

## 1. Ce este o variabilă?

- în programare folosim variabile pentru a putea reține anumite valori (informații) pe parcursul execuției unui program.
- exemplu:

variabila **scor** reține un număr.  
la început **scor <- 0**

dacă câștig atunci:

**scor <- scor + 1** (scorul crește cu 1)

dacă pierde atunci:

**scor <- scor - 1** (scorul scade cu 1)

## 2. Tipuri de date (variabile) uzuale în c++

Tip de variabilă	Exemplu de valoare	Semnificație
int	7 8 10 -2	Numere întregi (fără virgulă)
string	#include <string> "Ana are mere" "F" "-1" "10"	Șiruri de caractere (text de orice fel)
float	2.3 -4.5	Numere cu virgulă (precizie limitată)
double	5.999999	Numere cu virgulă (precizie crescută)
bool	true (1) false (0)	Valoare de adevăr (DA sau NU)
long	922337203685477580	Numere întregi foarte mari
auto	oricare de mai sus	Detectează automat tipul de variabilă după atribuirea acesteia.

### 3. Operații cu variabile

Operația	Sintaxa	Exemple
Declarare	<b>tip</b> nume_variabila;	<b>string</b> nume; <b>int</b> nr; <b>bool</b> e_frig;
Atribuire	nume_var = <b>valoare</b> ;	nume = "Ana"; nr = 5; e_frig = true;
Declarare + Atribuire	<b>tip</b> nume_var = <b>valoare</b> ;	<b>string</b> nume = "Maria"; <b>int</b> nr = 5; <b>bool</b> e_frig = false;
Extragere prin afișare (Consultare)	<b>cout</b> <<nume_var<<.. .	<b>cout</b> <<nume<< <b>endl</b> ; <b>cout</b> <<nr; <b>cout</b> <<e_frig<< <b>endl</b> ;
Extragere prin atribuire	var_a = var_b;	nume = "Ana"; alt_nume = nume; nr = nr+1;
Incrementare	var++; ++var; var = var+1;	nr++; ++nr; nr = nr+1;
Decrementare	var--; --var; var = var -1;	nr--; --nr; nr = nr - 1;

#### 4. Operatori

Operator	Sintaxa	Exemplu
<b>+</b>	<code>var = a + b;</code>	<code>int var; int a = 4; int b = 2; var = a + b; //var == 6</code>
<b>-</b>	<code>var = a - b;</code>	<code>var = a - b; //var == 4</code>
<b>* (produs)</b>	<code>var = a * b</code>	<code>8 == a * b;</code>
<b>/</b> <b>(div - împărțire</b> <b>întreagă pt int)</b>	<code>var = a / b;</code>	<code>var = a / b; //var == 2</code>
<b>%</b> <b>(modulo - restul</b> <b>împărțirii)</b>	<code>var = a % b</code>	<code>var = a % b; //var == 1</code>
<b>==</b> <b>(egal)</b>	<code>a == b //bool</code>	<code>int a = 5; int b = 2; var = a == b; //var == false</code>
<b>!=</b> <b>(diferit)</b>	<code>a != //bool</code>	<code>int a = 5; int b = 2; var = a != b; //var == true</code>
<b>&lt; / &gt;</b> <b>(mai mic/mare)</b>	<code>a &lt; b //bool a &gt; b</code>	<code>int a = 5; int b = 2; var1 = a &gt; b; //var1 == true var2 = a &lt; b; //var == false</code>
<b>&lt;= / &gt;=</b> <b>(mai mic/mare sau</b> <b>egal)</b>	<code>a &gt;= b //bool a &lt;= b</code>	<code>int a = 5; int b = 2; var1 = a &gt;= b; //var1 == true var2 = a &lt;= b; //var == false</code>
<b>!</b> <b>(not)</b>	<code>!var //(bool) //var: bool</code>	<code>var = true; var2 = !var; //var2 == false</code>

## 5. Aplicație 1

- se cere determinarea sumei a două numere naturale citite de la tastatură.
- exersare individuală pentru **diferență, produs și împărțire întreagă**

```
#include <iostream>

using namespace std;

int main()
{
    int numar1;
    int numar2;
    cout<<"dati primul numar"<<endl<<">>";
    cin>>numar1;
    cout<<"dati al doilea numar"<<endl<<">>";
    cin>>numar2;

    int suma = numar1 + numar2;
    int diferenta = numar1 - numar2;
    int produs = numar1 * numar2;
    int impartire = numar1 / numar2;

    cout<<"suma: "<<suma<<endl;
    cout<<"diferenta: "<<diferenta<<endl;
    cout<<"produsul: "<<produs<<endl;
    cout<<"catul: "<<impartire<<endl;

    return 0;
}
```

## 6. Structura de decizie

- în programare folosim structuri de decizie pentru a putea determina felul în care problema se rezolvă.
- anumite probleme se rezolvă diferit în funcție de datele conținute în variabile.

Exemplu:

Pentru a putea intra pe Facebook este nevoie de o parolă. Programul de la login **verifică dacă** parola introdusă este egală cu parola salvată în bazele de date Facebook.

```
//unde va în program:

string email_citire;
string parola_citire;

cout<<"email>>";
cin>>email_citire;
cout<<"parola>>";
cin>>parola_citire;

//se preia parola din baza de date:
string parola_corecta = Database().parola(email_citire);

//dacă parola este aceeași cu cea corectă:
if(parola_citire == parola_corecta){
    cout<<"Parola corecta!"<<endl;
}
//altfel (dacă ea este diferită de cea corectă):
else{
    cout<<"Parola gresita!"<<endl;
}
```

## 7. Aplicație 2

- exemplul de ora trecută verifica dacă un număr este par sau impar.
- un număr este par dacă restul împărțirii la 2 este 0.
- dacă restul împărțirii la 2 este 1, numărul este impar.

Exemplu:

5 : 2 = 2 r 1	5 este impar
<u>4</u>	
1	
4 : 2 = 2 r 0	4 este par
<u>4</u>	
0	

Română cu sintaxă ca în c++	C++
<pre> spune&lt;&lt;"introdu orice numar&gt;&gt;"; nr este int; citire&gt;&gt;nr;  dacă nr % 2 == 0 atunci{     spune&lt;&lt;nr&lt;&lt;" este par!"; }altfel{     spune&lt;&lt;nr&lt;&lt;" este impar!"; } </pre>	<pre> #include &lt;iostream&gt; using namespace std; int main() {     cout&lt;&lt;"introdu orice numar&gt;&gt;";     int nr;     cin&gt;&gt;nr;      if ( nr % 2 == 0){         cout&lt;&lt;nr&lt;&lt;" este par!";     }else{         cout&lt;&lt;nr&lt;&lt;" este impar!";     }     return 0; } </pre>

- ramura **else** este opțională.
- se poate introduce și ramura **else if** astfel
- mai multe despre compunerea structurilor discutăm în următoarea lecție.

```

if (conditie1){
    //conditie1 == true
}else if(conditie2){
    //conditie1 == false
    //conditie2 == true
}else if(conditie3){
    //conditie1 == false
    //conditie2 == false
    //conditie3 == true
}else{
    //conditie1 == false
    //conditie2 == false
    //conditie3 == false
}

```