

# La programmazione Object Oriented

quinta parte

*Ci siamo lasciati, la volta scorsa, con un impegnativo appuntamento, quello di parlare di MacApp, l'ambiente di sviluppo originale Apple, dedicato al Macintosh, che offre una piattaforma programmatoria estremamente facile da utilizzare nel creare un ambiente O.O. Uno dei maggiori benefici della programmazione di questo tipo è, infatti, la possibilità di creare una struttura, una ossatura di base applicativa; in altri termini, una applicazione già pronta, abbastanza universale, che può essere estesa facilmente e adattata ad un'ampia varietà di applicazioni reali e specifiche. MacApp è un tool specificamente costruito per fare questo*

## **Cosa è una struttura applicativa**

Come abbiamo già detto in precedenza MacApp può creare una struttura applicativa di base condividibile da diversi programmi e da diversi programmatori. Cosa vuol dire ciò? Intendiamoci con un esempio.

Prima di MacApp, tutto il software presente sul mercato era, per così dire, sviluppato individualmente pezzo per pezzo; in altre parole, ogni programmatore costruiva il suo codice originale per maneggiare, ad esempio, gli Event relativi al maneggio dei menu, delle icone, delle finestre di messaggio, dei bottoni e di tutto quello che fa di Mac una macchina unica, erano progettati singolarmente, volta per volta, a seconda delle singole esigenze e necessità del programma.

In altri termini, il programmatore era destinato, a priori, a dover sprecare tempo e fatica per poter aderire in maniera più o meno conforme a quello che, in gergo, viene chiamato standard di interfaccia utente Macintosh; la cosa era anche più frustrante se il programmatore appena pensava che probabilmente tutto questo lavoro preliminare di preparazione dei menu, delle finestre, dei bottoni, perfino degli shortcut, era magari già stato eseguito da un altro programmatore. Inoltre, cosa che faceva saltare i nervi, chi programmava pensava nello stesso momento che tutto questo lavoraccio (chi ha tentato di implementare, ab ovo, una interfaccia Mac anche minima sa quanto sudore e tempo si è costretti a impiegare per costruirsi una interfaccia appena decente) non aveva niente a che vedere con l'algoritmo di soluzione specifico del suo programma, che era ancora di là da venire.

Con questo principio, il passo a pensare di poter adottare una libreria di routine già pronte, messe a disposizione di tutti, fu breve. Il fine fu quello di creare una piattaforma di interfaccia elastica e sufficientemente efficiente per poter soddisfare a quasi tutte le esigenze; nacque così SmallTalk-80 destinata alla

interfaccia utente omonima, il Lisa toolkit per l'interfaccia utente Lisa e MacApp, dedicato all'interfaccia standard utente Macintosh. Il risultato fu la possibilità di costruire applicazioni espandibili che utilizzavano lo scheletro di base del toolkit, corredato dalle specifiche routine proprie del programma.

Una struttura applicativa ben realizzata dipende in gran parte dalle caratteristiche della programmazione object oriented. Ad esempio, la capacità di una routine principale di «chiamare», in una applicazione realmente espandibile, una specifica subroutine (o meglio dire, un «metodo», secondo la definizione assegnata a questo elemento nelle puntate precedenti) senza per questo dover modificare il progetto dell'intera applicazione, dipende essenzialmente dalle possibilità messe a disposizione dalla programmazione O.O. L'abilità del programmatore nell'espandere la sua applicazione (come avviene, ad esempio, negli upgrading delle applicazioni esistenti in commercio) sta nell'astuzia di aver saputo costruire un codice upgradabile in maniera facile e veloce, o, detta in termini O.O., nella abilità di sostituire un «metodo» di una «classe» con uno di una «classe» discendente. Tutto questo è possibile adottando un linguaggio dedicato all'object programming, ma non è questa la sola strada, anche se è la più agevole da percorrere. Un esempio è MacExpress, una piattaforma scritta in un linguaggio procedurale diffusissimo, che può essere usata come base di una applicazione nuova attraverso la modifica e la manipolazione di una serie di puntatori, forniti dal linguaggio stesso, che puntano, si perdoni la ripetizione, a procedure specifiche dell'ambiente stesso.

Per abbandonare le nostre divagazioni, MacApp è scritto in Object Pascal, e consiste in sei classi di basi, piuttosto articolate, che implementano la maggior parte delle routine e delle caratteristiche della interfaccia standard utente Macintosh. Chi conosce questo linguaggio si sarà reso conto che, attraverso di esso è molto facile intervenire per mo-

dificare quanto possa essere necessario; inoltre è possibile implementare routine di MacApp in un'applicazione specifica. Le sei classi di «eventi» manipolabili sono nell'ordine: TApplication, TDocument, TWindow, TFrame, TView e TCommand. Si tratta di classi di base già ben organizzate nel 1986 e che, nel tempo, si sono evolute e aggiornate mano a mano che nuovi tool erano messi a disposizione e in conseguenza degli aggiornamenti che la Apple faceva al suo Sistema Operativo e alle ROM di sistema.

Uno dei principi che anima la creazione di applicazioni costruite con tecniche di O.O. è di costruire il programma in modo abbastanza simile a quello che questo programma poi effettivamente dovrà svolgere. Ad esempio, nel costruire un programma di grafica, sembrerà logico creare nel programma stesso, una classe che manipola i poligoni, una che manipola le linee, una che determina le modalità di spostamento sullo schermo degli oggetti, una che gestisce i colori, e così via. Nel caso di costruzione di un linguaggio di programmazione, per continuare con gli esempi, si provvederà a realizzare una classe dedicata agli operatori numerici, una ai loop, una alle routine di gestione dei file (salvo poi a mettere assieme il tutto). Lo stesso principio è stato seguito da Apple nel mettere a punto MacApp; quindi, una classe per le finestre, una per i comandi (e i rispettivi menu), una per la gestione delle applicazioni e dei relativi file, ecc. In altri termini una finestra sa cosa fare quando viene premuto il close o lo zoom box, come eseguire l'upgrading dello schermo quando questo viene spostato o coperto da un'altra finestra, come si deve comportare quando si usano le scrollbar, e così via.

### Le classi fondamentali di MacApp La classe TApplication

Vediamo una per una queste differenti classi. La prima, la TApplication è incaricata del lancio dell'applicazione in uno dei tre modi noti (doppio click, chia-

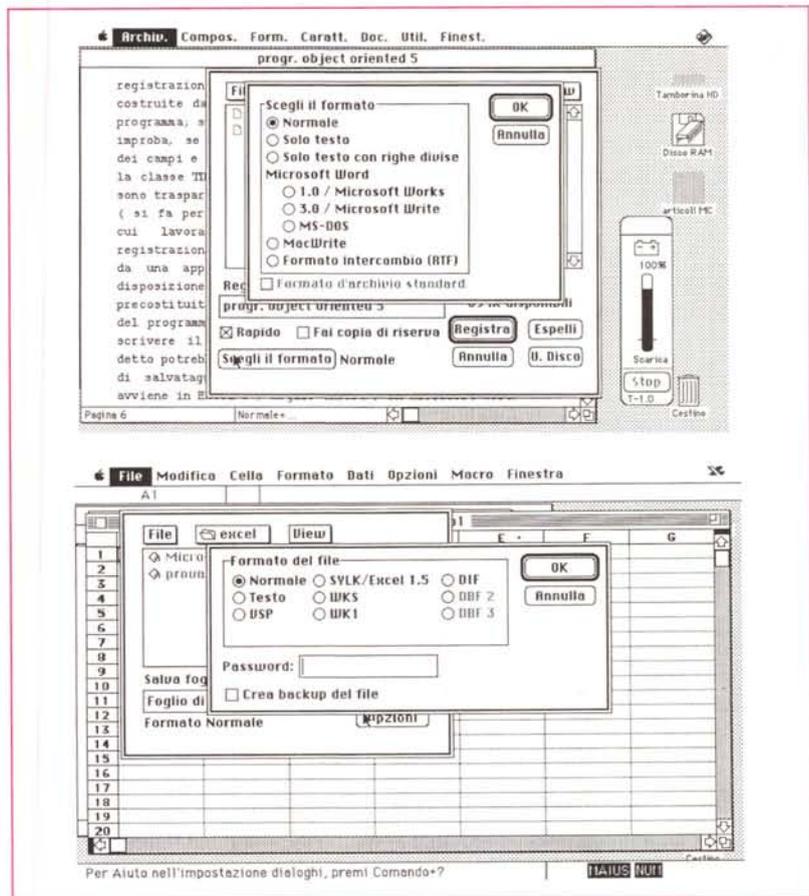
mata al comando «Apri» dal menu Finder, o shortcut); essa inoltre gestisce il percorso minimo di realizzazione della routine di stampa quando si desidera stampare un documento direttamente da Finder (lo standard di interfaccia Macintosh permette all'utente di stampare, come è noto, un documento senza aprire la relativa applicazione, direttamente dal File Menu di Finder). C'è da notare, ma si tratta di un'operazione trasparente all'utente normale, che l'applicazione, in questo caso non interviene nelle operazioni di stampa (tanto è vero che il Page Setup relativo non è quello specifico del documento ma quello del Finder).

In altri termini l'operazione di stampa non è svolta dall'applicazione stessa, ma dall'oggetto che trasforma l'immagine del documento, il View Object.

Altro compito di questa classe è la gestione dell'apertura del documento; questo si traduce nella chiamata alla finestra di dialogo che viene aperta chiamando dal menu il comando «Open». In

altri termini alla applicazione è destinato solo il compito di scegliere quale documento si desidera aprire; tutta l'operazione di scelta delle modalità di apertura, della gestione della lettura del file (si noti che, ad esempio, esistono documenti troppo grossi per essere letti nella loro interezza, per cui il programma deve anche gestire la gestione parziale in memoria di «fette» del documento). Ancora un altro compito di questa classe per così dire registra è la gestione del cosiddetto «Event Queue». L'applicazione esamina l'evento alla sommità della coda e invia un comando all'oggetto appropriato, per la gestione adeguata di esso (a meno che la gestione non sia eseguita a livello della root principale, direttamente dalla TApplication).

Ma non sono ancora finiti i compiti di questa classe; viene anche interessata la organizzazione del menu dell'applicazione, la risposta a certi comandi generali, come quelli ben noti di «About...» o di «Help», sovente presenti e disponibili



Un esempio di condivisione di documenti tra applicazioni diverse

per l'utente. Infine gestisce le relative finestre quando vengono selezionati comandi generali, come «Close», «Quit», ecc.

### La classe TDocument

Questa classe è destinata a gestire, controllare e organizzare tutte le opera-

zioni relative ai file di dati (ad esempio i documenti aperti da una applicazione, come Draw, Paint, Word fino ai più modesti «Score» dei giochi). Tra questi compiti ci sono quelli più importanti, come l'apertura e la chiusura del documento stesso. C'è da notare una differenza sottile tra la gestione in questo modo e quella attraverso MacApp. Immaginiamo di aver creato un file con un database; in un programma non O.O. l'apertura, l'aggiornamento dei record, la lettura e l'organizzazione dei puntatori ai record stessi, la registrazione, devono essere coordinati da routine costruite dal programmatore stesso e incorporate nel programma; si tratta, così di una faticaccia piuttosto improba, se si tiene conto delle infinite variabilità dei campi e dei record che possono sopravvenire. Usando la classe TDocument tutte queste operazioni massacranti sono trasparenti al programmatore; sua responsabilità (si fa per dire) è quella di individuare il file su cui lavorare e, eventualmente, il tipo di registrazione da adottare. Il resto del lavoro è svolto da una apposita routine di MacApp; essa mette a disposizione dell'utente tutta una serie di utility precostituite che maneggiano il file secondo i desideri del programmatore senza che esso debba preoccuparsi di scrivere il codice necessario. Un esempio di quanto detto potrebbe essere quello della gestione dei formati di salvataggio dei dati di uno spreadsheet, come avviene in Excel o, meglio ancora, in Microsoft Word. I programmatori che hanno costruito questo word processor non si sono certo preoccupati di definire un pezzo di codice per gestire le diverse opzioni di salvataggio del documento in base al formato. Essi hanno solo invocato una o più routine di MacApp (nella corretta maniera, ovviamente); dall'altro canto era inutile ripetere un lavoro che, magari, era stato già eseguito in maniera migliore e testato per un periodo senz'altro più lungo. In aggiunta a queste funzionalità, è destinata alla classe TDocument la gestione dei comandi di menu «Save», «Save as» e «Revert», la organizzazione degli «snapshot» presente in alcu-

La struttura di base della classe TDocument

Classe	Compiti specifici	Relazioni con le altre classi
TDocument 	<ul style="list-style-type: none"> <li>• maneggia tutto ciò che riguarda i file chiamati da una applicazione, tra cui:               <ul style="list-style-type: none"> <li>- apertura dei documenti</li> <li>- chiusura dei documenti</li> <li>- salvataggio su disco</li> <li>- ritorno alla precedente versione del documento</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• crea oggetti TView</li> <li>• crea oggetti TWindow</li> <li>• crea oggetti TFrame</li> <li>• gestisce alcune operazioni riguardanti oggetti del tipo view e window</li> </ul>

La struttura di base della classe TApplication

Classe	Compiti specifici	Relazioni con le altre classi
TApplication 	<ul style="list-style-type: none"> <li>• Maneggia tutte le interazioni tra le applicazioni e il Finder</li> <li>• maneggia la gestione degli event</li> <li>• mostra la barra di menu</li> <li>• maneggia i comandi presenti nel menu mela</li> </ul>	<ul style="list-style-type: none"> <li>• crea oggetti della classe TDocument</li> <li>• controlla la gestione di certi oggetti quando si verificano certe condizioni</li> </ul>

La struttura di base della classe TWindow

Classe	Compiti specifici	Relazioni con le altre classi
TWindow 	<ul style="list-style-type: none"> <li>• maneggia tutto ciò che riguarda le finestre, tra cui:               <ul style="list-style-type: none"> <li>- apertura delle finestre</li> <li>- chiusura delle finestre</li> <li>- spostamento delle finestre</li> <li>- dimensionamento e ridimensionamento delle finestre</li> <li>- attivazione delle finestre</li> <li>- deattivazione delle finestre</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• informa gli oggetti del tipo TFrame che certe operazioni (generalmente resa in attivo o dimensionamento) li hanno chiamati in causa</li> </ul>

La struttura di base delle classi descritte nell'articolo.

ni programmi come MacPaint II e la scelta del formato adatto di scrittura e lettura del file stesso. Ma la vera funzionalità di questa classe è quella di poter interagire con le altre classi per eseguire i seguenti compiti:

- gestione delle finestre per la visualizzazione dei contenuti del documento stesso (si pensi solo per un momento alle difficoltà insite nella gestione di documenti di grafica non in bitmap);
- gestione degli schermi che contengono le finestre (in collaborazione con TWindow e TFrame);
- gestione coordinata di finestre diverse, riferendosi allo stesso documento, come ad esempio la grafica di Fogli elettronici collegati a dati dello spreadsheet stesso, la gestione di utility come la sillabazione o lo spelling in word processor, la gestione di applicazioni condivise, il collegamento di documenti dipendenti da programmi diversi (come l'eccellente «Open Link — Collegamenti» di Excel, lo «Stampa unione» di Word o il caricamento di file ASCII in programmi di gestione di indirizzi, come in Silicon Press), o la gestione di finestre collegate (un esempio è ancora Silicon Press, do-

ve tre finestre sono collegate tra loro, o Word, dove la chiusura del documento comporta quella automatica di finestre come l'ortografia, il vocabolario, la sillabazione, la guida, e così via).

### La classe TWindow

Si tratta, questa, di una classe ibrida, che dimostra come la gestione delle finestre sia una delle cose più complesse dell'interfaccia Mac. Gli implementatori di MacApp hanno diviso la gestione delle finestre in due classi, questa e quella successiva (TFrame) proprio per evitare di creare un codice pesante da caricare e maneggiare. Così i compiti sono stati suddivisi. A questa classe ricorrono gli eventi che si rifanno a:

- apertura e chiusura delle finestre sullo schermo, con, cosa da non sottovalutare, gestione dei bordi (come fa un programma di grafica a «scrollare» quando il cursore «sbatte» contro il bordo dello schermo?);
- risposta alle richieste di ridimensionamento della finestra attraverso il resizing box;
- attivazione della finestra, come avvie-

ne quando si clicca nel suo interno per renderla principale e per metterla in primo piano;

- gestione, in parte, delle opzioni del Multifinder.

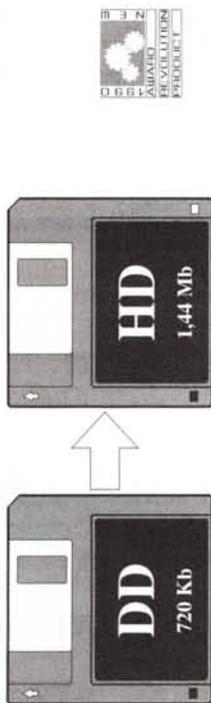
Anche TWindow è legato ad altre classi di MacApp, per eseguire insieme operazioni particolari, che richiedono l'intervento di tool delle due classi; alcune di esse sono:

- lettura del bottone del mouse, sia per quanto riguarda il suo stato (pressione o rilascio), sia per la determinazione del numero di volte schiacciato. Inoltre esso, in unione con le azioni descritte nella classe precedente permette il passaggio da window a window in applicazioni diverse (come accade in Multifinder o quando si utilizza un DA). Si immagini, in questo caso, la particolare complessità dell'operazione di scelta di apertura di una nuova applicazione mentre se ne mantiene attiva, ancorché in background un'altra;
- operazione di resizing della finestra.

Anche stavolta lo spazio è tiranno, continueremo così a trattare le restanti classi nella prossima puntata.

MS

**Dischi da 3"1/2 H.D. 1,44 MB, a sole 1.000 LIRE !!**  
Questa nuovissima invenzione crea l'impossibile.



NEW  
★★★★★  
1989  
IDEAS

## MAXIDISK CONVERTER

Se tu usi nuovi dischetti da 3"1/2 ad alta densità, sai che per averli devi pagare dalle 4000 alle 6000 lire! E' un po' troppo. addirittura quattro volte il prezzo di un vecchio floppy da 360K. Ora hai la possibilità di risparmiare e di avere un disco formattato a 1,44 Mb a sole 1.000 lire!

**COME E' POSSIBILE?** Hai mai provato a formattare un normale dischetto da 3"1/2 Doppia Densità in un drive da 1,44 Mb? Sicuramente sì, ma senza successo: il tuo PC ti segnalava un errore nella formattazione. In quanto il disco che hai inserito non è ad alta densità.

I nostri tecnici ed esperti in supporti magnetici hanno analizzato diverse decine di migliaia di dischetti da 3"1/2 per cercare le sostanziali differenze tra i supporti a Doppia Densità e quelli ad Alta Densità e il risultato finale è che conviene usare lo **STESSO SUPPORTO MAGNETICO** per entrambi i prodotti (Doppia Densità e Alta Densità).

**NON ESISTE ALCUNA DIFFERENZA!**  
Calcoli e studi di marketing hanno stabilito che per le Case produttrici è più conveniente usare lo **STESSO SUPPORTO MAGNETICO** per entrambi i prodotti (Doppia Densità e Alta Densità).

**LA DIFFERENZA E' SOLO NELLA PROTEZIONE DI PLASTICA!**  
E' infatti il secondo loro rettangolo presente sulla plastica a far sì che il dischetto venga riconosciuto dal drive come ad alta densità.

Dopo oltre un anno di test e di prove su migliaia di dischetti abbiamo messo a punto una particolare punzonatura di precisione in grado di riprodurre fedelmente il secondo foro e di non lasciare tracce e particelle di plastica che potrebbero danneggiare il disco.

Dall'inizio delle prove ad oggi sono stati convertiti, formattati e verificati più di 30.000 dischetti, e su questa quantità non abbiamo riscontrato una percentuale di dischetti difettosi maggiore di quella che normalmente si ha con i supporti ad alta densità.

**NON UNO DEI DISCHI CONVERTITI HA DATO ERRORI NELL'USO!** Il successo del **MAXIDISK** è dovuto soprattutto all'irresistibile offerta di guadagno in quanto con un piccolo investimento di 72.000 lire già puoi iniziare a risparmiare, infatti dopo aver convertito 100 dischi e dedotto il costo del **MAXIDISK** tu avrai risparmiato oltre 300.000 lire, un ritorno di tutto rispetto considerando l'investimento iniziale!

Il **MAXIDISK** è garantito per un anno e la garanzia vi tutela non solo sui difetti di fabbricazione ma anche sul successo della perfezione della conversione.

### ORDINA OGGI STESSO IL TUO MAXIDISK

Vogliate spedirmi:

N. \_\_\_\_\_ **Maxidisk**, a lire 72.000/cad IVA inclusa = lire \_\_\_\_\_

Pagamento contrassegno (lire 10.000) = lire \_\_\_\_\_

Spedizione corriere espresso 24 ore (lire 10.000) = lire \_\_\_\_\_

Spedizione contributo fisso = lire \_\_\_\_\_ 4.000,-

Totale del pagamento \_\_\_\_\_ = lire \_\_\_\_\_

NOME \_\_\_\_\_

INDIRIZZO \_\_\_\_\_

C.A.P. \_\_\_\_\_ CITTÀ \_\_\_\_\_ TEL. \_\_\_\_\_

PAGAMENTO:  ASSEGNO DI C/C NON TRASFERIBILE INCLUSO

PAGAMENTO IN CONTRASSEGNO (CONTRIBUTO LIRE 10.000)

VALIGIA POSTALE - ALLEGIO LA RICEVUTA O COPPIA

Per la fattura specificare codice fiscale e partita IVA; indirizzare a:

MAXIDISK

Info Data Communication

Via Chiaia, 112 - 80137 NAPOLI - Tel. (081) 7142208/7141994 FAX



Risparmia il tuo denaro...  
prendi anche tu un

### MAXIDISK CONVERTER™

- ✓ Made in Europe
- ✓ Costruzione metallica di prima qualità
- ✓ Nessun componente di plastica
- ✓ Converti oltre 200 dischi all'ora
- ✓ Testato per oltre 10.000 operazioni
- ✓ Punzone a doppio rinforzo
- ✓ Perforazione perfetta senza particelle
- ✓ Produce fori rettangolari, non circolari
- ✓ Identico ad un disco originale 1,44 Mb
- ✓ Styling moderno - colori tipo PC
- ✓ Per IBM compatibili e Macintosh
- ✓ Garanzia ufficiale di 12 mesi

Si cercano distributori per zone libere

IBM e Macintosh sono marchi registrati dei rispettivi proprietari

"20 ANNI DI ESPERIENZA NELL'INFORMATICA  
GARANTISCONO PRESTAZIONI E AFFIDABILITÀ"

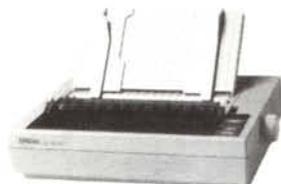


**A T A S T A R**

s.r.l. STABILIMENTO di Prato

Via Guicciardini, 29 - Tel. (0574) 38065/7 - Fax (0574)38068

VENITA ANCHE  
PER CORRISPONDENZA



24 aghi  
80 colori  
150 CPS

**EPSON**

L. 680.000



**NEC**

MULTISYNK  
3D L. 1.154.000  
4D L. 1.838.000



SCHEDA VIDEO: 1024x768

TSENG 512K 256	colori	L. 175.000
TSENG 1M 256	colori	L. 245.000
TRIDENT 1M 256	colori	L. 206.000



LaserJet III Hewlett-Packard

CON POSTSCRIPT  
DA L. 2.980.000



**tvm**®

MONITOR TVM MULTISYNK 14"

DP 0,28 colore 1024x768:

TVM 31-35KHz	L. 560.000
AOC 31-35KHz	L. 528.000
TVM 15-38KHz	L. 750.000
AOC 15-38KHz	L. 628.000



PREZZI INCLUSIVI DI SCHEDA MADRE, CONTROLLER AT-BUS, MEMORIA RAM, TASTIERA ESTESA microswitch CHERRY, CABINET MINI TORRE o DESKTOP\* con DISPLAY. FDD 1,2Mb o 1,4Mb

CPU	MHz	LANDM.	EXP. on board fino a	memor CACHE	RAM	LIRE
486DX	25 EISA	114 MHz	64M	128K	4Mb	6.279.000
486DX	25 ISA	114 MHz	8M		4Mb	5.319.000
386DX	33 EISA	58 MHz	32M		4Mb	2.402.000
386DX	33 ISA	58 MHz	16M	64K	4Mb	2.214.000
386DX	25	42 MHz	16M	64K	1Mb	1.690.000
386DX	25	42 MHz	16M		1Mb	1.594.000
386SX	20	27 MHz	17M		1Mb	984.000
*286	16	21 MHz	5M		1Mb	652.000
*286	12	16 MHz	5M		1Mb	628.000
V20NEC	12	12 MHz	1M		512K	495.000

I nostri COMPUTERS sono **AFFIDABILISSIMI, COSTANO MENO DI UN CLONE**. E queste sono alcune delle caratteristiche che li rendono **VELOCISSIMI** e di **QUALITÀ ALTAMENTE SUPERIORE**:

SHADOW RAM - EEMS VER 3.2 - LIM-EMS 4.0 - SHADOW RAM - VIRTUAL MODE 386 PER MULTITASKING - DISK CACHING - PIPELINE MODE - MEMORY FETCHING - BUS INTELLIGENT - ADVANCE NETWORK - VIRTUAL MODE 386 PER MULTITASKING - DATA BASE MANAGEMENT - M.M.U (MEMORY MANAGEMENT UNIT).

TUTTI I COMPONENTI ORIGINALI DI PRIMA SCELTA INTEL etc.  
GARANTITI 100% COMPATIBILI - UNIX - XENIX - LAN NOVELL.

DISCHI RIGIDI **Seagate**

42Mb	28ms	At-Bus	L. 385.000
90Mb	18ms	At-Bus	L. 726.000
124Mb	18ms	At-Bus	L. 846.000
211Mb	15ms	At-Bus	L. 1.336.000
337Mb	17ms	ESCDI	L. 2.130.000

GARANZIA 1 ANNO ESTENDIBILE A 3

INVIATECI IL VOSTRO COMPUTER.  
Il costo per l'installazione di un DISCO RIGIDO  
e di una SCHEDA VIDEO è di L. 98.000  
(Trasporti compresi)

GARANZIA 1 ANNO ESTENDIBILE A 3 - PREZZI IVA 19% ESCLUSA  
SU RICHIESTA INSTALLAZIONE E ASSISTENZA A DOMICILIO IN TUTTA ITALIA.  
TRASPORTO ECONOMICO TRAMITE CORRIERE.

**LAPTOP (3 KG)**

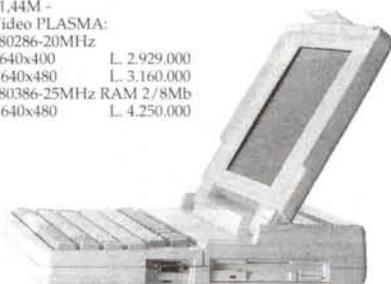
BATTERIE RICARICABILI  
286-16 MHz  
RAM 1MB EXP. 4MB  
FD 1,44M  
V. EGA HD 40M L. 2.360.000  
V. VGA HD 20M L. 2.980.000

con CPU 386SX - 16 MHz  
RAM 1MB EXP. 4MB  
FD 1,44M - HDD 20MB  
V. VGA HD 20M L. 3.360.000

Tutti con due porte ser. 1 par.  
per m. multisynk e tastiera

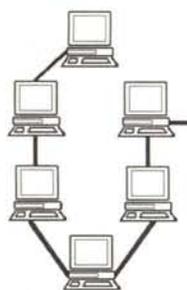
**LAPTOP (7 KG)**

CON 2 SLOTS D'ESPANSIONE  
RAM 1/4Mb - HDD 40M  
FDD 1,44M -  
con Video PLASMA:  
CPU 80286-20MHz  
EGA 640x400 L. 2.929.000  
VGA 640x480 L. 3.160.000  
CPU 80386-25MHz RAM 2/8Mb  
VGA 640x480 L. 4.250.000



**office automation**

**LAN**



**NOVELL RETE LOCALE**

Affidabilissima e Velocissima (Leader Mondiale)  
Il software NOVELL può collegare ogni PC, in qualunque linguaggio siano i programmi-utilizzatore.

Il prezzo comprensivo dell'installazione, di Software NOVELL e della Scheda ETHERNET 1Mbit/s per ogni posto di lavoro varia da L. 600.000 a L. 900.000.

**SISTEMI OPERATIVI XENIX - UNIX**

Il prezzo comprensivo dell'installazione, di Software e Scheda seriale per ogni posto di lavoro varia da L. 450.000 a L.800.000



**SUPER FAX 14** 9600 baud  
Tempo di trasmissione 15sec.  
142 memorie programmabili  
16 livelli di grigio  
Segreteria telefonica  
Orario programmabile L. 950.000  
SCHEDA FAX 2400-9600 baud L. 560.000  
Invio/ricezione. Confortevolissima.

**RIVENDITORI**

Informatevi sui vantaggi di una AFFILIAZIONE GRATUITA e del PROGRAMMA COMUNITARIO PUBBLICITÀ