metode za učenje ovoga procesa. Neki od primjera su e-učenje, računalne igrice i interaktivne web stranice.

Broj i vrste alternativnih instrumenata za učenje i razvoj vještina programiranja kontinuirano rastu, a jednaki trend očekuje se i u budućnosti. Ovi instrumenti namijenjeni su svima i lako su dostupni. Pri tome se misli da korisnici ili subjekti koji ih koriste ne moraju raspolagati s visokim i besprijekornim znanjem o informatici, matematici, engleskom jeziku ili ostalim područjima. Dovoljno je da bivaju motivirani za usvajanje i razvoj znanja te predani angažman.

Vjeruje se da ovi alati doprinose daljnjem razvoju informatike, uz poseban naglasak na programiranje, kao strateški proces suvremenoga doba. Od vodećih prednosti koje ih obilježavaju izdvajaju se laka dostupnost, nepostojanje troškova za probne verzije, primjenjivost na mobilnim uređajima i TV-u, niski troškovi licenciranih verzija, primijenjivost različitim skupinama dionika, kao što su na primjer djeca, te slično.

Daljnji razvoj ovih alata trebao bi unaprijediti pokret suvremenoga društva prema programiranju te doprinijeti brojnim poslovnim i životnim aktivnostima. Prema tome, smatra se da će sve više članova društva imati znanja i iskustva o ovome procesu, što će utjecati na ozbiljnije stvaranje društva programera u budućnosti.

Literatura

Knjige:

1. Kirriemuir, J., McFarlane, A. (2004.) Literature Review in Games and Learning. Bristol: Futurelab

Članici:

1. Markić, D. (2010.) Upotreba sustava e-učenja u nastavi informatike: programiranje u Pythonu – Diplomski rad. Zagreb: Prirodoslovno matematički fakultet u Zagrebu.
2. Miler, M.; Odošić, D. (2010.) Python – Uvod u programiranje za inženjere. Ekscentar. Br. 2. Str. 108.-110.
3. Murk, D. (2006.) Projektiranje korisničkih sučelja – programski jezik Delphi. Zagreb: Fakultet elektronike i računarstva Zagreb.

Mrežne stranice:

1. Android Studio (2019.) Android Studio. Dostupno na: <https://developer.android.com/studio>. Datum posjete: 18. rujna 2019.
2. Code Academy (2017.) Home. Dostupno na: <http://www.codecademy.com>. Datum posjete: 30. rujna 2017.
3. Coursera (2019.) Explore. Dostupno na: <https://www.coursera.org/>. Datum posjete: 18. rujna 2019.
4. Eclipse (2019.) Home. Dostupno na: <https://www.eclipse.org/>. Datum posjete: 18. rujna 2019.
5. Free Code Camp (2019.) Free Code Camp. Dostupno na: <https://www.freecodecamp.org/>. Datum posjete: 18. rujna 2019.
6. GitHub (2019.) Home. Dostupno na: <https://github.com/>. Datum posjete: 18. rujna 2019.
7. Hercigonja, Z. (2017.) Učenje programiranja grafičkim programiranjem igara. Dostupno na:

- <http://www.google.hr/amp/s/pogledkrozprozor.wordpress.com/2017/03/31/ucenje-programiranja-grafickim-programiranjem-igara/amp/?espv=1>. Datum posjete: 29. rujna 2017.
8. Infossremac (2017.) Pascal. Dostupno na: <http://www.infossremac.wordpress.com/informatika-iii-i-iv-programiranje/pascal/>. Datum posjete: 29. rujna 2017.
 9. Linux za sve (2017.) Uvod u GitHub. Dostupno na: <http://www.linuxzasve.com/uvod-u-github>. Datum posjete: 30. rujna 2017.
 10. Megabajt (2017.) Interaktivna web stranica. Dostupno na: <http://www.megabajt.org/rjecnik/interaktivna-web-stranica-dynamic-web-page/>. Datum posjete: 30. rujna 2017.
 11. Net Beans (2019.) Home. Dostupno na: <https://netbeans.org/>. Datum posjete: 18. rujna 2019.
 12. Nikola Tesla (2017.) Proces programiranja – etape. Dostupno na: <http://www.tesla.carnet.hr/mod/resource/view.php?id=5061>. Datum posjete: 28. rujna 2017.
 13. Novak, D. (2017.) NetBeans razvojnaokolina. Dostupno na: <http://www.darko-novak.from.hr/netbeans-razvojna-okolina/>. Datum posjete: 30. rujna 2017.
 14. Plavljančić, B. (2014.) Java – što, kako i zašto. Dostupno na: <http://www.pcchip.hr/softver/korisni/java-sto-kako-i-zasto/>. Datum posjete: 29. rujna 2017.
 15. Prirodnoslovno matematički fakultet u Zgarebu (2017.) Andorid. Dostupno na: <http://www.we.math.pmf.unizg.hr/android>. Datum posjete: 30. rujna 2017.
 16. Sinožić, S. (2017.) Taksonomija alata namijenjenih učenju programiranja. Dostupno na: https://www.bib.irb.hr/download/893989/Sebastian_Sinoi.pdf. Datum posjete: 18. rujna 2019.
 17. Srce (2017.) E-učenje. Dostupno na: <http://www.srce.unizg.hr/usluge/centar-za-e-ucenje/o-centru/e-ucenje>. Datum posjete: 29. rujna 2017.
 18. Srednja (2017.) Coursera: Besplatno online obrazovanje za koje se isplati izdvojiti vremena. Dostupno na: <http://www.srednja.hr/zabava/geek-kutak/tech-web/coursera-besplatno-online-obrazovanje-za-koje-se-isplati-izdvojiti-vremena-2/>. Datum posjete: 30. rujna 2017.
 19. Udemy (2017.) How to become a programmer. Dostupno na: <http://www.udemy.com/how-to-become-a-programmer/>. Datum posjete: 30.

rujna 2017.

20. Udemy (2019.) Udemy home. Dostupno na: <https://www.udemy.com/>. Datum posjete: 18. rujna 2019.
21. Wikibooks (2017.) Programiranje u BASICu. Dostupno na: http://www.wikibooks.org/wiki/Programiranje_u_BASICu/Uvod. Datum posjete: 29. rujna 2017.
22. Zovko, J. (2015.) Programiranje. Dostupno na: http://www.ef.sve-mo.ba/files/prog_10032015_1. Datum posjete: 28. rujna 2017.

Popis slika

Slika 1. Dijelovi računalnih programa	5
Slika 2. Tijek procesa programiranja.....	6
Slika 3. Glavni simboli dijagrama toga	8
Slika 4. Testiranje programa	9
Slika 5. Vrste programskih jezika.....	13
Slika 6. Code Academy interaktivna web stranica	26
Slika 7. Coursera	27
Slika 8. Udemy	28
Slika 9. W3Schools.....	29
Slika 10. Free Code Camp	30
Slika 11. GitHub.....	31
Slika 12. Net Beans	33
Slika 13. Android Studio.....	34
Slika 14. Eclipse	34

Popis tablica

Tablica 1. Aplikacije za programiranje računalnih igara – Game Maker – Studio 1.4 i Construct 2	24
Tablica 2. Eclipse i Android Studio.....	35

Sažetak

Suvremeno društvo uvelike je ovisno o tehnologiji, tehnološkim dostignućima i njezinom daljnjem razvoju. Poslovanje i svakodnevni život organiziraju se posredstvom ovoga resursa, čiji će značaj rasti i u budućnosti. Prema tome, smatra se strateškim značajem ulagati u daljnji razvoj znanja u informatici i njezinom razvoju na međunarodnoj razini.

Fukcioniranje računala i njihova primjena u praksi nezamislivi su bez postojanja programske podrške, odnosno adekvatnih računalnih programa koji upućuju naredbe i kodove računalima za izvršenje određenih radnji i procesa. Oni bivaju temelj u informatici pa im se pridaje sve veći značaj u današnjici. Kako bi se ovo područje kontinuirano unaprjeđivalo i dalje razvijalo, važno je posjedovati stručna i opsežna znanja o programiranju.

Tradicionalno programiranje sve više gubi na značaju i potražuju se neki suvremeni alati za usvajanje ovih znanja. E-učenje, računalne igrice i interaktivne web stranice tek su dio onoga što se primjenjuje u te svrhe. Broj i vrste ovih alata svakodnevno raste i pogoduju razvoju interesa brojnih sudionika za usvajanjem znanja iz područja programiranja.

Ključne riječi: suvremeno društvo, tehnologija, računalo, program, programski jezik, programiranje.

Summary

Contemporary society is very dependent on technology, innovations and technology development. Doing business and life activities are organised due this resource, which imperative importance will continue to grow in the future. Due that, the investment in the development of technology knowledge has an strategic importance on the international level.

Functioning and usement of computers in practice it is not possible without the program support, suitable programmes that send the codes and commands to the computer to execute some activity and process. To improve this area it is very important to develop wide knowledge about programming.

Traditional knowledge in this area is not so popular in contemporary time. Some of alternative instrument, such as e-learning, games and web interactive pages become more and more popular. The number of them will increase in the future which will develop the interest of all stakeholders in this area.

Key words: contemporary society, technology, computer, program, program language, programming.