



2. Datele

2.2. Operatorii

- Sunt **caractere speciale** (*, /, >, = etc.) sau **cuvinte cheie**(mod, and etc.) prin intermediul cărora se reprezintă operațiile care se efectuează în cadrul unui algoritm. Fiecare limbaj de programare are implementat propriul set de operatori.
- **Operatorii** care se folosesc în cadrul unui algoritm sunt:
 - Operatorul de atribuire
 - Operatorii matematici
 - Operatorul de concatenare a șirurilor de caractere
 - Operatorii relaționali
 - Operatorii logici.

- Operatorii pot fi aplicați numai pe anumite tipuri de operanzi, producând rezultate de un anumit tip. Dacă notăm cu a și b operanzii asupra cărora se aplică operatorul, cu \odot operatorul și cu c rezultatul:

$$a \odot b = c$$

atunci relația între a, b, \odot și c este dată de tabelul:

Tipul operanzilor a și b	Tipul operatorului \odot	Tipul rezultatului c
Numeric	Matematic (\mathcal{M})	Numeric
Numeric	Relațional (\mathcal{R})	Logic
Șir de caractere	Concatenare (\mathcal{C})	Șir de caractere
Șir de caractere	Relațional (\mathcal{R})	Logic
Logic	Logic (\mathcal{L})	Logic

Operatorii matematici

- $\mathcal{M} = \{+, -, *, /, **, ^, \text{mod}, \text{div}\}$
- Se aplică pe date de tip numeric și furnizează un rezultat de tip numeric

Operator	Semnificație	Exemplu
+ (adunare)	Adună matematic cei doi operanzi	$5+2=7$
- (scădere)	Scade al doilea operand din primul operand	$7-3=4$
/ (împărțirea reală)	Împarte primul operand la al doilea operand	$7/2=3.5$
* (înmulțire)	Înmulțește cei doi operanzi	$2*4=8$
** sau ^ (Ridicare la putere)	Ridică primul operand la puterea furnizată de al doilea operand	$2**3=8$
Mod (modulo)	Calculează restul împărțirii primului operand la al doilea operand	$19 \text{ mod } 4=3$
Div (împărțirea întreagă)	Calculează câtul împărțirii primului operand la al doilea operand	$19 \text{ div } 4 = 4$

Operatorii relaționali (de comparație)

- $\mathcal{R}=\{>, >=, <, <=, ==, !=, <>\}$
- Se aplică pe operanzi de tip numeric sau de tip șir de caractere și furnizează un rezultat de tip logic

Operator	Semnificație	Exemplu
== (egalitate)	Rezultatul este T dacă cei doi operanzi sunt egali	(5==5)=T (5==6)=F
<> != (diferit)	Rezultatul este T dacă cei doi operanzi sunt diferiți	(5!=5)=F (5!=6)=T
< (mai mic)	Rezultatul este T dacă primul operand este mai mic decât al doilea operand	(5<4)=T (5<6)=F
> (mai mare)	Rezultatul este T dacă primul operand este mai mare decât al doilea operand	(5>4)=T (5>6)=F
<= (mai mic sau egal)	Rezultatul este T dacă primul operand este mai mic sau egal decât al doilea operand	(5<=5)=T (7<=5)=F
>= (mai mare sau egal)	Rezultatul este T dacă primul operand este mai mare sau egal decât al doilea operand	(7>=5)=T (5<=7)=F

- Un operator relațional aplicat pe două șiruri de caractere realizează compararea celor două șiruri de caractere.
- Compararea a două șiruri de caractere este posibilă prin aplicarea numerică a codurilor celor două caractere. Astfel, codul ASCII al caracterului d este 100 iar a lui D este 68. Deci d este mai mare decât D. Acest lucru se scrie "d">"D" iar în urma aplicării operației de comparare, se produce rezultatul logic T deoarece operația de comparare se execută între cele două valori numerice.

- Compararea a două șiruri de caractere se face prin compararea codului ASCII al caracterelor din aceeași poziție a fiecărui șir. Dacă cele două șiruri nu au aceeași lungime, șirul de lungime mai mică este completat la sfârșit până la egalarea lungimilor cu caracterul care are codul ASCII 0. Operația de comparare începe cu prima poziție din șir și continuă cu următoarele poziții.

Operatorul de concatenare

- $C=\{+\}$
- Se aplică operanzilor de tip șir de caractere și furnizează un rezultat de tip șir de caractere

Operator	Semnificație	Exemplu
+ (concatenare)	Reunește două șiruri de caractere într-un singur șir de caractere	"Bună"+"ziua"="Bunăziua"

Operatorii logici

- Se aplică numai pe operanzi de tip logic și furnizează un rezultat de tip logic.
- $\mathcal{L}=\{\text{and, not, or}\}$

Operator	Semnificație	a	not a
not (negare)	Schimbă valoarea unui operand cu complementul său	T	F
		F	T

Operator	Semnificație	a	b	a and b	a or b
and (ȘI logic)	Rezultatul este T dacă ambii operanzi au valoarea T, altfel este F	T	T	T	T
		T	F	F	T
Or (SAU logic)	Rezultatul este T dacă cel puțin unui din operanzi are valoarea T, altfel este F	F	T	F	T
		F	F	F	F

- Operatorii logici sunt utili în construcții cu expresii logice care descriu anumite condiții ce vor fi testate, urmând ca, în funcție de rezultat, să se execute anumite operații.
- Relații importante:

$$\text{not}(a \text{ and } b) = (\text{not } a) \text{ or } (\text{not } b)$$
$$\text{not } (a \text{ or } b) = (\text{not } a) \text{ and } (\text{not } b)$$

Operatorul de atribuire

- Prin acest operator se poate atribui unei date, o numită valoare.

nume ← expresie

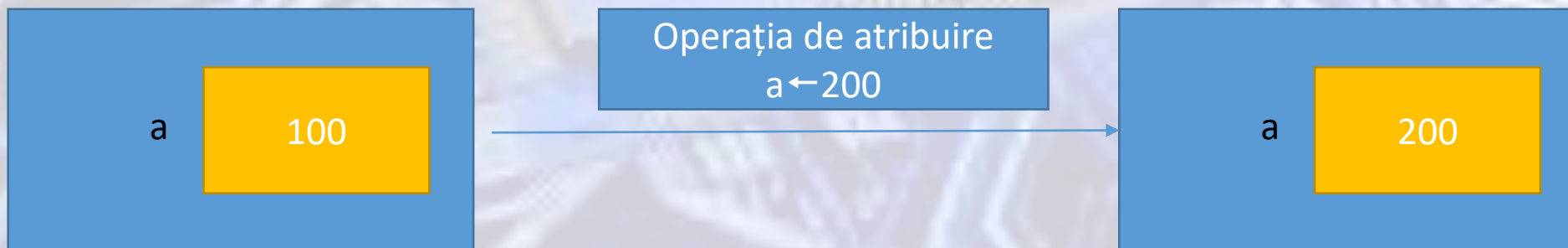
Operația de atribuire se desfășoară în două etape:

1. Se evaluează expresia
2. Valoarea obținută se atribuie datei identificate prin **nume**.

Rezultatul evaluării expresiei trebuie să fie de același tip cu data identificată prin **nume**.

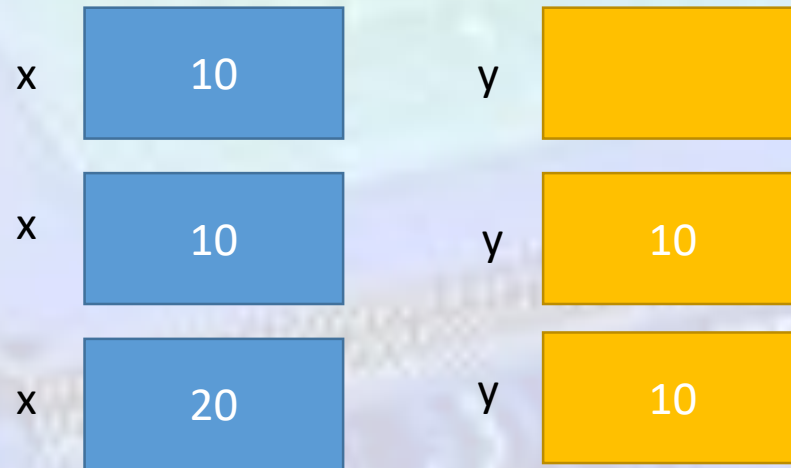
Exemplu	Semnificație
$n \leftarrow 100$	Datei de tip numeric i se atribuie valoarea 100
$n \leftarrow n + 100$	Se evaluează expresia adunându-se 100 la valoarea lui n. Valoarea obținută în urma evaluării este 200 și se atribuie ca nouă valoare datei n.
$\text{text} \leftarrow \text{"șir de caractere"}$	Datei de tip șir de caractere text i se atribuie valoarea "șir de caractere"

Prin operația de atribuire în zona de memorie alocată datei, se scrie noua valoare a datei, vechea valoare pierzându-se.



- Dacă se dorește păstrarea și a vechii valori, va trebui ca înainte de operația de atribuire sa o salvăm prin copierea într-o altă variabilă de memorie. Copierea se face tot prin atribuire:

$y \leftarrow x$
 $x \leftarrow 20$



- Operația de atribuire se folosește în cadrul algoritmilor pentru executarea operațiilor:
 - Inițializarea valorii unor variabile
 - Calculul iterativ al valorii unor variabile din memorie
- Calculul iterativ al valorii unei variabile de memorie înseamnă să-i atribuiți acelei variabile, valoarea unei expresii în care unul din operanzi este chiar numele acelei variabile de memorie.

- Contorul sau numărătorul: se folosește pentru a număra anumite cazuri care pot să apară. Contorul se inițializează cu valoarea 0 (nu este evidențiat nici un caz). Dacă în timpul prelucrării se întâlnește un caz, valoarea contorului se incrementează cu 1. $k \leftarrow k+1$.
- Suma: variabilă de memorie folosită pentru calculul iterativ al unei sume. $s \leftarrow 0$ și apoi $s \leftarrow s+a$
- Produsul: variabilă de memorie folosită pentru a calcula iterativ un produs. $p \leftarrow 1$, $p=p*a$.