



L'evoluzione del linguaggio

di Fabio Marzocca

Prima parte

Charles Babbage, l'inventore della prima macchina calcolatrice digitale della storia, viene considerato oggi come il padre della moderna tecnologia dei computer; ma tutta l'opera di Babbage non sarebbe giunta fino a noi se la sua infaticabile assistente, la contessa Augusta Ada Lovelace, non avesse pazientemente organizzato e trascritto tutto l'imponente lavoro lasciato dal matematico.

Ada Lovelace, la figlia del poeta inglese Lord Byron, ha inoltre tradotto e commentato alcuni studi del matematico italiano L.F. Menabrea, riguardanti le macchine analitiche e differenziali, meritandosi il titolo, da parte del mondo scientifico, di «grande signora dei computer».

Ed è alla Contessa Ada Lovelace, la grande signora dei computer, che si sono ispirati i ricercatori moderni per denominare il più recente linguaggio ad alto livello per computer: l'Ada.

L'U.S. Department of Defence (il Ministero della Difesa Statunitense), impiegando più di 400 diversi linguaggi e dialetti, è attualmente il maggior utente di software del mondo. Forse proprio a causa di questa grande «Babele» di linguaggi, il DoD ha commissionato nel 1975 lo studio e lo sviluppo di un nuovo linguaggio ad alto livello, atto a racchiudere tutte le proprietà di flessibilità operativa dei precedenti, integrato con nuove possibilità applicative.

Inizialmente si pensò ad adottare per tutti un linguaggio già esistente, ma ben presto fu abbandonata l'idea. Un gruppo del DoD, l'High Order Language Working Group, emise una serie di rapporti contenenti i requisiti fondamentali del linguaggio da mettere allo studio; uno di questi stabiliva che il nuovo linguaggio doveva essere fondato sulle strutture principali del Pascal, del PL/1 o dell'Algol 68.

Dei quattro finalisti (Softtech, Intermetrics, SRI International e CII Honeywell Bull), la spuntò l'affiliata dell'Honeywell nel 1979, presentando l'Ada. Il progettista principale dell'architettura del linguaggio, ed il curatore della veste finale, è stato il ricercatore Jean D. Ichbiah.

Il risultato fu un linguaggio estremamente comprensibile, quindi estremamente potente. Furono rispettati tutti i requisiti richiesti dal DoD, quali ad esempio i tipi, sottoprogrammi, input-output, parallel processing, gestione degli errori ed innumerevoli altri.

In verità l'Ada si è mostrato un linguaggio fortemente ispirato al Pascal, di cui è perciò il naturale sviluppo ed evoluzione. Ci sarà da aspettarsi nei prossimi anni un grande successo di questo nuovo linguaggio, anche in considerazione del fatto che sono stati immessi sul mercato alcuni compilatori Ada per microcomputer, funzionanti su sistemi operativi standard quali il CP/M e l'MS-DOS.

Tipi di dati

Per chi non è familiare con il Pascal, il concetto di «tipo» potrebbe non risultare immediatamente chiaro, per cui val bene la pena di analizzarlo a fondo.

Nello studio per la risoluzione di un problema, dovrà in generale essere posta la massima cura nella descrizione degli oggetti reali che compongono il problema stesso. Questo si riflette nel tradurre le proprietà a cui soddisfano gli oggetti reali, in termini di costrutti logici di un dato linguaggio di programmazione.

Non basta quindi descrivere gli oggetti nella loro forma e varietà, ma è necessario garantire che le operazioni su tali oggetti non violino le loro intrinseche proprietà: questo porta al concetto di tipo.

Ad esempio, mentre i canali di un radio possono essere descritti come un insieme di numeri interi, e due canali possono essere sottratti per dar luogo ad un intervallo, non ha senso effettuare la radice quadrata di un canale, oppure la sua elevazione a potenza.

In generale, un tipo rappresenta l'insieme dei valori che una variabile può assumere, e l'insieme delle operazioni che possono essere effettuate su tale variabile.

In Ada, la sintassi di una dichiarazione di tipo prende la forma:

type identificatore **is** definizione;

dove l'identificatore specifica il nome del tipo, e la definizione la classe di valori e, implicitamente, i modi in cui il tipo può essere impiegato.

Ad esempio, per i tipi enumerativi una delle proprietà è la successione dei dati stessi e la possibilità di confronto fra di essi. Così sul tipo:

type GIORNI is (LUN, MAR, MER, GIO, VEN, SAB, DOM);

l'operatore GIORNI-FIRST rappre-

```

Tipi enumerativi:
type GIORNO is (LUN, MAR, MER, GIO, VEN, SAB, DOM);

Tipi reali:
type CORRENTE is delta 0.1 range 0.0 .. 10.0;

Tipi derivati:
type RESISTENZA is new INTEGER;

Sottotipi:
subtype GIORNOFERIALE is GIORNO range LUN .. VEN;

Tipi record:
type DATA is
  record
    GIORNO : INTEGER range 1 .. 31;
    MESE   : NOMEMESE;
    ANNO   : INTEGER range 1 .. 2000;
  end record;

Tipi array:
type VETTORE is array (1 .. 100) of FLOAT;

Tipi access (puntatori):
type X is access array (1 .. 10) of FLOAT;
    
```

Figura 1 - Elenco dei tipi definibili da Ada.

sentia il primo valore del tipo (LUN), mentre GIORNI'LAST l'ultimo. Si avrà inoltre:

```

GIORNI'SUCC(MAR) = MER
GIORNI'PRED(SAB) = VEN
    
```

Introducendo un vincolo (ad esempio di intervallo), si genera il cosiddetto sottotipo:

```

subtype FERIALE is GIORNI range
LUN..VEN
    
```

Assegnando così una variabile:

```

OGGI : FERIALE;
    
```

sarà possibile scrivere:

```

for OGGI := LUN to VEN do ...
    
```

oppure:

```

for I in FERIALE loop
    
```

Un particolare tipo dell'Ada, detto «derivato», consente di definire un tipo B avente le stesse caratteristiche del tipo A, ma concettualmente diverso da esso.

La sintassi è la seguente:

```

type B is new A;
    
```

In questo modo sul tipo B possono essere applicate tutte le operazioni definite sul tipo A, pur mantenendo logicamente distinti i due insiemi.

Una delle caratteristiche più interessanti di Ada, e che colma una grave lacuna del Pascal, è la possibilità di definire tipi array di dimensioni non specificate, quindi variabili.

Nel seguente esempio si ha:

```

type MAT is array (INTEGER range < >,
INTEGER range < >) of INTEGER;
    
```

il tipo MAT è rappresentato da una matrice di interi di dimensioni indefinite su entrambi gli indici. Sarà però possibile definire le seguenti variabili:

```

PUNTEGGIO : MAT (1 .. 10, 1 .. 2);
TABELLA : MAT (1 ..100, 1 ..100);
    
```

Perciò il simbolo < > sta ad indicare l'indeterminatezza della dimensione finale del tipo array dichiarato, il quale acquista così un maggiore carattere di generalità.

Supponiamo di avere un tipo VETTORE definito come segue:

```

type VETTORE is array (INTEGER range
< >) of FLOAT;
    
```

e di voler creare una procedura generica che sia in grado di calcolare la somma delle componenti di un vettore di qualsiasi dimensione. Si potrà scrivere:

```

procedure SOMMA (V:in VETTORE;
SUM:out FLOAT) is begin
  SUM := V(V'FIRST);
  for I in (V'FIRST+1) .. V'LAST loop
    SUM := SUM + V(I);
  end loop;
end;
    
```

Abbiamo così visto in generale anche la struttura formale di una procedura; nella dichiarazione della proce-

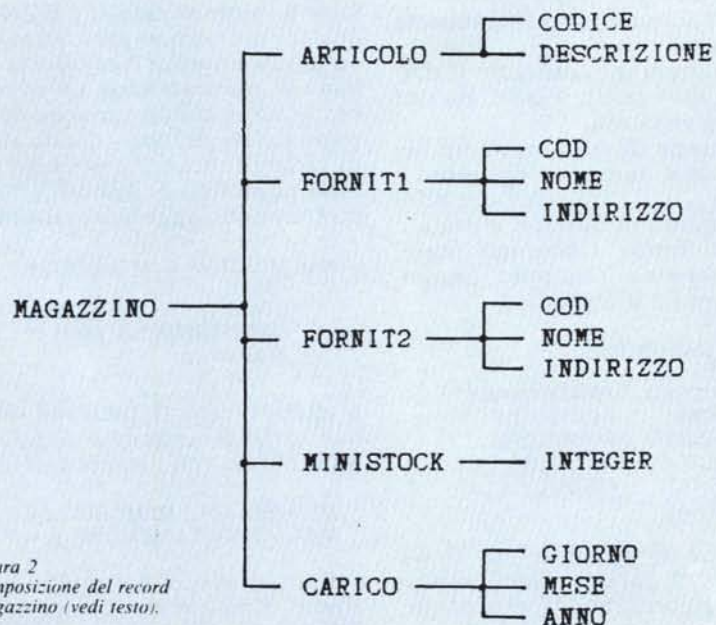


Figura 2
Composizione del record
Magazzino (vedi testo).

```
type NOMEMESE is (GEN, FEB, MAR, APR, MAG, GIU,
LUG, AGO, SET, OTT, NOV, DIC);
```

```
type PARTICOLARE is
record
CODICE : INTEGER;
DESCRIZIONE : STRING (1 .. 20);
end record ;
```

```
type FORNITORE is
record
COD : INTEGER ;
NOME : STRING (1 .. 25);
INDIRIZZO : STRING (1 .. 30);
end record;
```

```
type DATA is
record
GIORNO : INTEGER;
MESE : NOMEMESE;
ANNO : INTEGER;
end record;
```

dura vengono elencate le variabili di ingresso e uscita, rispettivamente V e SUM, con i relativi tipi di appartenenza (VETTORE e FLOAT). Lo stesso ordine andrà rispettato nella chiamata alla procedura: volendo ad esempio calcolare la somma delle componenti di un array TERMININOTI di tipo VETTORE così definito:

```
TERMININOTI: VETTORE (1 .. 10)
```

si dovrà specificare, nella chiamata alla procedura:

```
SOMMA (TERMININOTI, SUMTERMINI);
```

Questa modalità ricorda molto da vicino le chiamate a sottoprogramma del Fortran.

Passiamo ora ad analizzare il tipo record, simile a quello Pascal, ma con alcuni miglioramenti.

Supponiamo di voler creare un file di magazzino, composto da record i cui campi siano: l'articolo, i due principali fornitori, la quantità minima e la data di carico. Dovremmo avere, cioè, la seguente situazione (supponendo i tipi già dichiarati):

```
type MAGAZZINO is
record
ARTICOLO : PARTICOLARE;
FORNIT1 : FORNITORE;
FORNIT2 : FORNITORE;
MINSTOCK : INTEGER;
CARICO : DATA;
end record;
```

Per avere la possibilità di creare questo record, dovremo quindi dichiarare i tipi riportati nella figura di questa pagina.

La struttura è quindi chiaramente definita in modo intuitivo, ed il tipo record MAGAZZINO è composto a sua volta da numerosi altri tipi record, secondo lo schema rappresentato in figura 2.

Rispetto al Pascal, con Ada si ha la possibilità di inizializzare i campi di un record:

```
type DATA is
record
GIORNO : INTEGER := 1;
MESE : NOMEMESE := GEN;
ANNO : INTEGER := 1986;
end record;
```

in questo modo, la data viene ad assumere il valore di default 1 GEN 1986, qualora non venisse specificato un input diverso. Inoltre Ada offre la possibilità di parametrizzare un record secondo un elemento variabile da assegnare in fase di input; questo elemento variabile, detto *discriminante*, offrirà la possibilità di definire un record con campi di lunghezza variabili:

```
type MEMORIA (L:INTEGER) is
record
START : INTEGER;
CONTENUTO : STRING (1 .. L);
end record;
```

a questo punto si potranno definire due variabili, entrambe di tipo MEMORIA, ma con lunghezze diverse:

```
MINIMEMORY : MEMORIA (L = > 100);
MAXIMEMORY : MEMORIA (L = > 1000);
```

Rimandiamo alla figura 1 per un elenco completo dei tipi possibili in Ada.

Concessionari Memorex Computer Media

TORINO
COMPUTER MEDIA
Via Susa, 37 - Tel. 011/442261/441027

BIELLA (VC)
CO.FIN
Via Bengasi, 2 - Tel. 015/30237

CUNEO - VIOLA
B & C
Via Martini, 11/1 - Tel. 0174/73220

GENOVA
B & C
Via Col di Lana, 5/19 - Tel. 010/418719

MILANO
LOGOTEC
Via Pacini, 72 - Tel. 02/292677/235539

MILANO
GASPI
Via Pecchio, 1 - Tel. 02/225806

MONZA (MI)
COMPUTER CITY
Via San Gottardo, 84 - Tel. 039/326293

GALLARATE (VA)
EMMEQUATTRO
Via Pegoraro, 18 - Tel. 0331/795248

VIADANA (MN)
PAU
Via M. D'Azeglio, 29 - Tel. 0375/81874

CONEGLIANO VENETO (TV)
DAL CIN ELIO
Via Manin, 59/A - Tel. 0438/63144

PARMA
CHI-BO
Via Ravasini, 7 - Tel. 0521/995332

BOLOGNA
TRADER LINE
Via Morgagni, 8 - Tel. 051/271672

SAN LEONARDO (FO)
IL CENTRO EDP
Via Armellino, 19 - Tel. 0543/728091

LIVORNO
INFORMATICA
Via Scali degli Olandesi, 54 - Tel. 0586/30022

PERUGIA
R2 INFORM
Via XX Settembre, 70 - Tel. 075/61000-72266

ANCONA
PRISMA
Corso Carlo Alberto, 12 - Tel. 071/899262

PESCARA
SEFIN
Via Parini, 21 - Tel. 085/23632

ROMA
MEMORY LINE
Via Nomentana, 224 - Tel. 06/8320040-8320434

SALERNO
SYNCRON DATA
Via Paolo de' Granita, 14 - Tel. 089/241410

BARI
NICOLA ROBERTO CAVALLO
Via Durazzo, 17 - Tel. 080/330499

VIBO VALENTIA (CZ)
B. & B.
Via Pio XII, 14 - Tel. 096/343609

SASSARI
O.R.E.
Zona Industriale Predda Niedda
Tel. 079/260477

SARDEGNA
R & R ELECTRONICS
Via Fratelli Canepa, 94 - Serra Ricco (GE)
Tel. 010/750729-750866

PALERMO
BYTE'S HOUSE
Via Vann'Antò, 28 - Tel. 091/291154

**è importante scegli
MEMOREX**
A Burroughs Company

2/86

Teo Rusconi ha appena sfatato la leggenda secondo la quale i floppy disc sono tutti uguali

Difatti sembrano tutti uguali finchè non si osserva con attenzione il jacket. Qui termina l'uguaglianza.

La maggior parte delle società costruttrici sigillano i dischi un punto qui, un punto là, lasciando parte dei lembi non sigillati.

Prima o poi ai lembi accadono cose naturalissime: si gonfiano, si curvano, si raggrinziscono... in poche parole si aprono.



GLI ALTRI DISCHETTI

chiusi un punto qui, un punto là lasciano gran parte dei lembi aperti.



DISCHETTI MEMOREX

con lembi completamente saldati su tutta la superficie.

Con penne, matite, unghie persino un ragazzino di quattro anni come Teo può infilarsi in quegli spazi aperti.

Naturalmente è un danno enorme perchè se si inserisce qualcosa di molle e slabbrato nel disc-drive quest'ultimo può incepparsi; si può rovinare la testina e si possono perdere i dati. Questo può accadere con gli abituali sistemi di chiusura ma non con i dischetti Memorex che usa un procedimento esclusivo chiamato "Solid-Seam Bonding".

Con questo sistema ogni singolo millimetro quadrato dei lembi di tutti i dischi Memorex viene sigillato ermeticamente, rendendoli più rigidi e più resistenti.



È un sistema che consente al floppy disc di sostenere ogni assalto, che impedisce alla testina di rovinarsi e ai dati di andare perduti.

Il che sta a dimostrare che un floppy disc Memorex non è uguale a tutti gli altri: è migliore. E il sistema di saldatura è solo un esempio della cura infinita con cui viene prodotto ogni floppy disc Memorex; sia esso da 8", da 5 1/4" o il nuovo 3 1/2". Questa estrema accuratezza dà la garanzia che ogni disco Memorex è al 100% perfetto.

La prossima volta che acquistate un floppy disc - o qualche centinaio - ricordate: non tutti i dischetti sono uguali...

Memorex vi mette al riparo da qualsiasi inconveniente.



è importante scegli

MEMOREX

A Burroughs Company