EVOLUȚIA CALCULATOARELOR

- Instrucțiune
- Program

- Programe-scriere

- ordin elementar dat calculatorului
- un ansamblu de mai multe instr.
- Cod mașină: 0 1
- Lb. de asamblare (ASIRIS)
- Lb. nivel înalt (QBASIC, PASCAl, C)

HARD- SOFT

- Hardware
- Software
  - Firmware
  - Malware

- Fizicul
- Sufletul

BIT - UNITATEA DE INFORMAȚIE

- Bit
- Octet (byte)
- Kb
- Mb
- Gb
- Tb

- Cifră binară
- 8 biți
- 2^10 octeți
- 2^10 Kb
- 2^10 Mb
- 2^10 Gb

ISTORIC: I ABACUL

- 1617
- 1642
- 1942
- 1957
- 1961

- Vergele J. Napier
- Maşina lui B. Pascal
- ENIAC – SUA
- CIFA 1 – Bucureşti
- MECIPT - Timişoara

GENERAȚII DE CALCULATOARE

- G 1
- G2
- G3
- G4
- 1950: tuburi electronice
- 1960: tranzistori
- 1964-1970: circuite integr.
- 1970-1980: c. i. sc. largă
- Exemple FELIX C-256, TIM S, SPECTRUM
- La ora actuală: putere de calcul enormă.

LIMBAJUL C++

- Prima versiune> 1978: Dennis Ritchie și Brian Kernighan

Caracteristici:

- Este un limbaj structurat: unitatea de structură este funcția;
- Limbajul este CASE SENSITIV;
- Limbajul are un control slab la erori (I. E. are un control și mai slab la erori);
- Are 32 de cuvinte cheie (rezervate): avem voie să le utilizăm doar în contextul care le definește;
- Orice instrucțiune se termină cu semnul ;
- Comentarii: pe un rând // SCRIU CE VREAU
- Comentarii pe mai multe rânduri

/* SCRIU CE VREAU*/

LIMBAJUL C++: TIPURI DE DATE

Tip	Biți	Valori
unsigned char	8	- 127....127
char	8	0...255
int	16	- 32 7677 .... 32 767
unsigned int	16	0....65535
long int	32	+/- 2 miliarde (aprox.)
unsigned long int	32	0...4 miliarde (aprox.)
float	32	Nr. reale (cu max. 6 z. e.)
double	64	Nr. reale (cu max. 10 z. e.)

LIMBAJUL C++

- Identificatori: succesiuni de maxim 32 de caractere,litere și cifre (eventual _), încep obligatoriu cu literă, nu sunt cuvinte cheie: cin, cout, for, int etc.
- Constantele: sunt de aceleași tipuri ca și datele, ele nu își modifică valoarea pe timpul execuției programului.
- Variabilele: sunt zone de memorie desemnate prin identificatori, ele își pot modifica valoarea în timpul execuției programului și sunt de același tipuri ca și datele specificate anterior.

LIMBAJUL C++: OPERATORI

Mate	Info	Mate/Info
+	+	Diferit: !=
-	-	Mai mic sau egal <=
2x3	2*3	Mai mare sau egal >=
8:2	8/2	a%b=restul împărţirii lui a la b
Radical din 2	sqrt(2)	
şi	&&	
sau	||	
negaţia	!	

LIMBAJUL C++ INSTRUCȚIUNI

- **Instrucțiunea cout:**
- Format cout<<lista de expresii;
- Efect:
- Afișează pe ecran valoarea expresiilor din listă.
- **Instrucțiunea cin:**
- Format cin<<nume variabilă;
- Efect:
- Introduce în zona de memorie specificată de numele variabilei valorile date de la tastatură.
- **Instrucțiunea de atribuire:**
- Format variabilă=expresie;
- Efect:
- Calculează valoarea expresiei și o atribuie variabilei.

INSTRUCȚIUNEA IF (DACĂ)

- **Varianta 1:**
- Format: if (expresie)

```
        {
          bloc 1 de instrucțiuni;
        }
      else
        {
          bloc 2 de instrucțiuni;
        }
```

Efect: dacă expresia are valoarea logică de adevărat (nenulă), se execută blocul 1 și nu se execută blocul 2.

- dacă expresia are valoarea logică de fals (nulă), se execută blocul 2 și nu se execută blocul 1.

INSTRUCȚIUNEA IF (DACĂ)

- **Varianta 2:**
- Format: if (expresie)

```
{
  bloc 1 de instrucțiuni;
}
```

Efect: dacă expresia are valoarea logică de adevărat (nenulă), se execută blocul 1 și se continuă programul cu următoarea instrucțiune.

- dacă expresia are valoarea logică de fals (nulă), nu se execută blocul 1 și se continuă programul cu următoarea instrucțiune.

INSTRUCȚIUNEA SWITCH (COMUTĂ)

- Format:

```
switch(expresia_e)
  {
    case v1:
          {bloc_1 de instrucțiuni;
          break;}
    case v2:
          {bloc_2 de instrucțiuni;
          break;}
    …
    default:
          {bloc_n de instrucțiuni;
          break;}
    }
```

Efect: se calculează valoarea expresiei e și apoi se execută blocul de instrucțiuni corespunzător valorii obținute, și numai acesta,: v1, sau v2 etc.

Dacă valoarea expresiei e este diferită de valorile v1, v2, … atunci se execută blocul de instrucțiuni de la DEFAULT, și numai acesta.

INSTRUCȚIUNEA FOR

- Format:

```
for( v = e1; explog;einc)
{
   bloc 1 de instrucțiuni ;
}
```

Efect:

- Se atribuie variabilei v valoarea expresiei e1;
- Se execută blocul 1 de instrucțiuni, atât timp cât expresia logică explog are valoarea logică de ADEVĂRAT, modificând valoarea variabilei v, conform expresiei de incrementare einc.
- În momentul în care valoarea expresiei logice explog este de FALS, execuția programului continuă cu prima instrucțiune de după for.

INSTRUCȚIUNEA WHILE

- Format:

```
while(expresia_e)
{
   bloc 1 de instrucțiuni ;
}
```

Efect:

- Se execută blocul 1 de instrucțiuni atât timp cât expresia e este nenulă sau are valoarea logică de adevărat.
- Dacă expresia e este nulă sau are valoarea logică de fals, atunci blocul 1 de instrucțiuni nu se va executa, iar execuția programului va continua cu prima instrucțiune de după while.

INSTRUCȚIUNEA DO

- Format:

```
do
{
   bloc 1 de instrucțiuni ;
}
while(expresia_e)
```

Efect:

- Se execută blocul 1 de instrucțiuni atât timp cât expresia e este nenulă sau are valoarea logică de adevărat.
- Dacă expresia e este nulă sau are valoarea logică de fals, atunci blocul 1 de instrucțiuni nu se va executa, iar execuția programului va continua cu prima instrucțiune de după while.
- Spre deosebire de WHILE, blocul 1 de instrucțiuni se va executa în mod sigur cel puțin o dată.