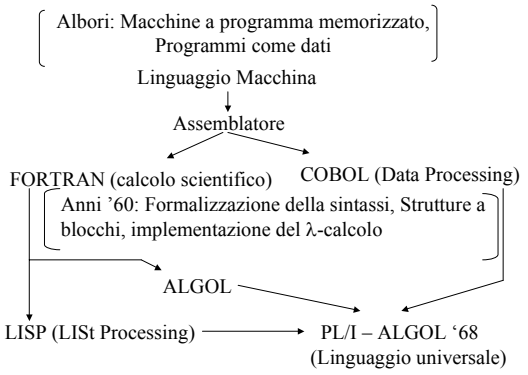
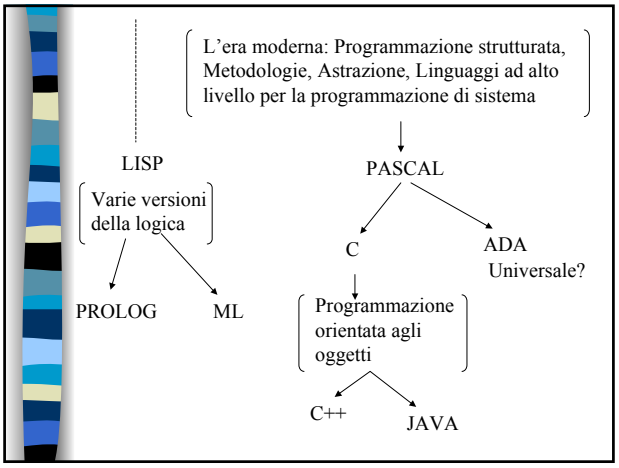
# Linguaggi di programmazione

Paradigmi di programmazione

## Linguaggi: un po' di storia




## Paradigmi

- Per paradigmi di programmazione si intendono i "modi" in cui vengono specificati i programmi
- Non si tratta tanto del tipo di linguaggio usato, ma del contesto più ampio al quale un certo linguaggio appartiene
- Parliamo di come viene organizzata la programmazione, con quali caratteristiche: stile, livello di dettaglio, "forma mentis" del programmatore

## Programmazione imperativa

- Classico: i programmi sono sequenze di comandi che agiscono sui dati o sull'ordine di esecuzione delle istruzioni.
- In genere viene studiato per primo ed è per questo che viene percepito come "più naturale"
- Esempi di linguaggi imperativi: Assembly, FORTRAN, C, COBOL, Pascal
- Costrutti tipici: assegnamento, cicli, if-then-else, procedure con passaggio di parametri

## Programmazione imperativa

- Il programmatore deve definire tutte le strutture dati e tutti gli algoritmi che operano su di esse
- Il programma non può gestire strutture dati parziali
- Meccanismi dinamici e chiamate di sottoprogrammi gestiti (anche se non sempre completamente) dal programmatore
- ...

## Programmazione orientata agli oggetti

- Di recente formalizzazione. Ha avuto molto successo.
- I programmi definiscono delle astrazioni (classi) di elementi del dominio di applicazione del programma
- Le classi contengono informazioni sui dati ma anche il codice per gestirli (in genere di tipo imperativo)
- Esempi: C++, Java, Smalltalk, Eiffel

## Programmazione Funzionale

- Il programma è una definizione di funzioni nel senso più matematico del termine
- Ordine superiore: gli argomenti delle funzioni definite possono essere sia valori di tipi primitivi che altre funzioni
- L'interpete si occupa di valutare, in base alle funzioni di base e a quelle definite, una qualsiasi espressione ben formata
- L'algoritmo di valutazione non è scritto dal programmatore, ma varia a seconda del linguaggio

## Programmazione Funzionale: esempio caml

```
#let rec fatt = function
#    0 -> 1
#  | n -> n * fatt n-1;;
fatt: int -> int = <fun>
#let rec map = function
#    [], f -> []
#  | (x::Xs), f -> (f x) :: map Xs;;
map: 'a list * ('a -> 'b) -> 'b list = <fun>
#map [2;0;3] fatt;;
- int list = [2; 1; 6]
#map [0;3] (function x -> x+1);;
- int list = [1;4]
```

## Programmazione Funzionale

- Primo linguaggio funzionale: $\lambda$-calcolo (nucleo della programmazione funzionale)
- Molto importante soprattutto dal punto di vista teorico (versione non tipata Turing-equivalente)
- Altri linguaggi: Miranda, Haskell, Scheme, Standard ML (varie implementazioni tra cui CAML), LISP
- Differenze nell'efficienza, nella vastità di librerie, nel tipo di valutazione (lazy vs eager)

## Programmazione Funzionale

- Approccio dichiarativo: il programma è una definizione di funzione, il "calcolo" è built-in
- I linguaggi funzionali possono essere usati come metalinguaggi per definire altri linguaggi ottenendo, da una definizione denotazionale della semantica, anche un interprete (gratis!)
- Le moderne implementazioni sono piuttosto efficienti

## Programmazione logica

- Un programma logico è la definizione di un certo numero di predicati della logica del primo ordine
- Limitazione: la definizione può avvenire solo con clausole Horn definite:

$$p_1(X_1^1,\ldots,X_n^1)\wedge\cdots\wedge p_k(X_1^k,\ldots,X_m^k)\rightarrow q(Y_1,\ldots,Y_h)$$

dove $n,m,k,h\geq 0$, $p_i$ sono simboli di predicato e gli $X_j^i, Y_i$ sono termini di un linguaggio del primo ordine: c, f(X), f(g(a)), g(c)...

## Programmazione logica: esempio

```
#append([],Ys,Ys).
#append([X|Xs],Ys,[X|Zs]) :-
      append(Xs,Ys,Zs).
```

:- sta per ←

```
?- append([1,2],[2,4],X).
  X = [1,2,2,4]
?- append(X,[3],[2,5,3])
  X = [2,5]
```

Ma c'è di più!

## Programmazione logica: esempio

```
?- append(X,Y,[3,4,5]).
  X = [], Y=[3,4,5] ;
  X = [3], Y=[4,5] ...
?- append(X,Y,[3,4|Z]).
  X = [3], Y = [4|Z] ;
  X = [3,4], Y = Z
```

Si definiscono relazioni che possono essere usate anche come funzioni, ma non solo!

## Programmazione logica

- Anche qui il formalismo è dichiarativo: i programmi sono definizioni, in questo caso di relazioni e non di funzioni
- Il motore di calcolo è un dimostratore di teoremi che cerca di dimostrare un goal (una clausola Horn senza la conseguenza) a partire dalle clausole del programma
- Il "risultato" viene fuori dagli assegnamenti alle variabili libere del goal che il dimostratore fa durante la visita (con backtracking) dell'albero di dimostrazione per il goal e il programma logico dati
- Linguaggio principale: PROLOG (diverse implementazioni)

## Paradigmi dichiarativi

- I linguaggi dichiarativi sono molto utili per realizzare velocemente un prototipo di un'applicazione complessa
- Con poche righe di codice si possono realizzare operazioni molto complesse
- Il problema principale resta sempre l'efficienza

## Programmazione con vincoli

- Nei linguaggi di programmazione con vincoli il motore principale di calcolo è un risolutore di vincoli in un certo dominio
- Esempio: risolutore di (dis)equazioni lineari e non nel dominio dei numeri reali
- Si associa bene con la programmazione logica
- Il programma aggiunge via via vincoli ad un insieme individuando uno spazio di soluzioni sempre più piccolo oppure vuoto

## Programmazione concorrente

- Un programma concorrente consiste di diversi processi che cooperano in un ambiente per raggiungere un risultato comune
- I processi possono avere meccanismi diversi di comunicazione (canali, memoria condivisa, multi-insieme di vincoli)
- L'attenzione della programmazione concorrente non è sul calcolo che effettuano i singoli processi, ma sulle loro interazioni

# Programmazione concorrente

- Numerosi formalismi possibili:
  - Algebre di processi: CCS, CSP
  - Set di costrutti concorrenti che si aggiungono a linguaggi classici: Linda, Pascal o C concorrenti
  - Programmazione concorrente con vincoli
- La formalizzazione è importante per effettuare verifiche formali di proprietà (es. mutua esclusione, raggiungibilità di stati, invarianze) sui programmi concorrenti