

Programiranje u fizici

9. Vrste programskih naredbi

Prirodno-matematički fakultet u Nišu
Departman za fiziku

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

Uvod

- Programske naredbe
- Elementarna naredba
- Blok naredbi
- Naredbe granjanja - if
- Naredbe granjanja - switch

Svaki program, napisan na nekom programskom jeziku, je sastavljen od niza pojedinačnih naredbi.

Poredak izvršenja programskih naredbi je obično sekvencijalan, dok se izmene redosleda izvršavanja vrši pomoću upravljačkih naredbi.

Upravljačke naredbe omogućavaju da se određeni delovi programa izvrše više puta, dok naredbe granjanja omogućavaju izvršenje određenih naredbi u zavisnosti od ispunjenja, odnosno neispunjenja datog uslova.

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

Uvod

Programske naredbe

Elementarna naredba

Blok naredbi

Naredbe granjanja - if

Naredbe granjanja - switch

Programski jezik C poseduje sledeće vrste programskih naredbi:

1. elementarna programska naredba
2. blok naredbi
3. naredbe granjanja, u koje spadaju:
 - if
 - switch
4. naredba iteracije, u koje spadaju:
 - while – petlja
 - for – petlja
 - do while – petlja
5. naredbe kontrolisanog skoka, u koje spadaju:
 - break
 - continue
6. naredba goto
7. naredba return

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

Uvod

Programske naredbe

Elementarna naredba

Blok naredbi

Naredbe granjanja - if

Naredbe granjanja - switch

Elementarna programska naredba se dobija kada se na kraju nekog izraza doda znak ;.

Primer:

```
suma = 7;  
printf ("Pozdrav");  
zbir--;
```

Specijalan slučaj elementarne programske naredbe predstavlja prazna naredba.

```
;
```

Prazna naredba nema nikakav uticaj na izvršenje program i ona se može pojaviti na svakom mestu u programu gde i elementarna naredba.

Ona se koristi u slučaju kada sa jedne strane, sintaksa jezika zahteva postojanje naredbe, kod sa druge strane, sam program na tom mestu ne vrši nikakvu logičku akciju.

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

Uvod
Programske naredbe
Elementarna naredba
Blok naredbi
Naredbe granjanja - if
Naredbe granjanja - switch

Blok naredbi predstavlja skup programskih naredbi koje čine jednu celinu. U bloku naredbi mogu da se pojave kako izvršive, tako i opisne naredbe. Blok naredbi se koristi u sledeće svrhe:

1. za grupisanje više naredbi u logičku celinu
2. kao telo neke funkcije
3. za ograničavanje uloge određenih opisnih naredbi na jedan lokalni deo programa

U programskom jeziku C svaki blok naredbi se označava parom vitičastih zagrada, pri čemu znak { stoji na početku a znak } na kraju bloka.

Blok naredbi se interpretira kao jedna jedina naredba.

Primer:

```
{  
    broj_a = 13;  
    broj_b = 3;  
    proizvod = broj_a * broj_b;  
    printf ("%d", proizvod);  
}
```

2.	Tipovi u programskom jeziku C
3.	Promenljive u programskom jeziku C
4.	Konstante u programskom jeziku C
5.	Operatori u programskom jeziku C
6.	Algoritamsko resavanje problema
7.	Vrste programskih naredbi

Uvod
Programske naredbe
Elementarna naredba
Blok naredbi
Naredbe granjanja - if
Naredbe granjanja - switch

U programskom jeziku C postoje dve vrste naredbi granjanja:

- if – naredba
- switch – naredba

Naredba if ima sledeći oblik:

```
if (izraz)
    naredba1
else
    naredba2
```

Izraz u zagradi može biti neka logička relacija ili promenljiva, a naredba1 i naredba2 mogu biti elementarne programske naredbe ili blok naredba.

Sintaksa programskog jezika C dozvoljava da se deo naredbe sa else izostavi. U tom slučaju naredba if ima oblik:

```
if (izraz)
    naredba1
```

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

- Uvod
- Programske naredbe
- Elementarna naredba
- Blok naredbi
- Naredbe granjanja - if**
- Naredbe granjanja - switch

```
if (izraz)
    naredba1
else
    naredba2
```

Naredba **if** se izvršava tako što se prvo izračunava vrednost izraza u zagradi.

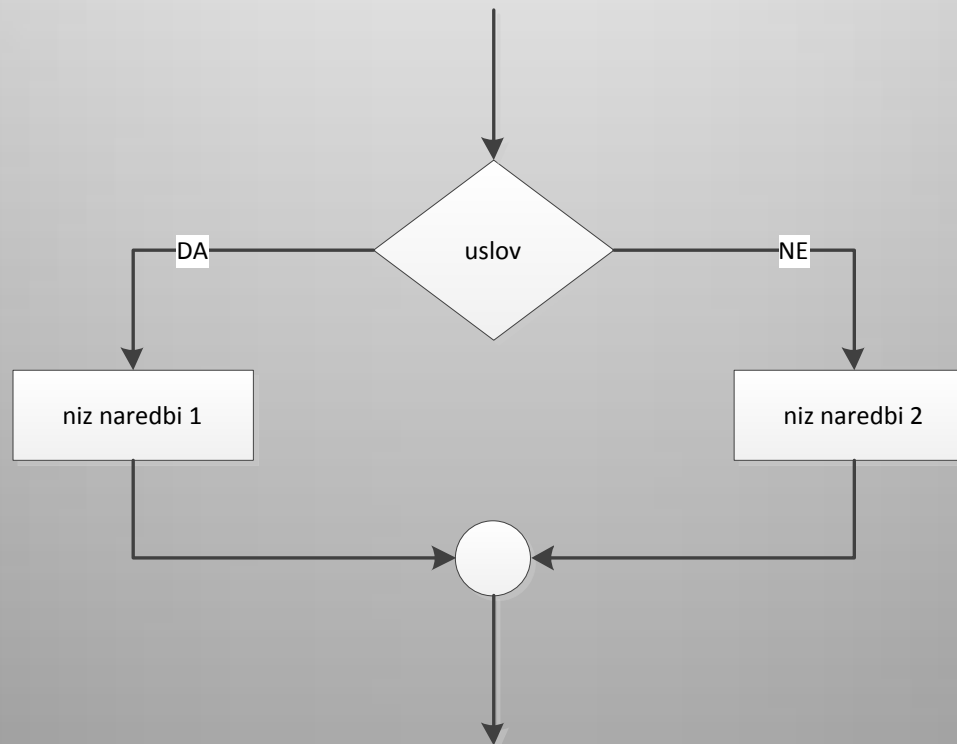
U slučaju kada je brojna vrednost izraza različita od nule (što odgovara logičkoj vrednosti tačan), izvršiće se **naredba1**, dok se deo naredbe sa **else**, ukoliko postoji, zanemaruje.

Ukoliko izraz ima brojnu vrednost nula (što odgovara logičkoj vrednosti netačan), izvršava se **naredba2**.

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

- Uvod
- Programske naredbe
- Elementarna naredba
- Blok naredbi
- Naredbe granjanja - if**
- Naredbe granjanja - switch

Korišćenjem dijagrama toka, naredba **if** se može predstaviti na sledeći način:



2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

- Uvod
- Programske naredbe
- Elementarna naredba
- Blok naredbi
- Naredbe granjanja - if**
- Naredbe granjanja - switch

Primer: Napisati program koji poredi dva broja osnovnog tipa float.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float pr_1, pr_2, pr_pom;

    printf ("Unesite prvi broj\n");
    scanf ("%f", &pr_1);

    printf ("Unesite drugi broj\n");
    scanf ("%f", &pr_2);

    if (pr_2 > pr_1) {
        pr_pom = pr_1;
        pr_1 = pr_2;
        pr_2 = pr_pom;
    }

    printf("\n\n Poredak brojeva je :\n");
    printf("%f\n", pr_1);
    printf("%f\n", pr_2);

    return 0;
}
```

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

Uvod
Programske naredbe
Elementarna naredba
Blok naredbi
Naredbe granjanja - if
Naredbe granjanja - switch

Pisanje „ugnježdene“ if naredbe.

```
if (broj_a <= 0) {  
    if (broj_a == 0) {  
        printf("broj je 0 \n");  
    }  
    else {  
        printf("broj je negativan \n");  
    }  
}  
else {  
    printf("broj je pozitivan \n")  
}
```

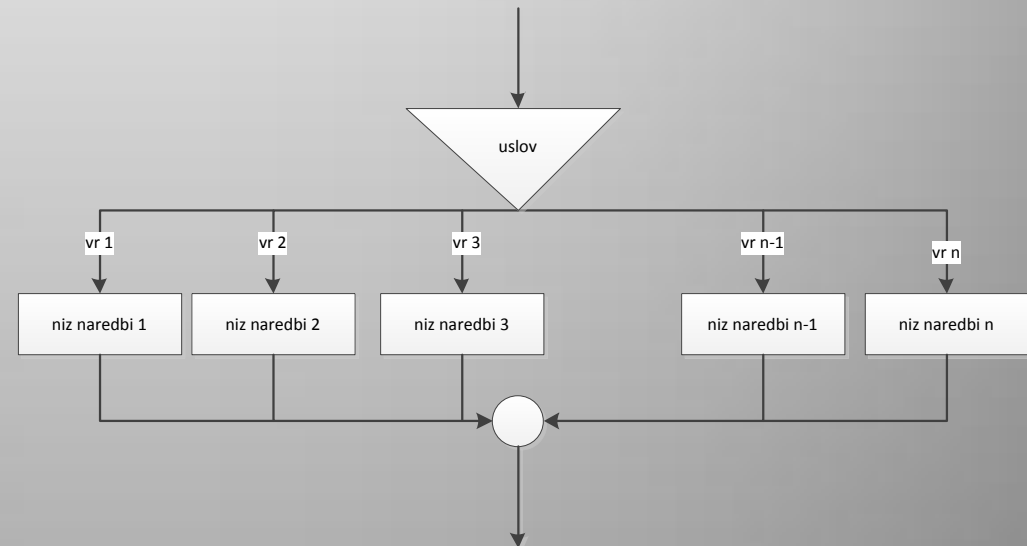
2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

Uvod
Programske naredbe
Elementarna naredba
Blok naredbi
Naredbe granjanja - if
Naredbe granjanja - switch

Naredba višestrukog granjanja – switch predstavlja uopštenje if naredbe, tj. ona omogućava izbor između više alternativa (a ne samo dve, kao kod if).

Ošti oblik naredbe switch je:

```
switch (izraz_1) {  
    case konst_1 : naredba_1  
    case konst_2 : naredba_2  
        . . .  
    case konst_n : naredba_n  
    default : naredba_d  
}
```



Pri čemu `konst_1`, `konst_2`, ... mogu biti konstante osnovnog tipa `int` ili `char`.

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

- Uvod
- Programske naredbe
- Elementarna naredba
- Blok naredbi
- Naredbe granjanja - if
- Naredbe granjanja - switch**

U naredbi **switch** se prvo izračunava **izraz_1**. Izračunata vrednost se poredi sa **konst_1**, **konst_2**, ..., **konst_n** sve dok se ne pronađe ekvivalentna vrednost ili se ustanovi da takva vrednost ne postoji ni u jednom od case delova.

Ukoliko je ekvivalentna vrednost pronađena, biće izvršene sve naredbe koje se nalaze između tog **case**-dela i kraj **switch** naredbe.

Drugim rečima, preostali **case**-delovi se neće porediti sa vrednošću datog izraza, ali će njihove pripadajuće naredbe biti izvršene.

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

- Uvod
- Programske naredbe
- Elementarna naredba
- Blok naredbi
- Naredbe granjanja - if
- Naredbe granjanja - switch**

U slučaju da treba izvršiti samo naredbu u odgovarajućem **case**-delu, mora se koristiti naredba **break**.

Svakom **case** i **default** delu je preporučljivo dodati naredbu **break**.

Na taj način će, posle pronalaženja ekvivalentne vrednosti izraza i izvršenja odgovarajuće naredbe, **switch**-naredba biti izvršena.

```
switch (izraz_1) {  
    case konst_1 : naredba_1; break;  
    case konst_2 : naredba_2; break;  
    . . .  
    case konst_n : naredba_n; break;  
    default :      naredba_d; break;  
}
```

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

- Uvod
- Programske naredbe
- Elementarna naredba
- Blok naredbi
- Naredbe granjanja - if
- Naredbe granjanja - switch**

Ukoliko ekvivalentna vrednost izraza u **case**-delovima nije pronađena, program će nastaviti sa izvršenjem naredbe koja se nalazi u **default**-delu.

U slučaju da taj deo ne postoji, naredba **switch** se preskače i program izvršava prvu naredbu, koja sledi **switch**-naredbu.

default-deo se može pojaviti najviše jedanput u **switch**-naredbi a mesto njegovog pojavljivanja je proizvoljno.

2. Tipovi u programskom jeziku C
3. Promenljive u programskom jeziku C
4. Konstante u programskom jeziku C
5. Operatori u programskom jeziku C
6. Algoritamsko resavanje problema
7. Vrste programskih naredbi

Uvod

Programske naredbe

Elementarna naredba

Blok naredbi

Naredbe granjanja - if

Naredbe granjanja - switch

Primer: Napisati program koji za unetu heksadecimalnu cifru štampa decimalnu cifru.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char hekza_cifra;

    printf ("Unesite heksadecimalnu cifru\n");
    scanf ("%c", &hekza_cifra);

    switch (hekza_cifra) {
        case 'A' :
        case 'a' : printf ("%10s", "10"); break;
        case 'B' :
        case 'b' : printf ("%10s", "11"); break;
        case 'C' :
        case 'c' : printf ("%10s", "12"); break;
        case 'D' :
        case 'd' : printf ("%10s", "13"); break;
        case 'E' :
        case 'e' : printf ("%10s", "14"); break;
        case 'F' :
        case 'f' : printf ("%10s", "15"); break;
        case '0' :
        case '1' :
        case '2' :
        case '3' :
        case '4' :
        case '5' :
        case '6' :
        case '7' :
        case '8' :
        case '9' : printf ("%10c", hekza_cifra); break;
        default : printf ("nije neksadecimalna cifra\n"); break;
    } // of switch

    return 0;
}
```