Programmazione I

A.A. 2002-03

Array

(Lezione XVIII , parte I)

Ciclo "for"

Prof. Giovanni Gallo
Dr. Gianluca Cincotti
Dipartimento di Matematica e Informatica
Università di Catania
e-mail : {gallo, cincotti}@dmi.unict.it

Ciclo "for"

- I tre *task* tipici di un ciclo controllato dal variare del valore di una variabile:
  - *inizializzazione variabile contatore,*
  - *espressione condizionale,*
  - *aggiornamento variabile e altre operazioni nel body.*
- L'istruzione **for** include tutte queste operazioni nello stesso costrutto:

```
for ( inizializzazione;  condizione;  iterazione )
    {
       istruzione;
    }
```

Ciclo "for" (cont.)

- La sintassi dell'istruzione *for:*

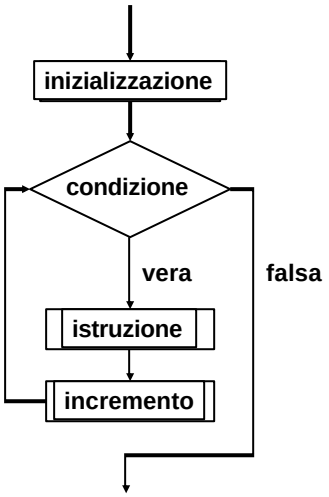
L'istruzione viene eseguita fino a che *condizione* diventa falsa

Espressione di *inizializzazione* è eseguita una volta prima di entrare nel ciclo

```
for ( inizializzazione; condizione; incremento )
   istruzione;
```

Espressione di *incremento* viene eseguita alla fine di ciascuna iterazione

Semantica del ciclo "for"



Quando utilizzare il "for"?

- La condizione di un ciclo *for* viene valutata prima di eseguire il ciclo, come nel ciclo *while*.
  - Il corpo del ciclo *for* può essere eseguito zero o più volte.

- Questo ciclo è indicato per eseguire istruzioni un numero di volte specifico che può essere *determinato a priori*

Equivalenza tra cicli

- Un ciclo *for* è equivalente al ciclo *while*:

```
inizializzazione;
while ( condizione )
  {
    istruzione;
    incremento;
  }
```

"for" vs. "while"

```
for ( i=0 ; i<10 ; i++ )
    faiQualcosa();
```

```
i=0;
while (i<10)
    {
        faiQualcosa();
        i++;
    }
```


Esempi

Es. 01

Es. 02

Es. 03

Es. 04

Espressioni mancanti

- Ogni espressione nella dichiarazione di un ciclo *for* è opzionale.
  - Se manca l'espressione di inizializzazione, allora nessuna inizializzazione viene effettuata.
  - Se manca l'espressione della condizione, allora si considera che sia sempre vera, e si realizza un ciclo infinito.
  - Se manca l'espressione di incremento, allora non si esegue nessun incremento.
- Ma i 2 caratteri "**;**" sono sempre necessari anche quando mancano le espressioni corrispondenti.

Ciclo infinito

- La forma più semplice di **for** è:

```
for ( ; ; )
   {
      istruzione;
   }
```

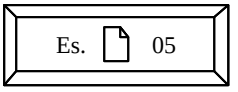
- Realizza un ciclo infinito perché la condizione mancante viene interpretata come vera.

Varianti del ciclo "for"

- Il "for" consente di utilizzare zero *o* *più* variabili di controllo, e quindi:
  - zero *o più* inizializzazioni,
  - una condizione booleana complessa,
  - zero *o più* istruzioni di assegnamento.

Esempio con due variabili

```
for ( i=0, j=5 ;   (i<10 && j>0) ;   i++, j-- )
    {
      faiQualcosa();
    }
```



Il ciclo descritto utilizza **due** variabili di controllo con **due** operazioni di assegnamento distinte.

Sia l'inizializzazione delle variabili di controllo che le operazioni di assegnamento utilizzano il carattere "," come separatore.

Quale ciclo utilizzare ?

***Definizione di ciclo (loop)**:*

Sequenza di istruzioni eseguita in modo ripetitivo sino a quando non si verifica una condizione di fine.

- ***DETERMINATI (for)***
- ***INDETERMINATI (while, do-while)***

Es. 06 Es. 07

Repetita juvant…

- La differenza fra **while** e **do-while** consiste nel fatto che il corpo del ciclo nel **do-while** viene sempre eseguito almeno una volta (la prima).
  - Nel **while** invece se la condizione booleana è falsa il corpo del ciclo non viene mai eseguito.
- Il ciclo **for**, viene in pratica utilizzato quando un dato blocco di istruzioni (il corpo del ciclo) deve essere ripetuto (*iterazione*) un determinato numero di volte noto a *priori*.

Programmazione I

A.A. 2002-03

Array

(Lezione XVIII , parte II)

Array multidimensionali

Prof. Giovanni Gallo
Dr. Gianluca Cincotti
Dipartimento di Matematica e Informatica
Università di Catania
e-mail : {gallo, cincotti}@dmi.unict.it

Array di due dimensioni

- Un array ad *1 dimensione* archivia una *lista di valori*.
- Un array a 2 *dimensioni* costituisce un tabella di valori organizzata per righe e colonne
  - Gli elementi vengono referenziati con due indici:

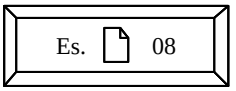
    A [2] [1];
  - Si può utilizzare una lista di inizializzazione.

Esempio

```
int [] []  tabella  =  new  int [3] [4];
     //Adesso posso accedere agli elementi
     //dell'array bidimensionale come segue:
tabella[2][2] = 5;
for (i=0;   i<3;   i++)
   for (j=0;   j<4;   j++)
         System.out.print (tabella[i][j]+",");
```

Una tabella di 3 righe e 4 colonne di interi

Output:
0,0,0,0,0,0,0,0,0,0,5,0,



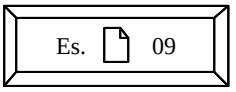
Array multidimensionali

- Un array può avere anche più di due dimensioni.
- Esempio:

```
int  [] [] [] cubo  =  new  int [3] [4] [2];
cubo [1][1][1] = 47;
```

Array frastagliati

- Sono array bidimensionali in cui le righe non hanno tutte lo stesso numero di colonne.
  - Permettono di risparmiare memoria !
  - Un array a 2 *dimensioni* in Java è un array di array …
    - In particolare, un array di riferimenti ad array …
      - Dunque, ognuno di questi ultimi array potrà avere lunghezza diversa !
        - L'attributo "length" si rivela molto utile.



Fine