

(Vaghi) Elementi  
di programmazione dei calcolatori  
elettronici e altro....



## Tipi di programmazione

- Procedural
  - C, Pascal, Basic, Fortran
- Functional
  - Lisp, ML
- Object-oriented
  - C++, Java, Smalltalk
- Rule-based (or Logic)
  - Prolog



Attenzione: i  
pacchetti software  
(FEA, Multibody,  
AutoCAD, tec.  
Stanno diventando  
programmabili molto  
gradualmente.



## Cos'è un programma di calcolo

Un insieme di

- Istruzioni da eseguire
- Espressioni da calcolare
- Regole da applicare
- Oggetti da manipolare
- Messaggi da inviare e ricevere
- ... e molto altro ancora



## Varietà dei linguaggi

- La scelta del linguaggio dipende da come il pensiero si sa meglio adattare ad un problema
- Inoltre si deve tener conto dei fattori di:
  - Efficienza
  - Compatibilità con codici esistenti
  - Disponibilità



## Varietà di applicazioni ... e pacchetti

- document languages, e.g. LaTeX, Postscript
- command languages, e.g. bash, MatLab
- markup languages, e.g. HTML, XML ed anche di nuovo il LaTeX (?)
- specification languages, e.g. UML



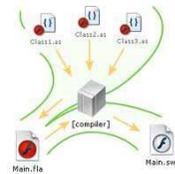
## Confronto



- Compilatore
  - I codici girano più velocemente a basso livello
  - La compilazione richiede spazio e tempo
  - C, Pascal, FORTRAN
- Interprete
  - La diagnostica è facilitata durante l'esecuzione
  - Basic, UNIX shells, Lisp, Ambienti in generale

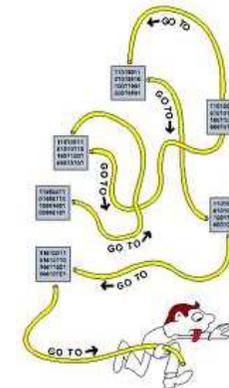
## Implementazione

- Compilatore
  - Passa dall'alto livello al basso livello (linguaggio macchina) con un codice ottimizzato
- Interprete
  - Traduce le istruzioni direttamente una alla volta



## Programmazione strutturata

- Istruzione
- Sequenza
- Selezione
- Iterazione
- ... ..



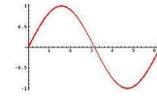
## Istruzione

- Assegnazione (  $a = a + 1$  )
- Stampa
- Lettura
- Calcolo espressione matematica
- Plot di un grafico
- Richiamo ad un sottoprogramma
- .....



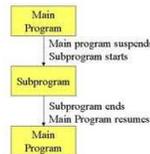
## Sottoprogramma - Esempi

- Procedura – OrdList(lista, listaordinata)
  - Riceve un array di numeri (lista) e riempie un nuovo array con i valori di lista ordinati dal minore al maggiore; i dati sono disponibili mediante il nuovo array listaordinata
- Funzione – CalcSeno(x)
  - Prende il valore x e ne calcola il seno; la funzione calcolaseno(x) fornisce al programma principale il valore  $\sin(x)$  pronto per l'uso.



## Sottoprogramma

- Procedura
  - È un programma vero e proprio che riceve dati dalle variabili input e restituisce dati mediante le variabili di output; può effettuare anche altre operazioni.
- Funzione
  - È una variabile particolare che assume un valore calcolato in base ai valori trasmessi dalle variabili input



## Sequenza

- Lista di istruzioni

```

...
zero = a * 0;
vB = zero;
termine = length(zero)-1;
for j = 1:1:termine
    VB(j+1) = vB(j) + dt * ( (a(j+1) + a(j))/2 );
end;
...

```

## Selezione (*if – then – else*)

```
if < condizione >
  – then < sequenza >
  – else < sequenza >
end;
```

la < condizione >  
si riduce ad una variabile booleana che può  
assumere il valore di vero o falso

## Iterazione - for

```
for < variabile = lista ordinata di valori > do
  – < sequenza >
end;
```

Esempio:

```
Somma = 0;
for i = 1 to 100 do
  somma = somma + i
end;
```

## Selezione (*case*)

```
case < variabile >
  • Valore1 : < sequenza >
  • Valore2 : < sequenza >
  • Valore3 : < sequenza >
  • Otherwise: < sequenza >
end;
```

## Iterazione - while

```
while < condizione > do
  – < sequenza >
end;
```

Esempio:

```
somma = 0;
i = 1;
while ( i < 101) do
  somma = somma + i
  i = i + 1
end;
```

## Iterazione - repeat

do repeat

– < sequenza >

until < condizione >

Esempio:

```
somma = 0;
```

```
i = 1;
```

```
repeat
```

```
    somma = somma + i
```

```
    i = i + 1
```

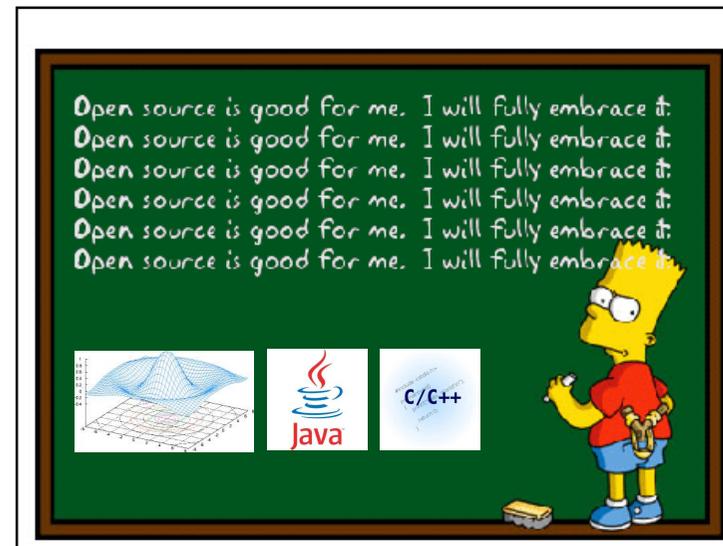
```
until (i = 101)
```

## Sintassi

- La sintassi delle varie istruzioni è estremamente variabile e dipende dal compilatore o dall'interprete adottato.

## Ricorsività

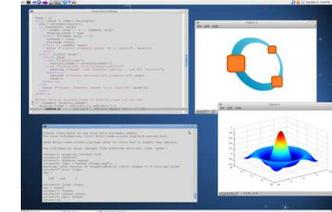
- Generalmente le procedure e le funzioni possono richiamare altre procedure e funzioni.
- In alcuni linguaggi ad una qualsiasi procedura (o funzione) è consentito richiamare sé stessa!





- KINSYNTH è uno spazio web creato per la cooperazione accademica dedicato allo sviluppo di codici di calcolo di tipo Open Source
- Gli Allievi hanno diritti di downloading ed uploading.
- Gli Allievi possono condividere I programmi ed ottenerne I corrispondenti Authoring Credits.

## Octave



- GNU General Public License
- Octave è un linguaggio di alto livello per il calcolo numerico.
- Offre notevoli routine di grafica
- Octave è usato in linea, come ambiente, oppure in modalità batch o script, come programma pronto per l'interpretazione.
- Altissima portabilità con MatLab

[http://dma.ing.uniroma1.it/users/m\\_app11\\_c1/index.html](http://dma.ing.uniroma1.it/users/m_app11_c1/index.html)

[http://dma.ing.uniroma1.it/users/m\\_app11\\_c1/index.html](http://dma.ing.uniroma1.it/users/m_app11_c1/index.html)

[http://dma.ing.uniroma1.it/users/lsm\\_progfunz/](http://dma.ing.uniroma1.it/users/lsm_progfunz/)

<http://www.dima.uniroma1.it/kinsynth/>

```
% primo esempio (C:\belf\esempio_00.m)
% sommo i primi 10 numeri naturali e verifico
% se tale somma è maggiore di % 100
% Se uso MatLab disp(' Inizio del programma ')
% se uso Octave
disp(" Inizio del programma ")
somma = 0
i = 0
while (i<10)
    i = i+1;
    somma = somma + i;
end;
% risultato
i
somma
% formula esatta
risultato = (10*(10+1))/2
if somma < 100
    disp("La somma cercata vale meno di 100 e precisamente : ")
    disp(somma)
% se uso MatLab disp('La somma cercata è inferiore a 100 e vale : ')
% se uso MatLab disp(somma)
end;
% se uso MatLab disp ('Ho finito .. ');
```

## Esercitazione N. 1

Dinamica di una massa localizzata

$$F = m a$$

Problema dinamico inverso,

Nota la forza  $F = A t^3 + B t^2 + C t$

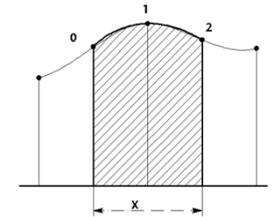
Determinare la legge del moto  $s(t)$

e le funzioni  $v(t)$ ,  $Q(t)$ ,  $E(t)$

## Integrazione numerica

- Formula di Simpson Cavalieri

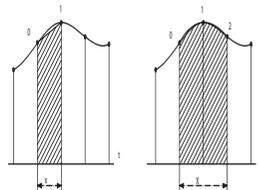
$$Y_{i+2} = Y_i + 2\Delta x \left( \frac{y_i + 4y_{i+1} + y_{i+2}}{6} \right)$$



## Integrazione numerica

- Formula di Bezout

$$Y_{i+1} = Y_i + \Delta x \left( \frac{y_i + y_{i+1}}{2} \right)$$



## Inizializzazione

```
% Se uso Octave
% #! ese01 -qf
%
% INIZIO DEL PROGRAMMA
%
% tutte le variabili eventualmente presenti in memoria
% sono messe a disposizione con le istruzioni clear e close
%
clear all
close all
```

### Soluzioni analitiche

```

dt = 0.2;
% Intervallo di definizione della funzione (definita in modalità discreta)
t = 0:dt:8;
%
% Definizione della funzione da integrare
a = - t.^3 + 10 * t.^2 - 20 * t ;
%
% ipotizzo v = 0 per t = 0
%
v = - (1/4) * t.^4 + (10/3)*t.^3 - 10*t.^2;
s = - (1/20)* t.^5 + (5/6)*t.^4 - (10/3)*t.^3;

```

### Excel

- E' disponibile nella pagina del corso anche un foglio Excel con la risoluzione della prima esercitazione.

### Formula di Bezout

```

...
zero = a * 0;
vB = zero;
termine = length(zero)-1;
for j = 1:1:termine
    VB(j+1) = vB(j) + dt * ( (a(j+1) + a(j))/2 );
end;
...

```