

Instructiuni de control

- Introducerea caracterelor de la tastatura
- Instructiuni de selectie (**if, switch**)
- Instructiuni repetitive (**for, while, do-while, foreach**)
- Instructiuni de salt (**break, continue, goto, return**)

Introducerea caracterelor de la tastatura

- Citirea de la tastatura se face cu metodele `Console.Read();` sau `Console.ReadLine();`
- Prima dintre ele returneaza un intreg, care trebuie convertit apoi la tipul de date `char`.
- Cea de-a doua returneaza un `string`.
- Datele de la consola sunt directionate intr-un buffer de linie. La apasarea tastei ENTER, in stream-ul de intrare se introduce secventa de caractere carriage return (`\r`) si line feed (`\n`). Faptul ca metoda `Read()` citeste dintr-un buffer de linie poate crea neplaceri daca este necesara o alta operatie de introducere de date, intrucat aceste date trebuie sterse din buffer (prin citire).

Exemplu:

```
/*introduceti un caracter de la tastatura iar  
apoi ENTER */  
using System;  
class citire  
{  
    public static void Main()  
    {  
        char ch;  
        Console.WriteLine("Apasati o tasta si  
apoi ENTER");  
        ch=(char)Console.Read();  
        Console.Read();  
        Console.Read();  
        Console.WriteLine("Ati tastat: "+ch);  
    }  
}
```

Instructiunea if

Forma generala

```
if (conditie)
    {instructiune1;}
else
    {instructiune2;}
```

- instructiune1** si **instructiune2** pot fi orice fel de instructiuni: de la instructiunea `vida(;)`, pana la blocuri complexe delimitate cu `{ }` sau alte instructiuni. In lipsa delimitarii corespunzatoare, va apare un mesaj de eroare! Se recomanda ca si o singura instructiune sa fie delimitata de `{ }`.
- daca expresia conditionala este adevarata se executa **instructiune1**, altfel (daca exista) se executa **instructiune2**.
- clauza `e/se` este optionala. In lipsa ei, daca conditia s-a evaluat cu false, se trece la urmatoarea instructiune din program.

Exemplu:

```
using System;
class Testif
{
    public static void Main()
    {
        char ch, rasp = 'A';
        Console.WriteLine("Introduceti un
        caracter");
        ch = Convert.ToChar(Console.ReadLine());
        if (ch == rasp)
        {
            Console.WriteLine("Ati nimerit
            raspunsul");
        }
        else
        {
            Console.WriteLine("Nu ati nimerit
            raspunsul");
        }
    }
}
```

Instructiunea **switch**

Valoarea expresiei *expresie*, care controleaza instructiunea si care poate fi de tip **char, short, byte, int** sau **string** se compara cu constantele (literalii) *exp_const*.

În caz de egalitate se execută instructiunea corespunzătoare.

Dacă valoarea determinată diferă de oricare din constantele specificate, se execută instructiunea specificată la **default, care apare o singură dată, nu** neaparat la sfârșit. Dacă **default lipsește se iese din switch.**

Valorile constantelor trebuie sa fie diferite; ordinea lor nu are importanță.

Contrar limbajelor C si C++, in C# daca un **case** este prevazut cu o serie de instructiuni atunci in mod obligatoriu acesta se termina cu **break;**

Acoladele ce grupeaza mulțimea **case-urilor sunt** obligatorii. După fiecare **case pot apare mai multe** instructiuni fără a fi grupate în acolade.

Instructiunea switch are forma generala:

```
switch (expresie)
{
case exp_const1:
    secventa instr1;
    break;
case exp_const2:
    secventa instr1;
    break;
case exp_constn:
    secventa instrn;
    break;
default:
    instructiune;
    break;
}
```

Instrucțiune **switch** validă pt. C și C++ însă neacceptată în C#

```
int i=2;  
switch(i)  
{  
case 1:Console.WriteLine(" 1");  
case 2: Console.WriteLine (" 2");  
case 3: Console.WriteLine (" 3");  
case 4: Console.WriteLine (" 4");  
default: Console.WriteLine (" etc.");  
}
```

In C și C++ Se va afișa: 2 3 4 etc!

In C# generează eroare.

Instrucțiunea **switch**

```
int i=2;  
switch(i)  
{  
case 1:Console.WriteLine(" 1");break;  
case 2: Console.WriteLine (" 2"); break;  
case 3: Console.WriteLine (" 3"); break;  
case 4: Console.WriteLine (" 4"); break;  
default: Console.WriteLine (" etc."); break;  
}
```

Se va afisa:

2

Chiar daca nu este permis ca o secventa case sa continue cu o alta, este posibil ca doua sau mai multe instructiuni case sa refere aceeasi secventa de cod. Exemplu:

```
switch (nota)
{
case 1:
case 2:
case 3:
case 4:
Console.WriteLine("Nota nesatisfacatoare.");
break;
//...
}
```

Se va obtine acelasi mesaj (Nota nesatisfacatoare) pt. oricare din valorile 1, 2, 3 sau 4.

Instructiunea **while**

Forma generala:

```
while (expresie)  
{  
    instructiuni;  
}  
instructiunea_urmatoare
```

Se evaluează *expresie*: dacă valoarea sa este *true* se execută *instructiunile* din blocul buclei și controlul este transferat înapoi, la începutul instrucțiunii **while**. Dacă valoarea expresiei este *false* se execută *instructiunea_urmatoare*.

Așadar *instructiune* se execută de zero sau mai multe ori.

Exemplu:

```
long factorial=1;  
int i=1, n;  
while (i++ < n)  
{  
    factorial *= i;  
}
```


Instructiunea **do...while**

Forma generala:

```
do
{
    instructiune
}
while (expresie);
instructiunea_urmatoare
```

Se execută *instructiune*. Se evaluează *expresie*: *dacă valoarea sa este nenulă* controlul este transferat înapoi, la începutul instrucțiunii *do..while*; **dacă valoarea este nulă** se execută *instructiunea_urmatoare*. **Așadar instructiune se execută o dată sau de mai multe ori**

```
using System;
class Testdo
{
    public static void Main()
    {
        char ch='a';
        Console.WriteLine("Introduceti un sir de caractere");
        do
        {
            ch = (char)Console.Read();
            Console.WriteLine("Ati introdus " + ch);
            Console.Read();
            Console.Read();
        }
        while (ch != 'x');
    }
}
```

Codul de mai sus citeste si afiseaza caracterele citite de la tastatura pana cand se citeste x.

Instructiunea **for**

Forma generala:

for (*initializare*; *conditie*; *iteratie*)

{*instructiune*}

instructiunea_urmatoare

Initializarea este de regula o instructiune de atribuire care fixeaza valoarea initial a variabilei de control al buclei. *Conditia* este o expresie de tip bool care stabileste daca bucla continua ciclarea. Expresia *iteratie* stabileste cantitatea cu care variabila de control al buclei se modifica la fiecare repetare a buclei.

Una, doua sau toate trei dintre expresii pot lipsi, dar cei doi separatori sunt obligatorii.

Exemple:

```
int i, suma=0;
```

```
for(i = 1; i<=100; i++)suma+=i;
```

```
int suma, i;
```

```
for(suma = 0, i=0; i <= 100; suma += i++);
```

```
int i = 0;
```

```
char c;
```

```
for(; (c = (char)Console.Read()) != '\n'; ++i)
```

```
Console.WriteLine(c);
```

Instructiunea **break**

Se utilizeaza in cadrul buclelor **for**, **while** sau instructiunii **switch**.
Produce ieșirea din bucla sau din **switch** și trece controlul la
instrucțiunea următoare

Exemplu:

```
for(int i=0;i<=10;i++)  
{  
    if(i==7)  
        break;  
    Console.WriteLine(i);  
}
```

Buclo for va rula incepand de la 0. Valoarea 7 nu va mai fi afisata deoarece instructiunea **if** se va evalua cu **true**, determinand iesirea din bucla **for**. Se va afisa 0,1,2,...,6

Instructiunea `continue`

Se utilizeaza in cadrul buclelor `for`, `while` sau instructiunii `switch`.
Intrerupe executia iteratiei curente și trece controlul la iteratia urmatoare.

Exemplu:

```
for(int i=0;i<=10;i++)  
{  
    if(i==7)  
        continue;  
    Console.WriteLine(i);  
}
```

De aceasta data nu se mai iese din bucla `for`, ci se sare la urmatoarea iteratie. Se va afisa 0,1,2,...,6,8,...,10

Instructiunea `goto`

Instructiunea `goto` muta controlul execuției programului către o instrucțiune etichetată (în cadrul aceleiași metode!). Această instrucțiune poate lua următoarele forme:

- **`goto etichetă;`**
- **`goto case expresie_constantă;`**
- **`goto default;`**

Folosirea acestei instructiuni NU este recomandata!

Exemplu (Utilizarea instructiunii goto)

```
using System;
class TestGOTO
{
    public static void Main()
    {
        const int a = 5;  const int b = -5;
        int x, cod_err = -1;
        x = int.Parse(Console.ReadLine());
        if (x == 0)
        {
            cod_err = 0;  goto eroare;
        }
        else if (x < b)
        {
            cod_err = 1;  goto eroare;
        }
        else if ( x > a)
        {
            cod_err = 2;  goto eroare;
        }
    }
}
```

```
eroare: //eticheta unde sare
switch (cod_err)
{
    case 0:
        Console.WriteLine(" Eroare x ={0}!", x);
        break;
    case 1: Console.WriteLine(" Eroare: Depasire inferioara!");
        break;
    case 2:
        Console.WriteLine(" Eroare: Depasire superioara!");
        break;
    default:
        Console.WriteLine("Nr. introdus este x={0}", x);
        break;
}
Console.ReadKey();
}
```

Funcție de valoarea variabilei x, programul afiseaza un anumit mesaj.