

# LINGUAGGI DI PROGRAMMAZIONE

## Storia

### Prima dell'elettronica

La nascita del calcolo e dei dispositivi di memorizzazione risale alla necessità di misurare dei terreni, di ricordare delle quantità e di sommarle. L'uomo ha quindi inventato dei dispositivi di memorizzazione e di aiuto al calcolo. Inizialmente erano tacche su dei bastoni o incisioni su pareti di caverne, poi cordicelle con nodi, abachi, e infine dispositivi meccanici, dei quali la calcolatrice di Pascal e l'addizionatrice di Leibnitz sono i dispositivi più noti, ma non isolati, infatti decine di artigiani hanno prodotto e venduto dei meccanismi funzionanti.

Il primo sistema concepito per svolgere più tipi di calcolo è l'Analytic Engine di Charles Babbage. In questo lavoro, peraltro non terminato, Babbage è aiutato nella programmazione dei calcoli, da Ada Augusta Byron contessa di Lovelace (1815-52) , figlia del poeta Byron e probabilmente la prima programmatrice nel mondo. Il "programma" di Ada Byron Lovelace, attorno al 1840, calcola i numeri di Bernoulli. L'Analytic Engine deriva dalle macchine Jacquard per tessere tappeti, in cui le trame ed i disegni, sono comandati da schede perforate.

Pochi anni dopo, nel 1847, George Boole (1815 – 1864), con "*Mathematical Analysis of Logic*", e 1854 "*The Laws of Thought*"<sup>1</sup> chiarisce il legame fra la logica (il calcolo proposizionale) ed una struttura algebrica con due valori, divenuta poi la base per le calcolatrici elettroniche.

Con l'avvento dell'elettromeccanica, diviene possibile costruire calcolatrici per trattare dati memorizzati su schede di cartoncino; Herman Hollerith costruisce una macchina di questo tipo per il Baltimore Department of Health nel 1886 e nel 1890 un'altra macchina per trattare i dati del censimento USA. Le schede sono vere e proprie memorie capaci di contenere, in genere, 80 caratteri; i "programmi" non sono su schede come nell'*Analytic Engine*, ma su appositi pannelli dotati di buchi con terminazioni conduttrici, una parte di questi è la rappresentazione della scheda. I buchi sulle schede permettono il passaggio di un segnale elettrico che può essere utilizzato per rilevare un valore o per comandare un'azione.

Nella decade iniziata nel 1930 si gettano le basi della teoria dei calcolatori: nel 1930 il Lambda Calcolo di Alonzo Church, e, nel 1936 Alan Turing (1912 – 1954) e Post, indipendentemente fra di loro, descrivono un apparato per eseguire qualsiasi tipo di algoritmo. Nello stesso anno appaiono altri lavori di Church e Kleene.

### I primi calcolatori

L'*Analytic Engine* e l'ENIAC, il primo calcolatore elettromeccanico del 1942, hanno in comune il fatto che per ogni calcolo da eseguire, occorre riprogrammare la macchina. L'idea di usare la memoria del calcolatore per contenere, oltre ai dati, le istruzioni per trattarli è attribuita a John Von Neumann<sup>2</sup> nel 1945 (*shared-program technique*), in realtà nasce nel gruppo che, alla Moore School, lavora a costruire l'ENIAC. Von Neumann fornisce altri importanti contributi concettuali, quali l'idea dell'esecuzione condizionata delle istruzioni, cioè la possibilità di interrompere la sequenza di istruzioni per eseguire istruzioni poste in un altro punto della memoria, concetto da cui segue quello di parti di programma riusabili e librerie di calcoli prestabiliti.

Negli anni 1942 – 1945 Konrad Zuse realizza, per il regime hitleriano, il primo calcolatore digitale e, durante il periodo in cui era nascosto sulle Alpi Bavaresi (1946-47), idea il linguaggio di

---

1 Il titolo completo è: *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*.

2 Un geniale matematico e un feroce anticomunista che auspicava l'utilizzazione della bomba atomica preventiva contro l'Unione Sovietica. Von Neumann morì in seguito alle radiazioni atomiche ricevute quando seguiva gli esperimenti nucleari. Egli è l'ispiratore del personaggio "Stranamore" di un famoso film antimilitarista.

programmazione PLANKALKUL, sviluppando una serie di programmi, fra i quali un programma per giocare un finale di partita di scacchi.

Lo sforzo iniziale dei programmatori si dirige alla costruzione di subroutines generali, da utilizzare in programmi più ampi, ma già nel 1948 David Wheeler scrive il primo compilatore assembler per il computer ESDAC dell'Università di Cambridge e l'anno successivo appare lo *Short Code*, più che altro una guida per tradurre i comandi manualmente in codice macchina. AUTOCODE di Alick E. Glennie dell'Università di Manchester nel 1952-54, è un rudimentale interprete scritto a tempo perso per il Mercuri Mark I.

I primi tentativi di costruire un linguaggio di programmazione, hanno origine nell'ambito scientifico: Heinz Ruthishauser (ETH Zurigo), sulla base dei lavori di Zuse, concepisce una procedura per tradurre espressioni matematiche in linguaggio macchina. Nel 1953 al MIT J. Halcombwe Laning e Nearl Zierler scrivono un programma per tradurre equazioni algebriche per il calcolatore Whirlwind.

Nel 1957 Alan Newell e Herbert Simon progettano il General Problem Solver. Anche dall'industria vengono strumenti per il calcolo scientifico: alla Eckert-Mauchly Computer Corporation, successivamente Remington Rand, costruttrice di UNIVAC, da un'idea di Mauchly sono sviluppati a partire dal 1951, A-0, A1, A-2, da cui deriva ARITH-MATIC, A-3, A-T3, ad essi lavora Grace Brewster Hopper (1906 – 1992) e nel 1957 il prodotto finito è rilasciato col nome MATH-MATIC.

Il primo linguaggio di successo tuttavia arriva dall'industria: John Backus assunto come programmatore presso l'IBM, nel 1953 propone di studiare una via per semplificare la programmazione. La proposta è accettata, e si forma un eclettico gruppo di giovani che, in un ambiente informale, realizza il FORTRAN. Dopo una campagna di presentazioni, inizia alla fine del 1957, la distribuzione del FORTRAN, e nel 1958 ne appare un aggiornamento, il FORTRAN II, che prevede subroutines e il richiamo di moduli scritti in assembler. Il successo del FORTRAN è immediato, tant'è che la "Guide to Programming FORTRAN" di Daniel McCracken apparsa nel 1961, è venduta in 300.000 copie. Forse il successo di FORTRAN stimola la nascita, nel 1958, del comitato scientifico che progetta il linguaggio ALGOL, il primo linguaggio descritto formalmente, in BNF (Backus-Naar Form 1959), ed il primo linguaggio a prevedere funzioni ricorsive. In attesa della definizione dell'ALGOL e con la necessità di dotare i propri calcolatori di un linguaggio di alto livello, il Navy Electronic Laboratori produce il NELIAC (*Navy Electronic Laboratori International fo Algol Compiler*), diffusosi su decine di elaboratori diversi ed utilizzato per una decina di anni. Il NELIAC è il primo compilatore ad autoriprodursi.

La teoria dei compilatori ha fra i suoi fondamenti i lavori di Noam Chomsky: "Logical Structures of Linguistic Theory" (1955) e "Syntactic Structures" (1957), la formalizzazione della descrizione dei linguaggi, dovuta a Backus è del 1959: "The Syntax on a Semantics of the Proposed International Algebraic Language ...".

Dopo i linguaggi "scientifici" Grace Hopper si dedica ad un linguaggio commerciale, FLOW-MATIC, fondato su circa trenta parole inglesi e l'utilizzo di nomi simbolici (COMPARE, IF, WRITE, ...) capaci di esprimere le esigenze di calcolo anche non scientifico e l'utilizzo di nomi per i dati, come UNIT-PRICE e INVENTORY. FLOW-MATIC è considerato uno degli ispiratori del COBOL.

## **Negli anni 60**

Se da una parte l'industria comprende che il calcolatore può fare ben altro che calcoli scientifici, gli ambienti accademici intravedono la possibilità di trattare non solo numeri ma anche entità più generali, e nel 1958 al MIT, ad opera di John McCarthy nasce il LISP (*LISt Processing*) come strumento per le ricerche nell'ambito dell'Intelligenza Artificiale.

L'utilizzo del calcolatore si estende in nuovi campi, con la nascita di molti *special-purpose languages*, come per la simulazione (1962 - 1967 Simula I), e il trattamento di testi, TRAC (1959 –

1964) e SNOBOL (1962 – 1967). Quest'ultimo è il primo linguaggio ad utilizzare tecniche di riconoscimento basate su *pattern matching*, ed il primo ad introdurre le matrici associative.

All'inizio degli anni 60 nasce APL (*A Programming Language*) un sofisticato linguaggio interprete per il calcolo scientifico, basato su operatori monocarattere la maggior parte dei quali non previsti dai dispositivi di input normali. APL è opera di Kenneth Iverson.

Nel 1964, al Dartmouth College sul calcolatore GE 225, è disponibile il BASIC sviluppato da John Kemeny e Thomas Kurtz con l'intento di disporre di un, un linguaggio semplice, adatto per i non informatici. Anche Niklaus Wirth, con lo scopo di insegnare la programmazione, ed in alternativa ad ALGOL 68, complesso e difficile da usare, crea il PASCAL. Con PASCAL vi è un corredo di strumenti per preparare e provare i programmi, inoltre, la sua cessione gratuita alle Università, ne facilita la diffusione su tipi di macchine diverse, favorendo la nascita della tecnica del p-code, cioè la compilazione che produce un codice per un "calcolatore virtuale" invece del codice in linguaggio macchina. Tale codice, tramite degli interpreti, molto più facili da costruire di un compilatore, può essere eseguito su macchine differenti.

Seymour Papert e Wallace Feurzeig iniziano lo sviluppo di LOGO (1967), "un linguaggio per imparare", con lo scopo di rendere familiare la programmazione ai bambini tramite comandi impartiti ad una tartaruga che si muove su uno schermo.

Nel 1968 Dijkstra<sup>3</sup> si pronuncia contro l'utilizzo dell'istruzione GOTO nei programmi (nel 1964 il teorema di Bohm-Jacopini<sup>4</sup> aveva già dimostrato che il GOTO non era necessario); segnando l'inizio della programmazione strutturata.

Nel 1968 compare Forth ad opera di Charles Moore. Forth è un linguaggio basato sulle memorie stack, e la notazione RPN (*Revers Polish Notation*), è relativamente complicato da programmare, ma l'interprete è facilmente implementabile ed estremamente veloce.

La necessità di mettere in collegamento i differenti siti del Ministero della difesa degli Stati Uniti (*Department of Defense* o *DoD*) conduce al progetto ARPANET (*Advanced Research Projects Agency Network*) e, alla fine del 1969, si ottiene la prima connessione fra due calcolatori, uno a Los Angeles e l'altro a Stanford.

## **Gli anni 70: il decennio d'oro**

Gli anni 1970 hanno influenzato fino ad oggi l'informatica con i lavori di Codd sul modello relazionale dei dati (1970)<sup>5</sup> e le innovazioni uscite dallo Xerox PARC, sotto la responsabilità di Alan Kay, come il linguaggio Smalltalk, uno dei primi linguaggi a oggetti (con SIMULA), le reti ETHERNET, il mouse e le interfacce grafiche. Appare anche il linguaggio PROLOG per Alain Colmeraur ed il linguaggio C ai Bell Labs (1972) ad opera di Dennis Ritchie, per il nuovo sistema operativo UNIX. Ancora nel 1970 Paul Allen e Bill Gates sviluppano il BASIC per il personal computer Atari; l'interprete occupa 4K di memoria.

Nel 1975 il dipartimento della difesa degli Stati Uniti promuove un gruppo di studio per individuare un linguaggio comune a tutti i settori della difesa, da esso scaturisce, cinque anni dopo, il linguaggio ADA.

Alla fine degli anni 70 nascono i linguaggi funzionali, Backus definisce FP (*Functional Program*), un modo di concepire la programmazione non più basato su istruzioni imperative ma sulla

---

3 Sigplan Notices Volume 17 No. 5, How do We Tell Truths that Might Hurt, espresse dei giudizi su alcuni linguaggi di programmazione. Alcuni di questi sono riportati nei paragrafi dei linguaggi relativi.

4 Böhm, C. and Jacopini, G.: "Flow diagrams, Turing machines and languages with only two formation rules", CACM 9(5), 1966.

5 E.T. F. Codd "A Relational Model of Data for large Shared Data Banks," in "the Association of Computer Machinery (ACM) journal, Communications of the ACM" June 1970.

combinazione di funzioni per evitare la dichiarazione di variabili e procedure<sup>6</sup>; contemporaneamente all'università di Edimburgo, nell'ambito delle ricerche sulla prova automatica dei teoremi, è definito ML (*Meta Language*). Ma non sono solamente gli ambienti accademici ad approfondire gli aspetti funzionali: il paradigma funzionale è alla base dei fogli elettronici, infatti VISICALC di Dan Bricklin è del 1978. Comunque questi aspetti "funzionali" erano già presenti in LISP e APL.

## **L'era del PC**

Nel 1980 dopo molti piccoli calcolatori nati dal lavoro di entusiasti per appassionati, IBM mette sul mercato il suo PC basato su microprocessore INTEL 8086 e con un sistema operativo della Microsoft. La crescita delle prestazioni dei sistemi, l'aumento di velocità dei microprocessori, delle capacità di memorizzazione dei dischi magnetici e l'utilizzo della grafica sui video, permette l'utilizzo di Sistemi operativi basati sulla grafica (GEM la prima interfaccia grafica per DOS della Digital Research, WINDOWS, X WINDOWS per gli ambienti UNIX). Nel 1984 nasce PostScript quale linguaggio di descrizione della pagina.

In Giappone nel 1981 viene lanciato il progetto per i computer della V generazione; il progetto è basato sull'utilizzo del Prolog.

E' anche l'epoca del consolidamento dei linguaggi ad oggetti: 1980 Smalltalk-80, Modula-2. Bjarne Stroustrup sviluppa una serie di linguaggi: "C With Classes" che sono la base del C++.

Nel 1984 Richard Stallman fonda la Free Software Foundation (FSF), con lo scopo di sviluppare un sistema operativo UNIX libero; il risultato è la creazione di molti programmi di utilità, fra i quali dei compilatori (C, Fortran, Pascal,...) e nel 1991 il Sistema Operativo LINUX ad opera di Linus Torvalds.

L'hardware permette di gestire facilmente grafica e suoni e favorendo la nascita di linguaggi come HYPERCARD di Apple nel 1987 che miscela multimedialità ed ipertestualità.

## **Internet**

Nel 1989 ai laboratori del CERN (*Centre Européen de Recherche Nucléaire*) di Ginevra Tim Berners Lee, per migliorare il sistema di diffusione della documentazione, basato su un semplice visualizzatore di files (GOPHER), inizia il progetto che origina il protocollo HTTP (*Hyper Text Transmission Protocol*), ed il linguaggio per le pagine HTML (*HyperText Markup Language*); nel 1990 questi prodotti diventano disponibili nella rete mondiale, con un immediato successo, grazie anche a Mosaic un programma cliente grafico del 1993.

Nel 1991 appare Java linguaggio ad oggetti che, tramite generazione di p-code, è eseguibile su qualsiasi piattaforma dotata di un interprete (Java virtual machine). Negli anni successivi si diffondono i linguaggi e gli strumenti di programmazione visuali.

Da Netscape Communications nel 1994 per opera di Brendan Eich, nasce JavaScript un linguaggio interpretato per dotare il browser Navigator di comportamenti dinamico ed interattivo, esso sarà rapidamente supportato dai maggiori browsers, diventando una componente fondamentale per le applicazioni INTERNET Interattive (AJAX).

---

<sup>6</sup> "Can programming be liberated from the von Neumann style?", John Backus, Communications of the ACM, 21, 8, pp.613-641, 1978