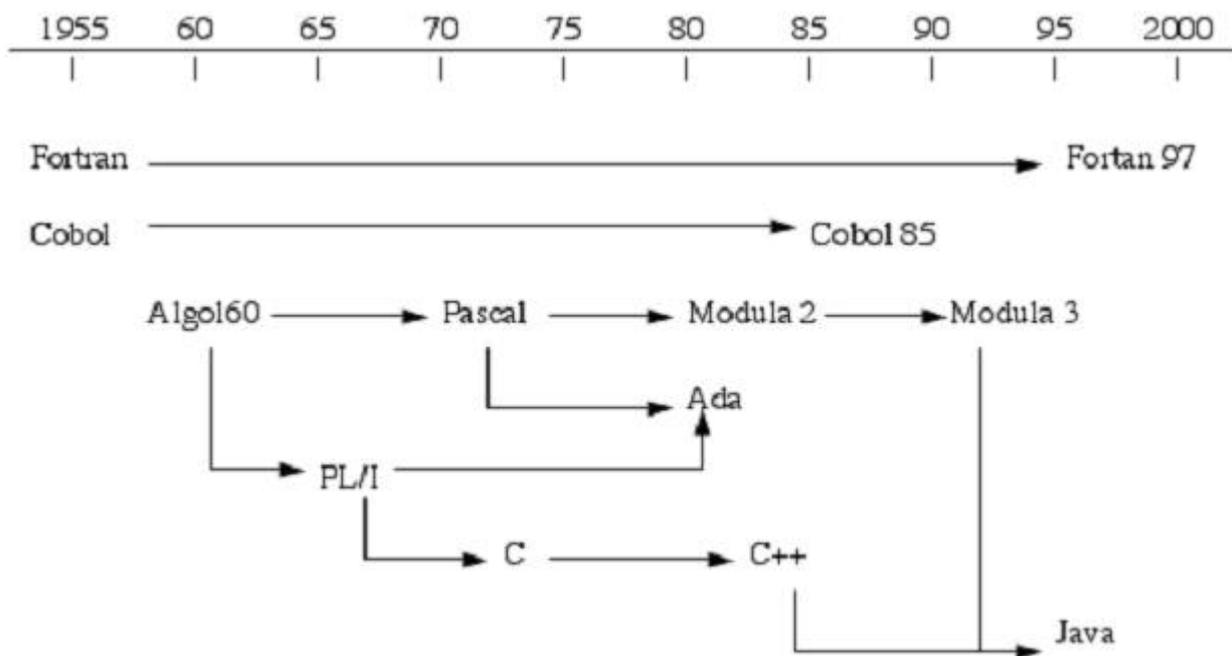


Tecnologie di programmazione Evoluzione dei linguaggi di programmazione

La seguente tabella mostra alcuni (pochissimi!) delle centinaia di linguaggi di programmazione ad alto livello che sono stati introdotti, a partire dal primo di essi, Fortran. Tutti i linguaggi indicati sono ancora in uso, in una qualche loro versione, con l'eccezione di Algol, PL/I e, probabilmente, Modula. Ma ciascuno di essi ha dato contributi essenziali per i successori. Algol è stato il primo linguaggio con costrutti di controllo veramente strutturati, Modula 3 è stato il banco di prova di alcuni concetti orientati agli oggetti che troviamo in Java.

Fortran e Cobol, due linguaggi che hanno dominato la scena fino alla fine degli anni 70, oggi rimangono in vita perché esistono grandi quantità di *software* scritto in essi che non è conveniente trasformare in altri linguaggi.



Linguaggi orientati agli oggetti

I linguaggi di programmazione come Pascal o C si basano su una chiara distinzione tra dati e procedure che agiscono su di essi.

è possibile adottare una prospettiva diversa:

1. un programma è costruito mediante una collezione di entità (**oggetti**);
2. ciascuna entità ingloba sia un dato che le procedure (**metodi**) che lo manipolano;
3. il dato non è immediatamente accessibile;
4. l'accesso al dato è permesso solo attraverso le procedure fornite dall'entità che lo ingloba;
5. le entità comunicano mediante **messaggi** coi quali richiedono al destinatario l'esecuzione di un metodo.

Un **oggetto** consiste dunque di:

1. dati nascosti;
2. operazioni pubbliche, dette metodi.

Un programma in un linguaggio orientato agli oggetti consiste in una serie di **messaggi** agli oggetti: un messaggio è la richiesta di eseguire un certo metodo.

Un esempio.

Possiamo immaginare un cerchio come un oggetto capace di rispondere a due messaggi, uno che richiede l'area del cerchio, un altro che richiede il raggio del cerchio. Abbiamo dunque un oggetto con due metodi: area e raggio. Se *c* è un cerchio, nello specifico senso appena definito e usando la sintassi del linguaggio Java, possiamo richiedere l'area del cerchio come:

```
c.area();
```

Osserviamo che non abbiamo detto come un cerchio è rappresentato all'interno dell'oggetto. Questo fa parte dei dati nascosti, che non sono manipolabili se non vi sono metodi che lo fanno. Questo incapsulamento dell'informazione è una caratteristica molto importante dei linguaggi orientati agli oggetti. Essa garantisce, per esempio, che ogni programma che usa un cerchio non dipende dallo specifico modo in cui un cerchio viene rappresentato.

Linguaggi orientati agli oggetti:

1. precursori: Simula, Smalltalk;
2. in uso: C++, Java.

Metodologia ad oggetti

L'oggetto è un costrutto di incapsulamento molto flessibile.

Possiamo applicare la prospettiva ad oggetti a realtà molto diverse:

1. numeri interi
2. strutture dati
3. *file systems*
4. basi di dati
5. finestre sullo schermo
6. ecc.

La metodologia ad oggetti permette la costruzione di sistemi facilmente estensibili, in modo particolare perché i linguaggi orientati agli oggetti forniscono strumenti sofisticati (ereditarietà) per il riuso del codice (cioè di parti di programma).

Ingegneria del software

Il costo dell'hardware diminuisce drasticamente ogni anno.

Il costo globale di un sistema di calcolo è di gran lunga dominato dal costo del suo *software*.

Costi di:

1. progettazione iniziale
2. realizzazione iniziale
3. aggiornamento
4. correzione degli errori

Di questi costi, quello della manutenzione, cioè dell'aggiornamento del sistema e della correzione dei suoi errori (veri o dovuti a specifiche mal poste), è di gran lunga quello più oneroso.

L'evoluzione dei linguaggi di programmazione può essere letta come una ricerca di ambienti linguistici nei quali:

1. sia semplice programmare;
2. sia semplice modificare programmi già scritti.

Semplice = poco costoso

Tecnologie *software*:

strumenti informatici per la produzione di grossi sistemi

Sono basate sui linguaggi di programmazione, ma comprendono molti altri aspetti. Alcuni di essi:

1. la specifica
2. la modularizzazione
3. la ricerca degli errori (*debugging*)
4. la prova (*testing*)
5. la gestione delle varie versioni
6. la produzione da parte di vari gruppi di lavoro
7. la documentazione
8. la manutenibilità

Della definizione, realizzazione e valutazione di queste tecnologie si occupa il settore dell'informatica detto **ingegneria del software**.

Conclusioni

I linguaggi di programmazione sono linguaggi artificiali che permettono di esprimere algoritmi in modo non ambiguo. I moderni linguaggi di programmazione da una parte forniscono al programmatore costrutti sofisticati per descrivere all'interno del programma la struttura del problema da risolvere. Dall'altra costringono il programmatore ad usare schemi di programmazione strutturata, che sono più facilmente verificabili della programmazione a spaghetti.

L'ingegneria del *software* ha come obiettivo la realizzazione e la valutazione di tecnologie mediante le quali la produzione di programmi sia sempre meno costosa e, soprattutto, meno soggetta al verificarsi di errori dopo la posa in opera del sistema.

La programmazione orientata agli oggetti è oggi una delle maggiori tecnologie in uso. Le caratteristiche fondamentali di un linguaggio orientato agli oggetti saranno affrontate in un approfondimento.

Sebbene non facciano parte direttamente dei linguaggi di programmazione, negli anni recenti hanno avuto grande sviluppo i linguaggi per la descrizione della struttura di documenti e, più in generale, di dati non strutturati. Tra questi il più noto è certamente HTML, un linguaggio che permette l'annotazione di documenti e dati multimediali. Moduli successivi affronteranno queste tecnologie (HTML, XML, ecc.), che stanno divenendo lo standard per la definizione e l'annotazione di dati non strutturati.