

The background of the slide is a stylized, glowing blue and green circuit board. A central square component, possibly a microchip or a sensor, is highlighted with a bright green glow. The circuit traces are intricate and spread across the entire frame, creating a sense of depth and complexity. The overall color palette is dominated by shades of blue and green, with a dark, almost black background that makes the glowing elements stand out.

# 2. Datele

## 2.3. Expresiile

## **Expresia este combinație validă de operatori și operanzi**

- Operanzii pot fi **nume de date, constante de tip numeric** sau **șir de constante** care, în urma evaluării furnizează un singur rezultat. În funcție de **tipul operatorilor**, rezultatul expresiei poate fi de tip numeric, șir de caractere sau logic.

- Într-o expresie, operatorii se folosesc într-un anumit scop:
  - Operatori aritmetici – pentru efectuarea calculelor
  - Operatori relaționali și logici – pentru efectuarea unor comparații și luarea unor decizii în cadrul algoritmilor
  - Operatorul de atribuire – pentru manipularea datelor
- Expresia este validă numai dacă operatorii care leagă operanzii corespund tipului de operanzilor pe care îi leagă.

# Exemplu de expresii valide și nevalide

- $e \leftarrow 22+75$  este validă pentru că operatorul matematic + leagă doi operanzi exprimați prin două constante de tip numeric
- $e \leftarrow \text{"Ana"}+\text{"Maria"}$  – este validă pentru că operatorul de concatenare + leagă doi operanzi exprimați prin două constante de tip șir de caracter
- $e \leftarrow 22+\text{"ani"}$  – nu este o expresie validă deoarece operatorul + leagă doi operanzi exprimați prin două constante: una de tip numeric (22) și una de tip șir de caracter. Operatorul + nu poate fi interpretat nici ca operator matematic, nici ca operatorul de concatenare.

# Funcții

- Dacă într-o expresie trebuie să se calculeze radicalul unei date numerice, nu se poate folosi nici un operator matematic ci trebuie folosită o **funcție**.
- **Funcția** este predefinită de autorii limbajului de programare și se poate folosi în cadrul unei expresii ca și **operand**. Ea este de fapt un **program** care, în urma execuției **furnizează o valoare** prin numele ei.
- Când se evaluează o expresie în care este apelată o funcție, funcția este înlocuită de valoarea returnată în urma executării ei. Funcția se identifică printr-un nume iar pentru apelarea ei, se scrie numele funcției urmat de o listă de **parametri** precizați între paranteze rotunde. Parametrii sunt valorile cu care se execută funcția în acel apel.

# Exemplu funcții

- Funcția  $\sin(x)$  returnează valoarea funcției trigonometrice sinus pentru numărul  $x$ . Numele funcției este **sin** iar  $x$  este parametru, de tip numeric.
- Dacă se evaluează funcția  $\sin(x)$  iar  $x$  are valoarea 2.5, în urma execuției programului asociat, funcția va furniza valoarea 0.598.

# Evaluarea expresiilor

- Pentru a evalua o expresie, calculatorul va executa într-o anumită ordine operațiile pe care le-am scris. Această ordine este dată de:
  - **precedența operatorilor** – ordinea în care se execută operațiile definite de operatori. Ea determină nivelul de prioritate al operatorilor
  - **asociativitatea operatorilor** – este ordinea în care se evaluează operatorii cu același nivel de prioritate.

# Tabela de precedență

- Fiecare limbaj de programare are implementată o tabelă de precedență (tabela nivelurilor de prioritate) și pentru fiecare nivel de prioritate o anumită asociativitate:
  1. se evaluează funcțiile
  2. Se evaluează operatorii matematici:
    - a. Operatorul \*\* (ridicare la putere) are nivel de prioritate 1, fiind primul care se evaluează
    - b. Operatorii \*,/,div, mod au nivelul de prioritate 2
    - c. Operatorii +,- au nivelul de prioritate 3
  3. Se evaluează operatorii de concatenare
  4. Se evaluează operatorii relaționali. Toți operatorii au același nivel de prioritate
  5. Se evaluează operatorii logici. Ordinea de prioritate a operatorilor logici este **not**, **and** și **or**, primul fiind cel mai prioritar.
- Asociativitatea operatorilor este de la stânga spre dreapta. Toți operatorii care au același nivel de prioritate se evaluează în ordine, de la stânga la dreapta.



# Exemplu de evaluare a unei expresii cu operatori aritmetici

$$5*7*4+4/2-2**3\text{mod}2/4=$$

$$5*7*4+4/2-2**3\text{mod}2/4=$$

$$5*7*4+4/2-8\text{mod}2/4=$$

$$35*4+4/2-8\text{mod}2/4=$$

$$140+4/2-8\text{mod}2/4=$$

$$140+2-0/4=$$

$$142-0/4=$$

$$142-0=$$

$$142$$

# Exemplu de evaluare a unei expresii cu operatori relaționali

$3 \leq 5$  or  $7 > 8$

$3 \leq 5$  or  $7 > 8$

T or  $7 > 8$

T or F

T

$3 \leq 5$  and not  $7 > 8$

$3 \leq 5$  and not  $7 > 8$

T and not  $7 > 8$

T and not F

T and T

T

# Exemplu

**$e \leftarrow a * b ** d > c$  or  $a/b = d$  and  $a + b < d$  or not  $a > c$**

$e1 \leftarrow b ** d$  (operator aritmetic de prioritate 1)

$e2 \leftarrow a * e1$  (operator aritmetic de prioritate 2)

$e3 \leftarrow a/b$  (operator aritmetic de prioritate 2)

$e4 \leftarrow a + b$  (operator aritmetic de prioritate 3)

$e5 \leftarrow e2 > c$  (operator relațional)

$e6 \leftarrow e3 = d$  (operator relațional)

$e7 \leftarrow e4 < d$  (operator relațional)

$e8 \leftarrow a > c$  (operator relațional)

$e9 \leftarrow \text{not } e8$  (operator logic)

$e10 \leftarrow e6 \text{ and } e7$  (operator logic)

$s11 \leftarrow e5 \text{ or } e10$  (operator logic)

$e \leftarrow s11 \text{ or } e9$  (operator logic)

# Paranteze

- Ordinea de evaluare a unei expresii poate fi schimbată prin folosirea parantezelor. Se folosesc numai paranteze rotunde care pot fi și imbricate. Numărul de paranteze deschise trebuie să fie egal cu numărul de paranteze închise.

$$E = \frac{a + b}{c} + \frac{b + d}{c + a} + \frac{a + c}{(a + b)^3}$$

- Se scrie:

$$E \leftarrow (a+b)/c + (b+d)/(c+a) + (a+c)/((a+b)**3)$$

E, a, b, c, d sunt identificatori pentru datele prelucrate de calculator. E este dată de ieșire iar a, b, c, d sunt date de intrare.

# Exemplu de expresie care conține paranteze și mai multe tipuri de operatori

- $E \leftarrow ((a/b+c)*4+a) \bmod b > a*b \text{ or not } (a+c)*b=a*c$
- $e1 \leftarrow a/b$  (operator aritmetic de prioritate 2)
- $e2 \leftarrow e1+c$  (operator aritmetic de prioritate 2)
- $e3 \leftarrow e2*4$  (operator aritmetic de prioritate 2)
- $e4 \leftarrow e3+a$  (operator aritmetic de prioritate 3)
- $e5 \leftarrow a+c$  (operator aritmetic de prioritate 3)
- $e6 \leftarrow e5 \bmod b$  (operator aritmetic de prioritate 2)
- $e7 \leftarrow a*b$  (operator aritmetic de prioritate 2)
- $e8 \leftarrow e5*b$  (operator aritmetic de prioritate 2)
- $e9 \leftarrow a*c$  (operator aritmetic de prioritate 2)
- $e10 \leftarrow e6 > e7$  (operator relațional)
- $e11 \leftarrow e8 = e9$  (operator relațional)
- $e12 \leftarrow \text{not } e11$  (operator logic)
- $e \leftarrow e10 \text{ or } e11$  (operator logic)

Exemplu de expresie matematică prin care se calculează valoarea lui e:

- $\frac{\sqrt{a+1}-\sqrt{a-1}}{\sqrt{a+1}+\sqrt{a-1}} \times (2-a)$

- Necesită calculul radicalului de ordin 2 pentru care se folosește funcția sqrt(x). Când data x are valoarea 9, funcția sqrt(x) furnizează valoarea 3 iar când x are valoarea 49, furnizează valoarea 7

- Pentru a putea fi evaluată, expresia se scrie:

- $e \leftarrow ((\text{sqrt}(a+1)-\text{sqrt}(a-1))/(\text{sqrt}(a+1)+\text{sqrt}(a-1)))*(2-a)$

•  $e \leftarrow ((\sqrt{a+1}-\sqrt{a-1})/(\sqrt{a+1}+\sqrt{a-1}))*(2-a)$

$e1 \leftarrow a+1$

$e2 \leftarrow a-1$

$e3 \leftarrow a+1$

$e4 \leftarrow a-1$

$e5 \leftarrow \sqrt{e1}$

$e6 \leftarrow \sqrt{e2}$

$e7 \leftarrow \sqrt{e3}$

$e8 \leftarrow \sqrt{e4}$

$e9 \leftarrow e5-e6$

$e10 \leftarrow e7+e8$

$e11 \leftarrow e9/e10$

$e12 \leftarrow 2-a$

$e \leftarrow e11*e12$

# Precondițiile unei expresii

- Expresia definește cum trebuie să prelucreze calculatorul anumite valori ale datelor. De fiecare dată când calculatorul evaluează expresia, sunt prelucrate valorile curente ale datelor variabile care apar ca operanzi în cadrul expresiei. În unele cazuri trebuie stabilite condiții pentru valorile acestor date, pentru a se putea evalua expresia adică trebuie stabilite precondițiile expresiei
- Precondițiile expresiei reprezintă un ansamblu de restricții și constrângeri impuse datelor care apar în expresii ca operanzi.



- Constrangerile **operației de împărțire**. Operatorii folosiți pentru împărțire sunt  $/$ ,  $\text{div}$  și  $\text{mod}$ . Precondiția este ca împărțitorul să fie diferit de 0.
- Exemplu: în expresia  $a/(c+d)$  condiția este ca  $(c+d) \neq 0$

- Constrângerile **argumentului unei funcții**. Argumentele unei funcții trebuie să îndeplinească o anumită condiție. De exemplu  $\text{sqrt}(x)$  nu se poate aplica pe valori negative ale lui  $x$ . De exemplu, pentru o ecuație de gradul 2
- $-b + \text{sqrt}(b^2 - 4ac)$ , precondiția este ca  $(b^2 - 4ac) \geq 0$
- $-b + \text{sqrt}(b^2 - 4ac) / 2a$  precondițiile sunt  $(b^2 - 4ac) \geq 0$  and  $(a \neq 0)$

- Constrangerile pentru ca formula să fie validă. Formula folosită pentru a calcula o valoare poate fi validă numai pentru anumite valori ale datelor care sunt folosite ca operanzi. De exemplu, operatorii **div** și **mod** se pot aplica numai pe date de tip întreg