«I linguaggi di programmazione»
Antonio Tufano

Introduzione alla programmazione

Definizione 1 – Un linguaggio è un insieme di parole e di metodi di combinazione delle parole usati e compresi da una comunità di persone

- È una definizione poco precisa perché…
  - …non evita le ambiguità dei linguaggi naturali
  - …non si presta a descrivere processi computazionali automatici
  - …non aiuta a stabilire proprietà

Introduzione alla programmazione

Definizione 2 – Il linguaggio è un sistema matematico che consente di rispondere a domande come:

- quali sono le frasi lecite?
- si può stabilire se una frase appartiene al linguaggio?
- come si stabilisce il significato di una frase?
- quali sono gli elementi linguistici primitivi?

Introduzione alla programmazione

Lessico, sintassi e semantica

- **Lessico**: l'insieme di regole formali per la scrittura di parole in un linguaggio
- **Sintassi**: l'insieme di regole formali per la scrittura di frasi in un linguaggio, che stabiliscono cioè la grammatica del linguaggio stesso
- **Semantica**: l'insieme dei significati da attribuire alle frasi (sintatticamente corrette) costruite nel linguaggio
- Nota: una frase può essere sintatticamente corretta e tuttavia non avere significato!

Introduzione alla programmazione

Cenni storici

- Benché siano macchine in grado di compiere operazioni complesse, i calcolatori devono essere “guidati” per mezzo di istruzioni appartenenti ad un linguaggio specifico e limitato, a loro comprensibile.

Introduzione alla programmazione

Cenni storici

- Un linguaggio di programmazione è costituito, come ogni altro tipo di linguaggio, da un *alfabeto*, con cui viene costruito un insieme di parole chiave (il vocabolario) e da un insieme di regole sintattiche per l’uso corretto delle parole del linguaggio.

Introduzione alla programmazione

Cenni storici

- A livello hardware, i calcolatori riconoscono solo comandi semplici, del tipo *copia un numero*, *addiziona due numeri, confronta due numeri.*

Introduzione alla programmazione

Cenni storici

- I comandi realizzati in hardware definiscono il set di istruzioni macchina e i programmi che li utilizzano direttamente sono i programmi in linguaggio macchina
- In linguaggio macchina…
  - ogni “operazione” richiede l’attivazione di numerose istruzioni base
  - qualunque entità, istruzioni, variabili, dati, è rappresentata da numeri binari: i programmi sono difficili da scrivere, leggere e manutenere
- Il linguaggio macchina riflette l’organizzazione della macchina più che la natura del problema da risolvere

Introduzione alla programmazione

Cenni storici

- Negli anni ‘50, tutti i programmi erano scritti in linguaggio macchina o in assembly
  - ogni istruzione è identificata da una sigla piuttosto che da un codice numerico
  - il riferimento alle variabili viene effettuato per mezzo di nomi piuttosto che mediante indirizzi di memoria
- I programmi scritti in assembly necessitano di un apposito programma assemblatore per tradurre le istruzioni tipiche del linguaggio in istruzioni macchina

Introduzione alla programmazione

Cenni storici

- Oggi si utilizza l'assembly solo se esistono vincoli stringenti sui tempi di esecuzione; viceversa, si usano linguaggi più vicini al linguaggio naturale, i linguaggi di alto livello
- I linguaggi di alto livello sono elementi intermedi di una varietà di linguaggi ai cui estremi si trovano il linguaggio macchina, da un lato, ed i linguaggi naturali, come l'italiano e l'inglese, dall'altro.
- I linguaggi di programmazione differiscono comunque dai linguaggi naturali: sono infatti meno espressivi ma più precisi
- Sono semplici e poveri (poche parole chiave, poche regole), ma privi di qualsiasi ambiguità

Linguaggi di programmazione

- Un linguaggio di programmazione è costituito, come ogni altro tipo di linguaggio, da un alfabeto con cui viene costruito un insieme di parole chiave (il vocabolario) e da un insieme di regole sintattiche (la grammatica) per l'uso corretto delle parole del linguaggio
- I microprocessori presenti all'interno della macchina sono stati progettati per riconoscere ed eseguire un insieme piuttosto ristretto di istruzioni; tali istruzioni costituiscono il cosiddetto linguaggio macchina
- Il linguaggio macchina è basato su una codifica estremamente compatta e poco intuitiva

Linguaggi di programmazione

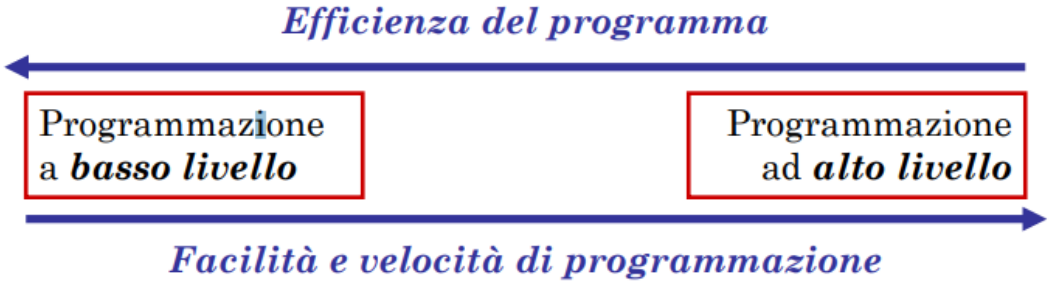
- Codificare un programma utilizzando il linguaggio macchina è assai arduo e richiede una conoscenza approfondita del funzionamento di un particolare calcolatore (o meglio: del microprocessore che costituisce la CPU della macchina)
- Per ovviare a questo problema, che ha costituito per molti anni un grosso limite alla diffusione della programmazione e quindi anche dell'uso dei calcolatori, sono stati sviluppati dei linguaggi di programmazione più evoluti, che si pongono a metà strada fra il nostro linguaggio naturale ed il linguaggio macchina
- Sono semplici e poveri (poche parole chiave, poche regole), ma privi di qualsiasi ambiguità

Linguaggi di programmazione

- In informatica si parla di programmazione a basso livello quando si utilizza un linguaggio molto vicino alla macchina, al suo funzionamento interno
- Si parla invece di programmazione ad alto livello quando si utilizzano linguaggi più sofisticati ed astratti, slegati dal funzionamento fisico della macchina
- Si viene così a creare una gerarchia di linguaggi, dai meno evoluti (il linguaggio macchina e l'assembler) a quelli più evoluti (Pascal, Fortran, Cobol, Java, Perl, Python, Go)
- Il linguaggio C si pone ad un livello intermedio

Linguaggi di programmazione

- La programmazione a basso livello è più ardua e meno intuitiva, ma consente di sviluppare programmi molto efficienti su uno specifico sistema hardware/software, sfruttandone a fondo le caratteristiche
- Ad alto livello la programmazione è più “naturale” e rapida, ma è possibile che non consenta di produrre software particolarmente efficiente



I linguaggi di alto livello

- Consentono di trattare oggetti complessi senza doversi preoccupare dei dettagli della macchina sulla quale il programma viene eseguito
- Richiedono un compilatore o un interprete in grado di tradurre le istruzioni del linguaggio di alto livello in istruzioni macchina di basso livello, eseguibili dal calcolatore
- Un compilatore è un programma traduttore simile ad un assemblatore, ma più complesso, infatti…
  - esiste una corrispondenza biunivoca fra istruzioni in assembler ed istruzioni macchina
  - ogni singola istruzione di un linguaggio di alto livello corrisponde a molte istruzioni in linguaggio macchina

I linguaggi di alto livello

- I linguaggi che non dipendono dall'architettura della macchina offrono due vantaggi fondamentali:
  - i programmatori non devono cimentarsi con i dettagli architetturali di ogni calcolatore
  - i programmi risultano più semplici da leggere e da modificare

⇒ **portabilità, leggibilità, manutenibilità**

I linguaggi di alto livello

- **Portabilità**: i programmi scritti per un calcolatore possono essere utilizzati su qualsiasi altro calcolatore, previa ricompilazione
- **Leggibilità**: la relativa similitudine con i linguaggi naturali rende i programmi più semplici, non solo da scrivere, ma anche da leggere
- **Manutenibilità**: facilità nell'effettuare modifiche di tipo correttivo, perfettivo, evolutivo e adattivo

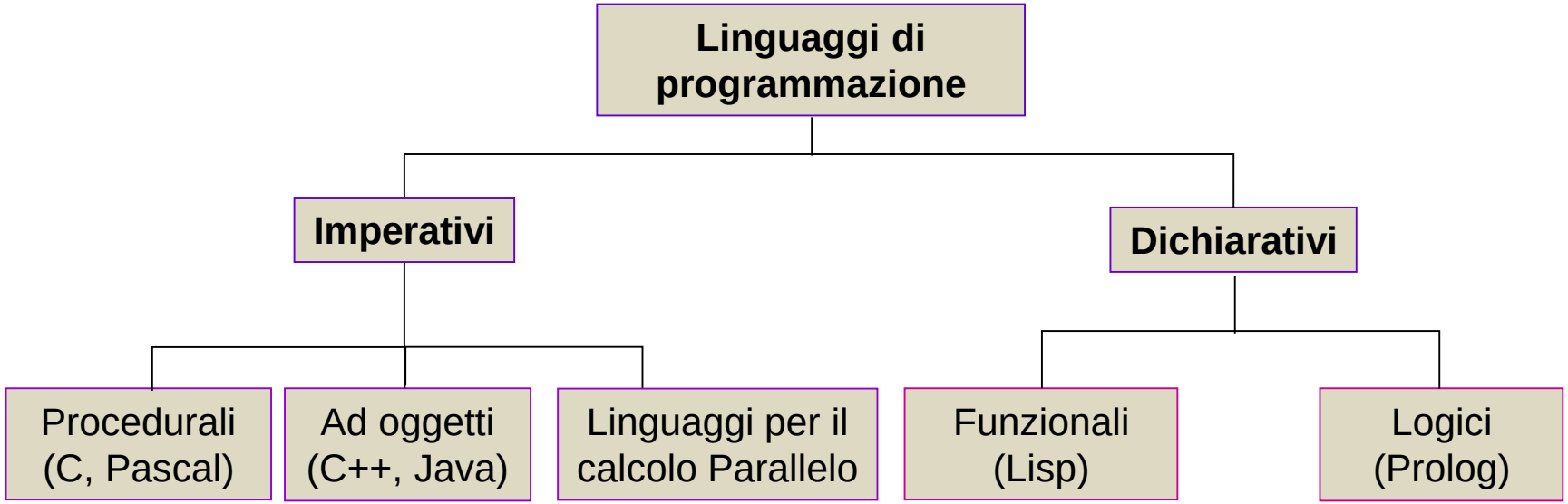
- La possibilità di codificare algoritmi in maniera astratta si traduce in una migliore comprensibilità del codice e quindi in una più facile analisi di correttezza

I linguaggi di alto livello

- Eventuale svantaggio dell'uso dei linguaggi di alto livello è la riduzione di efficienza
  - È possibile utilizzare successioni di istruzioni macchina diverse per scrivere programmi funzionalmente equivalenti: il programmatore ha un controllo limitato sulle modalità con cui il compilatore traduce il codice
  - Tuttavia… compilatori sofisticati ricorrono a trucchi di cui molti programmatori ignorano l'esistenza
- La ragione fondamentale per decretare la superiorità dei linguaggi di alto livello consiste nel fatto che la maggior parte dei costi di produzione del software è localizzata nella fase di manutenzione, per la quale leggibilità e portabilità sono cruciali

Tipi di linguaggi ad alto livello

Possiamo aggregare i numerosi linguaggi di programmazione esistenti sulla base del modello astratto di programmazione che sottintendono e che è necessario adottare per utilizzarli



Tipi di linguaggi ad alto livello

- **Linguaggi imperativi**
  - Il modello computazionale è basato sul cambiamento di stato della memoria della macchina
  - È centrale il concetto di assegnazione di un valore ad una (variabile) locazione di memoria
  - Il compito del programmatore è costruire una sequenza di assegnazioni che producano lo stato finale (in modo tale che questo rappresenti la soluzione del problema)

Tipi di linguaggi ad alto livello

- **Linguaggi dichiarativi**
  - Il modello computazionale è basato sui concetti di funzione e relazione
  - Il programmatore non ragiona in termini di assegnazioni di valori, ma di relazioni tra entità e di valori di una funzione

Tipi di linguaggi ad alto livello

- Sulla base dell'ambito in cui si colloca il problema da risolvere, è opportuno adottare un linguaggio piuttosto che un altro:
  - Calcolo scientifico: Fortran, C
  - Intelligenza Artificiale: Prolog, Lisp, C
  - Applicazioni gestionali: Cobol, SQL, C
  - Sistemi operativi: Assembler, C
  - Applicazioni visuali: C++, Java, Visual Basic
  - Applicazioni Web: Java, PHP, ASP