

Pogojni stavek

Pogojni (if) stavek

Tip bool

Primerjanje

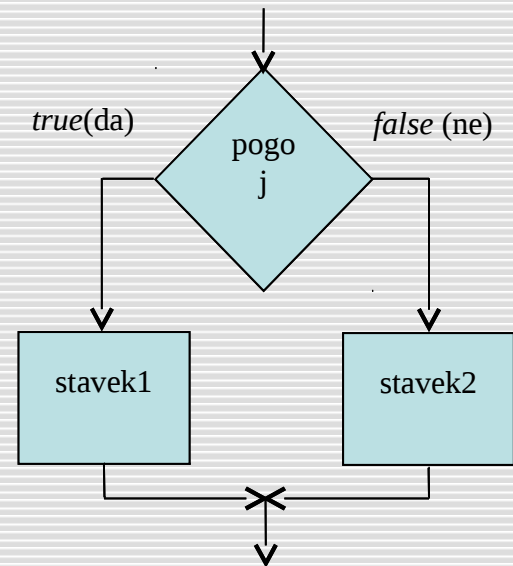
If stavek

- ❑ Vsi dosedanja programi so se izvajali zaporedoma, ni bilo nobenih vejitev
 - ❑ Program razvejimo na osnovi odločitev – pogojnega stavka
 - Če to, potem naredimo eno
 - Če drugače, naredimo drugo
 - ❑ Primer: Program nas ob zagonu vpraša po imenu, nato po uri in nas v skladu z uro pozdravi
 - Če je ura < 8 , z *Dobro jutro!*
 - Drugače z *Dober dan!*
-

Popoln pogojni stavek

- Popolni pogojni stavek: vejitev na dve veji
 - Preverimo pogoj `p`
 - Če je pogoj resničen (`true`), izvedemo stavek1 (ali več stavkov)
 - Če pogoj ni resničen (`false`), izvedemo stavek2 (ali več stavkov)
 - Veji se združita, program se nadaljuje

```
if (pogoj)
    stavek1;
else
    stavek2;
```



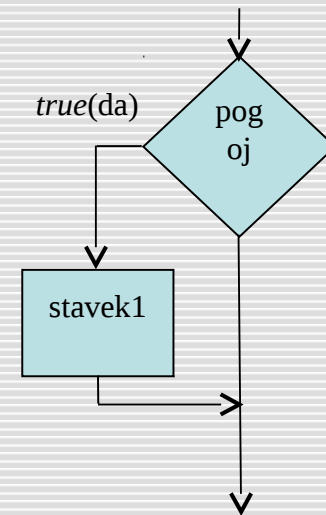
Popoln pogojni stavek - zgled

```
Console.Write("Tvoje ime: ");
string ime = Console.ReadLine(); // preberemo ime
Console.Write("Koliko je ura: ");
string uraNiz = Console.ReadLine();
int ura = int.Parse(uraNiz); //pretvorba niza v celo število
string pozdrav;
if (ura < 8)
{
    pozdrav = "Dobro jutro";
}
else
{
    pozdrav = "Dober dan";
}
Console.WriteLine(pozdrav + ", " + ime + "!");
Console.WriteLine("\t\t\t... končamo z Enter ...");
//napovemo kako končamo, uporabimo tri tabulatorje
Console.ReadLine();
```

Nepopoln pogojni stavek

- Nepopolni pogojni stavek: to obliko uporabimo, če takrat, ko pogoj ni izpolnjen, ne naredimo ničesar
 - Preverimo pogoj `p`
 - Če je pogoj resničen (`true`) izvedemo stavek1 (ali več stavkov v tem primeru stavke zapišemo v blok = {...})
 - Če pogoj ni resničen (`false`), se ne zgodi nič
 - Veji se združita, program se nadaljuje

```
if (pogoj)  
    stavek1;
```



Pogojni stavek

- Če je stavkov v obeh vejah več, moramo obvezno uporabiti oklepaje – **blok { ... }**

```
if(pogoj)    //ce je pogoj izpolnjen, se izvedejo
{
    stavki stavek1 do stavekn
    stavek1;
    stavek2;
    ...;
    stavekn;
}
else        //ce pa pogoj NI izpolnjen, se izvedejo
{
    stavki stavekA do stavekX
    stavekA;
    stavekB;
    ...;
    stavekX;
}
```

Pogoji – relacijski operatorji

- Primerjanje (relacijski operatorji)
 - > večje
 - < manjše
 - >= večje ali enako (vrstni red pomemben!)
 - <= manjše ali enako
 - == enako (pozor dva (2) enačaja)
 - != različno
-

Logične vrednosti

- Tip `bool`
- Vrednosti le `true` in `false`.

```
bool vRedu, konec;  
vRedu = true;  
konec = false;
```

- Vaja: kakšno vrednost dobita spremenljivki `v` `p` in `q`?

```
int a = 17;  
int b = 5;
```

```
bool p = (a < b);  
bool q = (a <= b + 12);
```

Logične operacije

- Logične vrednosti lahko združujemo z operatorji
 - `&&` in
 - `||` ali
 - `!` Ne
 - `A && B`: res, če sta res in A in B (sta oba `true`)
 - `A || B`: res, če je vsaj eden res oziroma narobe le, če sta oba `false`
 - `!A`: res (`true`), če je A napačen (`false`)
-

Logične operacije - povzetek

- Vrednosti
 - true res
 - false ni res
 - Operacije (p in q sta logični vrednost oz. izraza, ki vrneti logično vrednost)
 - p && q p in q
 - p || q p ali q
 - !p negacija p
 - Logične vrednosti najpogosteje dobimo kot rezultat primerjav
 - izrazA == izrazB izrazA je enak (ima isto vrednost) kot izrazB
 - izrazA != izrazB izrazA ni enak (nima iste vrednosti) kot izrazB
 - izrazA < izrazB izrazA je manjši kot izrazB, oz. izrazA ima manjšo vrednost kot izrazB
 - izrazA <= izrazB izrazA je manjši ali enak kot izrazB
 - izrazA > izrazB izrazA je večji kot izrazB, oz. izrazA ima večjo vrednost kot izrazB
 - izrazA >= izrazB izrazA je večji ali enak kot izrazB
-

Zgled - delitelj

- Napiši program, ki prebere celi števili a in b ter ugotovi, ali je število a delitelj števila b
 - Če a deli b izpiši `true`, če pa a ne deli b , izpiši `false`.
 - Število je delitelj drugega, če je ostanek pri deljenju enak 0
$$deli = (b \% a == 0);$$
 - Izvajanje (npr. b je 16, $a = 3$)
 - Izračunamo $b \% a \rightarrow$ dobimo 1
 - $1 == 0 \rightarrow$ rezultat je `false`
 - V spremenljivko `deli` shranimo `false`
-

Zgled – delitelj: program v C#

```
Console.Write("Vnesi število, katerega deljivost te  
zanima: ");  
string beri = Console.ReadLine(); // preberemo prvo  
int deljenec = Convert.ToInt32(beri);  
Console.Write("Vnesi delitelja: ");  
beri = Console.ReadLine(); // preberemo delitelja  
int delitelj = Convert.ToInt32 (beri);  
bool deli = (deljenec % delitelj == 0);  
Console.WriteLine("Trditev, da " + delitelj + " deli " +  
deljenec + " je " + deli );  
Console.WriteLine("\n\n\nKončamo z Enter ...");  
//napovemo kako končamo  
Console.ReadLine();
```

Zgled – prestopno leto

- Napišimo program, ki prebere leto (celo število) in pove, ali je prestopno.
 - Leto je prestopno, če je deljivo s 4.
 - Izjema so leta deljiva s 100, ki niso prestopna.
 - Dvojna izjema so leta deljiva s 400, ki so prestopna.

 - Leto 1980 je prestopno, ker je deljivo s 4 in ni deljivo s 100.
 - Leto 1700 ni prestopno, ker je deljivo s 100.
 - Leto 2000 je prestopno, ker je deljivo s 400.
-

Zgled – prestopno leto

- ❑ Pogoji, da je leto prestopno, je torej, da je
 - deljivo s 4 IN NI deljivo s 100 ALI
 - Je deljivo s 400
 - ❑ Deljivo s 4
`leto % 4 == 0 // POZOR NA dvojni enacaj== !!!`
 - ❑ NI deljivo s 100:
`leto % 100 != 0 // ali (!(leto % 100 == 0))`
 - ❑ Deljivo s 400
`leto % 400 == 0`
 - ❑ Deljivo s 4 IN NI deljivo s 100
`((leto % 4 == 0) && (leto % 100 != 0))`
 - ❑ Prestopno:
`((leto % 4 == 0) && (leto % 100 != 0)) || (leto % 400 == 0)`
-

Zgled – prestopno leto: program v C#

```
Console.Write("Vnesi letnico: ");
string beri = Console.ReadLine(); // preberemo prvo
int leto = int.Parse(beri);
bool prestopno = (((leto % 4 == 0) && (leto % 100 != 0))
    || (leto % 400 == 0)); // formula za prestopnost
string odgovor = "Leto " + leto; // sestavimo odgovor
if (prestopno)
    odgovor = odgovor + " je ";
else
    odgovor = odgovor + " ni ";
Console.WriteLine(odgovor + "prestopno leto!");
Console.WriteLine("\n... končamo z Enter ...");
Console.ReadLine();
```

Zgled – največja številka

- ❑ Ustvarimo naključno naravno število med 100 in 999 Ugotovi največjo številko v tem številu!
 - ❑ $437 \rightarrow 7$, $454 \rightarrow 5$, $661 \rightarrow 6$, $781 \rightarrow 8 \dots$
 - ❑ Ideja
 - Zaporedoma jemljemo številke
 - enice (%)
 - /
 - Če je trenutna številka večja od doslej največje, si jo zapomnimo!
 - Ko smo pregledali vse 3, poznamo največjo!
-

Največja številka: program v C#

```
Random naklj = new Random();//generator naključnih števil
int stevilo = naklj.Next(0, 900) + 100;//naključno trimestno število
Console.WriteLine("Ustvarjeno število je "+stevilo);
int maxstevka = stevilo % 10;
stevilo = stevilo / 10;

int desetice = stevilo % 10;
if (desetice > maxstevka)
    maxstevka = desetice;

stevilo = stevilo / 10; //ostanejo še stotice
if (stevilo > maxstevka)
    maxstevka = stevilo;
Console.WriteLine("Največja številka je " + maxstevka);

Console.WriteLine("\n\n\nKončamo z Enter ...");
//napovemo kako končamo
Console.ReadLine();
```

Sestavljeni stavek

- Je poljubno zaporedje stavkov med { }
{
 stavek₁;
 stavek₂;
 ...
 stavek_n;
}
 - Na koncu, za zavitim zaklepajem, NI podpičja
 - Za lažje delo, oz. priporočilo: pri pogojnem stavku VEDNO uporabimo { }
 - Najpogosteje takoj naredimo oba oklepaja – ni težav s pozabljanjem
-

Gnezdeni pogojni stavek

- Znotraj pogojnega stavka je lahko poljuben stavek – tudi pogojni stavek!

```
if (g == 10)
{
    g = 0;
    dag = dag + 1;
    if (dag == 100)
    {
        dag = 0;
        kg = kg + 1;
    }
}
```

Primerjava dveh števil po velikosti

- Izpiši v kakšnem "odnosu" sta si števili
 - Manjše
 - Večje
 - Enako
- $st1 < st2$
 - DA: vemo odgovor
 - NE: dve možnosti!

```
if (st1 < st2)
{
    odg = "manjše";
}
else
{
    if (st1 == st2)
    {
        odg = "enako";
    }
    else
    {
        odg = "večje";
    }
}
```

Kaj naredi del programa?

```
Console.Write("Število točk: ");
točke = Console.ReadLine();
st_tock = Convert.ToInt32(točke);
if (st_tock >= 75)
{
    if (st_tock < 90) ocena = "prav dobro";
    else ocena = "odlično";
}
else
{
    if (st_tock >= 60) ocena = "dobro";
    else
    {
        if (st_tock < 40) ocena = "nezadostno";
        else ocena = "zadostno";
    }
}
odg = "Za " + točke + " točk dobiš oceno " + ocena;
```

Pogojni stavek - zaključek

- Pomen:
če je pogoj p izpolnjen, se izvedejo $stavek_{1a}, \dots, stavek_{na}$ sicer pa $stavek_{1b}, \dots, stavek_{mb}$
 - Pazi na oklepaje okoli pogoja.
 - Stavki se izvedejo, če je pogoj izpolnjen.
 - Nepopoln pogojni stavek uporabimo, če takrat, ko pogoj ni izpolnjen, nimamo opraviti ničesar.
 - Stavki v $\{ \}$ se torej izvedejo le, če je pogoj izpolnjen. Če ni izpolnjen, se ne zgodi nič.
 - Pogoj je lahko sestavljen – pri tem uporabimo logične operatorje $\&$ (logični IN), $\|$ (logični ALI) ali $!$ (negacija)
-

Vaje

- Sestavimo program, v katerega preko tipkovnice vnesemo število stranic pravilnega večkotnika in dolžino stranice. Program izračuna in izpiše njegov obseg. Če je število stranic premajhno, da bi lahko tvorile lik, izpiše: "*Napaka! Število stranic je premajhno.*".
 - Program naj zahteva vnos starosti neke osebe, nato pa naj izpiše za kakšno vrsto osebe gre in sicer

do 2 leti	Dojenček
3-10 let	Mladoletnik
11-19 let	Najstnik
20 in več	Odrasla oseba
-

Vaje

- Ugotovi pravilnost oz. nepravilnost naslednjega pogoja, če je $x = 5$
 $((3 > x) \parallel (5 \leq x)) \&\& (x \neq 8)$
- Kakšna je vrednost spremenljivke N po izvedbi naslednjega stavka if?

```
int N = 1;  bool B = true;
if ((N < 5) && B)
{
    N = N + 1;
}
else
{
    N = 0;
}
```


Vaje

- Kakšno vrednost ima spremenljivka potem, ko se izvede prireditveni stavek:
 - `bool f = !(3 <= 5 && 6 != 7);` _____
 - `bool g = (3 > 2) || (3 > 5 && 6 > 3);` _____
 - `bool b = 12 <= 4 * 3;` _____
 - `b = true && (2 < 1);` _____
 - `b = (7 < 4) || (4 >= 5);` _____
 - `bool n = !(5 < 5) && (2 <= 3);` _____
 - `n=!n`
-

Vaje

- Katere kombinacije znakov izmed navedenih so relacijski operatorji v C#

- =<
 - !=
 - >=
 - <>
 - ==!
 - >
 - =
-

Vaje

- ❑ Sestavi logični izraz, ki vrne true, če je spremenljivka x sodo število večje od 50 in manjše od 100, ki ni deljivo s sedem.
 - ❑ Sestavi logični izraz, ki ima vrednost true, če je spremenljivka x sodo število med (vključno) 50 in 100 ali pa lihi večkratnik števila 3.
 - ❑ Napišite logični izraz, ki ima vrednost true takrat, če je pozitivno celo število v spremenljivki semTromestno res tromestno in false sicer. 123 je tromestno, števila 23, 1245, 1123 pa niso.
-

Vaje

- Preberi dve dolžini v metrih, decimetrih in centimetrih in ugotovi, katera je večja. Namig: če boš obe dolžini pretvoril v centimetre, bo odločanje veliko lažje!

 - Sestavite program, ki bo prebrano oceno po ameriški lestvici (torej znak A, B, ...) pretvoril v slovensko. Pri tem upoštevajte, da velja:
 - *A odlično,*
 - *B prav dobro,*
 - *C in D dobro,*
 - *E zadostno in*
 - *F nezadostno.*
 - Oceno izpišite na zaslon z besedo.
-

Vaje

- Napiši program, ki zahteva vnos stranic trikotnika in ugotovi, ali tak trikotnik sploh obstaja, ali je trikotnik pravokoten, ali je trikotnik enakokrak in ali je trikotnik mogoče enakostraničen.
 - Preberi poljubno celo število. Ugotovi in izpiši, ali je sodo ali liho!
 - Napiši program, ki prebere tromestno celo število in ugotovi, ali je število desetice večje od vsote enic in stotic tega števila.
-

Vaje

- Napiši program, ki prebere tromestno število in ugotovi, če je število palindrom. Število je palindrom, če sta število in njegov obrat enaka.
 - Napiši program, ki ustvari naključno dvomestno število. Ugotovi in izpiši, ali je to število večje od 50, je sodo in ni deljivo s 5!
-