Linguaggi di programmazione

Linguaggi di programmazione

Descrizione di un **problema** → Individuazione di un **ALGORITMO** → **Programma**

Un linguaggio di programmazione è una notazione formale per descrivere algoritmi

Linguaggio macchina

Il linguaggio macchina è il linguaggio direttamente eseguibile dall'elaboratore, senza alcuna intermediazione

•Ogni istruzione è una sequenza di bit

- unità elementare dell'informazione trattata da un elaboratore
- unità di definizione di uno stato logico
  - rappresentazione logica del bit: {0, 1}.

•Computer con architetture interne (CPU) differenti hanno linguaggi macchina differenti

•Pertanto, un programma scritto nel linguaggio macchina di un computer non è eseguibile su di un computer con un'architettura differente

(**Non è portabile**!)

Linguaggi di alto livello

- Si basano su una "macchina" le cui "mosse" non sono quelle della macchina hardware

→ realizzano una "***macchina virtuale***"

- x=x+y; ⇨ 00000010 000000011011100
  00000110 000000011111100
  00000100 000000011011100

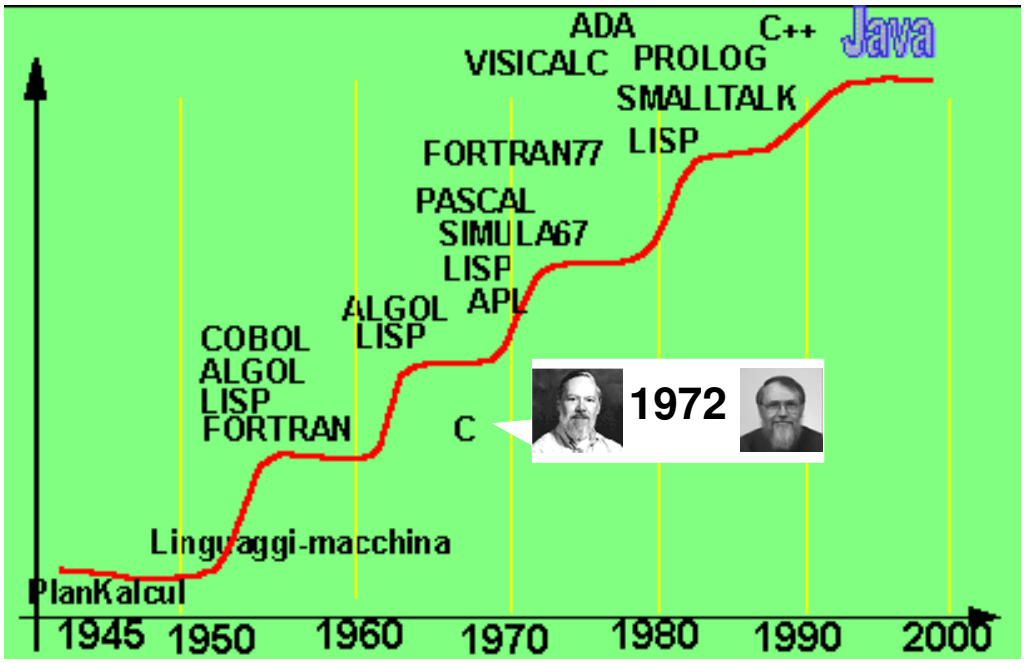
- Supportano concetti ed astrazioni
- Promuovono metodologie per agevolare lo sviluppo del software da parte del programmatore
- Hanno capacità espressive molto superiori rispetto a quelle del linguaggio macchina

- **Esistono centinaia di linguaggi di programmazione**!
  (anche se pochi sono in uso)

Linguaggi di alto livello (*cont.*)

Linguaggi di alto livello (*cont.*)

Evoluzione dei linguaggi



Paradigma imperativo

I linguaggi imperativi adottano uno ***stile prescrittivo***.

Un programma scritto in un linguaggio imperativo prescrive

✓le operazioni che il processore deve compiere (ad es. assegnamento).

✓l'ordine di esecuzione

Ambiente di programmazione

IDE (Integrated Development Environment)

È l'insieme degli strumenti (*tool*) che consentono la codifica, la verifica e l'esecuzione di nuovi programmi (*fasi di sviluppo*)

Traduttore

- Affinché un programma scritto in un qualsiasi linguaggio di programmazione ad alto livello sia comprensibile (e quindi eseguibile) da un calcolatore, ***occorre tradurlo*** **dal linguaggio di programmazione originario al linguaggio comprensibile al calcolatore** (**linguaggio macchina**)
- Questa operazione viene normalmente svolta da speciali programmi, detti **traduttori**

Processo di traduzione

Programma		**Traduzione**
main()		
{ int A;		**00100101**
...	**...**	
A=A+1;		**11001..**
if ...		**1011100 ...**

I traduttori convertono il testo dei programmi scritti in un particolare linguaggio di programmazione (***programmi sorgenti***) nella corrispondente rappresentazione in linguaggio macchina (***programmi eseguibili***)

Tipi di Traduttori

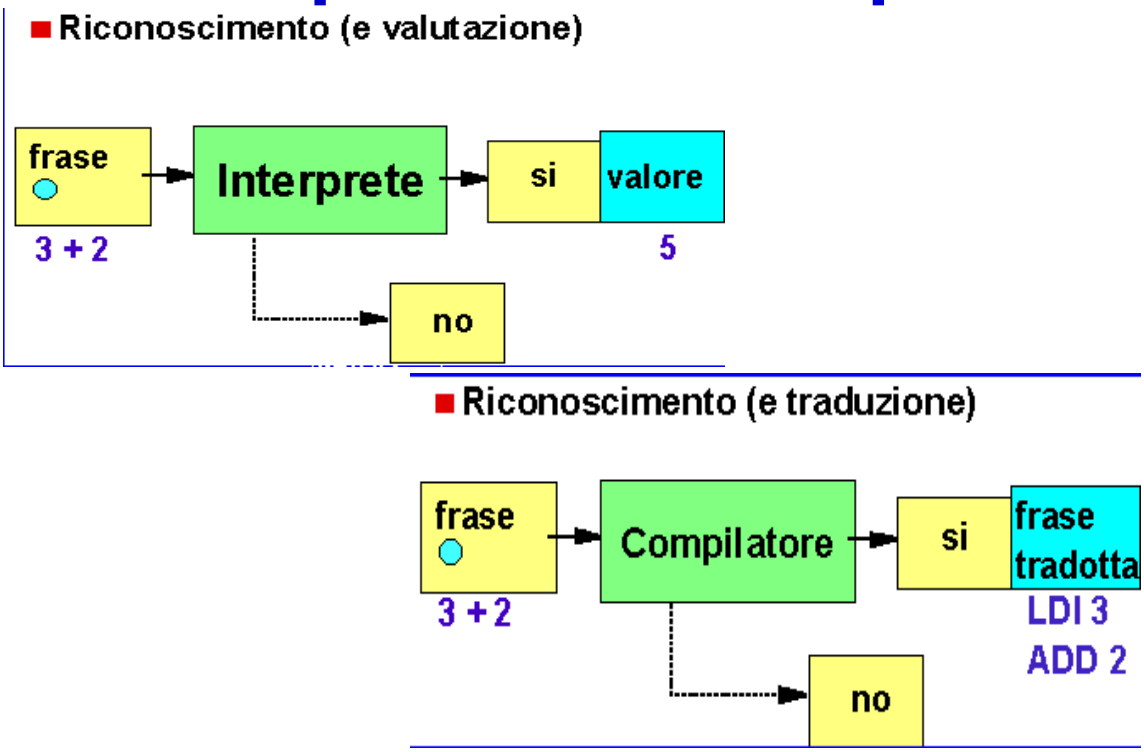
Due categorie di traduttori:

- ***Compilatori*** traducono l'intero programma (senza eseguirlo!) e producono in uscita il programma convertito in linguaggio macchina
- ***Interpreti*** traducono ed eseguono immediatamente ogni singola istruzione del *programma sorgente*

Quindi,

- **Nel caso del compilatore**, lo schema traduzione-esecuzione viene percorso una volta sola prima dell'esecuzione
- **Nel caso dell'interprete**, lo schema traduzione-esecuzione viene attraversato tante volte quante sono le istruzioni che compongono il programma
  Ad ogni attivazione dell'interprete su di una particolare istruzione segue l'esecuzione dell'istruzione stessa.

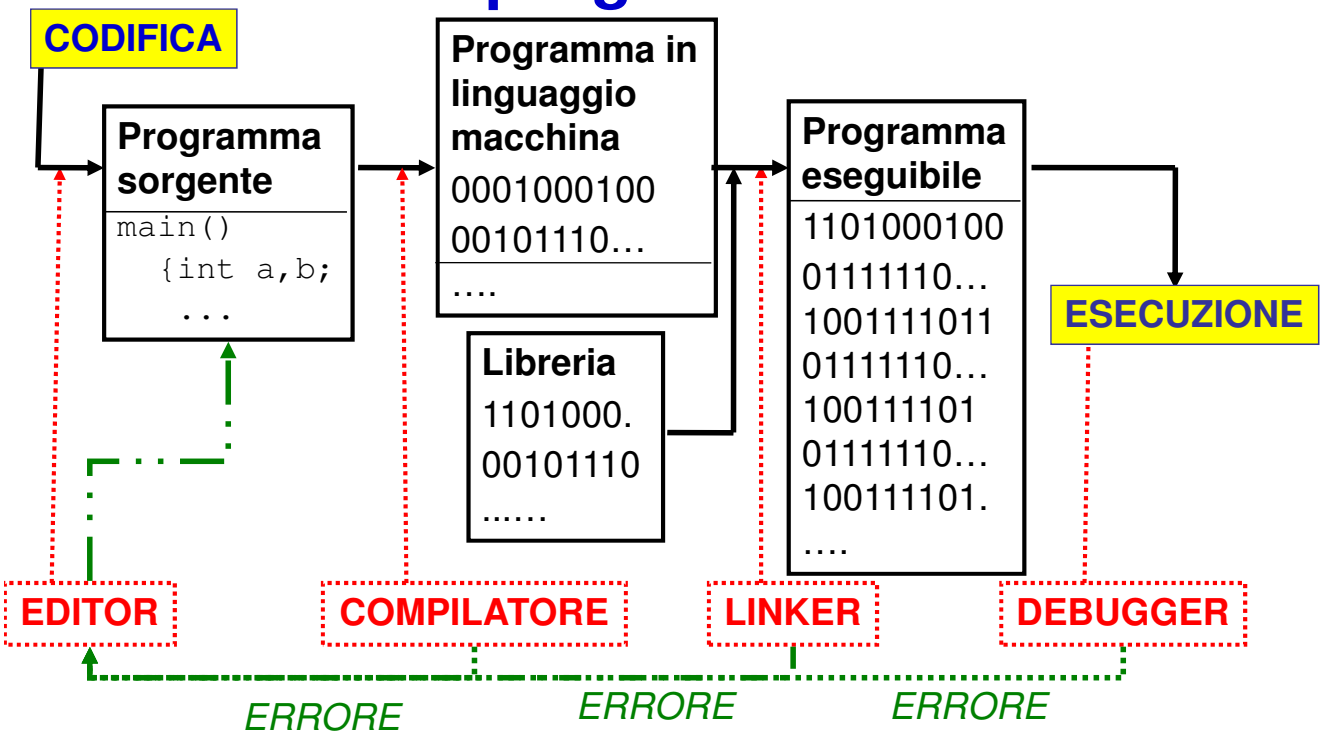
Compilatore e Interprete



Compilatore e Interprete (*cont.*)

- Sebbene in linea di principio un qualsiasi linguaggio possa essere tradotto sia mediante **compilatori** sia mediante **interpreti**, nella pratica si tende verso una differenziazione già a livello di linguaggio:
  - Tipici linguaggi interpretati: **Basic**, **Javascript, Perl**, ...
  - Tipici linguaggi compilati: **C**, **Fortran**, **Pascal**, **ADA**, … (*NOTA:* **Java** *costituisce un caso particolare*)
- L'esecuzione di un programma *compilato* è più veloce dell'esecuzione di un programma *interpretato*
- Un programma interpretato è più **portabile** di un programma compilato
  (***portabile*** = può essere eseguito su piattaforme diverse)

Ambiente di programmazione per linguaggi compilati (IDE)

- ***Editor*:** serve per creare file che contengono testi (cioè sequenze di caratteri). In particolare, l'editor consente di scrivere il *programma sorgente*.
- ***Compilatore*:** opera la traduzione di un programma sorgente scritto in un linguaggio ad alto livello in un *programma oggetto* direttamente eseguibile dal calcolatore.
- ***Linker*** ("*collegatore*"): nel caso in cui la costruzione del programma oggetto richieda l'unione di più moduli (compilati separatamente), provvede a collegarli per formare un unico *programma eseguibile.*
- ***Debugger*** ("*spulciatore*"): consente di eseguire passo-passo un programma, controllando via via quel che succede, al fine di scoprire ed eliminare errori non rilevati in fase di compilazione.

Sviluppo ed esecuzione di un programma