1. Osnove programiranja

1.1. Programiranje i programski jezici

1.1.1. Uvod

-**programiranje** omogućuje upotrebu sve **složenijih elektroničkih uređaja s ciljem njihove poboljšane iskoristivosti** (npr. efikasnost, mogućnosti,…)
-korištenje funkcija u **Excelu** (npr. min, max, average, if, sum,...) omogućuje jednostavno programiranje, a u njemu postoji i pravi način programiranja (**Visual Basic for Applications (VBA)**)
-programiranje omogućuje **automatiziranu primjenu** nekih **programskih alata** (npr. u Windowsima)
-**programska oprema** (engl. **software**) je **skup svih programa instaliranih** na nekom računalu
-**program** (engl. **program**) je niz **naredbi** (uputa) koje se izvode **točno određenim redoslijedom** da bi se **izvršio neki zadani cilj** (npr. sviranje pjesme)
-**naredba** (engl. **instruction**) je **nalog računalu** za izvršenje neke **jednostavne radnje** (npr. zbrajanje dva broja)
-da bi se **program** mogao **koristiti** na računalu potrebno ga je **instalirati**
-**instalacija programa** je postupak kojim se napisani program **priprema za rad**
-program pišu **programeri** u nekom **programskom jeziku** (npr. C++)
-**programski jezik** je program koji prepoznaje **unaprijed zadane nazive** (**ključne riječi**) čijim **kombiniranjem po unaprijed zadanim pravilima** pišemo nove programe
-postupak **pisanja programa** zovemo **programiranjem**
-programiranje je **složeni umni postupak**
-mi ćemo na vježbama koristiti besplatni programski alat **wxDev-C++** koji možete skinuti s internetske adrese **http://wxdsgn.sourceforge.net**

1.1.2. Programski jezici

-razvoj programskih jezika pratio je **razvoj računala**
-**programske jezike** dijelimo na:

a) **strojne**
-naredbe se pišu u **binarnom obliku** (niz 0 i 1)
-strojni jezik pisan je **samo za određeni procesor**
-programiranje u strojnom jeziku je vrlo **složeno**, teško i nerazumljivo
-programi **nisu prenosivi** na drukčije građena računala
-time se bave **stručnjaci**
-primjer naredbe: 1110101001011010

b) **simboličke jezike niske razine (asemblere)**
-**asembler** je simbolički jezik u kome je **naredba strojnog jezika predočena odgovarajućim simbolom koji je kombinacija nekoliko slova**
-primjer:CMP - usporedi (engl. ***compare)***
-te **kombinacije slova se lako pamte** (podsjećaju na **značenje** naredbe), a zovu se **mnemonici**
-svakim program mora biti **preveden** u binarni oblik da bi ga procesor mogao izvršiti
-**to radi jezični prevoditelj**
-primjer asemblerske naredbe: ADD A, #100
-programi u asembleru su **čitljiviji i lakši za razumijevanje** od binarnog zapisa, ali ih je **teško pisati i ispravljati**
-**ovise o vrsti i unutrašnjoj građi računala** (procesora)
-imaju **veliku brzinu izvršavanja**

c) **više programske jezike**
-naredbe su **nalik govornom jeziku**, **lake su za pamćenje i upotrebu**
-**više naredbi asemblera** predočeno je **jednom simboličkom naredbom**
-**programiranje** je olakšano
-program se izvršava na **različitim računalima** (procesorima)
-primjer programa u jeziku više razine (jezik C):

```
int a=5;
int b=7;
int c;
```

c=b-a;

-**simbolički jezici visoke razine** mogu biti **po namjeni**:

a) **opće namjene**

-**za bilo koje zadatke**

b) **prilagođeni određenoj vrsti problema**

-**prilagođeni za određeno područje primjene**

-**C jezik** je **opće namjene**, **velikih mogućnosti**, **neovisan o računalu**, **s malo ključnih riječi**

-C je **modularan jezik** (**podjela programskog zadatka na manje cjeline**)

-poboljšanjem jezika C s **objektno orijentiranim programiranjem** nastao je programski jezik **C++** (**najkorišteniji programski jezik opće namjene)**

-program u **jeziku više razine** se prije pokretanja pomoću **jezičnog prevoditelja pretvara u oblik pogodan za procesor**

-**jezični prevoditelji** su:

1.) **interpreteri**

-naredbu višeg programskog jezika pretvara u strojne naredbe **tijekom izvršavanja** (npr. **QBasic**)

2.) **kompajleri**

-naredbe višeg programskog jezika pretvara u strojne naredbe **prije izvršavanja** (npr. **C++**)

-rezultat je **izvršna datoteka** koja se **pokreće**

d) **programski jezici prilagođeni krajnjim korisnicima**

-njima se **ubrzava programiranje** i za **neprogramere** (npr. upitni jezici za baze podataka - SQL)

e) **programski jezici neovisni o sklopovlju i operativnom sustavu**

-naredbe se **izvršavaju na bilo kojem računalu s bilo kojim operativnim sustavom**

-programi su **prenosivi** (npr. **Java)**

-**programske jezike po građi** (strukturi) su:

a) **proceduralni**

-program se **dijeli na niz manjih cjelina**, a **svaka je dio ukupnog zadatka** (C, Basic, Pascal,...)

b) **objektno orijentirani**

-u programu **definiramo objekte koji se sastoje od podataka i operacija na njima**

-**vanjske radnje definiraju događanja među objektima**, a time i **tijek programa** (Visual Basic)

1.2. Načela programiranja

-u rješavanju zadataka **čovjek** se služi

a) znanjem
b) iskustvom
c) logičkim rasuđivanjem
d) intuicijom
e) pamćenjem
f) osjećajima itd.

-u rješavanju zadataka **računalo** koristi samo:

a) **pamćenje**
b) **logičko rasuđivanje**

-da bi računalo riješilo zadatak, treba ga **pretvoriti** u oblik koji uključuje samo **pamćenje i logičko rasuđivanje**

-u **pretvorbi nam** pomažu **pomoćni postupci, a to su:**

a) **planiranje**
b) **analiza zadatka**
c) **algoritam**
d) **pseudokôd**
e) **dijagram tijeka**

-**složeniji zadatak** traži **više pomoćnih postupaka**

-planiranjem se **određuje tko će, kada i što raditi**

-njime se **predviđaju i raspoređuju pojedine faze izrade programa**

-**analiza zadatka** je **raščlanjivanje i potpuno razumijevanje zadatka i željenih rezultata**

-rezultat analize je tzv. **specifikacija zadatka**

-to je dokument koji sadrži **podroban opis zadatka i željenih rezultata**

-**algoritam** je **naputak kako riješiti zadatak**
-**cilj algoritma** je **zadatak svesti na niz jednostavnih, manjih radnji**
-**algoritam** se **prikazuje**:
a) **dijagramom tijeka**
b) **pseudokôdom**
-**dijagram tijeka** je **grafički prikaz algoritma**
-**pregledno prikazuje algoritam**, **omogućava lakšu analizu i provjeru predloženog rješenja,** te **pronalaženje boljih postupaka rješavanja zadatka**
-sastoji se od nekoliko jednostavnih **geometrijskih likova** spojenih **usmjerenim crtama koje** pokazuju **tijek rješavanja zadatka**
-**pseudokôd nalikuje na računalni program**, ali **nije napisan u programskom jeziku**
-sastoji se od **kratkih izraza na govornom jeziku** koji **opisuju i ukratko objašnjavaju pojedine zadatke** algoritma
-svaki programski jezik ima vlastiti **ograničeni skup riječi** koje imaju **posebna značenja** i **ne smiju se koristiti u druge svrhe**
-to su **ključne** (**rezervirane**) **riječi**
-**sintaksa** programskog jezika su **propisana pravila slaganja** (**pisanja**) **ključnih riječi u naredbe**
-ako se ne zadovolji propisana sintaksa, program će biti **neispravan** i **neće se moći izvršiti**
-da bi program bio **uporabno koristan**, mora biti **logički ispravan**
-za otkrivanje **logičkih pogrešaka** potrebno je **provjeravati** (**testirati**) **program**
-program provjerava autor programa, više ljudi kod proizvođača progama ili neovisni ispitivači
-**održavanje programa** je **postupak mijenjanja programa** tijekom njegovog "životnog vijeka"
-**održavanje** može biti:
a) **izravno** (npr. temeljem ugovora o održavanju)
b) **neizravno** (npr. izdavanjem novih inačica i ispravaka programa)
-**dokumentacija** je važan dodatak programu, a **sastoji se** od:
a) **uputa za instaliranje programa**
b) **priručnika za korisnike**
c) **tehničkog opisa programa**
-**programska struktura** opisuje **način i redoslijed izvršavanja pojedinih radnji** koje dovode do **konačnog rješenja zadatka**

1.3. Algoritam - teorija i vježbe

-računalo postavljeni zadatak može riješiti samo ako dobije **upute kako to učiniti**
-uputa se mora sastojati samo od različitih **operacija** i **pamćenja** vrijednosti koje uvrštavamo u program (**ulazni podaci**) ili dobivamo kao **međurezultate** i **rezultate** (**izlazni podaci**)
-niz takvih uputa tvore **algoritam**
-**cilj algoritma** je cjelokupni zadatak svesti na **niz jednostavnih, manjih postupaka** čijim **kombiniranjem** rješavamo cijeli zadatak
-**izvršavanjem** tih **osnovnih radnji** na temelju **ulaznih podataka** dobivamo **rezultat** (vidi sliku)

-zadatak se može riješiti na **više različitih načina** pa za to postoji **više različitih algoritama**
-algoritam treba **najbrže, najučinkovitije i najsigurnije** dati rezultat
-mora se pisati **jasno i detaljno** **svakodnevnim govorom**
-**algoritam** mora **zadovoljiti uvjete da**:
a) postoje **jasno definirani ulazni podaci** (tzv. **početni objekti**)
b) **završetkom** algoritma moramo dobiti **rezultat** (**izlazni podaci** ili tzv. **završni objekti**)
c) algoritam **za iste ulazne podatke uvijek mora dati iste rezultate**
d) algoritam mora imati **konačan broj postupaka tijekom izvođenja**
e) **algoritam završava u konačnom vremenu**, tj. mora postojati **kraj algoritma**

f) **svaki korak** (instrukcija) u algoritmu mora biti **izvediv** (npr. ne smije doći do djeljenja s 0)
g) **instrukcije algoritma** moraju biti **jednoznačne**, tj. rezultat je uvijek isti za iste ulazne podatke
-**algoritmi po namjeni** su:
a) **specijalizirani**
-rade **samo za neke ulazne podatke**
b) **općeniti**
-rade s **više skupina** (klasa) **ulaznih podataka**
-često se pod **algoritmom** misli samo na **određene postupke** kao **dijelove programa**
-u algoritmu se **ulazni podaci, međurezultati i rezultati** pamte **proizvoljnim imenima** (**čim kraćim** radi bržeg pisanja, npr. slova abecede) koje zovemo **varijablama**
-to označava **mjesto u memoriji** čiji se **sadržaj u tijeku izvođenja programa može mijenjati**
-nastojimo koristiti **čim manje varijabli** da bi trošili **čim manju količinu memorije**
-kod algoritama je potrebno paziti što je **bitnije**:

- **veća brzina** (**uobičajeno**)
- **manji utrošak memorije** (**rijeđe**)

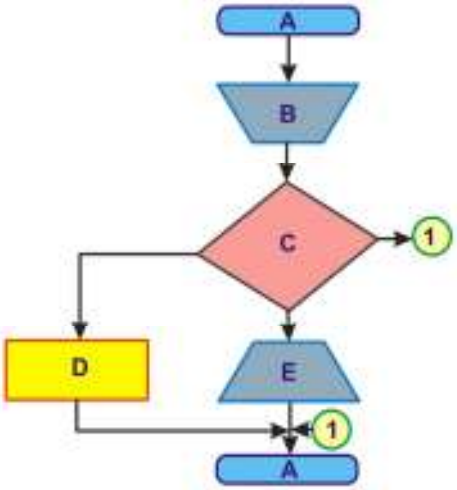
-**primjer algoritma**:
1.) Napišite algoritam koji unosi dva broja a i b, računa njihov zbroj i ispisuje rezultat na ekranu
-rješenje:

učitaj a
učitaj b
c=a+b //računanje zbroja
ispiši c na ekranu

-dvije kose crte (//) označavaju da **iza njih slijedi opis trenutnog koraka** (**komentar**)
-**varijabla s desne strane znaka** = označava vrijednost te varijable **neposredno prije trenutnog koraka**, a **s lijeve strane nakon izvršenja trenutnog koraka**

1.4. Dijagram tijeka - teorija i vježbe

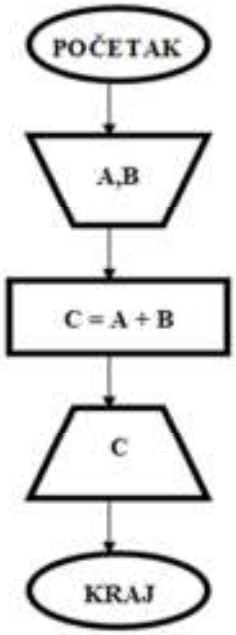
-**dijagram tijeka** služi za **grafičko predočavanje algoritama** pomoću **jednostavnih grafičkih simbola spojenih usmjerenim crtama** (strelicama) radi **lakšeg praćenja** funkcioniranja algoritma
-**usmjerene crte** (**strelice**) pokazuju **tijek rješavanja zadatka**
-dijagram tijeka **olakšava izradu programa**, te se **lakše uklanjaju pogreške** u algoritmu
-pogodan je za **analizu problema** i **traženje najboljih rješenja**
-**neovisan** je o **programskom jeziku i računalu**
-**najčešće korišteni simboli u dijagramu tijeka** su:



-**slovo A** (**pravokutnik** **zaobljenih vrhova** ili **elipsa**) predstavlja **oznaku početka, prekida ili kraja programa**
-taj simbol se **uvijek** koristi barem na dva mjesta u programu: za **početak i kraj**
-**slovo B** (**trapez s dužom gornjom stranicom**) označava **unošenje podataka u program**
-**romb** označen **slovom C** je simbol tzv. **grananja** u programu
-**grananje** označava da se **ovisno o napisanome u rombu** (**uvjet ili vrijednost neke varijable**) **program nastavlja samo s jednom izlaznom strelicom**
-time se može **promijeniti način odvijanja programa**

-u **grananju** **najčešće** postoje **samo dva izlaza** iz romba koji odgovaraju **točnom** (**DA**) ili **netočnom** (**NE**) odgovoru na pitanje u rombu
-**pitanje postavljeno u rombu** zovemo **uvjet**
-**uvjet** je najčešće **neka provjera** (npr. nešto veće od nečega, manje, jednako i sl.
-**grananje ovisno o uvjetu** zove se **uvjetno grananje**
-ukoliko **iz romba izlaze više od dvije strelice**, tada se radi o tzv. **višestrukom odabiru**
-tu je **pitanje u rombu** da li je **podatak jednak unaprijed zadanoj cijeloj vrijednosti** (npr. 1, -3), a ta **vrijednost** se **piše na početku strelice koja izlazi iz romba**
-za slučaj **višestrukog odabira** pitanje u rombu zadaje se u obliku **varijabla=?**
-**simbol običnog pravokutnika** (slovo **D**) označava **bilo koju naredbu** (**operaciju**) ili **više njih**, **osim** onih za koje postoje **posebni simboli**
-često se njime prikazuju **matematičke operacije**
-ako je u pravokutniku **više operacija**, pišu se **od gore prema dolje redom kako se izvršavaju**
-**simbol trapeza s dužom donjom stranicom** (slovo **E**) označava **izlaženje podataka iz programa** (npr. ispis na ekranu ili printeru)
-**kružići s upisanim brojem** (npr. broj 1) su **priključne točke za povezivanje raznih dijelova programa u cjelinu**
-brojeve u njima zovemo **veznim brojevima**
-**priključne točke** koristimo kod **dužih algoritama**
-**usmjerene crte** (**crte sa strelicama**) zovu se **linije tijeka programa**
-one **označavaju kojim redom se naredbe izvršavaju**
-**primjer**: Unesi dva broja A i B, te ispiši njihov zbroj C.
-algoritam:

početak
  upiši brojeve A i B
  C=A+B
  ispiši C
kraj



-napomena: **uvlačenja pojedinih izraza u algoritmima** su zbog **preglednosti i grupiranja zajedničkih dijelova**