Pojam i etape programiranja

Svi smo svjesni činjenice kako danas razvoj programske podrške prati razvoj fizičkih dijelova računala. No, na samom početku razvoja računala, to nije bilo tako. U to vrijeme čak i namjenu računala određivalo je sklopovlje. Kada se namjena računala trebala promijeniti, mijenjani su određeni dijelovi u računalu. Za takva računala, kojima namjenu određuju fizički dijelovi, kažemo da nisu „programibilna“. Pokušajte zamisliti kako bi to izgledalo danas! Umorili ste se od pisanja referata iz povijesti i želite odigrati kratku igricu. Ili vam je potrebna informacija za koju znate da postoji na Internetu, ne možete promijeniti program koji koristite, već neki od fizičkih dijelova računala! Mislite li da bi računala bila dio naše svakodnevnice da je to tako? Bi li uopće postojao Internet u obliku u kakvom ga poznajemo danas?

Nedostaci takvih računala brzo su uočeni i krenulo se u razvoj programa i višenamjenskih računala.

Program

Najjednostavnije rečeno, program je skup naredbi ili instrukcija poredanih po strogo određenom redoslijedu nakon čijeg se izvršavanja dobije određeni rezultat. Osnovni element programa je naredba ili instrukcija.

Osobe koje pišu programe su programeri, a proces nastajanje programa – programiranje. Program se piše umjetno kreiranim jezicima koje nazivamo programski jezici.

Proces nastanka programa

Kako računala postaju jeftinija, a time i raširenija, tako se povećava i broj programa koji određuju njegovu namjenu. No, svaki od tih programa, preciznije skupova programa (aplikacija) najprije je morao biti razvijen.

Bilo da program radi jedna osoba ili čitav tim, u pravilu se taj proces sastoji od istih koraka:

Zadavanje zadatka

Analiza zadatka

Sastavljanje algoritma

Pisanje programa u odabranom programskom jeziku

Prevođenje programa

Testiranje

Izrada dokumentacije

Održavanje programa

U sljedećim lekcijama pojasnit ćemo svaki od navedenih koraka. Vrlo je važno da shvatite da programiranje nije samo upisivanje naredbi u računalo, već vrlo složen i zahtjevan proces u kojem pisanje naredbi (popularno – kodiranje) zauzima samo jedan manji dio. Naučiti naredbe programskog jezika nije teško. Ono što je u programiranju teško jest primijeniti te naredbe na način da se riješi određen problem.

Zadavanje zadatka

Kao što smo vidjeli ranije, prvi korak u izradi programa je zadavanje zadatka. Zadatak zadaje naručitelj programa. U stvarnom svijetu to su pojedinci ili predstavnici različitih tvrtki. U ovom udžbeniku to će biti njegovi autori, a osim njih, naručitelji programa bit će i vaši nastavnici.

Prilikom zadavanja zadatka potrebno je voditi računa o tome da on dobro definiran i potpuno jasan. Programer ili tim programera mora dobiti sve informacije: što točno program treba raditi, s kakvim podacima, te koji su prihvatljivi rezultati.

Primjer 1:

Tvoj prijatelj ima mlađeg brata koji ide u četvrti razred osnovne škole i ne ide mu geometrija. Uči računanje površine i opsega pravokutnika. Kako bi njegova mama bila sigurna da mu je zadaća dobra, tvoj prijatelj ju mora svaki dan provjeriti. Ponekad zbog toga ne stigne s tobom na trening. Kada bi mu netko pomogao i napisao program koji će mu omogućiti da u računalo unese duljinu stranica pravokutnika i na zaslon ispiše njegovu površinu i opseg, to bi mu pomoglo da bratovu zadaću provjeri za nekoliko minuta i stigne svaki put na trening.

Ovom pričom je zadan zadatak: potrebno je napraviti program koji će omogućiti unos duljina stranica pravokutnika, a na zaslon ispisati njegovu površinu i opseg. Potrebno je još samo dodati da učenici u četvrtom razredu osnovne škole ne znaju za realne brojeve, tako da duljine stranica moraju biti cijeli brojevi.

Analiza zadatka – skiciranje rješenja

Nakon što naručitelj zada zadatak, programer mora napraviti skicu rješenja. Tek nakon potpune analize može biti siguran da je zadatak dobro zadan. Tijekom analize definira se skup ulaznih podataka, pronađu potrebne formule za izračunavanje, a na temelju toga definira se i skup izlaznih vrijednosti. Ukoliko se ustanovi da nedostaju još neki potrebni podaci za rješavanje problema, kontaktira se naručitelj i zadatak se dopuni. Dok radi analizu programer često mora koristiti dodatnu literaturu ili se savjetovati sa stručnjakom za određeno područje.

Za one koji žele znati više…

Algoritam je pojam puno stariji od računala. Dolazi od latinskog prijevoda imena perzijskog znanstvenika, matematičara i astronoma Abu Abdullah Muhammad ibn Musa al-Khwarizmi-a. On se smatra ocem algebre jer je definirao osnovna pravila rješavanja linearnih i kvadratnih jednadžbi. Rođen je najvjerojatnije 783. godine, a umro oko 840.

Zadatak:

Pronađite barem jedan algoritam koji ima veze sa svakodnevnim životom i zapišite ga u bilježnicu. Na satu usporedite svoj algoritam sa algoritmima ostalih učenika. Napravite izmjene u svom algoritmu, ukoliko se za to pokaže potreba.

Algoritmi vezani uz programiranje su vrlo slični algoritmima iz svakodnevnog života. Sastoje se od tri glavna dijela:

Unos podataka (sastojci)

Obrada (priprema)

Ispis rezultata (posluživanje)

Načini zapisivanja algoritma

Algoritam može biti prikazan grafički (dijagram tijeka) ili u pseudokodu.

Zapisivanje algoritma u pseudokodu:

Pseudokod je prijelazni oblik između govornog i programskog jezika. Sadrži samo instrukcije u ispravnom redoslijedu. Zbog toga je lako razumljiv i vrlo se jednostavno može prevesti u bilo koji programski jezik. Najčešće se koristi kod kraćih i manje složenih problema.

Pisanje programa – kodiranje

Proces pisanja programa često zovemo i kodiranje. Pri izradi programa možemo se koristiti različitim programskim jezicima. Programski jezik čine skup naredbi i pravila njihova pisanja.

Programski jezik može se usporediti s bilo kojim govornim jezikom, jer svaki jezik ima dozvoljen skup znakova (abecedu), riječi, pravila njihovog slaganja i značenje!

Skup svih znakova i riječi programskog jezika naziva se leksička struktura (leksik). Pravila slaganja dozvoljenih riječi je sintaksa dok značenje ispravno napisanih naredbi određuje semantika (semantička struktura). Leksikom, semantikom i sintaksom potpuno je određen svaki programski jezik.

Programski jezici razvijaju se paralelno sa razvojem računala, a možemo ih podijeliti na strojne i simboličke.

Strojni jezik predstavlja najnižu razinu definiranja programa. Građen je od niza naredbi zapisanih u binarnom obliku. To je ujedno i jedini jezik koji računalo razumije. Jedan od njegovih nedostataka je činjenica da je naporan za programera koji mora pamtiti nizove 0 i 1. Uz to, proces pronalaženja pogrešaka je mukotrpan i dug. Sljedeći, ne manje važan, nedostatak je u tome što strojni jezik nije univerzalan, već ovisi o centralnoj jedinici za koju se piše.

Zbog navedenih nedostataka programeri su u to vrijeme tražili optimalni algoritam. Jer, bolji algoritam značio je i manje kôda. Manje kôda povlačilo je za sobom i manju vjerojatnost pogreške!

Kako je ljudima ipak jednostavnije pratiti i pamtiti riječi, logično je da su programeri koji su programirali u strojnom jeziku težili za razvijanjem programskih jezika u kojima bi nizove nula i jedinica zamijenila slova ili riječi. Takvi se jezici nazivaju simbolički.

Postoje simbolički jezici niže i više razine.

Asemblerski jezik (Asembler) je jezik niže razine. Nastao je polovicom 50-ih godina prošlog stoljeća. U njemu se pojedine naredbe označavaju skraćenicama koje podsjećaju na svoju namjenu (LD ili MOV - učitavanje, MUL – množenje, ADD - zbrajanje…). Iako je razumljiviji čovjeku od strojnog jezika, također pripada skupini jezika orijentiranih računalu. Svaka asemblerska naredba predstavlja jednu naredbu u strojnom jeziku.

Kod programskih jezika više razine (nastalih oko 1960. godine) naredbe su kratke riječi, najčešće vezane uz englesko govorno područje. One se lako pamte i bliže su ljudskom načinu razmišljanja. Za razliku od jezika niže razine, većina naredbi viših programskih jezika zamjenjuje čitav niz strojnih naredbi.

Sada nije teško zaključiti da korištenjem viših programskih jezika izrađujemo kraće i razumljivije programe. Najpopularniji viši programski jezici su BASIC, FORTRAN, Pascal, C, C++… To su jezici opće namjene. Osim njih postoje jezici koji su prilagođeni određenoj vrsti problema (PHP, JavaScript).