

**T**rade magazine

mensile di speculazione finanziaria

trade magazine • numero 8 • giugno / luglio 2008

Lo **Scalping**

Luca Lazzaretti

Dollar **Sensitive**

Marco Procoli

Prospettive di **breve...**

Iride Research & Consulting

**Acqua**

Trimagazine

Il prezzo di **apertura**

Igor Boggio

**SPECIALE**

Vacanze **ad Ischia**

TripleHead**2go**

Gabriele Durante

Titoli **Caldi**

Gialuca De Fendi

# DALL'IDEA ALL'ALGORITMO

Maurizio Zuzzaro

**Algoritmo**

Il termine deriva da Mohammed Al Khwarizmi, un famoso matematico persiano da cui si ritiene essere uno dei primi autori ad aver fatto riferimento apertamente a questo concetto, nel libro Kitab al-djabr wa 'l-muqabala (libro sulla ricomposizione e sulla riduzione); dal quale tra l'altro prende anche le origini la parola algebra.



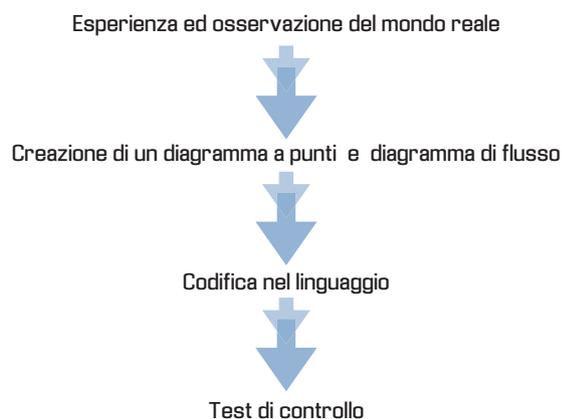
# Dall'IDEA

# all'ALGORITMO

“ Le idee di trading nascono dall'osservazione della realtà ”

**Se guardiamo un grafico, possiamo notare delle caratteristiche di comportamento che possono essere proprie di quel sottostante o comuni anche ad altri mercati. Possiamo osservare ad esempio come un titolo aumenti la volatilità in certi periodi o come ci sia un mercato correlato che ne anticipa i movimenti. E' da queste idee che si sviluppa tutto il processo di costruzione di un algoritmo.**

## SCHEMA DEL PROC. DI COSTR. DI UN SIST. DI TRADING

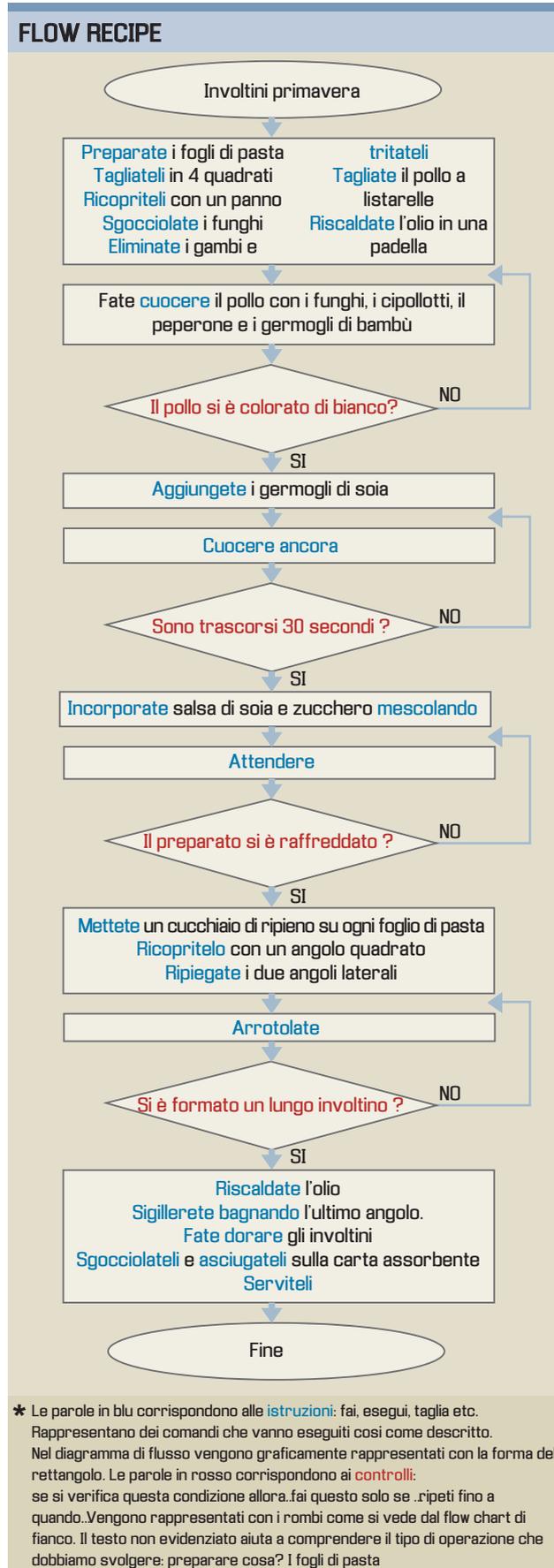


**P**er algoritmo intendiamo una sequenza ordinata e composta da azioni o istruzioni che generano un determinato risultato in un tempo finito. Lo schema sotto rappresenta il processo di costruzione di un sistema di trading e più generalmente di qualsiasi sistema esperto. Le idee possono ovviamente essere più o meno complesse ed integrate con altre. Possiamo avere ad esempio un sistema che contempli solo delle operazioni Long e non la vendita allo scoperto, oppure che utilizzi, a seconda del sottostante, dei filtri diversi o delle intere procedure. L'importante è che i concetti che dobbiamo esprimere in codice siano:

## CONCETTI ESPRIMIBILI IN CODICE

- Azioni eseguibili e non dubbie.**  
Non sono ammessi "un pò" e "a piacere".
- Determinismo.**  
Eseguito un passo, il successivo deve essere univoco, certo. Ovviamente sono possibili delle scelte ma non l'incertezza: "se il titolo è maggiore di ... allora ..." "se minore di .allora..." e non "forse"

Facciamo un esempio di come si costruisce un algoritmo partendo da una procedura reale e consolidata. Immaginiamo di voler preparare degli involtini primavera.



## \*Involtni primavera per 4 persone

**Preparate** i fogli di pasta, **tagliateli** in 4 quadrati e **ricopriteli** con un panno.

**Sgocciolate** i funghi, **eliminate** i gambi e **trititeli**. **Tagliate** il pollo a listarelle.

**Riscaldare** l'olio in una padella e fate **cuocere** il pollo con i funghi, i cipollotti, il peperone e i germogli di bambù **fino a quando** il pollo non si colora di bianco.

**Aggiungete** i germogli di soia e **cuocete ancora un 30 secondi**.

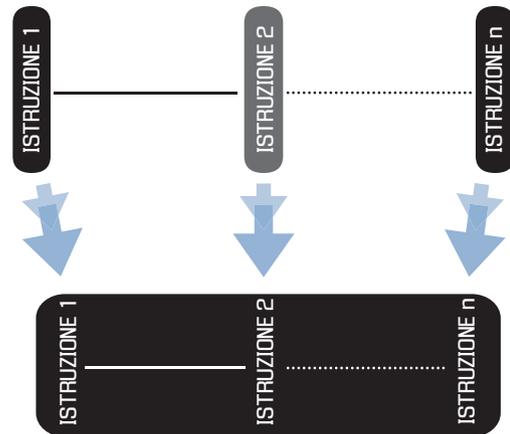
**Incorporate** salsa di soia e zucchero **mescolando**.

**Proseguite solo se** il preparato si è raffreddato.

**Mettete** un cucchiaino di ripieno su ogni foglio di pasta, **ricopritelo** con un angolo quadrato, **ripiegate** i due angoli laterali ed infine arrotolate **fino a formare** un lungo involtino che **sigillerete bagnando** l'ultimo angolo.

**Riscaldare** l'olio e **fate dorare** gli involtini, **sgocciolateli** e **asciugateli** sulla carta assorbente e **serviteli** subito.

E' stata migliorata la leggibilità del flusso raggruppando i blocchi di istruzioni in sequenza, in altre parole dove esiste un solo punto di ingresso ed uno in uscita.



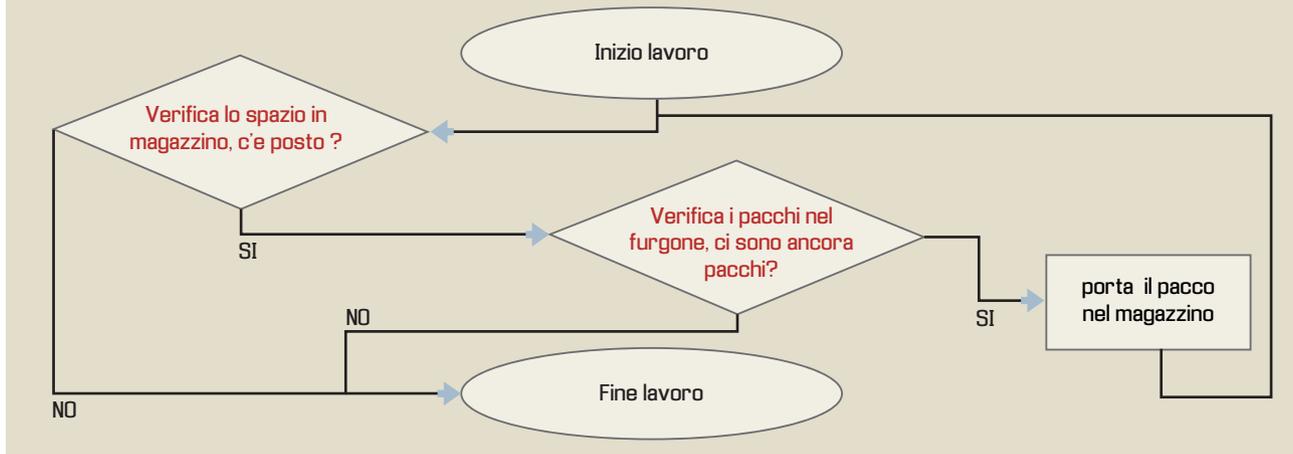
Supponiamo di dover scaricare un furgone pieno di pacchi e di doverli mettere in magazzino e terminare il lavoro quando o il magazzino è pieno o non abbiamo più pacchi sul furgone.

Le istruzioni sarebbero le seguenti:

- inizio lavoro
- verifica lo spazio in magazzino
- se ci sono ancora pacchi nel furgone prendili
- deposita il pacco in magazzino
- fine del lavoro

Una considerazione importante da fare è sulla semplicità delle istruzioni, "sgocciolare i funghi" può essere un passo valido per un algoritmo di cucina in quanto possiamo presupporre che chi lo andrà ad eseguire sa esat-

## ISTRUZIONI INIZIO LAVORO "MAGAZZINO"



tamente come lo si deve eseguire. Al contrario un'istruzione come "preparare un pentolino di mascarpone" o "guidare un'auto" può non essere eseguibile da tutti ma rimandando alla consultazione di ulteriori manuali.

Un passo di un algoritmo può essere definito anche tramite un altro algoritmo: in questo caso si parlerà di sottoalgoritmo o per usare un termine informatico subroutine. I sottoalgoritmi servono a suddividere un'istruzione in tante piccole fasi che la compongono.

Vediamo ad esempio come scomporre il problema "ritornare a casa dall'ufficio".

L'algoritmo "va dall'ufficio a casa" si compone delle seguenti istruzioni:

- 1 esci dal ufficio**
- 2 curva a sinistra**
- 3 prosegui per la via fino all'ultima rotonda**
- 4 gira a destra**

L'esempio che è stato riportato è sicuramente di semplice comprensione per un essere umano, ma nel caso di un automa richiederebbe di specificare ulteriormente i passaggi.

L'algoritmo "esci dal ufficio" si compone di:

- 1 controlla se la porta è aperta**
- 2 nel caso che la porta sia aperta salta il passo seguente**
- 3 apri la porta**
- 4 avanza fino al parcheggio**
- 5 prendi la macchina**
- 6 esci dal parcheggio**

L'algoritmo "prendi la macchina", compreso nel precedente, a sua volta si compone di:

- 1 prendi la chiave dell' auto**
- 2 inserisci la chiave nella serratura dell' auto**

- 3 afferra la maniglia**
- 4 toglì le chiavi dalla portiera**
- 5 entra in auto**
- 6 inserisci le chiavi ruotandole**
- 7 premi la frizione**
- 8 inserisci la marcia**

Un modo dettagliato di rappresentare l'algoritmo "va dal ufficio a casa" è allora il seguente:

- 1 esci dal ufficio**
  - 1.1 controlla se la porta è aperta**
  - 1.2 nel caso che la porta sia aperta salta il passo seguente**
  - 1.3 apri la porta**
    - 1.4 avanza fino al parcheggio**
  - 1.5 prendi la macchina**
    - 1.5.1 prendi la chiave dell' auto**
    - 1.5.1 inserisci la chiave nella serratura dell' auto**
    - 1.5.1 afferra la maniglia**
      - 1.5.1 toglì le chiavi dalla portiera**
    - 1.5.1 entra in auto**
    - 1.5.1 inserisci le chiavi ruotandole**
    - 1.5.1 premi la frizione**
    - 1.5.1 inserisci la marcia**
  - 1.6 esci dal parcheggio**
- 2 curva a sinistra**
- 3 prosegui per la via fino all'ultima rotonda**
- 4 gira a destra**

Proviamo a seguire questo metodo per cercare il valore massimo di una serie storica. Più precisamente vogliamo trovare il prezzo di chiusura più alto dei primi due mesi del 2006 del titolo Fiat; per farlo manualmente si scorre semplicemente la tabella e man mano che troviamo un valore più alto lo teniamo a mente..

## FLOW RECIPE

Data	Prezzo Chiusura
01/02/2006	7,47
01/03/2006	7,59
01/04/2006	7,59
01/05/2006	7,44
01/06/2006	7,52
01/09/2006	7,62
01/10/2006	7,86
01/11/2006	8,09
01/12/2006	8,04
01/13/2006	8,00
01/16/2006	8,25
01/17/2006	8,44
01/18/2006	8,39
01/19/2006	8,29
01/20/2006	7,84
01/23/2006	7,81
01/24/2006	8,08
01/25/2006	8,27
01/26/2006	8,18
01/27/2006	8,18
01/30/2006	8,21
01/31/2006	8,07
02/01/2006	8,17
02/02/2006	8,12
02/03/2006	8,13
02/06/2006	8,17
02/07/2006	8,15
02/08/2006	8,15
02/09/2006	8,24
02/10/2006	8,24
02/13/2006	8,42
02/14/2006	8,33
02/15/2006	8,35
02/16/2006	8,37
02/17/2006	8,60
02/20/2006	8,62
02/21/2006	8,63
02/22/2006	8,75
02/23/2006	8,69
02/24/2006	8,98
02/27/2006	9,30
02/28/2006	9,08

## Inizio procedura

## Ricerca del prezzo maggiore

Prima riga contenente i valori è sulla tabella la 2°; R = 2

## Posizionati sulla riga R

Leggi il valore della data corrente; DataCorrente = colonna 1, riga R

Leggi il valore della prezzo corrente; PrezzoCorrente = colonna 2, riga R

Se il valore della data corrente è maggiore, memorizza il valore, IF PrezzoCorrente > Valore Max THEN ValoreMax = PrezzoCorrente

Se la data attuale è precedente il 02/28/2006 passa al giorno successivo; IF DataCorrente < 02/28/2006 THEN R = R + 1 PUNTO 3

## Fine procedura

Possiamo facilmente testare la procedura utilizzando la tabella per la lettura dei valori, e seguendo il percorso logico dell' algoritmo fino alla fine. Se la procedura è corretta il valore di **ValoreMax** è di 9,30.

Alla fine della prova avremo trovato il valore più grande. In apparenza questo sistema può sembrare complicato, in realtà è più o meno quello che accade nel nostro cervello in modo inconscio ed automatico.

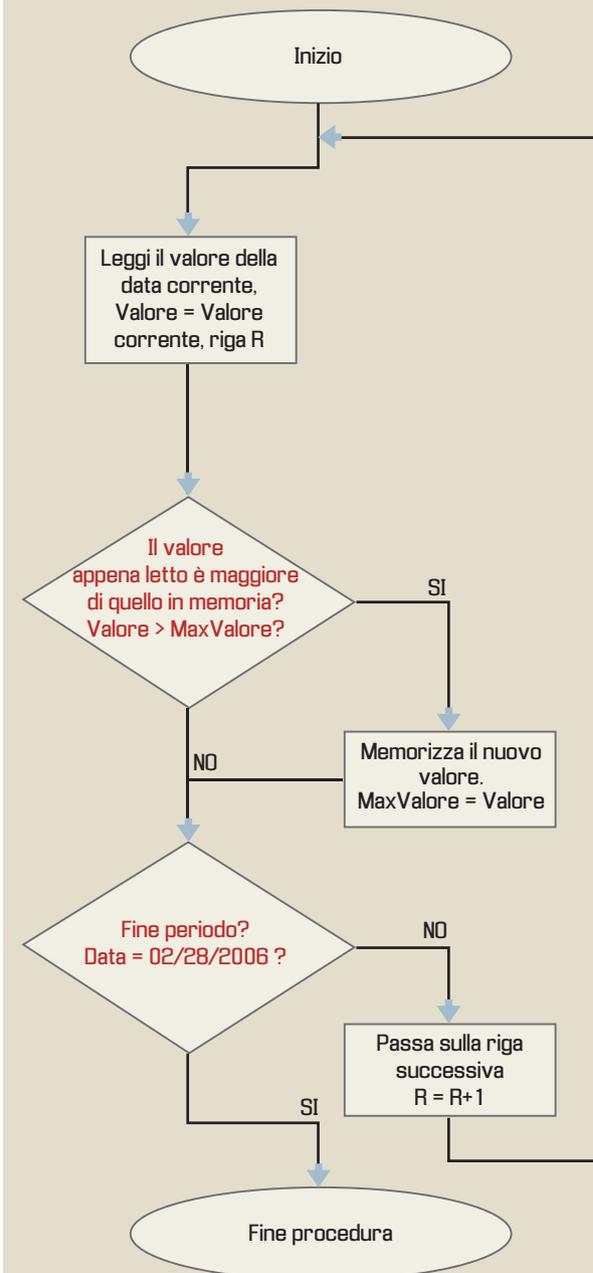
Negli esempi finora esposti, abbiamo visto condizioni da cui scaturiva una sola sequenza di eventi: se è vera questa condizione allora esegui...se è falsa, passa al primo nodo seguendo il flusso della struttura principale.

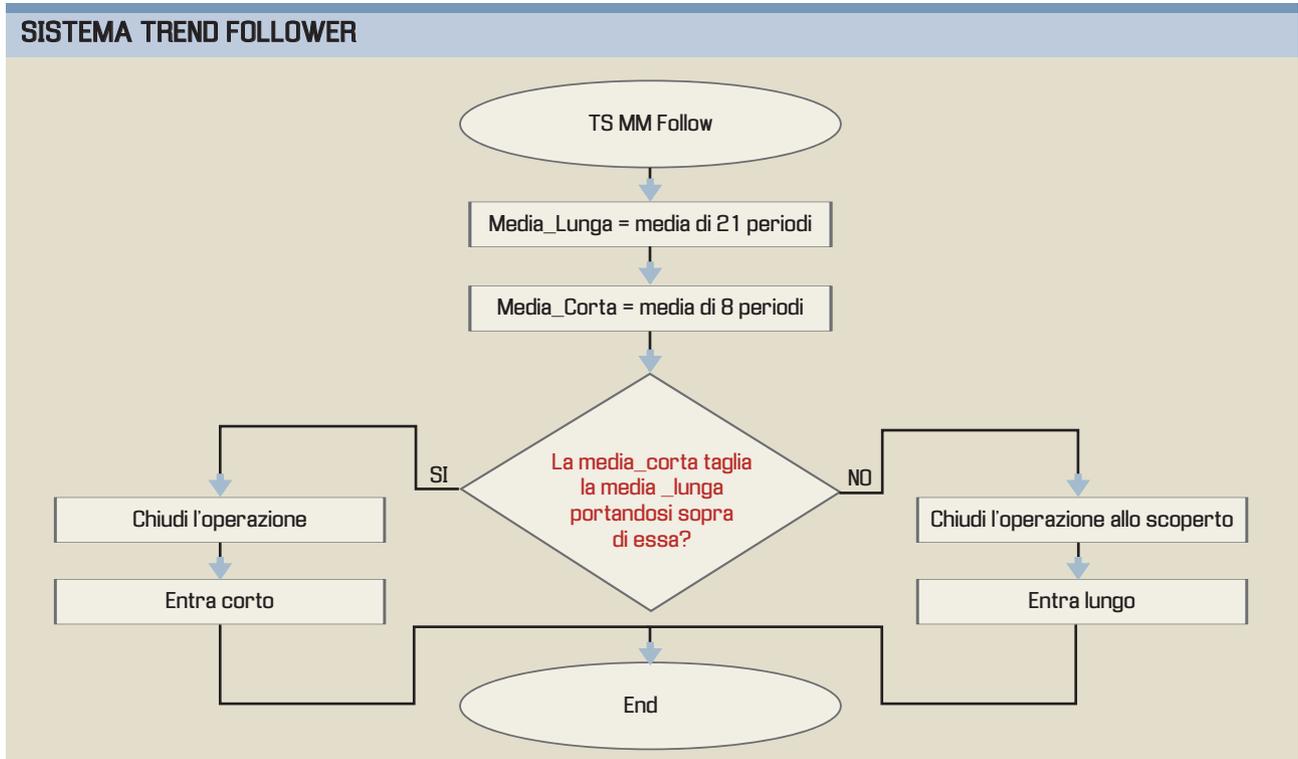
Un blocco condizionale da

cui nascono due procedure distinte è riscontrabile ad esempio trovare nella costruzione di un sistema trend follower basato su medie mobili.

Se due medie mobili, una a 21 periodi e l'altra a 8, si incrociano, e la media a 8 passa sopra quella a 21, entriamo sul mercato (lunghi); viceversa apriamo un'operazione allo scoperto. In entrambe le condizioni le posizioni aperte vanno chiuse.

## STORICA PREZZI





## SINTASSI E SEMANTICA DEI LINGUAGGI DI PROGRAMMAZIONE

Tutti i linguaggi, inclusi quindi quelli per la programmazione software, si compongono di due caratteristiche: la sintassi e la semantica. La sintassi definisce la struttura (o la forma) del relativo linguaggio, mentre la semantica definisce il significato delle strutture sintattiche cosiddette "well-formed", ovvero formalmente corrette. Ogni linguaggio di programmazione ha le sue caratteristiche e le sue regole, ma la scrittura del codice in un linguaggio è solo una delle fasi terminali nella costruzione di un Trading System. L'importante non è conoscere un linguaggio di programmazione piuttosto che un'altro, ma scegliere il linguaggio giusto per i nostri scopi. È chiaro che la maggior padronanza di un linguaggio rispetto ad un altro è da tenere in considerazione ai fini della tempistica di sviluppo.

È proprio dalla necessità di abbassare i tempi di sviluppo che sono nati linguaggi studiati ad hoc per l'utilizzo in campo speculativo finanziario come *Easy Language* della *Omega Research* e quelli generici come Java utilizzato anche dalla piattaforma di trading e-signal. Anche senza voler ora addentrarci nelle specifiche di alcun linguaggio, possiamo in ogni caso iniziare a dare delle generiche linee guida. Per chiarire meglio vediamo un esempio: se vogliamo assegnare ad una variabile il valore di 5 per alcuni linguaggi la sintassi sarà:  $a=5$ , per altri  $a:=5$ . Per i matematici, la sottile differenza tra i due

tipi di notazioni è un abisso dal punto di vista logico. In ogni caso tutti i linguaggi di programmazione contengono questi concetti base: variabili, istruzioni, condizioni, espressioni.

### Variabili

Le variabili sono semplicemente dei contenitori di informazioni, come dei cassetti di un armadio a cui è posta una etichetta per ritrovarne il contenuto.

### Operatori

Gli operatori permettono la connessione tra due parole o numeri. La connessione può essere di tipo matematico o di tipo relazionale.

### Condizioni

Servono a creare le scelte all'interno delle procedure logiche. Rispondono all'esigenza di eseguire uno o più comandi se si verifica una determinata situazione. Vengono rappresentate nei diagrammi di flusso con la forma del rombo.

### Parole riservate (o token)

Qualsiasi linguaggio ha delle parole che hanno dei significati particolari e non utilizzabili per altri scopi.

### Istruzioni

L'istruzione è un'operazione che l'interprete esegue in

maniera atomica. Le istruzioni possono essere o delle espressioni o più semplicemente comandi che vengono eseguiti immediatamente, come ad esempio il lancio di una stampa o la registrazione sul disco di un file. Le istruzioni vengono rappresentate nei diagrammi di flusso con la forma dei rettangoli.

### Espressioni

Un' espressione è una combinazione di valori, variabili e operatori. Come ad esempio l'assegnazione di un valore ad una variabile o la differenza tra due matrici. Anche le espressioni sono rappresentate con la forma del rettangolo.

### I separatori

Ogni istruzione, espressione, condizione è separata dalle altre (a seconda dei casi e dal linguaggio) da caratteri speciali come parentesi, virgole, punti e virgole etc..

## IL CONCETTO DI VARIABILE E TIPO DI DATO

**E** sistono vari tipi di variabili a seconda del contenuto che devono memorizzare le più note sono:

### Integer

contengono dei numeri interi. Possono essere utilizzati ad esempio per memorizzare i giorni trascorsi dalla rottura di un supporto ordine o per contare il numero di trade.

### Float

numeri in virgola mobile. Per memorizzare i prezzi è sicuramente consigliabile utilizzare questo tipo di variabile e mai quella di tipo integer che troncherebbe le cifre decimali.

### Boolean

possono assumere solo due valori come: acceso o spento, 0 o 1, vero o falso etc. Un impiego potrebbe essere quello di registrare se abbiamo delle posizioni aperte. Quindi la variabile booleana posizioni\_aperte = vero.

### Stringa

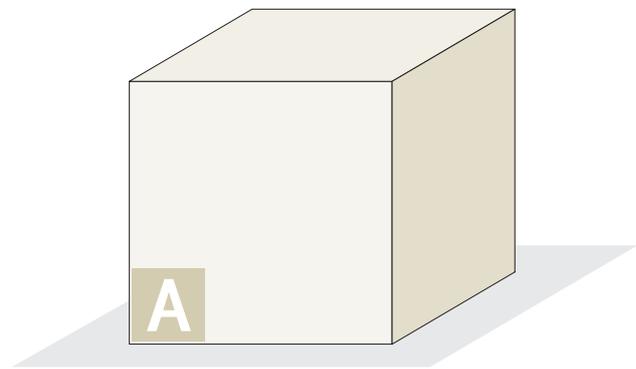
valori alfanumerici in pratica del testo. L'utilizzo di questo tipo di variabili è molto ampio, l'importante è che non vengano scelte per memorizzare dei valori che ci servono per fare dei calcoli di tipo matematico. Infatti se memorizziamo all'interno delle variabili stringa il prezzo di apertura e di chiusura e facciamo una somma tra di loro otteniamo la somma dei

caratteri.

- apertura= "1,20"
- chiusura= "1,25"
- somma= apertura + chiusura
- somma da come risultato: "1,20+1,25"

La maggior parte dei linguaggi di programmazione richiede che venga dichiarata obbligatoriamente il tipo di variabile prima dell' utilizzo. Questo tipo di approccio consente all' elaboratore di ottimizzare la memoria. In VBA (Visual Basic for Applications) e Visual Basic questa condizione è opzionale.

## RAPPRESENTAZIONE DI UNA VARIABILE A



### Array

Gli array o vettori sono semplicemente una serie di variabili indicizzate, hanno in comune tutte lo stesso nome e vengono identificate da un indice.

### MioTitolo(i)

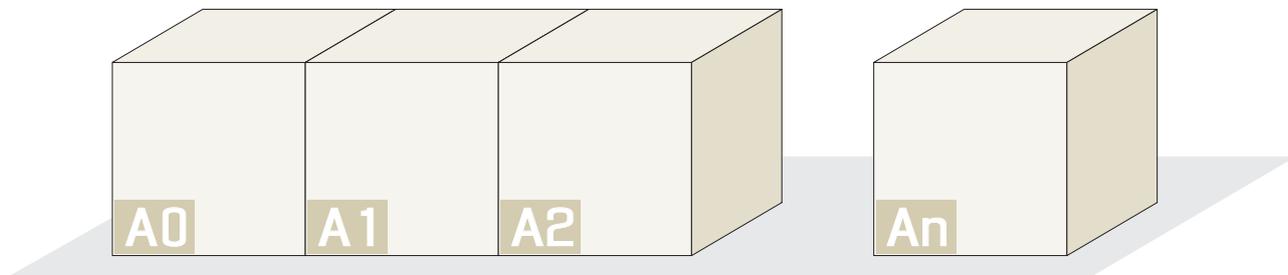
dove **i** è l'indice di sessione

E' come assegnare un nome a tutta una cassetta e distinguere i cassetti solo dal numero. Un esempio di utilizzo potrebbe essere la memorizzazione delle serie storiche dei dati di borsa, i dati sono tutti dello stesso tipo, quello che cambia è solo il periodo.

Ad esempio se vogliamo memorizzare i prezzi di chiusura di un titolo di una settimana avremo ad esempio:

- MioTitolo(0) = 1,20
- MioTitolo(1) = 1,23
- MioTitolo(2) = 1,25
- MioTitolo(3) = 1,23
- MioTitolo(4) = 1,24
- MioTitolo(5) = 1,23
- MioTitolo(6) = 1,25

## RAPPRESENTAZIONE DI UN VETTORE A



### Matrici

Le matrici si differenziano dai vettori, perché mentre quest'ultimi hanno solo un indice, le matrici ne possono avere più d'uno e si estendono quindi per più dimensioni. Potremmo ad esempio creare la matrice

#### MioTitolo(i,a,c,min,max,v)

- dove **i** è l'indice di sessione
- **a** è il prezzo di apertura
- **c** quello di chiusura
- **min** è il minimo di giornata
- **max** è il massimo di giornata
- **v** il volume

### OPERATORI ARITMETICI, LOGICI E RELAZIONALI

La simbologia utilizzata può cambiare a seconda del linguaggio di programmazione. Si rimanda quindi ad esempi più specifici per una maggiore precisione.

### Operatori aritmetici

Gli operatori aritmetici consentono di effettuare delle operazioni aritmetiche che comportano il calcolo di valori numerici rappresentati da valori letterali, variabili, altre espressioni.

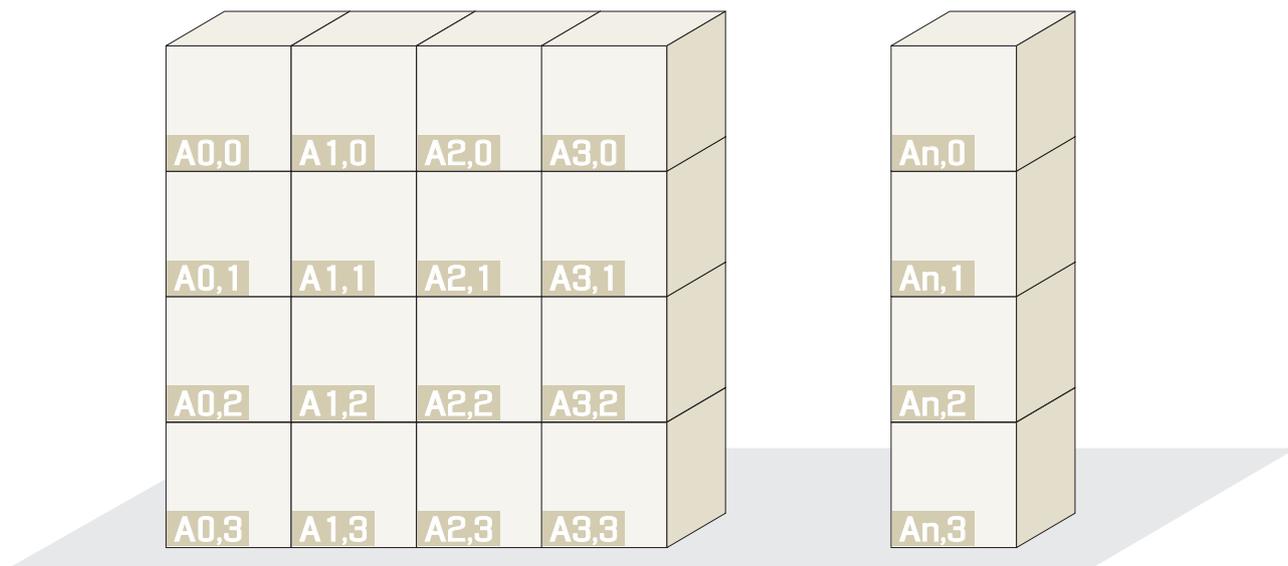
#### OPERATORI ARITMETICI

+	addizione
-	sottrazione
*	moltiplicazione
/	divisione
+	+ unario
-	- unario
=	assegna il valore destro alla variabile sinistra)
--	decremento in C++
++	incremento in C++

#### Esempi

- a = 4+6; assegna alla variabile A il valore di 10
- a = c\* 6-b; assegna alla variabile a il valore di c moltiplicato per 6 meno la variabile b

## RAPPRESENTAZIONE DI UNA MATRICE BIDIMENSIONALE A



## Operatori relazionali

Gli operatori relazionali si usano per confrontare i valori di due espressioni. Il risultato di un operatore relazionale è di tipo booleano e può assumere soltanto uno dei due valori true o false, a seconda del risultato del confronto.

OPERATORI RELAZIONALI	
== , =	Uguale
!= , <>	Diverso
>	Maggiore
<	Minore
>=	Maggiore o uguale
<=	Minore o uguale

### Esempi

7 == 3 , 7 = 3  
risultato false  
5 > 2  
risultato true  
3 = 1  
risultato true  
5 >= 5  
risultato true  
4 < 4  
risultato false

Ovviamente possiamo usare qualunque espressione valida, comprese le variabili.

Supponiamo che a=3, b=4 e c=12  
a == 5  
risultato false.  
a\*b >= c  
risultato true in quanto (2\*4 >= 12).  
(b+4 > a\*c)  
risultato false in quanto (4+4 > 3\*6).

## Operatori logici o booleani

Gli operatori logici o booleani prendono il loro nome da George Boole, matematico inglese della prima metà dell'800 che formalizzò la logica binaria che sta alla base dei moderni calcolatori.

OPERATORI LOGICI O BOOLEANI	
AND	e
OR	oppure
NOT	negazione

### Esempi

Supponiamo che a=3, b=4:

a = 3 and b = 3

Risulta falso in quanto tutte e due le espressioni devono essere vere, mentre in questo caso è vera solo la prima:

a = 3 or b = 3

Il risultato è vero perché in questo caso è sufficiente che sia vera solo una delle due:

not (a = 3 or b = 3)

Risulta falso in quanto nega il valore dell'espressione

## Priorità degli operatori

Quando scriviamo espressioni complicate con molti operatori e operandi possono creare dei dubbi sull'ordine in cui gli operatori vengono valutati. Ad esempio, con l'espressione:

a = 2 + 7 \* 2

può sorgere il dubbio tra le seguenti due interpretazioni:

a = 2 + (7 \* 2) con risultato 16

a = (2 + 7) \* 2 con risultato 18

In realtà gli operatori con la stessa precedenza sono valutati da sinistra a destra anche se per una maggiore chiarezza nella lettura del codice è meglio usare le parentesi.

Continua...



# MASTER IN SCALPING DINAMICO ad ISCHIA

Con **Leonardo Gioacchini**

Un'occasione unica per partecipare alla versione estiva (Settembre) del **Master in Scalping Dinamico**: un vero e proprio ritiro di 4 giorni ad Ischia in un'albergo dotato di tutti i confort, dove sarà possibile istituire un clima ideale ad apprendere efficacemente la metodologia.

Il costo del master è di euro **1450 + Iva tutto compreso** (albergo 4 stelle in mezza pensione, seminario, dispense, 4 mesi di e-learning successivi al master): sarà possibile portare gratuitamente un'altro accompagnatore/trice esterna non partecipante al master.

info e prenotazioni | [scalpingweb@yahoo.it](mailto:scalpingweb@yahoo.it)