

Računarstvo i informatika

III razred

1. Transformacija problema na oblik pogodan za rešavanje na računaru.

Prva faza u rešavanju problema pomoću računara je postavka zadatka.

Sledeća faza je matematička formulacija zadatka. U ovoj fazi potrebno je izvršiti analizu postavljenog zadatka i na osnovu raspoloživih podataka preciznije opisati odnose između podataka. Tu se često koriste i neke matematičke formule, pa otuda i ovaj naziv.

Sledeća faza je izbor metoda i ona je ključna za opis postupka rešavanja zadatka.

Nakon izbora metoda sledi faza opisa algoritma. Algoritam je jedan od osnovnih pojmova matematike i računarstva. Algoritam je jasno uputstvo o tome kakve radnje i u kojem poretku treba izvršiti da bi se rešio svaki zadatak iz zadate klase zadataka.

Algoritmi zajedno sa strukturama programa čine programe.

2. Pojam apstrakcije objekta i algoritma. Programski jezici i njihova sintaksa i semantika.

Algoritmi se opisuju pomoću algoritamskih šema (blok dijagrama). One predstavljaju grafički prikaz toka programa i sastoje se od simbola povezanih linijama u jednu celinu.

Programski jezici su sredstva za pisanje programa i saopštavanje programa računaru. Programski jezici su veštački jezici namenjeni notaciji i prenosu specifičnih informacija. Najbitnija karakteristika programskih jezika jeste njihova jednoznačnost. Programi napisani na programskim jezicima moraju biti prevedeni na mašinski jezik.

Najpoznatiji programski jezici su: FORTRAN, COBOL, LISP, PL-1, Pascal, PROLOG, Ada, Modula 2, C, C++, Java i drugi.

Sintaksa je nauka o jeziku u kojoj se izučava obrazovanje gramatički korektnih konstrukcija jezika.

Semantika je nauka o jeziku u kojoj se izučava značenje konstrukcija jezika.

pdfMachine

Is a pdf writer that produces quality PDF files with ease!

Produce quality PDF files in seconds and preserve the integrity of your original documents. Compatible across nearly all Windows platforms, if you can print from a windows application you can use pdfMachine.

Get yours now!

Sintaksu jezika čini skup pravila za obrazovanje pravilnih konstrukcija jezika. Ukoliko je program napisan bez poštovanja pravila o zapisu konstrukcija programskog jezika, javljaju se sintaksne greške.

Semantiku jezika čini skup pravila za utvrđivanje značenja konstrukcija jezika. Za razliku od sintakasnih grešaka, većinu semantičkih grešaka ne može otkriti prevodilac, već to mora da uradi čovek.

Za formalno opisivanje sintakse programskih jezika najčešće se koriste:

- Bekusova notacija i
- Sintaksni dijagrami.

Semantika programskih jezika se obično formalno opisuje pomoću:

- Bečkog definicionog jezika i
- Van Vajngardenovih gramatika.

3. Klasifikacija programskih jezika

Prema stepenu zavisnosti od računara programske jezike delimo na :

- mašinski zavisne i
- mašinski nezavisne jezike.

Mašinski zavisni jezici se dele na:

- Mašinske jezike i
- Mašinski orijentisane jezike.

Mašinski orijentisani jezici se dele na:

- simboličke jezike i
- makrojezike.

Mašinski nezavisni jezici se dele na:

- proceduralno orijentisane jezike i
- problemski orijentisane jezike.

4. Proceduralni i neproceduralni jezici

Kod neproceduralnih programskih jezika nije primaran opis algoritma jer su oni slični prirodnom jeziku čoveka.

pdfMachine

Is a pdf writer that produces quality PDF files with ease!

Produce quality PDF files in seconds and preserve the integrity of your original documents. Compatible across nearly all Windows platforms, if you can print from a windows application you can use pdfMachine.

Get yours now!

Kod proceduralnih jezika se koristi procedura opisana nizom naredbi. U proceduralnim programskim jezicima uvek mora da se opiše kako se problem rešava. Proceduralni jezici su: Pascal, C, FORTRAN, Ada, Modula-2...

pdfMachine

Is a pdf writer that produces quality PDF files with ease!

Produce quality PDF files in seconds and preserve the integrity of your original documents. Compatible across nearly all Windows platforms, if you can print from a windows application you can use pdfMachine.

Get yours now!

5. Rešavanje problema pomoću računara – vežbanje

- Kreirati algoritam za sabiranje dva cela broja
- Kreirati algoritam za izračunavanje apsolutne vrednosti broja
- Kreirati algoritam za ispitivanje parnosti broja
- Kreirati algoritam za ispitivanje znaka broja.

6. O jeziku Pascal. Objašnjenje strukture jednostavnog programa u Pascal-u.

Krajem sedme decenije XX veka pojavio se programski jezik Pascal. Njegov tvorac je Niklaus Wirth. Programski jezik Pascal dobio je ime u čast velikog francuskog matematičara i fizičara Bleza Paskala – konstruktora jedne od prvih računarskih mašina.

Pascal je uticao na nastanak čitavog niza programskih jezika specijalizovanih za razne oblasti.

Da bi se rešio zadatak pomoću Pascal-a, mora da se napiše program kojim se opisuje rešenje.

Program u Pascalu ima sličnu ulogu kao i u drugim proceduralnim jezicima, t.j. treba da opiše:

- podatke koji se obrađuju i
- radnje koje se izvršavaju nad podacima.

Program čine tri komponente:

- zaglavlje,
- blok (ili telo) i
- tačka koja označava fizički kraj programa.

Zaglavlje ima svoju strukturu i može se definisati na sledeći način:

Program ime (ime, ime);

Ime se u zaglavlju pojavljuje ispred zagrade gde označava ime programa i u zagradama gde označava ime datoteke.

Sledeća zaglavlja su korektno definisana:

Program prvi (input, output);

Program zzz;

Program obradadat (input, brojevi, slova);

Program prost (output);

Preko bloka ostvaruje se osnovna uloga programa. Blok se sastoji iz:

- odeljka deklaracija i definicija i
- odeljka naredbi.

U odeljku deklaracija i definicija opisuju se (ili navode) podaci sa kojima se operiše, a u odeljku naredbi opisuje se algoritam (preko niza naredbi) za rešenje zadatka.

Odeljak deklaracija i definicija sadrži posebne odeljke za:

- deklaraciju obeležja (labela),
- definiciju konstanti,
- definiciju tipova podataka,
- deklaraciju promenljivih,
- deklaraciju procedura ili funkcija.

Nijedan od ovih odeljaka nije obavezan, što znači da program može funkcionisati i bez njih.

Odeljak naredbi je obavezan i njegova struktura je sledeća:

```
Begin  
Naredbe  
End.
```

Između begin i end nalazi se niz naredbi koje su odvojene tabulatorom i zapetom.

Primer:

```
Begin  
Writeln('prvi program');  
End.
```


.1.

```
Program prvi (input, output);  
Var  
A,b,c: integer;  
  
Begin  
Writeln ('Unesite dva cela broja');  
Readln (a,b);  
C:=a+b;  
Writeln('Zbir je', c);  
End.
```

.2.

```
Program drugi (input, output);  
Var  
A,b,c,p,o: real;  
  
Begin  
Writeln('Unesite katete pravouglog trougla');  
Readln(a,b);  
C:=sqrt(sqr(a)+sqr(b));  
O:=a+b+c;  
P:=(a*b)/2;  
Writeln('obim iznosi', o);  
Writeln('povrsina iznosi', p);  
End.
```

.3.

```
Program treci (input, output);  
Const  
Pi=3.14;  
Var  
R,h,p,v:real;  
  
Begin  
Writeln('Unesite poluprecnik osnove i visinu valjka');  
Readln(r,H);  
P:=sqr(r)*pi*2+2*r*pi*H;  
V:=sqr(r)*pi*h;  
Writeln('povrsina iznosi', p);  
Writeln('zapremina iznosi', v);  
End.
```

. 4.

Program trougao (input, output);

Var

A,h,P: real;

Begin

Writeln('Unesite stranicu jednakostrani nog trougla');

Readln(a);

H:=(a*sqr(3))/2;

P:=(sqr(a)*sqr(3))/4;

Writeln('Visina iznosi', h);

Writeln('Povrsina iznosi', P);

End.

. 5.

Program lopta(input, output);

Const

Pi=3.14;

Var

R, p, v: real;

Begin

Writeln('Unesite poluprecnik lopte');

Readln(r);

P:=4*sqr(r)*pi;

V:=(4/3)*r*r*r*pi;

Writeln('Povrsina je', p);

Writeln('Zapremina je', V);

1. Write . 10

Program ime;

Var
I: integer;

Begin
For i:= 1 to 10 do
Writeln('Ana');
End.

For do, i
, .
(10).

.2.

n .

Program petlja (input,output);

Var
I, n: integer;

Begin
Writeln('Koliko puta zelite da se ispise vase ime');
Readln(n);
For i:=1 to n do
Writeln('Ana');
End.

For do . : While do Repeat
Until.

repeat until ' ' i n . While
(until).
do. while
(i:=i+1).

1

10

while do.

Program prvi;

Var
I: integer;

Begin
I:=1;
While i<=10 do
Begin
Writeln('Nenad');
I:=i+1;
End;

2

10

repeat until.

Program drugi;

Var
I:integer;

Begin
I:=0;
Repeat
I:=i+1;
Writeln('Nenad');
Until i>10;
End.

Nizovni tip

Definisanje nizovnog tipa

Type

Niz = array [1..50] of integer – definisani tip podataka 'niz' ima najviše 50 članova koji su celobrojnog tipa (integer).

Posle definisanja tipa 'niz' potrebno je u odeljku za deklarisanje promenljivih 'var' dodeliti promenljivoj x da pripada tipu podataka niz, I onda je nizovni tip definisan. Promenljiva x može u sebi sadržati do 50 članova (u ovom slučaju celih brojeva) I svaki ima svoj redni broj: x[1], x[2], x[3],..., x[n], gde je n broj članova niza.

Primer 1.

Napisati program u Pascalu koji učitava niz od najviše 50 članova ($n \leq 50$), I ispisuje te članove niza u obrnutom redosledu.

```
Program prvi;  
Type  
Niz = array [1..50] of integer;  
Var  
X:niz;  
I, n: integer;
```

```
Begin  
WriteLn('Koliko članova ima niz?');  
ReadLn(n);  
WriteLn('Unesite članove niza');  
For i:=1 to n do  
ReadLn(x[i]);  
For i:= n downto 1 do  
WriteLn(x[i]);  
End.
```

Primer 2.

Napisati program u pascalu koji računa zbir članova niza I računa aritmetičku sredinu članova niza.

```

Program drugi;
Type
Niz=array [1..50] of integer;
Var
X:niz;
I, n, s: integer;
Sred: real;

Begin
Writeln('Koliko clanova ima niz?')
Readln(n);
Writeln('Unesite celobrojne clanove niza');
For i:=1 to n do
Readln(x[i]);
S:=0;
For i:=1 to n do
S:=s+x[i];
Sred:=s/n;
Writeln('aritmeticka sredina clanova niza je',sred);
End.

```

Primer 3.

Napisati program u pascalu koji ucitava niz od najvise 50 clanova I pronalazi najveći član niza, I ispisuje najveći član niza kao I njegov redni broj u nizu.

```

Program treci;
Type
Niz=array [1..50] of integer;
Var
X:niz;
I, n, max, rb: integer;

Begin
Writeln('Koliko clanova ima niz?');
Readln(n);
Writeln('Unesite clanove niza');
For i:= 1 to n do
Readln(x[i]);
Max:=x[1];
For i:= 2 to n do

```

```

If x[i]>max then
Begin
Max:=x[i];
Rb:=l;
End;
Writeln('najveci clan niza je', max);
Writeln('njegov redni broj u nizu je', rb);
End.

```

Program koji pronalazi najmanji clan niza razlikuje se samo u tome sto cemo promenljivu max nazvati promenljivom min I uslov ce biti if $x[i]<min$.

Primer 4.

Napisati program u pascalu koji ucitava niz od n clanova ($n \leq 50$) I sortira dati niz u rastuci poredak.

```

Program cetvrti;
Type
Niz=array [1..50] of integer;
Var
X:niz;
l, j, n, pom: integer;

Begin
Writeln('koliko clanova ima niz?');
Readln(n);
Writeln('Unesite clanove niza');
For i:=1 to n do
Readln(x[i]);
For i:= 1 to n-1 do
For j:= i+1 to n
If x[i]>x[j] then
Begin
Pom:=x[i];
X[i]:=x[j];
X[j]:=pom;
End;
Writeln('rastuci niz:');
For l := 1 to n
Writeln(x[l]);

```

End.

Primer 5.

Napisati program u pascalu koji ucitava niz od najvise 50 clanova I od tog niza kreira dva niza; jedan od pozitivnih clanova niza, a drugi od negativnih clanova niza.

Program peti;

Type

Niz=array [1..50] of integer;

Var

X, a, b: niz;

I, n, j, k: integer;

Begin

Writeln('koliko clanova ima niz?');

Readln(n);

Writeln('unesite clanove niza');

For i:= 1 to n do

Readln(x[i]);

J:=0; k:=0;

For i:= 1 to n do

If x[i]>=0 then

Begin

J:=j+1;

A[j]:=x[i];

End

Else

Begin

K:=k+1;

B[k]:=x[i];

End;

Writeln('Pozitivni:');

For i:=1 to j do

Writeln(a[i]);

Writeln('Negativni:');

For i:=1 to k do

Writeln(b[i]);