Poznavanje engleskog jezika u programskom jeziku

Pićan, Lea

Undergraduate thesis / Završni rad

2015

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:641816

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Image not found or type unknown

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

zir.nsk.hr

Image not found or type unknown

Sveučilište Jurja Dobrile u Puli

Fakultet ekonomije i turizma

«Dr. Mijo Mirković»

LEA PIĆAN

POZNAVANJE ENGLESKOG JEZIKA U PROGRAMSKOM JEZIKU

Završni rad

Pula, 2015.

Sveučilište Jurja Dobrile u Puli

Fakultet ekonomije i turizma

«Dr. Mijo Mirković»

LEA PIĆAN

POZNAVANJE ENGLESKOG JEZIKA U PROGRAMSKOM JEZIKU

Završni rad

JMBAG:0303032906, redoviti student
Studijski smjer: Informatika

Predmet: Engleski jezik
Mentor: prof. dr. sc. Moira Kostić – Bobanović

Pula, lipanj 2015.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Lea Pićan kandidat za prvostupnika Informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega 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, 17. 06. 2015. _____

SADRŽAJ

Uvod 1

1 Programski jezici 2

1.1 Povijest i razvoj 3

1.2 Programski i prirodni jezik 5

2 Engleski jezik i programski jezik 6

2.1 Važnost engleskog jezika za programera 6

2.2 Programski jezici koji nisu bazirani na engleskom jeziku 8

2.2.1 Kineski programski jezici 9

2.2.2 Znakovni programski jezici 10

3 Učenje programskog jezika 11

3.1 Učenje programskog jezika za one koji ne razumiju engleski 12

4 Engleski jezik i C++ 13

4.1 Objektno orijentirano programiranje 13

4.2 Ključne riječi 14

4.3 Sintaksa 18

4.4 Programiranje u C++ 14

4.5 Semantika 19

5 Poznavanje engleskog jezika u programiranju (anketa) 20

6 Zaključak 24

Sažetak 25

Summary 26

Literatura 27

Popis slika, tablica i grafova 29

Popis priloga 29

UVOD

Programski jezik je formalni jezik za prezentaciju kompjuterskih programa. Pomoću njega se kompjuterskom sistemu, čovjeku čitljivim tekstom, daju instrukcije i naredbe za izvršenje unutarnjih zadataka, opis ulaznih i izlaznih podataka kao i njihovu strukturu u zavisnosti od unutrašnjih i vanjskih događaja.

Programski jezik C++ je programski jezik opće namjene i srednje razine s podrškom za objektno orijentirano programiranje.

Engleski jezik je jezik kojim govori preko 328 milijuna ljudi, a poznaje ga 508 milijuna širom svijeta te je kao takav „službeni“ jezik programera. O utjecaju engleskog jezika na programske jezike ne piše se mnogo.

Cilj ovoga rada je ukazati na važnost engleskoga jezika u programskom jeziku C++ ali i ostalim programskim jezicima te samoj informatici kao znanstvenoj disciplini. Ovaj rad sastoji se od pet glavnih poglavlja nakon kojih slijede zaključak i sažetak.

U prvome poglavlju objasniti ću što su to programski jezici te ukratko proći njihovu povijest te ukazati na razlike između programskog i prirodnog jezika.

U drugome poglavlju bavit ću se utjecajem engleskog jezika na programske jezike, važnosti engleskoga jezika za svakog programera te programskim jezicima koji nisu bazirani na engleskome jeziku.

Treće poglavlje govori o samom učenju programskih jezika te o razlikama u učenju kod osoba koje imaju predznanje engleskoga jezika te kod onih koji nemaju.

Četvrto poglavlje započet ću definicijom objektno orijentiranog programiranja te programiranja u C++. Nakon toga slijedi analiza ključnih riječi u odnosu na engleski jezik te sintaksa i semantika u programskom jeziku C++.

Zbog premalog broja istraživanja te materijala odlučila sam sama napraviti anketu među C++ programerima te njene rezultate smjestiti u peto poglavlje ovoga rada. Anketu sam postavila na forume a istu je ispunilo 50 sudionika.

1 PROGRAMSKI JEZICI

Svakodnevno se u životu susrećemo s problemima koji nam se nameću i koje moramo riješiti na način koji smo odabrali kao najpogodniji, bilo da je to nešto posve novo ili rutinski dio za koji ćemo iskoristiti već poznata saznanja i rješenja. Da bi automatizirali rješavanje problema možemo koristiti programske jezike.

Programiranje je davanje uputa računalu što i kako učiniti. Davanje uputa može se ostvariti pisanjem koda u nekom od programskih jezika. Programiranje je umjetnost i umijeće u stvaranju programa za računala. Stvaranje programa sadrži u sebi pojedine elemente dizajna, umjetnosti, znanosti, matematike kao i inženjeringa. Osoba koja stvara program zove se programer.

Bavljenje razvojem programske potpore iziskuje obrazovanje te stjecanje znanja i vještina u svrhu razvijanja sposobnosti apstrakcije, odnosno sposobnosti raščlanjivanja i sagledavanja problema ili zadatka s različitim pristupima glede njegovog rješavanja uz odvajanje bitnih od nebitnih svojstava.

Programer treba imati širok spektar znanja, mora biti domišljat, pronicljiv, vješt, maštovit, kreativan, uporan, tvrdoglav, sistematičan i nepokolebljiv te voljeti taj posao. Mora neprekidno pratiti nove tehnologije i učiti na vlastitim i tuđim greškama. Poznavanje engleskog jezika, matematike, fizike, tehničkog crtanja, likovnog, glazbenog i sličnih 'nepotrebnih' znanja u programiranju nije nepotrebno, cjeloživotno obrazovanje, školsko i samostalno, preduvjet je za uspješan rad dobrog programera kojemu programiranje može postati posao.

Postoji više programskih jezika za različite namjene, a neke od njih su: dizajn i administracija baza podataka, dizajn i programiranje sučelja (desktop), programiranje web aplikacija i web dizajn, programiranje mrežne programske potpore (NOS), programiranje upravljačkih programa uređaja (driver), programiranje uslužnih-pomoćnih programskih alata (utility), programiranje multimedijskih alata, programiranje i dizajn igara, simulatora, programiranje razvojnog okruženja - programerskih alata, programiranje operativnih sustava, programiranje mikrokontrolera, programiranje BIOS-a i slično.

1.1 POVIJEST I RAZVOJ

Prva računala koja su se pojavila bila su vrlo složena za korištenje. Njih su koristili isključivo stručnjaci koji su bili osposobljeni za komunikaciju s računalom. Kako su se računala počela koristiti za sve raznovrsnije namjene tako su stvarani programski jezici. Fortan koji je bio pogodan za matematičke proračune, Basic koji se vrlo brzo učio, te Cobol koji je bio namijenjen upravljanju bazama podataka.

Povijesni pregled razvoja programskih jezika

- **1957-1959**

FORTAN (Formula Translation) , LISP (List processor), COBOL (common bussines oriented language)– poslovni softver koji su se najčešće koristili za bankomate i kreditne kartice.

- **1970**

PASCAL (dobio je ime po matematičaru/fizičaru Blaiseu Pascalu) – kreator ovog programskog jezika je Niklaus Writh. Pascal se primarno koristio za učenje programiranja. Koristio ga je Skype.

- **1972**

C (baziran na njihovom prethodnom jeziku zvanom B). Kreator C-a je Dennis Ritchie iz Bell labsa. Prethodi današnjem jeziku C++. To je bio prvi jezik opće namjene te je postigao veliki uspjeh zbog više razloga:

- jednostavan za učenje
- omogućavao je modularno pisanje programa
- brzina pretvaranja koda
- korisnik je sam mogao dodavati funkcije koje mu trebaju

Programski jezik **C** dosta se mijenjao tokom godina te je u više navrata neformalno i formalno standardiziran. Prva važnija inačica poznata je pod nazivom "K&R C", što je kratica prezimena dvaju autora najpoznatijeg C priručnika "The C Programming Language", a to su Brian Kernighan i Dennis Ritchie. Kao jedan od najvažnijih jezika u povijesti komercijalne računalne industrije, C je do danas ostao jedini programski jezik prilagođen za sve računalne platforme, od malih sustava pa do mrežnih superračunala.

- **1983**

C++ (Nadograđeni C jezik, ++ označava povećanje u programskom jeziku) - Bjarne Stroustrup zajedno s drugim istraživačima u Bell Labs proširuje C dodavajući sposobnosti objektno orijentiranog programiranja, a naziv ovog novog programskog jezika je C++. Većina Internet preglednika (Google Chrome, Mozzila, Internet Expolorer) programirani su u ovom jeziku. Zbog kompliciranе sintakse, C++ često prati glas vrlo nečitljivog i kompliciranog jezika.

- **1995**

JAVA (zbog količine kave koju su popili dok su stvarali programski jezik) – kreator Jave je James Gosling. Java je stvorena te se najčešće koristi za programiranje visoke tehnologije. Velika prednost u odnosu na većinu dotadašnjih programskih jezika je to što se programi pisani u Javi mogu izvoditi bez preinaka na svim operativnim sustavima za koje postoji **JVM**[1] dok je klasične programe pisane primjerice u C-u potrebno prilagođavati platformi na kojoj se izvode. Time i bogatim skupom klasa za rad s mrežnim komunikacijama u jednom trenutku je Java bila najbolji izbor za široku lepezu mogućih aplikacija.

[1] Java Virtual Machine

1.2 PROGRAMSKI I PRIRODNI JEZIK

Programski jezik je određen skupom simbola i pravilima njihovog slaganja kojim se opisuje postupak računanja. Opis računanja u danom programskom jeziku interpretira, odnosno tumači, stroj kojeg nazivamo računalo. Poznati sinonimi za postupak računanja su algoritam i program.

Programski simbol je nedjeljiva cjelina koju računalo, odnosno tumač, "razumije" i prevodi u rad stroja. Niz simbola, složeni po pravilima programskog jezika, čine jezične izraze tog programskog jezika koji se zatim zapisuju na podatkovni medij i predaje računalu koje na svaki otkucaj sata čita i tumači jedan simbol programa zapisanog na podatkovnom mediju.

Jedan od senatora u Americi proglasio je da bi se programski jezici u školama trebali smatrati izbornim stranim jezicima[2] (poput Španjolskog, Talijanskog itd.) . Ima nekih sličnosti u učenju kodiranja (na primjer jezika kao što je C++ kojeg su predložili za „novi strani jezik“) i učenju novog jezika. Budući da je strani jezik obavezan predmet učenici bi kao alternativu mogli izabrati programski jezik koji će učiti. Na taj način zainteresiralo bi se više mladih za informatiku te programiranje.

Pisanje programa ima slična pravila kao i pisanje u običnom govornom jeziku. Programski jezik određuju tri jezične komponente:

- Leksika - niz znakova koji čini prepoznatljivu nedjeljivu cjelinu
- Sintaksa - određuje kojim redom se leksemi slažu u programske iskaze
- Semantika - određuju značenje programskih iskaza

Učenje prirodnih jezika vrši se leksičkom, gramatičkom (sintaksa i fonetika) i semantičkom analizom. Kod programskih se jezika ne razmatra fonetska struktura jezika pa se učenje gramatike svodi na sintaktičku analizu. Za opis semantičkih značajki programskih jezika najčešće je u upotrebi običan govorni jezik.

Sintaksu i semantiku programskog jezika C++ objasnit ću u četvrtom poglavlju ovog rada.

[2] http://www.wave3.com/story/24534657/senate-committee-says-computer-programming-is-foreign-language

2 ENGLESKI JEZIK I PROGRAMSKI JEZIK

Engleski jezik je jezik kojim govori preko 328.008.000 milijuna ljudi, a poznaje ga 508,000,000 širom svijeta. Engleski jezik je nastao iz jezika germanskih plemena koja su se u kasnom starom vijeku naselila na jugoistoku otoka Velike Britanije. S razvojem britanskog kolonijalnog carstva, u područjima koja je kolonizirala njen je jezik postajao dominantan, a često i službeni. U Sjedinjenim Američkim Državama na saveznoj razini ne postoji službeni jezik ali svi znaju engleski. Engleski je postao globalni jezik zbog zabave, filmova, muzike te naravno Interneta.

Na engleskom jeziku se temelje ključne riječi u informatici. Engleski povećava efektivnost studiranja informatike, programiranja, istraživanja i komunikacije na Internetu. Programskim jezicima daje razumljivost i jednostavnost.

2.1 VAŽNOST ENGLESKOG JEZIKA ZA PROGRAMERA

Danas se sve više informatičkih poslova radi u online timovima koji ne moraju nužno biti u istom gradu, čak ni na istom kontinentu. Uz odlično znanje programiranja da bi se priključili te napredovali u takvom tomu bitno je znati i engleski jezik. Dobra komunikacija predstavlja dobru suradnju, dobrog voditelja te dobro vodstvo. Engleski jezik je globalni jezik za komuniciranje i sporazumijevanje sa nekime s kime ne znate kako biste se inače sporazumjeli.

Svaki programer bi trebao znati engleski iz više razloga:

1. **Učinkovita komunikacija na sastancima**

Svojstvo svakog dobrog programera je njegova sposobnost za rad u timovima. Često se dogodi u tvrtkama za razvoj softvera da klijenti nisu istog govornog područja. Bitno je uvjeriti klijenta da će dobiti ono što je htio, a to je mnogo lakše koristeći engleske stručne izraze i dobar govor.

2. **Značaj u prezentacijama i informatičkim izrazima**

U hrvatskome jeziku dostupni su neki prijevodi engleskih informatičkih izraza no nitko ih ne koristi niti ne prihvaća. Svojstvo dobrog programera je znati engleske a ne hrvatske izraze. Tako je na primjer „programming language“ preveden u naputni jezik, „text file“ kao orječna datnica i slično. Da bi se sastavila dobra prezentacija o programskome jeziku potrebno je koristiti nazive koje koristi i program te ih je sasvim nepotrebno prevoditi.

3. **Svaki programer treba moći napisati dokumentaciju za svoj program koju će tržište razumjeti**

Dokumentiranje programskih rješenja je jedno od ključnih aktivnosti programskog inženjerstva. Dokumentacija programskog projekta ima višestruke uloge. Na primjer, može služiti kao komunikacijski medij među članovima razvojnog tima ili kao priručnik koji će korisnici moći koristiti prilikom rada i administracije sustava. Također, može poslužiti kao jedinstveni repozitorij informacija razvojnim inženjerima koji održavaju sustav.

Često se koristi kao izvor informacija menadžmentu koji će na temelju tih informacija donositi odluke vezane uz budžet, poslovni plan i daljnju razradu rasporeda razvoja projekta. Dobra praksa je imati dvojezičnu dokumentaciju. Dokumentacija na engleskome jeziku povećat će tržište te omogućiti lakše korištenje softverskoga rješenja korisnicima cijeloga svijeta.

4. **Može se efektivnije snalaziti na Internetu**

Kod programiranja te općenito informatike nerijetko se dolazi do problema koje je već netko riješio te objavio svoje rješenje na informatičkim forumima. Dozvoljeno je koristiti ta rješenja u vlastitim programima. Na stranicama kao što je Stack overflow[3] (trenutno najpopularnija stranica sa pitanjima i odgovorima za programere) moguće je pronaći rješenja gotovo svih problema. Stranica funkcionira na način da netko postavi pitanje na koje bilo tko može odgovoriti. Svi dobri odgovori pozitivno se ocjenjuju. Odgovor s najviše pozitivnih ocjena prikaže se prvi a zatim ostali poredani po broju ocjena. Sva komunikacija odvija se na engleskome jeziku. Znatko[4] je hrvatska verzija takve stranice no broji puno manje članova te odgovora.

5. **Literatura na engleskom jeziku**

Pri učenju programskog jezika teško je doći do literature pisane na hrvatskom jeziku. Sve knjige i novija istraživanja pisana su na engleskom jeziku te se vrlo malo te materije prevodi. Ukoliko budući programer ne razumije engleski, bit će mu vrlo teško doći do literature.

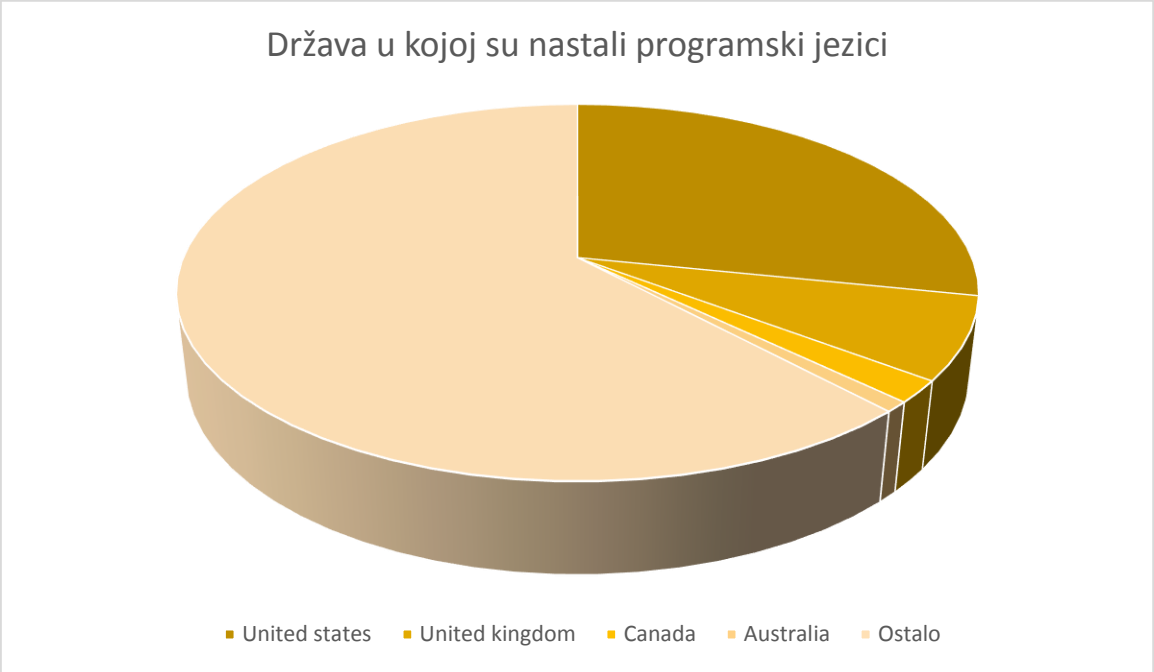
[3] http://stackoverflow.com/

[4] http://znatko.com/

2.2 PROGRAMSKI JEZICI KOJI NISU BAZIRANI NA ENGLESKOM JEZIKU

Programski jezici koji nisu bazirani na engleskom su programski jezici koji ne koriste ključne riječi uzete ili inspirirane engleskim rječnikom. Trend u izradi programskih jezika je da se ključne riječi uzimaju iz engleskoga jezika. Otprilike postoji više od 8500 programskih jezika od kojih je više od jedne trećine nastalo u državama u kojima je engleski primarni jezik. U državama kojima engleski nije materinji jezik najčešće programski jezici imaju engleske ključne riječi.

Graf 1: Vizualni prikaz programskih jezika po državi u kojoj su stvoreni

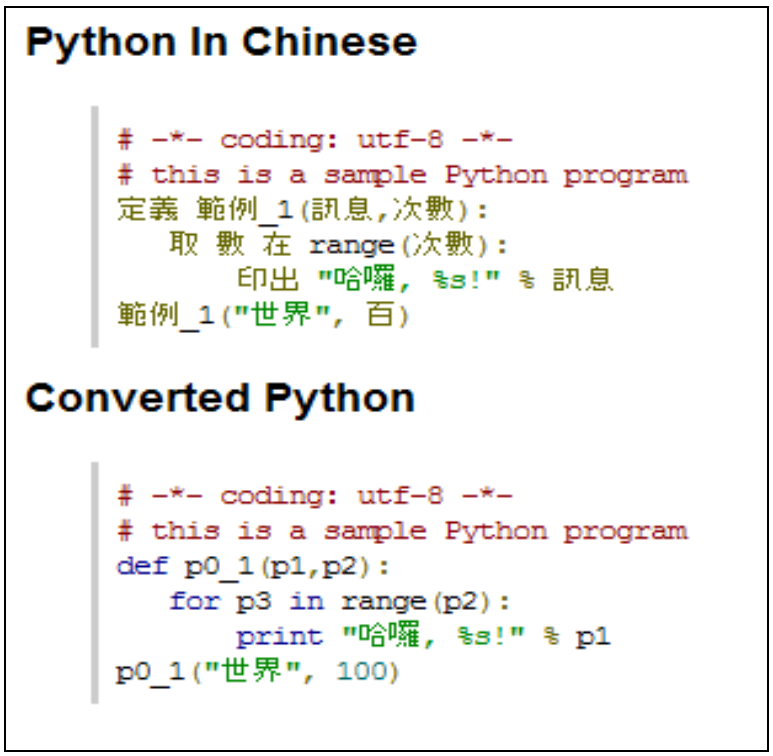


Izvor: http://www.levenez.com/lang/

2.2.1 Kineski programski jezici

Osim engleskog, postoje programski jezici koji koriste ključne riječi iz ostalih jezika. Jedan od najzastupljenijih je i kineski jezik. Zbog posebnog pisma u Kini nužno je bilo prevesti programske jezike na kineski. Kineski programski jezici nastali su da bi se olakšala proizvodnja softvera na kineskom jeziku te da bi se približili programski jezici i logika onima koji ne znaju engleski jezik.

Slika 1: Razlika između programskoga jezika baziranog na engleskom i programskoga jezika baziranog na kineskom



Izvor: http://musicsophist.com/wp-content/uploads/2012/12/zhpy1.bmp

EPL (easy programming language) je kineski programski jezik. EPL broji najveću zajednicu programera koji nisu iz engleskog govornog područja. EPL je popularan u Kini budući da je kinezima teško naučiti engleski jezik zbog prevelike različitosti.

2.2.2 Znakovni programski jezici

Programski jezici mogu umjesto ključnih riječi koristiti znakove. Simbol je u programskom jeziku primarni tip podataka. Simboli se mogu koristiti za identifikaciju. Takvi jezici najčešće koriste matematičke simbole te se koriste za izračune. Osim njih postoje i manje korisni „Esoteric programming languages“ koji su stvoreni za eksperimentiranje sa čudnim idejama ili jednostavno kao šala. Prioritet takvim jezicima je „beskorisnost“ i zabava.

Neki od programskih jezika koji koriste simbole umjesto ključnih riječi:

- APL (A programming language) - je interaktivni redno-orijentirani programski jezik i integrirano razvojno okruženje koje je na raspolaganju mnogobrojnih komercijalnih i nekomercijalnih prodavača za većinu računarskih platformi.

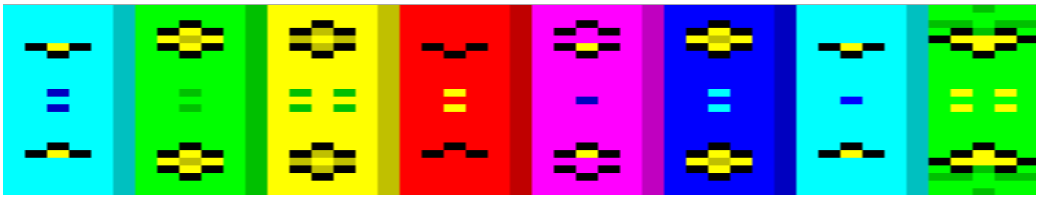
Slika 2: APL programski jezik

```
[6]    L←(L⍳':')↓L←,L          ⍝ drop To:
[7]    L←LJUST VTOM',',L       ⍝ mat with one entry per row
[8]    S←¯1++/∧\L≠'('          ⍝ length of address
[9]    X←0⌈⌈/S
[10]   L←S⍴(-(⍴L)+0,X)↑L       ⍝ align the (names)
[11]   A←((1↑⍴L),X)↑L          ⍝ address
[12]   N←0 1↓DLTB(0,X)↓L       ⍝ names)
[13]   N←,'α',N
[14]   N[(N='_')/⍳⍴N]←' '      ⍝ change _ to blank
[15]   N←0 ¯1↓RJUST VTOM N     ⍝ names
[16]   S←+/∧\' '≠⍴N            ⍝ length of last word in name
```

Izvor: http://www.chilton.com/~jimw/a2ap1g1.gif

- Piet – ezoterični programski jezik. To je programski jezik čiji programi izgledaju kao apstraktne slike. Jezik je nazvan po Pietu Mondrianu koji je doktorirao umjetnost geometrije.

Slika 3: primjer programa u Pietu



Izvor:http://www.retas.de/thomas/computer/programs/useless/secunet_contest/entry_3/s2_8.gif

3 UČENJE PROGRAMSKOG JEZIKA

Danas smo okruženi nevjerojatnim brojem različitih programskih jezika i teško je razabrati koji od njih je najbolji. Oni koji tek ulaze u programerske vode htjeli bi saznati koji im je jezik najisplativije učiti, no to pitanje teško može dobiti krivi ili točan odgovor. U osnovnim školama učenje programiranja počinje programskim jezikom Logo, u srednjim školama Pascalom te na fakultetima C++.

Učenje programskoga jezika lako je i zabavno ako imamo razvijenu logiku te barem osnovno znanje engleskoga jezika. Programiranje se može naučiti u školi, na fakultetu te na raznim tečajevima. No osim toga, programiranje se može učiti i doma uz pomoć Interneta a taj način je mnogo efikasniji.

1. Code academy (http://www.codecademy.com/) - Codecademy je interaktivna web stranica gdje vi uživo programirate i izvršavate zadatke. Jako zanimljiva i korisna web stranica, odličnog dizajna i koncepta.
2. Web monkey (http://www.webmonkey.com/) - Odlični tutorijali i lijepo organizirana stranica koja sadrži sve najbitnije programske jezike.
3. Dream in code (http://www.dreamincode.net) - Tutorijali i diskusije na tu temu. Odlično za razmjenu iskustva sa ostalim članovima.

Stariji programeri dati će vam savjet da se radije fokusirate na to da pohvatate sve koncepte programiranja. Pisanje koda samo po sebi iziskuje iskustvo u određenim problemskim situacijama, što će mnogi procijeniti kao najvažniji čimbenik. S druge strane, mnogi smatraju da programski jezik nije toliko bitan jer kada znate kako programirati onda ćete se ‘lako’ prebaciti iz jednog jezika u drugi.

3.1 UČENJE PROGRAMSKOG JEZIKA ZA ONE KOJI NE RAZUMIJU ENGLESKI

Većina programerskih kolegija ili tečajeva nude učenje programskih jezika koji su bazirani na engleskom jeziku te koriste engleske ključne riječi. Mnogo je znanstvenih ustanova u Europi prihvatilo engleski kao službeni jezik programiranja te svoja predavanja tome prilagođavaju.

Učenje programiranja teško je za studente koji ne razumiju ključne riječi pa samim time imaju poteškoća sa praćenjem nastave te pamćenjem što koja ključna riječ znači. Stoga se preporuča da studenti imaju barem osnovno predznanje engleskoga jezika te znanje ključnih riječi kako bi imalo bolje rezultate.

Engleski jezik postao je internacionalni jezik komunikacije, znanosti, informacijske tehnologije i poslovanja. Mnogo profesija uključujući i informatiku zahtijevaju radnu snagu sa znanjem engleskoga jezika.

Prilikom učenja programskoga jezika ukoliko se ne zna engleski jezik najbolje je prvo završiti tečaj engleskog jezika u informatici te zatim početi s učenjem programiranja.

4 ENGLESKI JEZIK I C++

Programski jezik C++ je programski jezik opće namjene i srednje razine s podrškom za objektno orijentirano programiranje. Prvotno je razvijan u Bell Labs (laboratoriju telekomunikacijske tvrtke Bell) pod rukovodstvom Bjarnea Stroustrupa tokom 1980-ih, i to kao proširenje programskom jeziku C pa mu je originalno ime bilo "C with classes" (engl. C s klasama). Zbog velike potražnje za objektno orijentiranim jezicima te izrazitim sposobnostima istih, specifikacija programskog jezika C++ ratificirana je 1998. kao standard ISO/IEC 14882.

4.1 OBJEKTNO ORIJENTIRANO PROGRAMIRANJE

Objektno orijentirano programiranje ili kraće OOP je jedan od mogućih pristupa programiranju računala. Za razliku od ostalih pristupa, u kojima je težište na akcijama koje se vrše na podatkovnim strukturama, ovdje je težište na projektiranju aplikacije kao skupa objekata koji izmjenjuju poruke.

U OOP metafori osnovna jedinica je objekt, što je konkretna realizacija klase. U objektno orijentiranom svijetu izbjegava se korištenje globalnih varijabli, nego svaki objekt ima svoje varijable, koje još zovemo i podatkovni članovi klase. Isto tako nema više ni samostojećih funkcija, nego funkcije pripadaju klasi, bilo da vanjskom svijetu nešto kazuju o stanju objekta, ili mijenjaju stanje objekta, a zovemo ih funkcijski članovi klase ili metode.

Čest je slučaj da se neki objekti minimalno razlikuju, i zasebno definiranje svakog od njih je redundantno. Osim toga za samo efikasno projektiranje praktično je pojednostavljivanje konkretnog problema. Kad već definiramo neki objekt, a zatreba nam neki sličan objekt koji je zapravo podskup početnog objekta, moguće je naslijediti početni objekt, čime štedimo vrijeme za programiranje.

4.2 PROGRAMIRANJE U C++

Računalni program je skup uputa računalu što treba učiniti i kako to izvesti. Postoje razne vrste programa ovisno o zadatku koji želimo da program izvrši. Tako postoje tekst procesori za pisanje teksta, proračunske tablice za izradu tablica, baze podataka i dr.

Postoje četiri faze izrade programa[5]:

1. **Pisanje izvornog koda** – izvorni kod može se pisati u bilo kojem programu za uređivanje teksta međutim većina današnjih prevoditelja i povezivača isporučuje se kao cjelina s programom za upis i ispravljanje koda te su poznati pod nazivom *integrirane razvoje okoline* (integrated development enviroment IDE).
2. **Prevođenje izvornog koda** – prevoditelj prilikom prevođenja provjerava sintaksu izvornog koda te u slučaju pogreške ispisuje odgovarajuće poruke o pogreškama
3. **Povezivanje u izvedbeni kod** – nakon što je kod ispravno preveden pristupa se povezivanju objektnih kodova i izvedbeni. Objektni kod treba spojiti s bibliotekama (datoteke u kojima se nalaze već gotove prevedene funkcije). Nakon što se isprave sve pogreške kod treba opet ispraviti i provjeriti.
4. **Testiranje programa** – iako je kod sad provjeren to i dalje ne znači da će program raditi ono što smo mi zamislili. Postoje programi za debagiranje iako kod možemo i ručno testirati.

Kao i za svaki drugi posao, pisanje programa zahtjeva određeno znanje i vještinu. Uoči li se pogreška u nekoj od faza izrade programa izvorni kod se treba doraditi i ponoviti sve prethodne faze. Zbog toga postupak izrade programa nije pravocrtan nego se stalno vrtimo u krug.

Da bi se za neki program moglo reći da uspješno završen treba proći uspješno kroz sve četiri faze.

[5] Šribak, Julijan; Motik, Boris. DEMISITIFICIRANI C++. 3. dopunjeno izdanje. Zagreb: Element d.o.o. 2010.

4.3 KLJUČNE RIJEČI

Svaki programski jezik koristi vlastiti, ograničeni skup riječi koje imaju posebna značenja. Takve riječi nazivamo ključnim riječima. Tablica prikazuje analizu ključnih riječi u programskom jeziku C++. Može se uočiti kako je engleski jezik uvelike utjecao na izradu ključnih riječi ovoga jezika.

Tablica 1: Analiza ključnih riječi programskoga jezika C++

KLJUČNA RIJEČ	ENGLESKA RIJEČ	ZNAČENJE RIJEČI
ASM	assembly	ubacivanje asembler instrukcije
AUTO	automatic	deklariranje lokalne varijable
BOOL	boolean	deklariranje logičke varijable
BREAK	break	izađi iz petlje
CHAR	character	deklariranje znakovne varijable
CLASS	class	deklariranje klase
CONST	constant	delariranje nepromijenjive varijable
CONTINUE	continue	nastavlja petlju
DELETE	delete	odlobađa memoriju
DO	do	naziv petlje
DOUBLE	double	delkariranje duple cijelobrojne varijable
ELSE	else	alternativno riješenje if petlje
EXTERN	externally	deklariranje varijabli iz drugog programa
FALSE	false	logička vrijednost za laž
FLOAT	float	deklariranje float varijable
FOR	for	naziv petlje
GOTO	go to	skoči na drugi dio programa

IF	if	naziv petlje
INT	integer	deklariranje cijelobrojne varijable
LONG	long	deklariranje duge cijelobrojne varijable
NAMESPACE	name space	deklaracija elemenata standardne C++ biblioteke
NEW	new	otvara novo mjesto u memoriji
PRIVATE	private	privatni članovi klase
PROTECTED	protected	zaštičeni članovi klase
PUBLIC	public	javni članovi klase
RETURN	return	povratak iz funkcije
SHORT	short	deklariranje kratke cijelobrojne varijable
SIGNED	signed	modifikacija deklaracije tipa varijable
SIZEOF	size of	vraća duljinu varijable
STATIC	statical	kreira parametar mjesta pohrane za varijablu
STRUCT	structural	kreiranje nove strukture
SWITCH	switch	naziv petlje
TEMPLATE	template	kreira generičke funkcije
THROW	throw	ubacuje iznimku
TRUE	true	logička vrijednost za istinu
TRY	try	izvršava iznimku
TYPEDEF	type definition	kreiranje imena tipa podataka
TYPEID	type id	opis objekta
USING	using	deklarianje namespace-a
VOID	void	delkariranje funkcije koja ne vraća rezultat
WHILE	while	naziv petlje

\N	new row	novi redak
\T	tabulator	horizontalni tabulator
\V	vertical tabulator	vertikalni tabulator
\R	carriage returm	povrat na početak reda
\A	alert	zvučni signal
\B	backspace	pomak za mjesto unazad
\F	form feed	nova stranica

4.4 SINTAKSA

Za svaki su programski jezik propisana pravila slaganja ključnih riječi u naredbe. Takva se pravila nazivaju sintaksa. Ako se ne zadovolji propisana sintaksa, program će biti neispravan i neće se moći izvršiti. Najraširenija formalna notacija sintakse je „Extended Backus-Naur Form (EBNF)“. Počinje se s entitetom najvećega nivoa entiteta te se onda koriste naredbe za dekompozaciju entiteta . Na primjer if petlju bi definirali na slijedeći način

If (uvjet) naredba [else naredba]

Što znači ako je uvjet zadovoljen izvrši prvu naredbu, inače izvrši drugu naredbu.

Mnogo programera se bori sa sintaksom nekog jezika no sintaksa je ipak najmanje važan dio programskoga jezika. Lako je naučiti sintaksu koju možemo razumjeti. Greške u sintaksi su česte ali često će nas i naš kompajler[6] upozoriti na njih. Ovo su najčešće greške u sintaksi:

1. Treba se paziti na dužinu varijabli. Gleda se samo prvih 10 znakova, stoga ako napišemo current_winner i current_width imat će istu identifikaciju.
2. Puno programskih jezika uključujući i C++ su osjetljivi na slova što znači da current_win i CURRENT_WIN nije isto.
3. Uvijek treba provjeriti kako pišemo komentare u programskom jeziku, u C++ učiniti ćemo to ovako */* *komentar* **/*.
4. Treba paziti i na znak jednakosti. *(a=b)* postavlja vrijednost **b** da bude jednaka vrijednosti **a**. *(a==b)* provjerava da li je vrijednost **a** identična vrijednosti **b**.
5. Osim ako koristimo naredbu za ispis te u navodnike stavimo razmake, programski jezici će ih obrisati, neće ih prepoznati kao razmake već kao nepotrebno prazno mjesto.

Uobičajeno se kaže da gramatika programskih jezika spada u klasu bez kontekstne gramatike. Za opis sintakse nekog jezika koristi se posebni jezik koji se naziva meta jezik. Jezik koji se opisuje meta jezikom naziva se ciljni jezik.

[6] programski prevodilac, eng. compiler

4.5 SEMANTIKA

U teoretskom računarstvu, formalna semantika je disciplina koja se bavi rigoroznim matematičkim proučavanjem značenja programskih jezika i modela računanja. Formalna semantika jezika je dana matematičkim modelom koji opisuje moguća računanja koja jezik opisuje. Postoje mnogi pristupi formalnoj semantici, koji se mogu razvrstati u tri glavna razreda:

Denotacijska semantika u kojoj je svaka fraza jezika prevedena u denotaciju, tj. frazu u drugom jeziku. Denotacijska semantika slabo odgovora kompilaciji, iako je "ciljni jezik" obično matematički formalizam umjesto nekog drugog računalnog jezika.

Operacijska semantika, u kojoj se izravno opisuje izvršavanje jezika (umjesto prevođenjem). Operacijska semantika slabo odgovora interpretaciji, iako je opet "jezik izgradnje" interpretera općenito matematički formalizam. Operacijska semantika može definirati apstraktni stroj (kao što je SECD stroj) i dati značenje frazama opisujući prijelaze koje one induciraju nad stanjima stroja. Alternativno, kao sa čistim lambda računom, operacijska semantika može biti definirana sintaksnim pretvorbama nad frazama samog jezika.

Aksiomatska semantika, u kojoj se značenje daje frazama opisujući logičke aksiome koji se na njih primjenjuju. Aksiomatska semantika ne razlikuje značenje fraze i logičkih formula koji je opisuju; njezino značenje jest točno ono što o njoj može biti dokazano u nekoj logici. Kanonski primjer aksiomatske semantike jest Hoareova logika.

Razlika između triju širokih razreda pristupa može nekad izgledati mutna, ali svi poznati pristupi formalnoj semantici koriste gornje tehnike, ili neku njihovu kombinaciju. Osim izbora između denotacijskog, operacijskog i aksiomatskog pristupa, većina varijacija u sustavima formalne semantike izniče iz izbora podržavajući matematički formalizam.

Područje proučavanja formalne semantike uključuje sljedeće:

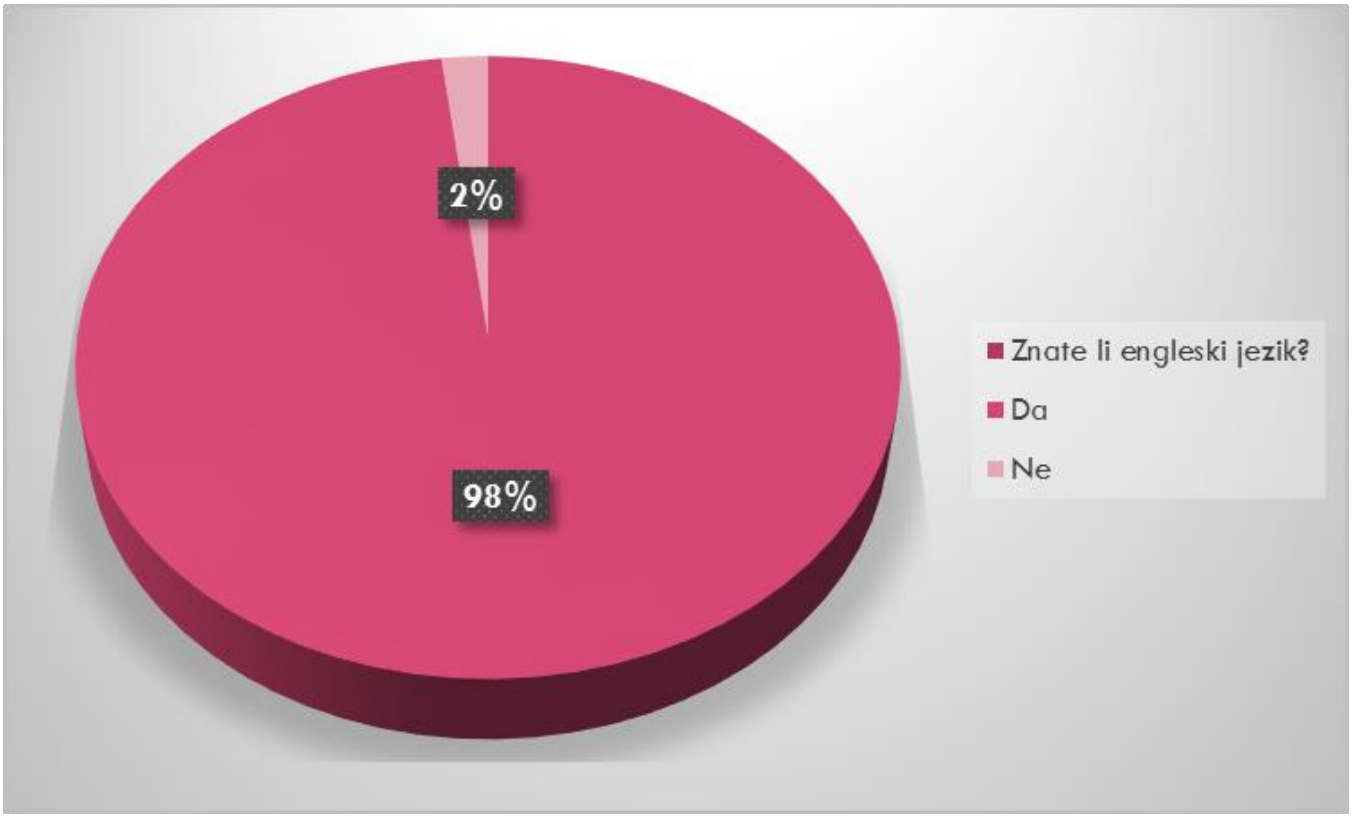
- definiciju semantičkih modela
- odnose između različitih semantičkih modela
- odnose između različitih pristupa značenju
- odnose između računanja i temeljnih matematičkih struktura iz polja kao što su logika, teorija skupova, teorija modela, teorija kategorija

5 POZNAVANJE ENGLESKOG JEZIKA U PROGRAMIRANJU (ANKETA)

Izradila sam te analizirala anketu na temu poznavanja engleskog jezika u programiranju. Anketa je postavljena za ciljanu skupinu ljudi – programera C++ jezika. Sveukupno je anketi pristupilo pedeset ispitanika hrvatskoga govornog područja.

Prvo pitanje je bilo znate li engleski jezik, od 50 ispitanika njih 49 se izjasnilo da zna engleski jezik a samo jedan da ne zna.

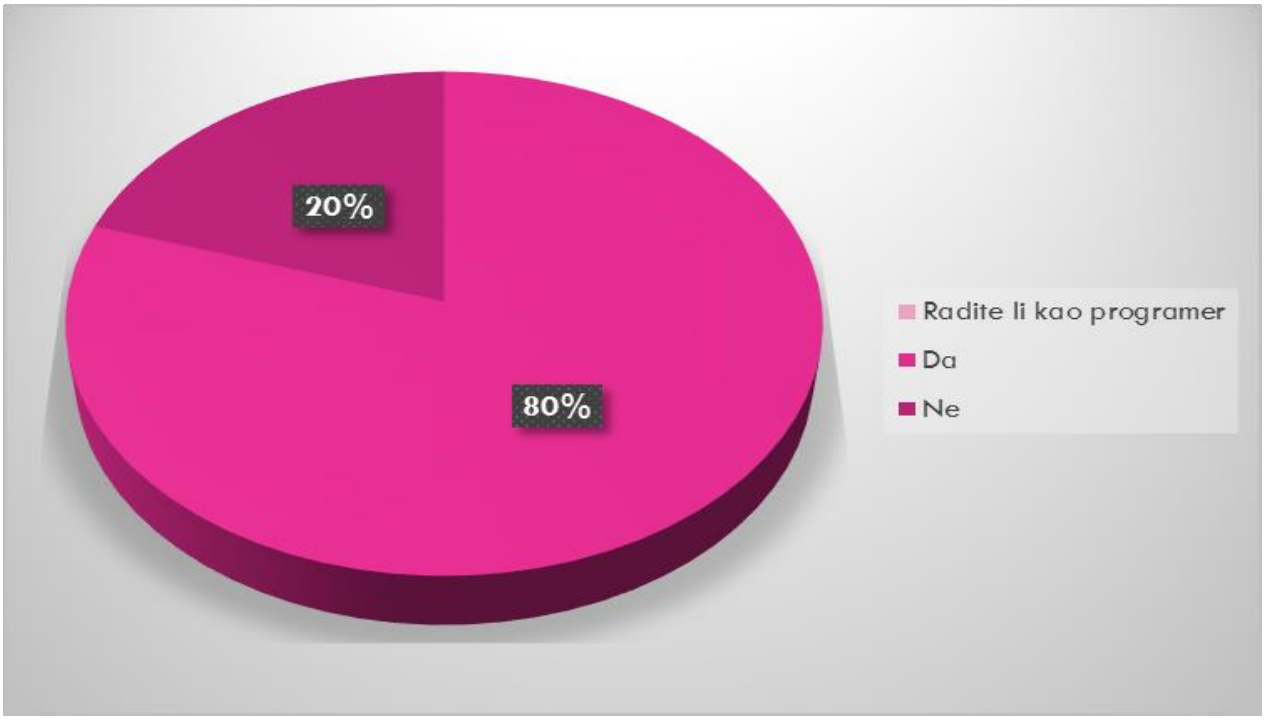
Graf 2: Znate li engleski jezik?



Izvor: analiza ankete

Od 50 ispitanika 80% rade kao programeri dok se ostalih 20% bavi programiranjem iz hobija. Na pitanje da li im je engleski jezik pomogao pri učenju programiranja svi su odgovorili potvrdno. Najčešći odgovori bili su da im je engleski jezik pomogao pri čitanju literature koja je većinom dostupna samo na engleskome jeziku, prilikom razumijevanja ključnih riječi, razumijevanjem sintakse, mogućnosti suradnje s ostalim programerima i aktivnosti na forumima. 60% ispitanika preferira programiranje u C++ programskom jeziku dok ostalih 40% preferira programiranje u Javi.

Graf 3: Radite li kao programer?



Izvor: Analiza ankete

Nakon općih pitanja uslijedila su pitanja u kojima su ispitanici trebali ocjenom od jedan do pet (po ljestvici nevažan (1), dovoljno važan (2), važan(3), vrlo važan(4), iznimno važan(5)) ocijeniti koliko je bitan engleski jezik u programiranju te pojedinim aspektima programiranja.

Na pitanje koliko je engleski jezik važan za programere 49% je odgovorilo je da iznimno važan, 36% da je vrlo važan te 18% da je dovoljno važan.

Važnost engleskoga jezika u pronalasku programerskog posla 38 ispitanika ocijenilo iznimno važnim te 10 vrlo važnim.

Važnost engleskog jezika za kodiranje 30 ispitanika je ocijenilo iznimno važnim, 9 vrlo važnim te ostali nižim ocjenama.

Za razumijevanje programskog jezika 27 ispitanika smatra da je iznimno važan engleski jezik, njih 12 da je vrlo važan, sedmero da je važan dok ostalih troje ispitanika misle da je nevažan.

Engleski jezik u timskom radu među programerima svi ispitanici ocijenili su iznimno važnim ili vrlo važnim.

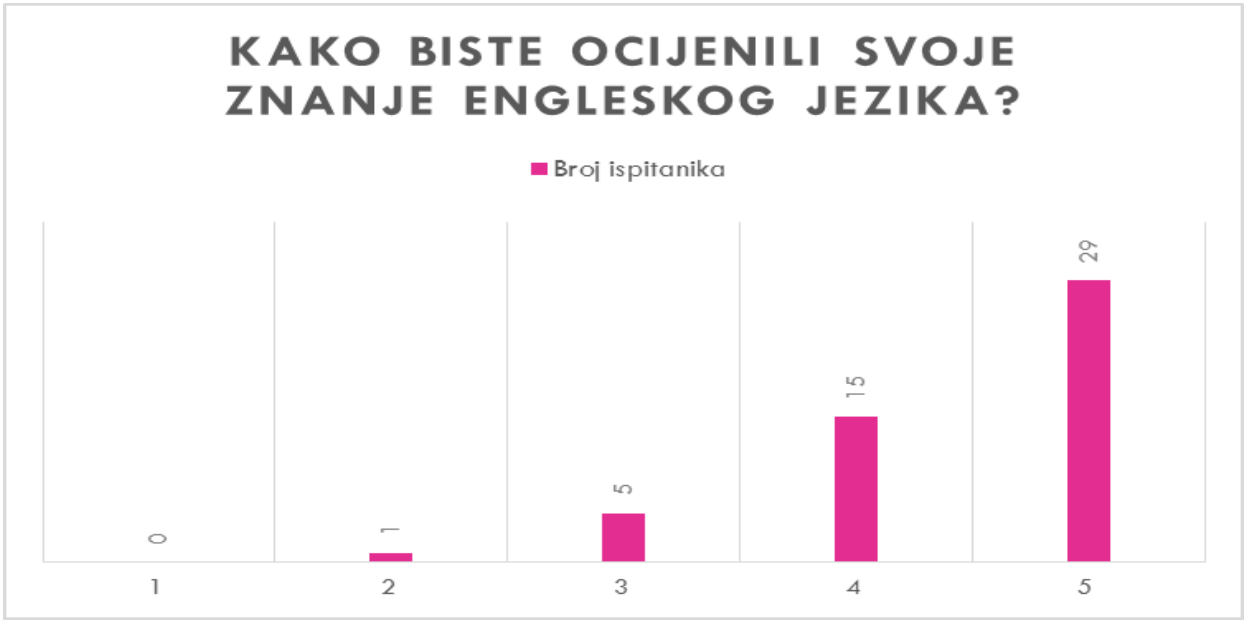
Graf 4: Koliko je engleski jezik važan za programere?



Izvor: analiza ankete

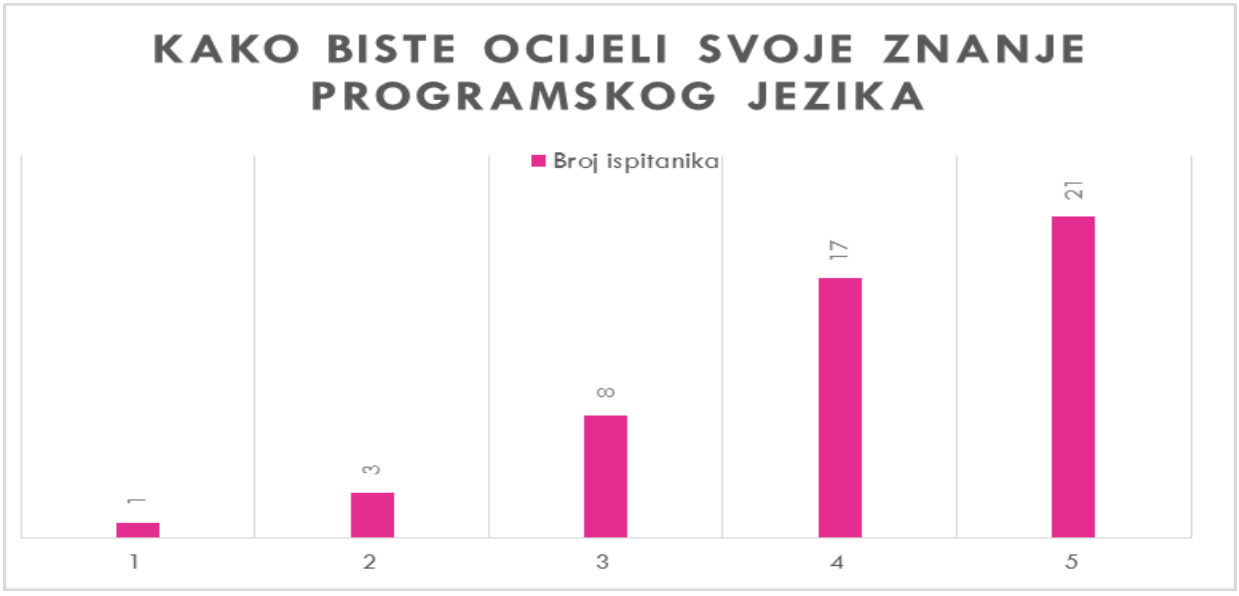
Kroz zadnja dva pitanja htjela sam vidjeti da li znanje engleskoga jezika utječe na efikasnost u programskog jezika. Kao što možemo vidjeti grafovi 5 i 6 gotovo su isti iz čega možemo zaključiti da ispitanici koji su izvrsni u engleskome jeziku postižu izvrsne rezultate u programiranju.

Graf 5: Kako biste ocijenili svoje znanje engleskog jezika?



Izvor: analiza ankete

Graf:6: Kako biste ocijenili svoje znanje programskog jezika?



Izvor: analiza ankete

6 ZAKLJUČAK

Posao programera je danas jedan od najtraženijih poslova. Dobar programer treba imati širok spektar znanja, mora biti domišljat, pronicljiv, vješt, maštovit, kreativan, uporan, tvrdoglav, sistematičan i nepokolebljiv i voljeti taj posao. Programer automatizira rješavanje problema.

Za posao programera nužno je znati engleski jezik. Budući da se trendovi u programiranju često mijenjaju, rijetko ćemo naći literaturu na hrvatskome jeziku. Sve ključne riječi i sintaksa su na engleskome jeziku.

Svaki programer bi trebao znati engleski iz više razloga: učinkovita komunikacija na sastancima, značaj u prezentacijama i informatičkim izrazima, svaki programer treba moći napisati dokumentaciju za svoj program koju će tržište razumjeti, efikasnije snalaženje na Internetu, sva literatura je na engleskom jeziku i slično.

Što se tiče C++ jezika engleski jezik je uvelike utjecao na izradu ključnih riječi. Za svaki su programski jezik propisana pravila slaganja ključnih riječi u naredbe. Takva se pravila nazivaju sintaksa koja se isto bazira na engleskome jeziku.

Iz ankete zaključila sam da je engleski jezik vrlo važan u svim segmentima informatike i programiranja. Informatika je posao budućnosti a engleski je svjetski jezik te jedno bez drugog ne idu. Smatram da svaka osoba koja želi napredovati u bilo kojem području treba znati engleski jezik. Programeri koji proučavaju materijale na engleskome jeziku te najnovija istraživanja sigurno će biti uspješni.

SAŽETAK

Cilj ovog rada je dokazati da engleski jezik ima veliku važnost u programskom jeziku. Informatika svake godine sve više napreduje te su programerska zanimanja sve više tražena. Engleski jezik utjecao je na programiranje već od samog njihovog razvoja tako što su sve ključne riječi uzete iz engleskoga jezika. Zbog literature i materijala na engleskome jeziku utjecao je i na samo učenje programskoga jezika.

Programski jezik C++ je programski jezik opće namjene i srednje razine s podrškom za objektno orijentirano programiranje. Svaki programski jezik koristi vlastiti, ograničeni skup riječi koje imaju posebna značenja. Učenje programiranja mnogo je lakše kada se razumiju ključne riječi.

Programer bi trebao znati engleski zbog učinkovite komunikacije na sastancima, prezentacija i informatičkih izraza, svaki programer treba moći napisati dokumentaciju za svoj program koju će tržište razumjeti, literature na engleskom jeziku i slično.

Engleski jezik važan je kako u programiranju tako i u informatici. Sve češće koristimo engleske riječi u poslovnom govoru te ih postaje besmisleno prevoditi na hrvatski. Po analizi ankete vidimo da nivo znanja engleskoga jezika utječe na znanje programskoga jezika.

Ključne riječi: Engleski jezik, Programski jezik, C++, programeri, učenje na engleskom, višejezičnost

SUMMARY

The purpose of this paper is to prove that English language is very important for a programming language. Information technology keeps advancing and programming positions are increasingly in demand. English language has influenced computer programming since its conception, meaning all of the keywords were taken from English.

Due to vast literature and reading material in English, it has influenced the very learning of programming languages. C++ programming language is a mid-level general-purpose programming language with support of object-oriented programming.

Each programming language uses its own, limited choice of special meaning keywords. Learning programming is much easier when these keywords are understood.

A programmer should know English to efficiently communicate in meetings, presentations and use IT expressions. He should also be able to write documentation for his program so the market can understand it and any other necessary literature.

English language is important in programming as in information technology in general. More and more english words are used in business jargon and soon it will be pointless trying to translate them.

Poll analysis brings us to conclusion that higher english language knowledge level influences programming language knowledge.

Keywords: English language, Programming language, C++, programmers, english learning, multilingual

LITERATURA

Knjige:

- Šribak, Julijan; Motik, Boris**. DEMISITIFICIRANI C++**. 3. dopunjeno izdanje. Zagreb: Element d.o.o. 2010.
- Stroustrup, Bjarne. **THE C++ PROGRAMMING LANGUAGE**. Fourth edition. United states: Pearson Educations Inc., 2013.
- Huang, Jack; Huang Timothy. **AN INTRODUCTION TO CHINESE, JAPANESE AND KOREAN COMPUTING.** Singapur: World scientific publishing, 1991.
- Panian, Željko. **ENGLESKO-HRVATSKI INFORMATIČKI ENCIKLOPEDIJSKI RJEČNIK**. Zagreb: Europapress holding, 2005.
-

Znanstveni članci:

- Kumar Veerasamy, Ashok; Shillabeer, Anna: **TEACHING ENGLISH BASED PROGRAMMING COURSES TO ENGLISH LANGUAGE LEARNERS/NON-NATIVE SPEAKERS OF ENGLISH.**
  URL: http://www.ipedr.com/vol70/004-ICEMI2014_H00006.pdf

Internet:

- **HISTORY OF PROGRAMMIG LANGUAGE INFORGRAPHIC**
  URL: http://www.veracode.com/blog/2013/04/the-history-of-programming-languages-infographic/
- **10 RULES TO BUILD ENGLISH COMMUNICATION SKILLS FOR DEVELOPERS**
  URL:http://www.c-sharpcorner.com/UploadFile/84c85b/10-rules-to-build-english-communication-skills-for-developer/
- **THE UGLY AMERICAN PROGRAMMER**
  URL: http://blog.codinghorror.com/the-ugly-american-programmer/
- **RJEČNIK INFORMATIČKIH POJMOVA**
  URL: http://www.informatika.buzdo.com/rjecnik/
- **HOW LANGUAGE WORKS** (TUTORIAL)
  URL: http://www.indiana.edu/~hlw/Introduction/intro.html

- **COMPUTER CODE != HUMAN LANGUAGE**

URL:http://linguisticpulse.com/2014/02/03/computer-code-human-language-why-coding-cant-replace-language-education/

- **SYNTAX AND SEMANTICS OF PROGRAMMING LANGUAGES**

URL: http://homepage.cs.uiowa.edu/~slonnegr/plf/Book/

POPIS SLIKA

Slika 1: Razlika između kineskog i engleskog programskoga jezika……...9

Slika 2: APL programski jezik……..10

Slika 3: primjer programa u Pietu……..10

POPIS TABLICA

Tablica 1: Analiza ključnih riječi programskoga jezika C++……...….14

POPIS GRAFOVA

Graf 1: Vizualni prikaz programskih jezika po državi u kojoj su stvoreni……...……8

Graf 2: Znate li engleski jezik?.....20

Graf 3: Radite li kao programer?.....21

Graf 4: Koliko je engleski jezik važan za programere?.....22

Graf 5: Kako biste ocijenili svoje znanje engleskog jezika?.....23

Graf:6: Kako biste ocijenili svoje znanje programskog jezika?.....23

POPIS PRILOGA

Prilog 1: Anketa – poznavanje engleskoga jezika u programskom jeziku…….30

Prilog 2: Rezultati i analiza ankete……....32



Engleski jezik u programiranju

Znate li engleski jezik?

- ◯ da
- ◯ ne

Radite li kao programer?

- ◯ da
- ◯ ne

Koj programski jezik preferirate?

- ◯ c++
- ◯ java
- ◯ Ostalo: ____

Da li vam je engleski jezik pomogao pri učenju programskog jezika?

- ◯ da
- ◯ ne

Ako je vaš odgovor da, kako?

____

Koliko je engleski jezik važan za programere?

	1	2	3	4	5	
Nije važan	◯	◯	◯	◯	◯	Iznimno važan

Koliko je engleski jezik važan u pronalasku programerskog posla?

	1	2	3	4	5	
Nije važan	◯	◯	◯	◯	◯	Vrlo važan

Da li je engleski jezik važan za kodiranje?

	1	2	3	4	5	
Nije važan	◯	◯	◯	◯	◯	Iznimno važan

Da li je engleski jezik važan za timski rad u programiranju?

	1	2	3	4	5	
Nije važan	○	○	○	○	○	Vrlo važan

Da li je engleski jezik važan za razmijevanje programskog jezika?

	1	2	3	4	5	
Nije važan	○	○	○	○	○	Iznimno važan

Kako biste ocijenili svoje znanje engleskog jezika?

	1	2	3	4	5	
Nije važan	○	○	○	○	○	Iznimno važan

Kako biste ocijenili svoje znanje programskog jezika?

	1	2	3	4	5	
Nije važan	○	○	○	○	○	Iznimno važan

Pošalji

Nikada ne šaljite zaporke putem Google obrazaca.

Omogućuje Google Forms

Google nije izradio niti podržava ovaj sadržaj.
Prijava zloupotrebe - Uvjeti pružanja usluge - Dodatni uvjeti

Rezultati ankete

Redni broj:	1	2	3	4	5	6	7	8	9	10	11	12
Rodni jezik	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski
Znate li engleski jezik	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da
Radite li kao programer?	Da	Da	Ne	Da	Ne	Ne	Ne	Da	Da	Ne	Ne	Da
Koj programski jezik preferirate?	C++	C++	C++	Java	C++	Java	Java	Java	C++	C++	C++	C++
Da li vam je engleski jezik pomogao pri učenju programskog jezika?	Da	Da	Da	Da	Da	Da	Da	Da	Ne	Da	Da	Da
Koliko je engleski jezik važan za programere?	5	5	5	5	5	5	4	2	5	5	5	5
Koliko je engleski jezik važan za pronalazak programerskog posla?	5	3	5	4	5	5	4	5	4	5	5	5
Da li je engleski jezik važan za kodiranje?	5	1	5	5	5	4	4	2	4	3	3	3
Da li je engleski jezik važan za timski rad u programiranju?	5	2	4	4	5	3	5	2	4	5	5	5
Da li je engleski jezik važan za razmijevanje programskog jezika?	5	1	4	5	5	5	4	3	2	5	4	5
Kako biste ocijenili svoje znanje engleskog jezika?	3	5	4	5	5	4	5	2	4	3	3	3
Kako biste ocijenili svoje znanje programskog jezika?	3	5	4	2	4	5	4	5	5	1	3	3

Redni broj:	13	14	15	16	17	18	19	20	21	22	23	24
Rodni jezik	Hrvatski	Hrvatski	Hrvatski	Grčki	Engleski	Hrvatski	Portugalski	Rumunjski	Hindi	Engleski	Engleski	Hrvatski
Znate li engleski jezik	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da
Radite li kao programer?	Da	Da	Da	Ne	Da	Da	Da	Ne	Ne	Da	Ne	Ne
Koj programski jezik preferirate?	C++	C++	C++	C++	C++	Java	C++	Java	C++	C++	C++	C++
Da li vam je engleski jezik pomogao pri učenju programskog jezika?	Da	Da	Da	Da	Da	Da	Da	Da	Ne	Da	Ne	Da
Koliko je engleski jezik važan za programere?	5	5	5	5	5	4	5	5	4	5	1	5
Koliko je engleski jezik važan za pronalazak programerskog posla?	5	5	5	4	1	4	3	3	4	5	1	5
Da li je engleski jezik važan za kodiranje?	5	5	5	3	1	4	3	3	4	5	1	5
Da li je engleski jezik važan za timski rad u programiranju?	1	4	5	5	1	5	1	3	1	5	1	5
Da li je engleski jezik važan za razmijevanje programskog jezika?	4	5	4	3	5	4	4	4	5	1	1	5
Kako biste ocijenili svoje znanje engleskog jezika?	5	5	5	4	5	4	5	2	3	4	4	4
Kako biste ocijenili svoje znanje programskog jezika?	5	5	5	4	5	4	5	2	3	4	4	2

Redni broj:	25	26	27	28	29	30	31	32	33	34	35	36
Rodni jezik	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski
Znate li engleski jezik	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da
Radite li kao programer?	Da	Da	Ne	Da	Da	Da	Da	Da	Da	Da	Da	Ne
Koj programski jezik preferirate?	C++	C++	C++	Java	Java	Java	C++	Java	C++	C++	C++	Java
Da li vam je engleski jezik pomogao pri učenju programskog jezika?	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da
Koliko je engleski jezik važan za programere?	5	5	4	5	5	5	5	5	5	5	5	5
Koliko je engleski jezik važan za pronalazak programerskog posla?	5	5	4	5	4	5	4	5	5	5	5	4
Da li je engleski jezik važan za kodiranje?	3	5	3	5	3	5	4	5	5	5	5	3
Da li je engleski jezik važan za timski rad u programiranju?	4	5	4	5	4	5	4	5	5	5	5	5
Da li je engleski jezik važan za razmijevanje programskog jezika?	4	3	2	5	3	4	3	5	5	5	5	5
Kako biste ocijenili svoje znanje engleskog jezika?	4	4	5	5	4	4	5	4	5	5	5	4
Kako biste ocijenili svoje znanje programskog jezika?	4	3	3	4	5	4	4	5	5	5	5	3

A	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY
Redni broj:	37	38	39	40	41	42	43	44	45	46	47	48	49	50
Rodni jezik	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski	Hrvatski
Znate li engleski jezik	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da
Radite li kao programer?	Ne	Da	Da	Da	Da	Ne	Da	Da	Da	Ne	Da	Da	Da	Ne
Koj programski jezik preferirate?	Java	C++	C++	C++	C++	C++	C++	C++	C++	C++	C++	C++	C++	C++
Da li vam je engleski jezik pomogao pri učenju pro	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da
Koliko je engleski jezik važan za programere?	4	5	5	5	4	5	5	5	5	5	5	5	5	5
Koliko je engleski jezik važan za pronalazak progr	4	5	4	5	5	5	5	4	3	5	4	3	3	5
Da li je engleski jezik važan za kodiranje?	4	3	5	5	4	4	4	3	4	5	4	4	3	5
Da li je engleski jezik važan za timski rad u progra	4	4	4	5	5	4	2	5	5	5	4	5	4	5
Da li je engleski jezik važan za razmijevanje progr	5	5	3	5	4	5	3	4	5	5	4	4	3	5
Kako biste ocijenili svoje znanje engleskog jezika	3	4	3	5	3	3	5	4	5	4	5	3	2	5
Kako biste ocijenili svoje znanje programskog jezi	4	4	3	5	5	4	4	4	5	5	4	3	2	5

Analiza ankete:

ZNATE LI ENGLESKI JEZIK?	
DA	49
NE	1

RADITE LI KAO PROGRAMER	
DA	40
NE	10

KOJI PROGRAMSKI JEZIK PREFERIRATE	
JAVA	20
C++	30

KOLIKO JE ENGLESKI JEZIK VAŽAN ZA PROGRAMERE?	
2	1
4	3
5	46

DA LI VAM JE ENGLESKI JEZIK POMOGAO PRI UČENJU PROGRAMSKOG JEZIKA?	
DA	49
NE	1

KOLIKO JE ENGLESKI JEZIK VAŽAN U PRONALASKU PROGRAMERSKOG POSLA?	
1	1
3	1
4	10
5	38

DA LI JE ENGLESKI JEZIK VAŽAN ZA TIMSKI RAD U PROGRAMIRANJU?	
1	2
2	2
3	3
4	11
5	32

DA LI JE ENGLESKI JEZIK VAŽAN ZA KODIRANJE?	
1	3
2	1
3	7
4	9
5	30

KAKO BISTE OCIJENILI SVOJE ZNANJE PROGRAMSKOG JEZIKA?	
1	1
3	2
8	3
17	4
21	5

KAKO BISTE OCIJENILI SVOJE ZNANJE ENGLESKOG JEZIKA	
0	1
1	2
5	3
15	4
29	5