

Наредби за избор

1. Наредба if

Во програмскиот јазик C++ за еднократно разгранување се користи исказот if кој има општ облик:

```
if(logicki_izraz) iskaz;  
    iskaz_po_razgranuvanjeto;
```

Ако вредноста на логичкиот израз (logicki_izraz) е вистинита (true, 1), се изведува исказот по изразот if(logicki_izraz). Ако вредноста на логичкиот израз (logicki_izraz) е неvistинита (false, 0), исказот по изразот if(logicki_izraz) се прескокнува и се изведува следниот исказ по разгранувањето.

Со следниов исказ се проверува дали дадениот број x е негативен:

```
if(x<0)  
    cout<<"Brojot "<<x<<" e negativen";
```

Ако x има негативна вредност на пр. -10, логичкиот израз $x < 0$ ќе биде точен, односно ќе има вредност 1, па ќе се изврши исказот

```
cout<<"Brojot "<<x<<" e negativen";.
```

На екранот ќе се прикаже: „Brojot -10e negativen“.

Ако x нема негативна вредност, на пр. 10, тогаш логичкиот израз $x < 0$ ќе биде неточен, односно ќе има вредност 0, па исказот

```
cout<<"Brojot "<<x<<" e negativen";
```

нема да се изврши. На екранот ништо нема да се прикаже.

Важно!

По изразот if (logicki_izraz) не се става точка и запирка (;). Ако ова го направиш преведувачот ќе мисли дека знакот ; припаѓа на наредба која треба да се изврши ако условот е исполнет, па ќе имаш несакан ефект. Помеѓу условот и знакот ; нема ништо, па преведувачот гледа празна наредба и програмата нема ништо да изврши.

Често се случува грешка и наместо операторот == се користи операторот =. Ова може да доведе до неисправно работење на програма.

2. Наредба if-else

Во програмскиот јазик C++ за двократно разгранување се користи наредба if-else која има општ облик:

```
if(logicki_izraz) iskaz1;  
    else iskaz2;  
    iskaz_po_razgranuvanjeto;
```

Ако вредноста на логичкиот израз (logicki_izraz) е вистина (true, 1), тогаш се извршува наредбата iskaz1. Откако таа ќе се изврши се извршуваат наредбите по исказот if else.

Ако вредноста на логичкиот израз е неvistина (false, 0), наредбата iskaz1 се прескокнува и се извршува наредбата по else – iskaz2. Откако таа ќе се изврши се извршуваат наредбите по исказот if else.

По изразите if и else не се става знакот точка и запирка (;).

Пр. Програмата за два дадени различни цели броја го прокажува поголемиот број:

```
# include <iostream>  
using namespace std;  
int main()  
{  
    int a,b;  
    cout<<"Vnesi dva razlicni celi broja";  
    cin>>a>>b;  
    if (a>b) cout<<"Brojot "<<a<<" e pogolem "<<endl;  
    else cout<<"Brojot "<<b<<" e pogolem "<<endl;  
    return 0;  
}
```

Пр. Програмата проверува дали даден природен број е едноцифрен:

```
# include <iostream>
using namespace std;
int main()
{
int x;
cout<<"Vnesi cel broj";
cin>>x;
if (0<=x && x<=9) cout<<"Brojot "<<x<<" e ednocifren "<<endl;
else cout<<"Brojot "<<x<<" ne e ednocifren "<<endl;

return 0;
}
```

3. Блок од искази

Важно е да се знае дека исказот `if` контролира извршување само на првиот исказ кој се наоѓа веднаш по него. Веќе следниот исказ не е дел од исказот `if` па тој исказ ќе се изврши независно дали условот е точен или не. Да го погледнеме следниот пример:

```
if(x<0)
    cout<<"Brojot "<<x<<" e negativen."<<endl;
    cout<<"Negovata apsolutna vrednost e "<<-x;
```

Втората нареба по исказот `if` ќе се изврши и за позитивните броеви па за нив ќе се испише погрешна апсолутна вредност.

Овој проблем се решава со градење на *блок од искази*. **Кога исказот `if` контролира два или повеќе искази, сите тие се ставаат меѓу големи загради.**

Претходниот пример исправно се запишува на следниот начин:

```
if(x<0)
{
    cout<<"Brojot "<<x<<" e negativen."<<endl;
    cout<<"Negovata apsolutna vrednost e "<<-x;
}
```

Пр. Да се напише програма со која се одредува дали даден број е парен.

```
# include <iostream>
using namespace std;
int main()
{
int n;
cout<<"vnesi cel broj";
cin>>n;
if (n%2 == 0) cout<<n<<" e paren broj";
else cout<<n<<" ne e paren broj";
return 0;
}
```

4. Техника на вгнездување на искази

Во блокот од искази може да се користат било кои искази, па така и исказот за избор од две можности. Впрочем, и во секојдневниот живот често се случува еден условен настан да зависи од друг условен настан и е потребно да се испитаат повеќе услови. Тогаш еден услов е надворешен, а друг е внатрешен или вгнезден.

Во програмските јазици дозволено е да се користат повеќе `if` искази вгнездени еден во друг било во блокот на искази по `if` било во блокот од искази по `else`. Ова се нарекува техника на вгнездени искази.

Во програмските јазици дозволено е да се користат повеќе `if` искази вгнездени еден во друг, било во блокот на искази по `if`, било во блокот од искази по `else`. Ова се нарекува техника на вгнездени искази.

5. Структура за избор од повеќе можности

Повеќекратното разгранување е добро решение кога се работи за избор од повеќе можности, но може да биде заморно и кодот да стане предолг ако се работи за поголем број на можни избори. Во програмските јазици во такви прилики се користи структурата за избор од повеќе можности.

Наредбата за повеќекратно разгранување е:

```
switch (izraz)
{
    case konst_1:          blok_na_iskazi_1;
                          break;
    case konst_2:          blok_na_iskazi_2;
                          break;
    ....
    case konst_n:          blok_na_iskazi_n;
                          break;
    default:
                          posledniot_blok_na_iskazi;
}
```

Вредност на изразот се споредува со целобројните константи: konst_1, konst_2, konst_3, итн. Ако вредноста на изразот е еднаква на некоја од дадените константи, ќе се изврши блокот на искази придружен на таа константа. По извршувањето на тој блок, исказот break го прекинува извршувањето на switch-case исказот и програмата продолжува со првата наредба по овој исказ. Ако вредноста на изразот не е еднаква на ниту една од дадените константи, се извршува блокот на искази по default.

Пр.1. Да се напише програма за прикажување на општиот успех на ученикот ако е даден бројниот успех:

```
1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int oценка;
7      cout<<"Vnesi oценка od 1 do 5! "<<endl;
8      cin>>ocenka;
9      switch (ocenka)
10     {
11         case 1:
12             cout<<"Vasiot uspeh e nedovolen!"<<endl;
13             break;
14         case 2:
15             cout<<"Vasiot uspeh e dovolen!"<<endl;
16             break;
17         case 3:
18             cout<<"Vasiot uspeh e dobar!"<<endl;
19             break;
20         case 4:
21             cout<<"Vasiot uspeh e mnogu dobar!"<<endl;
22             break;
23         case 5:
24             cout<<"Vasiot uspeh e odlicen!"<<endl;
25             break;
26         default:
27             cout<<"Vnesenata oценка ne e regularna!"<<endl;
28     }
29     system ("PAUSE");
30     return 0;
31 }
```

Пр.2. Да се напише програма со кој се внесува природен број помал од 10 и да се отпечати дали тој е прост, делив со 2, делив со 3 или е совршен. (бројот е совршен ако е делив со збирот на своите

делители без 1 и самиот себе, пр. 6)

```
#include <iostream>
using namespace std;
int main()
{
int n;
cout<<"vnesi proroden broj pomal od 10"<<endl;
cin>>n;
switch (n)
{
case 1:cout<<n<<" ne e ni prost ni slozen broj";
break;
case 2:
case 3:
case 5:
case 7: cout<<" brojot e prost";
break;
case 4:
case 8: cout<<"brojot e deliv so 2";
break;
case 6:cout<<"brojot e sovršen";
break;
case 9:cout<<"brojot e deliv so 3";
break;
default: cout<<"gresno vnesen broj";
}
return 0;
}
```

Пр.3. Да се напише програма со кој се внесуваат два броја и оператор +, -, * или / и да се отпечати резултатот во зависност од внесениот оператор.

```
1  #include <cstdlib>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      float a,b; char znak;
7      cout<<"Vnesi dva broja: ";
8      cin>>a>>b;
9      cout<<"Vnesi znak za operacija: ";
10     cin>>znak;
11     switch (znak)
12     {
13         case '+':
14             cout<<a<<znak<<b<<"="<<a+b<<endl;
15             break;
16         case '-':
17             cout<<a<<znak<<b<<"="<<a-b<<endl;
18             break;
19         case '*':
20             cout<<a<<znak<<b<<"="<<a*b<<endl;
21             break;
22         case '/':
23             if (b == 0)
24             {
25                 cout<<"Ne e dozvoleno delenje so 0!"<<endl;
26                 break;
27             }
28             cout<<a<<znak<<b<<"="<<a/b<<endl;
29             break;
30         default:
31             cout<<"Pogresno vnesen znak!"<<endl;
32     }
33     system ("PAUSE");
34     return 0;
35 }
```

Пр. Да се напише програма со која се внесуваат два различни броја, а потоа се печати помалиот, па поголемиот број.

```
# include <iostream>
using namespace std;
int main()
{
    int k,p;
    cout<<"vnesi dva broja";
    cin>>k>>p;
    if (k<p) cout<<k<<" "<<p;
        else cout<<p<<" "<<k;
    return 0;
}
```

пр. да се напише програма со која се печати средната цифра на трицифрен цел број.

```
# include <iostream>
using namespace std;
int main ()
{
    int n,n1,n2;
    cout<<"vnesi cel broj";
    cin>>n;
    n1=n/10;
    n2=n1%10;
    cout<<"srednata cifra e "<<n2;
    return 0;
}
```

Пр. Која вредност ќе ја добие променливата а по извршување на следните наредби:

```
float a = 1.56;
if (a < 1)
a = a+1;
else
{
a = a+4;
a = -a+7;
}
a=1.44
```

Пр. Што ќе се прикаже по извршување на следниве искази?

```
int broj = -6;
char bukva='K';
if (broj <0)
{
    if (bukva<'S')
        cout<<1;
    else
        cout<<2;
}
else
cout<<3;
```

1

Пр. Да се напише алгоритам со се одредува колку денови има даден месец во годината ако месецот се внесува со неговиот реден број.

```
# include <iostream>
using namespace std;
int main()
{
```

```
int n;  
cin>>n;  
switch(n)  
{  
  case 1:  
  case 3:  
  case 5:  
  case 7:  
  case 8:  
  case 10:  
  case 12: cout<<31;  
  break;  
  case 2:cout<<28;  
  break;  
  case 4:  
  case 6:  
  case 9:  
  case 11:cout<<30;  
  break;  
}  
return 0;  
}
```