Corso alternativo di programmazione

A proposito di date, del millennium bug, dei calendari

In questo secondo articolo dedicato a temi "alternativi di programmazione" parleremo di date, da sempre considerate elementi da trattare con le "molle" da tutti i programmatori, sempre pronte a tirar loro qualche "colpo basso", ultimo della serie il famigerato "millennium bug", che fortunatamente non ha causato i disastri temuti, ma che ha costretto le aziende a notevoli investimenti che hanno in gran parte contribuito ad arricchire numerosi consulenti.

di Francesco Petroni

Cosa è una data

La data può essere considerata un modo convenzionale per indicare un giorno.

Una data, secondo il nostro calendario, è composta da un numero che rappresenta l'anno, da un numero che rappresenta il mese (ed i mesi sono sempre 12), da un numero che rappresenta il giorno (ed i giorni nel mese possono essere 31, 30, 28 oppure 29).

Ogni sette giorni costituiscono una settimana, entità che quindi non è un sottomultiplo né del mese né dell'anno. Un calendario rappresenta una sequenza di giorni variamente organizzati, per giorno, per settimana, per mese e per anno.

Poi esistono altri concetti legati alle date, ieri, oggi e domani, ad esempio, che possono essere anche questi manipolati in un programma.

La data dal punto di vista del programmatore

Insomma, ma perché per i programmatori le date sono difficili da maneggiare?

- Innanzi tutto perché per rappresentarle si usano caratteri numerici oppure letterali, oppure un misto tra i due e quindi facilmente equivocabili: es.

  12052001
  12/5/2001
  12MAG01
  5/12/2001
  2001/5/12
  sabato 12 maggio 2001

Inoltre per affermare che queste sei righe rappresentino una data e che la data sia la medesima, ci servono assolutamente altre informazioni, ad esempio quella che indichi la modalità di rappresentazione utilizzata.

- La data è un concetto relativo nel senso che necessita di un punto di riferimento. L'età di una persona altro non è che la differenza tra due date, i dinosauri sono vissuti 50.000.000 milioni di anni fa, le nozze d'oro corrispondono all'età di un matrimonio, scoccano 50 anni dopo il matrimonio, ecc.
- Esiste una data di "partenza" convenzionale per il nostro calendario che è la nascita di Cristo. E' talmente importante che anche le date precedenti alla nascita di Cristo fanno riferimento a tale data convenzionale.
- Nel trattare le date spesso ci imbattiamo in espressioni non immediatamente traducibili in codice. Eccone alcuni esempi.

La frase "ci vediamo tra tre mesi" non può essere tradotta in un numero fisso di giorni (possono essere 92, 91, 90 oppure 89), né di settimane, in quanto dipende dal giorno in cui viene

Figura 1 - In che anno Jimi Hendrix ha suonato a Roma? Questo è un "volantino", a dir poco rudimentale, ma che conservo gelosamente, del concerto tenuto a Roma da Jimi Hendrix al quale ho assistito. Sul volantino, di cui vedete solo la parte superiore, manca l'indicazione non solo dell'anno, che doveva essere il 1968 o il 1969 (Hendrix è morto nel 1970, il 18 settembre), ma anche del mese. Chi lo guarda, ad esempio i miei figli, che sono nati negli anni '80, non ne ricavano nessuna indicazione e quindi nessuna sensazione temporale. In questo articolo parliamo di date. Non preoccupatevi, non ne parliamo in termini filosofici - il passare del tempo, il loro rapporto con i ricordi, ecc. -, ma in termini tecnici, da rudi programmatori quali siamo e quali vogliamo essere.

pronunciata.

Alcuni calcoli finanziari utilizzano dei mesi "ideali" di 30 giorni, per cui il 28 febbraio capita tre giorni prima del giorno successivo, mentre il 31 di ogni mese è del tutto equivalente al giorno precedente.

In certe procedure è necessario calcolare il primo giorno feriale del mese.

Nelle banche si usa il concetto di data valuta, che indica quanti giorni dopo un versamento la somma versata è realmente disponibile.

Alcune feste non capitano gli stessi giorni di ogni anno: la Pasqua e il Natale si comportano diversamente. Pasqua cambia di anno in anno ma capita sempre di Domenica, Natale... penso che lo sappiate.

Nella gestione del personale l'anzianità, e quindi il diritto ad andare in pensione, si misura in anni, mesi e giorni. Es. 20 anni, 6 mesi e 1 giorno.

I giorni lavorativi di un mese non sono sempre gli stessi. Ad esempio un mese può avere 21 giorni lavorativi e lo stesso mese dell'anno successivo 2 giorni in più. Se ipotizziamo una produzione giornaliera fissa, la produzione mensile invece è del 9,5% superiore. Il dato mensile diventa non significativo, anzi palesemente errato.

Un uomo nato il 28 agosto 1945, il 31 maggio 2001 ha compiuto 54 anni, 9 mesi e 2 giorni, ma anche 2857 settimane, oppure 20.000 giorni. Non esiste il compigiorno, né il "buon non compleanno" di Alice nel Paese delle Meraviglie...

Dopo questa serie di amenità, eccovi alcune premesse importanti prima degli esercizi.

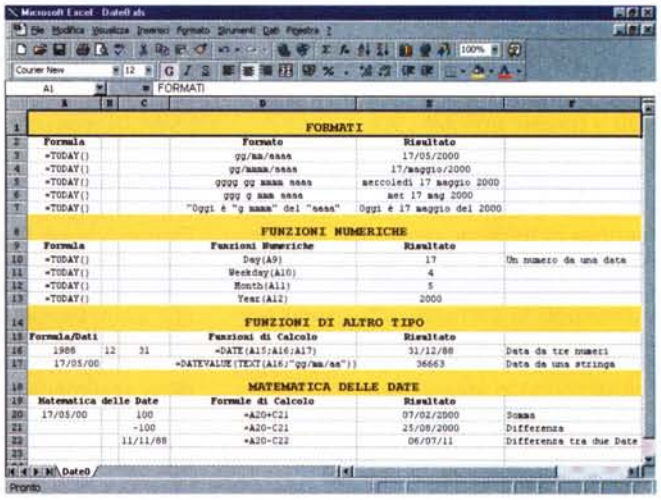
Come Windows vede le date

Sotto sotto il calendario di Windows è fatto di giorni, conseguentemente una data corrisponde ad un semplice numero intero. Se anche noi ragionassimo solo con i giorni, tutti i calcoli con le date sarebbero molto più semplici; inoltre, poiché tutti i calcoli tra le date sono calcoli "relativi", basta anche in questo caso stabilire un giorno convenzionale per la "partenza" della numerazione, ad esempio, come fa Window, il primo gennaio 1900, che è quindi il giorno numero 1.

Tutti i prodotti per Windows, comunque, dispongono di strumenti per "tradurre" questo numero nella data, così come la vediamo noi umani. Tali strumenti sono le funzioni ed i formati di visualizzazione. E' importante non confondere gli uni con le altre.

Aprite un foglio Excel e scrivete il numero 37000. Nel calendario di Windows questo numero corrisponde a

Figura 2 - Sperimentazioni sulle date con il foglio di MS Excel.
In questo articolo parliamo quindi di date, intese sia come "tipo di dato", da non confondere né con le stringhe né con i numeri, sia come elemento da maneggiare, con la dovuta cura, nelle varie applicazioni. Il primo esercizio che vi proponiamo ha come finalità una prima familiarizzazione con il concetto di data, con i suoi formati di visualizzazione, con le funzioni che manipolano le date e con le operazioni "aritmetiche" che le possono riguardare. Per il nostro primo esercizio utilizziamo il foglio di Excel, che contiene tutto il necessario per iniziare.



			FORMATI		
	Formula		Formato	Risultato	
	=TODAY()		gg/mm/aaaa	17/05/2000	
	=TODAY()		gg/mmm/aaaa	17/maggio/2000	
	=TODAY()		gggg gg mmm aaaa	mercoledì 17 maggio 2000	
	=TODAY()		ggg g mmm aaaa	mer 17 mag 2000	
	=TODAY()		"Oggi è "g mmm" del "aaaa"	Oggi è 17 maggio del 2000	
			FUNZIONI NUMERICHE		
	Formula		Funzioni Numeriche	Risultato	
	=TODAY()		Day(A9)	17	Un numero da una data
	=TODAY()		Weekday(A10)	4	
	=TODAY()		Month(A11)	5	
	=TODAY()		Year(A12)	2000	
			FUNZIONI DI ALTRO TIPO		
	Formula/Dati		Funzioni di Calcolo	Risultato	
	1988	12 31	=DATE(A15;A16;A17)	31/12/88	Data da tre numeri
	17/05/00		=DATEVALUE(TEXT(A16;"gg/mm/aa"))	36663	Data da una stringa
			MATEMATICA DELLE DATE		
	Matematica delle Date		Formule di Calcolo	Risultato	
	17/05/00	100	=A20+C21	07/02/2000	Somma
		-100	=A20-C21	25/08/2000	Differenza
		11/11/88	=A20-C22	06/07/11	Differenza tra due Date

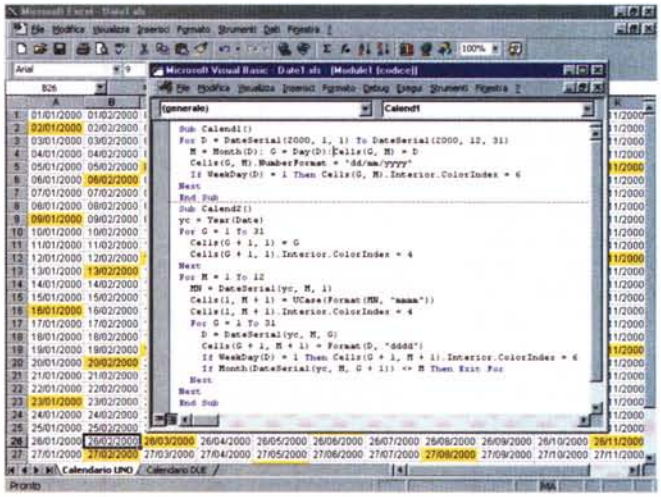


Figura 3 - Macro di MS Excel per creare dei calendari – Primo esempio e codice.
Usiamo ancora Excel per imparare a maneggiare, anche da un programma, le date. Excel dispone sia di funzioni usabili direttamente sul foglio, che abbiamo in parte visto nell'esempio precedente, sia di funzioni VBA (Visual Basic for Application), usabili all'interno delle macro. Ecco due programmi che creano, su 12 colonne di un foglio, un calendario. Nel primo caso eseguiamo un semplice ciclo (istruzione For Next) numerico tra due date, sfruttando l'affermazione, meglio approfondita nel testo, che "per Windows una data equivale ad un numero". Nota una data, ne sono noti sia il relativo mese (che corrisponde alla colonna) che il relativo giorno (che corrisponde alla riga) e quindi la stessa è facilmente posizionabile sul foglio.

giovedì 19 aprile 2001. Per controllarlo basta formattare la cella con un formato data. Possiamo provare a controllare il contrario. Scriviamo la data digitandola secondo uno dei formati riconosciuti come data, ad esempio 19/04/2001, oppure 19-apr-2001, e poi formattiamola come numero generico. Rivedremo il numero 37000.

La stessa data può essere vista in tanti formati differenti, ma rimane sempre una data, la stessa data, e il contenuto della cella rimane sempre di tipo data. Insomma il formato di una cella è solo un fatto "estetico" che non ne altera il valore reale.

Le **funzioni** invece producono generalmente una modifica del tipo di dato. Ad esempio:

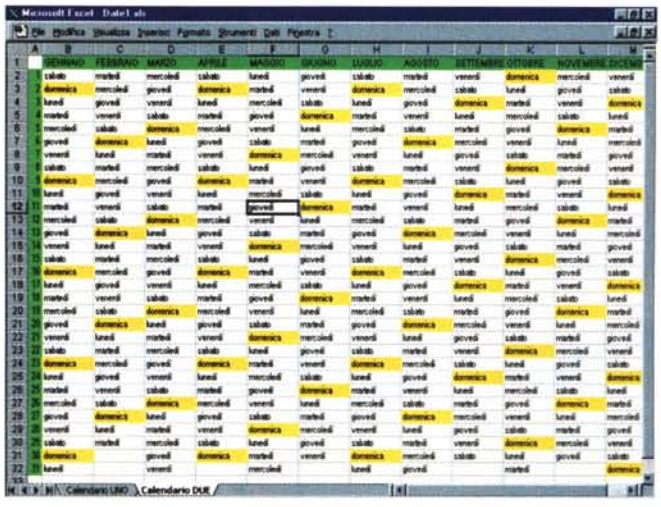


Figura 4 - Macro di MS Excel per creare dei calendari – Secondo esempio.
Nella seconda macro, invece di utilizzare un solo scorrimento tra due date sfruttiamo due scorrimenti, il primo che genera i 12 mesi di un anno ed il secondo che genera tutti i giorni del singolo mese. In questo secondo caso, quindi, costruiamo la data partendo dal giorno e dal mese, usando l'utilissima funzione DateSerial. Nota una data ne è, ovviamente, noto il giorno della settimana (funzione WeekDay, che restituisce un numero, 1 per la Domenica e così via). Se vogliamo invece il giorno della settimana "in chiaro", possiamo utilizzare le funzioni di formattazione, come la ben nota Format.

```
MM = Month(DD)              DD è una data, MM un numero, entra una data esce un numero
MN = MonthName(NN)          NN è un numero, MN una stringa
YY = Year(DD)               DD è una data, YY un numero
GG = Day(DD)                DD è una data, GG un numero, il giorno del mese
GG = WeekDay(DD)            DD è una data, GG un numero, il giorno della settimana, 1 è domenica
FF = Format(DD,"gggg gg mmmm aaaa")  DD è una data, FF una stringa
DD = Dateserial(YY,MM,DD)   YY, MM, DD sono numeri, DD una data
```

Tre precisazioni.

La prima è che in Excel ci sono due famiglie di funzioni, purtroppo simili ma non identiche: quelle usabili sul foglio e quelle usabili in un programma VBA. Alcune sono in comune, altre sono presenti nel foglio e non in VBA e viceversa. Altre sono differenti come sintassi, ma svolgono lo stesso lavoro.

Ad esempio la **DateSerial** vale in VBA, mentre sul foglio lo stesso servizio è svolto dalla funzione **Date**.

La seconda precisazione riguarda il fatto che il patrimonio di funzioni in generale evolve da versione a versione del prodotto.

Quelle citate ed utilizzate nell'articolo sono abbastanza standard, ad esempio nei prodotti Office 97, e sono presenti da tantissime versioni. In MS Office 2000, nel suo VBA, ed in MS Visual Basic 6.0 ce ne sono di nuove. Citiamo - anche se, come detto, non ne mostriamo esempi - **MonthName**, dal significato ovvio, e **DateDiff** (alla quale va passato un parametro che indica quale unità visualizzare, tra anni, mesi, settimane, giorni, ecc.)

La terza precisazione riguarda il fatto che un programmatore tende a risolvere "in proprio" i problemi, ad esempio costruendosi proprie funzioni personalizzate. Ve ne proponiamo immediatamente una che serve a contare quanti giorni feriali ci sono in un dato mese di un dato anno.

I parametri in entrata sono quindi il numero dell'anno e il numero del mese, mentre in uscita abbiamo il numero dei giorni feriali, quelli che non sono né Domenica né Sabato. Nella funzione, che abbiamo chiamato ContaFer, calcoliamo la data con la funzione DateSerial, alla quale passiamo il numero di giorno, di mese e di anno. Poi con la funzione WeekDay controlliamo quale giorno della settimana è e se è feriale lo contiamo. Aumentiamo via via i giorni a partire dal primo giorno del mese poi, per verificare se abbiamo "sforato", usiamo la funzione Month per verificare se il mese della data è ancora quello che ci interessa.

Questo è reso possibile dalla caratteristica della funzione DateSerial di gestire i "riporti", quindi il mese 13 è il gennaio dell'anno dopo, il giorno 32 è il primo giorno del mese successivo, se il mese precedente è di 30 giorni, ecc.

Ecco il listato:

```
Function ContaFer(AA As Integer, MM As Integer) As Integer
ContaFer = 0
DC = DateSerial(AA, MM, 1)                    ' Data iniziale
Do
  If Weekday(DC) = 1 Or Weekday(DC) = 7 Then  ' Controllo sabato o domenica
  Else
    ContaFer = ContaFer + 1                   ' Conta i feriali
  End If
  DC = DC + 1                                 ' Giorno successivo
  If Month(DC) <> MM Then Exit Do             ' Controllo mese successivo
Loop
End Function
```

Questa funzione non tiene conto delle feste infrasettimanali.

Per tenerne conto occorrerebbe inserire un elenco, insomma una semplice matrice, con le date dei giorni comunque festivi, come Pasqua, Natale, Ferragosto, la Festa del Patrono, ecc.

Sempre rimanendo in tema di funzioni, ne cito altre due molto utili.

La prima è la IsDate(DD), che restituisce un valore true o false a seconda che DD sia o meno un dato di tipo data.

Questa funzione, che appartiene ad una numerosa famigliola di funzioni Is..qualcosa, è comodissima per tutte le problematiche di controllo dell'input.

La seconda, presente purtroppo solo nel foglio di Excel, è la Days360(D1,D2), che esprime la differenza in giorni tra due date. Tale differenza è calcolata considerando mesi di 30 giorni e quindi anni di 360 giorni e facilita tutti i calcoli dell'anzianità nelle problematiche del personale e nei calcoli dei ratei e degli interessi nelle problematiche finanziarie. In uno degli esercizi proposti abbiamo ricostruito la funzione.

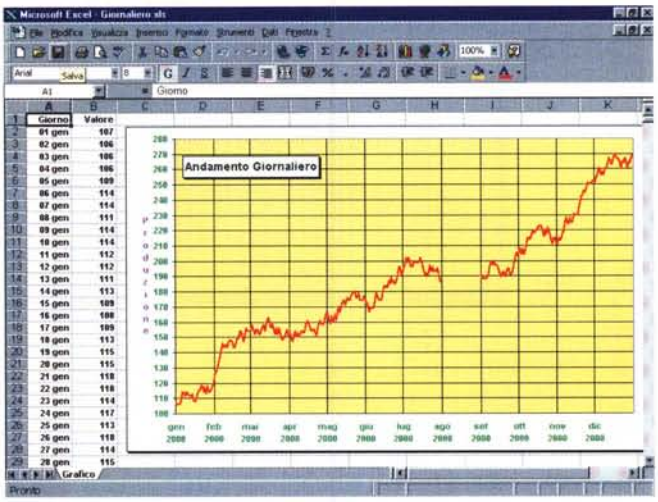


Figura 5 - Grafo di MS Excel con tutti i giorni di un anno sull'asse delle X. La conoscenza delle date non è mai fine a se stessa. La data è spesso uno dei tanti elementi presenti in un certo problema che deve essere trattato insieme agli altri. In questo esempio, solo apparentemente semplice, sono stati affrontati e risolti una serie di problemi legati alle date, ad esempio la realizzazione della scala del grafico che indica il singolo mese, anche se i dati visualizzati sono giornalieri, oppure il "buco" nella produzione di Agosto, perché ad Agosto la produzione viene sospesa, ecc. Sapreste fare altrettanto?

Applicazione del concetto di data

Altro tema che ci sembra opportuno premettere agli esercizi è quello che riguarda l'uso che si fa delle date.

Ad esempio, se in una tabella di un database è presente un campo di tipo data, lo si può usare per eseguire selezioni (ne vediamo alcuni esempi nelle varie figure), per eseguire ricerche, per eseguire ordinamenti, raggruppamenti per giorno, settimana, giorno della settimana, mese, ecc.

Anche nella grafica di tipo Business si fa molto uso delle date, in tutti i grafici statistici, in tutti i grafici che mostrano le quotazioni di Borsa nel tempo, ecc.

Anche un calendario è in definitiva un modo di utilizzare le date, sia perché è un modo per impaginarle, sia perché un calendario, normalmente, visualizza altre informazioni, ad esempio il santo del giorno, o permette di inserirne altre, anche a mano, ad esempio gli appuntamenti. Spesso una vista in chiave temporale è il miglior modo per analizzare un dato.

Passiamo agli esercizi

Essendoci dilungati nelle premesse, saremo brevi nella descrizione degli esercizi che troverete nel CD in dotazione alla rivista.

Esercizio 1 - Pratica con Excel.

Lo vediamo in figura 2. E' un esercizio libero, che ha come scopo quello di provare i formati di visualizzazione delle date, di provare le funzioni che maneggiano date, disponibili sul foglio di Excel, di provare un po' di aritmetica delle date:

una data più un numero X restituisce una seconda data di X giorni successiva

la differenza tra due date restituisce un numero di giorni solari.

resse principale degli esercizi sta nella differenza "filosofica" delle due soluzioni. Nella prima eseguiamo un ciclo numerico tra due date. Dalla generica data ricaviamo tutto quello che ci occorre, ovvero giorno della settimana, giorno del mese e mese, queste ultime due informazioni necessarie al posizionamento del risultato sul foglio. Nella seconda "cicliamo" su 12 colonne, che corrispondono al mese, e su 31 righe, che corrispondono al giorno. Con la funzione DateSerial (confesso, è la mia preferita) costruiamo la data e la posizioniamo al suo posto.

Dobbiamo controllare se sforiamo

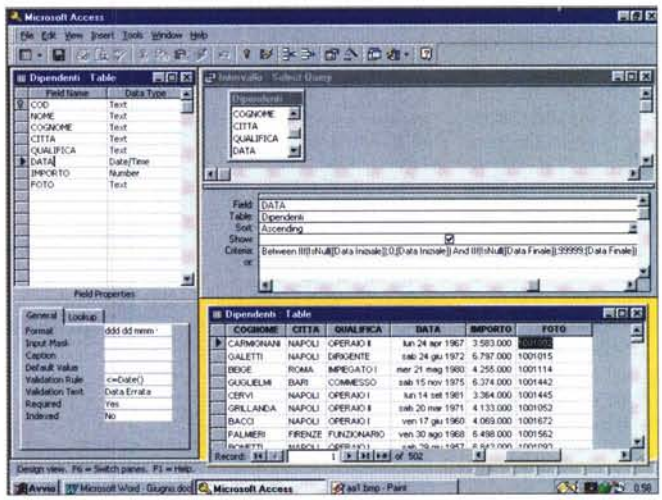


Figura 6 - MS Access – Cosa si può fare con una data.
In questa videata vediamo, in un'applicazione realizzata con MS Access, come sia possibile creare un campo di tipo data, come di questo sia possibile impostare un formato di visualizzazione direttamente nella struttura della tabella, come sia possibile impostare una regola di validità in fase di immissione. Tutte queste impostazioni varranno "per sempre", nel senso che in nessun caso sarà possibile trasgredirle. Sulla destra vediamo inoltre la struttura di una query un po' "spinta". Si tratta di una query parametrizzata, che esegue una selezione di dati sulla base di un intervallo di date. Se non si immette uno dei due parametri viene presa una data lontanissima, sia per quanto riguarda la data iniziale che la data finale. Nel testo la descrizione dettagliata di questo utile meccanismo, sfruttabile anche per i campi numerici.

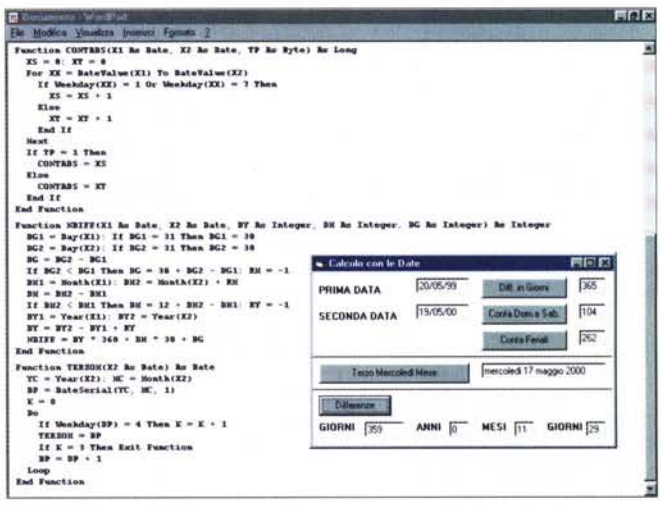


Figura 7 - MS Visual Basic – Costruzione di funzioni personalizzate – Output con il listato sullo sfondo.
Molto spesso nelle nostre applicazioni occorrono delle funzioni particolari non presenti nella dotazione del prodotto che stiamo usando. Si debbono quindi realizzare delle funzioni personalizzate. Nel nostro caso ne proponiamo tre, la prima, CONTADS, che conta i giorni festivi oppure i giorni feriali tra due date e quindi restituisce un numero; la seconda, TERZOM, che trova il terzo mercoledì di un dato mese e di un dato anno, e quindi restituisce una data; infine la funzione NDIFF, che calcola la differenza tra due date, esprimendola in anni, mesi e giorni. Nel nostro caso restituisce i tre numeri ed anche un numero con il totale dei giorni.

Esercizio 2 e 3 – Due macro Excel.

Scriviamo due Macro per realizzare due "calendari" sul foglio di Excel. L'intecon il mese utilizzando la funzione **Month**, poi dobbiamo controllare il giorno della settimana perché vogliamo evi-

denziare le domeniche e quindi usiamo la **WeekDay**.

Esercizio 4 – Un grafico di Excel in scala temporale.

Immaginate delle rilevazioni giornaliere (tipico le quotazioni della Borsa) eseguite per un anno intero da riportare in un grafico. Il problema principale è la visualizzazione delle etichette sull'asse X: non si possono vedere tutti i giorni, né alcuni giorni opportunamente intervallati. Ci aiuta Excel, che "capisce il nostro imbarazzo" e ci permette di visualizzare i mesi.

Esercizio 5 – Una tabella di Access.

Questo esercizio ricade tra quelli di "avvicinamento" al problema delle date. In Access la data esiste come tipo di dato e quindi si può indicare a livello di struttura. Poi se ne può impostare, una volta per tutte, il formato di visualizzazione, se ne possono impostare il valore di default e le regole di validità. Tali impostazioni così rigorose servono a garantire "nei secoli" la correttezza del dato immesso e non potranno mai essere trasgredite, qualsiasi sia il sistema o il prodotto che si utilizzi per accedere ai dati.

Sulla destra della videata abbiamo sistemato una "query" un po' spinta. Si tratta di una query parametrizzata che esegue una selezione in intervallo di date. Chiede quindi una data iniziale ed una data finale.

Permette alcune varianti.

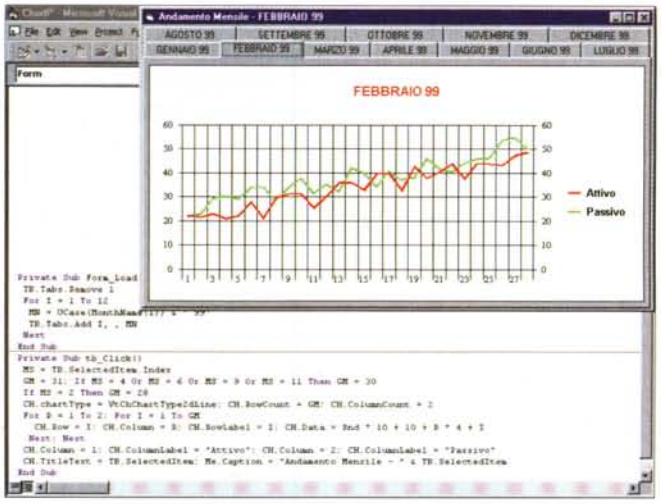


Figura 8 - MS Visual Basic – Esempio di applicazione delle date. Come abbiamo già detto in una delle precedenti didascalie, una data non è mai fine a se stessa. Qui vediamo un'applicazione VB che mostra dati sintetici organizzati per mese. Il mese, scelto dalla linguetta (tab) del controllo TabStrip di Visual Basic, diventa elemento per la selezione dei dati.

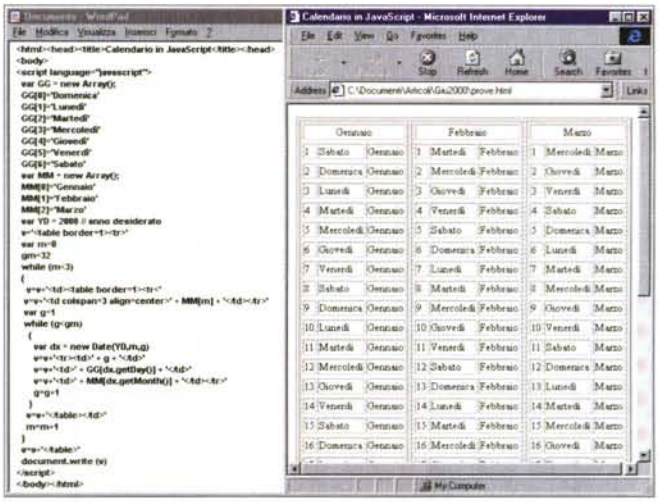


Figura 9 - Calendario realizzato in JavaScript. Anche tutti i linguaggi di programmazione per Internet permettono di manipolare date. I linguaggi per Internet sono tutti di tipo "interpretato" e si dividono in due categorie, quelli client, che vengono quindi interpretati dal browser che incorpora il motore di interpretazione, e quelli server, che vengono interpretati da un motore sul server. In questo secondo caso al client viene inviato il risultato del calcolo e non il codice, mentre nel primo viene inviato il codice. Il linguaggio più diffuso "lato client" è il JavaScript, di cui mostriamo un esempio. In pratica si tratta della traduzione in JavaScript delle routine di produzione del calendario.

Inserimento di ambedue le date	i record la cui data sia compresa nell'intervallo
Inserimento della sola data iniziale	tutti i record a partire dalla data iniziale
Inserimento della sola data finale	tutti i record fino alla data finale
Inserimento di nessuna data	tutti i record.

Il trucco, un po' scorretto per i puristi della query, ma efficace e riutilizzabile con qualsiasi tipo di campo, si basa sul fatto che la data è un numero, quindi con la funzione IIF si controlla che la data iniziale - lo stesso discorso vale per la data finale - sia stata digitata. Se non è stata digitata si prende un valore numerico piccolissimo. La sintassi SQL è la seguente:

```
Between IIf(IsNull([Data Iniziale]);0;[Data Iniziale])
And IIf(IsNull([Data Finale]);99999;[Data Finale])
```

Un maggior controllo si potrebbe ottenere usando la funzione IsDate, che permette di intercettare se è stata digitata una data oppure una espressione scorretta.

Esercizio 6 – Tre funzioni personalizzate realizzate con il Visual Basic.

Nella figura 7 vediamo solo le tre funzioni e non il programma che le utilizza.

Alla funzione CONTADS vanno passate due date ed un parametro numerico, che può valere 1 o 2 e che indica se la funzione stessa deve contare i giorni feriali o quelli festivi tra le due date.

Alla funzione TERZOM, che ritorna il terzo mercoledì del mese, va passata una data dalla quale viene estratto l'anno ed il mese necessari per calcolare quanto richiesto. La funzione restituisce una data. La funzione NDIFF ha un mare di parametri, due in entrata: le due date, iniziale e finale, per le quali calcolare la differenza in anni, mesi e giorni, e quattro in uscita, il numero di anni, il numero di mesi ed il numero di giorni; inoltre, la funzione restituisce direttamente il totale di giorni considerando mesi di 30 giorni e quindi anni di 360 giorni. Da notare lo sfruttamento della clausola ByRef (che è quella di default) nel passaggio dei tre parametri numerici, che in questo modo diventano parametri in uscita.

Anche di Visual Basic proponiamo un esempio in cui le date vengono usate come elemento di elaborazione. Lo vediamo in figura 8. Usiamo i mesi per impostare le tab di un controllo TabStrip, e quindi per selezionare i dati e produrre un grafico al click sulla "linguetta".

Passiamo alla programmazione per Internet

Penso che ormai la maggioranza dei programmatori sia impegnata a lavorare su applicazioni che girano sul WEB e quindi nei nostri articoli dobbiamo pensare anche a loro.

Fortunatamente quanto detto fino ad ora sulle date vale anche su Internet, soprattutto per il fatto che per programmare per tale ambiente vengono sfruttati tutti i linguaggi tradizionali che si affiancano e collaborano con i linguaggi specifici, tipo **HTML** e **DHTML**.

I due linguaggi più diffusi sono il **Java-**

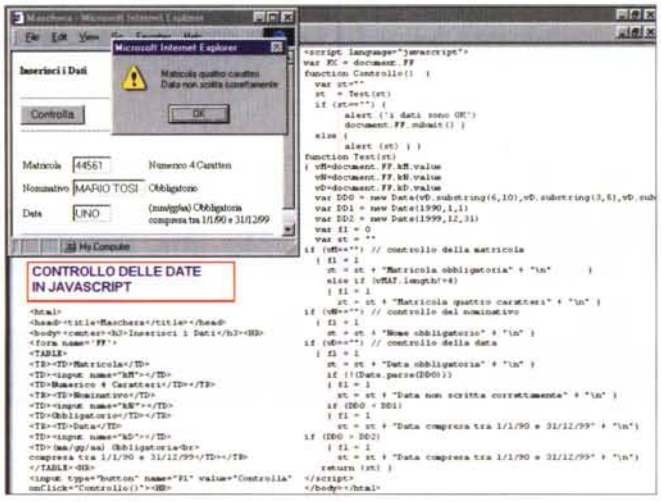


Figura 10 - Maschera con controlli in JavaScript.
Uno dei principali compiti dei programmi lato client è quello di controllare "il più possibile" quanto i vari utenti inseriscono nelle form. Tutti i controlli formali, ad esempio che un campo obbligatorio sia stato effettivamente riempito, che un campo numerico contenga effettivamente un numero, che un campo data contenga effettivamente una data, che un campo in cui inserire un indirizzo e-mail contenga il carattere @, ecc., vanno eseguiti sul client. Nel nostro esempio vediamo come, in JavaScript, sia possibile controllare che la data sia una data e che la stessa sia compresa tra due date prefissate. Altri tipi di controlli, ad esempio che una password sia stata digitata correttamente e corrisponda all'identificativo dell'utente, devono essere eseguiti lato server.

Script e il **Visual Basic Script** e ambedue possono essere utilizzati sia lato client, ed in questo caso è il browser che si occupa di interpretare il codice, che lato server, ed in questo caso è il motore del server che interpreta il codice e produce il codice HTML da inviare verso client.

Esercizio 7 - Calendario realizzato con JavaScript (figura 9).

Per quanto riguarda le date, vale tutto quanto detto fino ad ora: si usano formati e funzioni, per cui le nostre soluzioni assomigliano terribilmente a quelli realizzate con il Visual Basic o con il Visual Basic for Application.

Anche in JavaScript esiste il tipo data, anche se è espresso in millisecondi ed a partire dal 1 gennaio 1970 e quindi contiene in sé anche le informazioni sull'orario.

In JavaScript per costruire una data si può usare l'istruzione:

var DX = new Date(YY,MM,GG)

dall'ovvio significato.

Invece se si vuole impostare la data di oggi:

var DX = new Date()

Ecco due funzioni che lavorano sulla variabile DX, che è di tipo data:

DX.GetDay()
DX.GetMonth()
DX.GetYear()

che restituiscono numero del giorno, numero del mese, numero dell'anno. Per italianizzare mesi e giorni, per il nostro esercizio, abbiamo creato delle semplici array, in cui abbiamo memorizzato i nomi dei giorni della settimana e quelli dei mesi.

Esercizio 8 - Controllo dei dati immessi.

Sempre in JavaScript è stato realizzato l'esercizio che presentiamo per affrontare la problematica "controllo dati in fase di immissione". Vi proponiamo una routine per controllare che una data digitata sia stata immessa, che sia formalmente valida e che sia compresa tra due date estreme. La funzione:

Date.Parse(DX)

è una funzione logica che ritorna **true** o **false** a seconda che DX sia effettivamente una data o meno. Superato il controllo formale per eseguire i successivi confronti, bisogna conoscere la simbologia JavaScript per gli operatori booleani:

! = Not
| = Or
& = And
ecc.

Il programma mostrato esegue anche controlli sul campo matricola, obbligatorio e lungo 4 caratteri, nominativo, obbligatorio, oltre che sul campo data, obbligatorio, di tipo data e compreso tre 1/1/90 e 31/12/99.

Nel caso in cui uno o più di uno di questi controlli non vengano superati, viene preparato e mostrato un messaggio. In caso contrario può essere eseguita la submit della form, ovvero i dati digitati possono essere inviati al server per la loro registrazione.

Esercizio 9 - Programma ASP, eseguito lato server, che crea un calendario.

Per finire vi presentiamo un calendario applicato nel senso che contiene anche dei dati. In figura 11 vi mostriamo solo il risultato del programma, mentre il listato, troppo lungo per essere pubblicato, lo potrete trovare sul CD allegato alla rivista.

Il dinamismo dell'applicazione sta ne fatto che è possibile "navigare" tra i mesi e gli anni attraverso una apposita pulsantiera. In base al mese e all'anno così determinati vengono calcolati, sfruttando la dotazione di funzioni per la manipolazione delle date del VBS, tutti gli elementi necessari alla costruzione della tabella, come quale sia il giorno della settimana del primo giorno del mese, indispensabile per determinare la casella di partenza, oppure il meccanismo di sincronizzazione tra i giorni della settimana e del mese, che serve per scalare di una riga quando la data corrisponde ad un sabato.

Il problema di cosa inserire nella singola cella, che è indipendente dal problema della costruzione del calendario, lo abbiamo risolto in modo elementare. Ad ogni giorno corrisponde anche un contenuto testuale che abbiamo inserito in una matrice di stringhe per poi riversarlo nelle varie celle. La matrice può essere ovviamente letta da un file, selezionando solo i dati necessari in base all'accoppiata mese e anno. Nel nostro caso l'abbiamo riempita "a mano", non avendo trovato un database con i santi. MC

Figura 11 - Calendario "programmato" in una pagina ASP.
Diamo per nota, almeno dal punto di vista concettuale, la tecnologia ASP, che è parte integrante dell'IIS, Internet Information Server, il server Internet della Microsoft. Una pagina ASP (Active Server Pages) in genere contiene, oltre al normale codice HTML, del codice Visual Basic Script, oppure JavaScript, che viene eseguito dal server e che produce ed invia al client, su richiesta di quest'ultimo, pagine HTML. Dall'interno del codice ASP si possono richiamare funzioni presenti nelle DLL di Windows, sia quelle in dotazione al sistema, sia quelle realizzate "in proprio" dal programmatore. La più importante è la ADO Library (siamo alla versione 2.5, in Windows 2000), che permette di accedere ai database. Nel nostro caso, ipotizziamo la produzione di un calendario contenente dati disponibili in un database interrogato "dinamicamente" allo scorrere delle pagine.