

**ISTITUTO TECNICO INDUSTRIALE
G. M. ANGIOY
SASSARI**



CORSO DI PROGRAMMAZIONE

TIPI VALORE E TIPI RIFERIMENTO

DISPENSA 05.03

05-03_Tipi_Rif_e_Val_[ver_15]



Questa dispensa è rilasciata sotto la licenza Creative Common CC BY-NC-SA. Chiunque può copiare, distribuire, modificare, creare opere derivate dall'originale, ma non a scopi commerciali, a condizione che venga riconosciuta la paternità dell'opera all'autore e che alla nuova opera vengano attribuite le stesse licenze dell'originale.

Versione del: **07/11/2015**
Revisione numero: **15**

Prof. Andrea Zoccheddu
Dipartimento di Informatica

DIPARTIMENTO INFORMATICA E TELECOMUNICAZIONI





TIPI RIFERIMENTO E TIPI VALORE

DIFERENZE TRA I TIPI RIFERIMENTO E I TIPI VALORE

CONCETTO DI ISTANZA

PROGETTO GUIDATO

- Si prepari un Form1 simile al seguente:
- Preparare le variabili globali seguenti

```
int numero ;
int [] numeri ;
```

- Doppio clic su **intero** e associare il seguente codice:

```
numero = 13;
listBox1.Items.Add(numero);
```

- Doppio clic su **vettore** e associare il seguente codice:

```
numeri[0] = 13;
listBox2.Items.Add(numeri[0]);
```

- Prova il progetto coi soli due primi pulsanti; il primo funziona ma il secondo mostra un errore di questo tipo:

L'errore è un errore a tempo di esecuzione, ovvero non c'è un errore nel programma che il compilatore possa evitare

il programmatore distratto non ha tenuto conto che l'uso del vettore era vietato se prima non viene allocata la memoria.

Il vettore, come suggerisce il messaggio, è un tipo riferimento; invece il numero intero è un tipo valore.

- Arresta l'esecuzione del programma col Menu **Debug | Termina debug**.

- Associa al pulsante **new intero** il codice seguente:

```
numero = new int();
numero = 17;
listBox1.Items.Add(numero);
```

- Associa al pulsante **new array** il codice seguente:

```
numeri = new int[100];
numeri[0] = 17;
listBox2.Items.Add(numero);
```

- Prova il progetto usando solo i secondi due pulsanti; nessun errore questa volta!





RIFLESSIONI SUI TIPI RIFERIMENTO ED SUI TIPI VALORE

In Visual C# ci sono due categorie diverse di tipi: i tipi Riferimento ed i tipi Valore. Un **tipo valore** di dati contiene direttamente i dati nella propria locazione di memoria. Un **tipo riferimento** invece contiene solo un riferimento (puntatore) a un'altra posizione di memoria che contiene realmente i dati.



Questo implica che, quando si dichiara una locazione di un tipo valore (per es. si dichiara una variabile intera) la si può usare immediatamente, poiché il suo spazio dei dati è già pronto. Quindi posso fare:

```
int i;
i = 13; //Nessun errore
```

Viceversa quando si dichiara un tipo riferimento, si dispone del solo spazio che contiene l'eventuale indirizzo che inizialmente è un indirizzo nullo. Per questo il tentativo di usarlo solleva un errore:

```
int [] v ;
v [0] = 17; //Errore!
```

Per poter usare la locazione di tipo riferimento è necessario prima allocare la memoria, ovvero farne richiesta al sistema operativo che trova la memoria, la prepara per la locazione ed infine aggancia il riferimento alla variabile. Quindi si deve fare così:

```
int [] v ;
v = new int[10];
v [0] = 17; //Nessun
errore
```

ESEMPI DI TIPI VALORE

Fra i tipi di valore sono inclusi:

- Tutti i tipi di dati numerici
- Boolean, Char e Date
- Tutte le strutture, anche se i relativi membri sono tipi riferimento
- Le enumerazioni (poiché il loro tipo sottostante è sempre numerico)

ESEMPI DI TIPI RIFERIMENTO

Fra i tipi di riferimento sono inclusi:

- Tutti i vettori e le matrici, anche se i relativi elementi sono tipo valore
- Il tipo String, anche se in alcuni casi si comporta come un tipo valore (per esempio non necessita della new)
- I tipi oggetto e di classe, come ad esempio Random, Form, Button, Label, TextBox, ecc...

RIEPILOGO DI TIPI E AZIONI

PROGETTO GUIDATO

- Si prepari un Form1 simile al seguente:
- Dichiarare le variabili globali seguenti

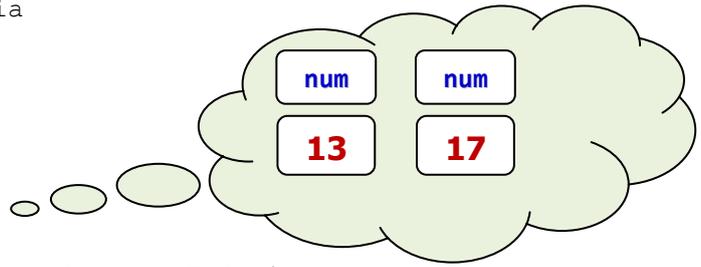
```
int numa, numb ;
string stringa, stringb ;
int [] rifa, rifb ;
```





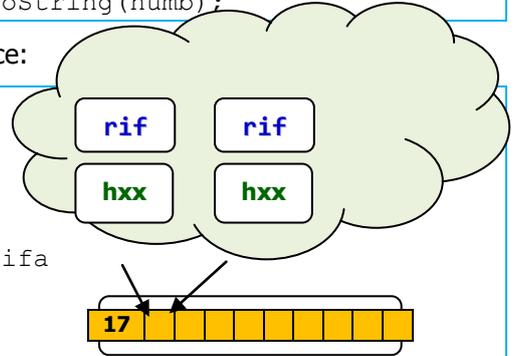
- Doppio clic su **Tipo Valore** e associare il seguente codice:

```
//posso usarla senza allocare memoria
numa = 13;
//l'assegnazione copia il valore
numb = numa; //numb prende 13
//ciascuno ha spazi separati
numb = 17; //non modifica numa
//visualizzazione
Text = Convert.ToString(numa) + "; " + Convert.ToString(numb);
```



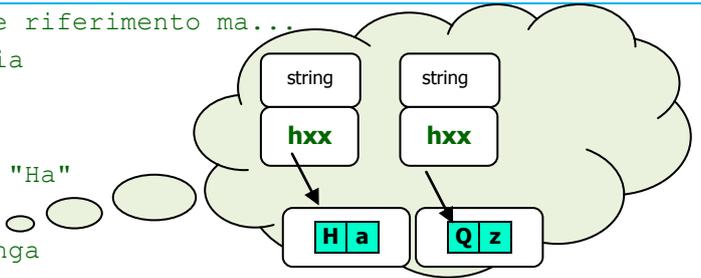
- Doppio clic su **Tipo Riferimento** e associare il seguente codice:

```
//prima di usarla devo allocare memoria
rifa = new int[100];
rifa[0] = 13;
//l'assegnazione copia l'indirizzo ma non i dati
rifb = rifa; // numb fa riferimento ai dati di rifa
//possono avere spazi condivisi
rifb[0] = 17; // modifica anche numa
//visualizzazione
Text = Convert.ToString(rifa[0]) + "; " + Convert.ToString(rifb[0]);
```



- Doppio clic su **Tipo Stringa** e associare il seguente codice:

```
//le stringhe sono classificate come riferimento ma...
//posso usarla senza allocare memoria
stringa = "Ha";
//l'assegnazione copia il valore
stringb = stringa; //stringb prende "Ha"
//ciascuno ha spazi separati
stringb = "Qz"; //non modifica stringa
//visualizzazione
Text = stringa + "; " + stringb;
```



RIFLESSIONE SUGLI SPAZI DELLE DICHIARAZIONI

PROGETTO GUIDATO SULLE DICHIARAZIONI

- Si prepari un Form1 simile alla figura:
- Si preparino le variabili globali seguenti:

```
string[] g1, g2;
```

- Doppio clic su **Assegnazione tra globali** e associare il seguente codice:

```
g1 = new string[10];
g1[0] = "globale";
g2 = g1;
textBox1.Text = g2[0];
```





- Doppio clic su **Assegnazione tra locali** e associare il seguente codice:

```
string[] l1, l2;  
l1 = new string[10];  
l1[0] = "locale";  
l2 = l1;  
textBox2.Text = l2[0];
```

- Provare il programma

PROGETTO GUIDATO

- Si prepari un Form1 simile alla figura:
- Si preparino le variabili globali seguenti:

```
TextBox[] vettore = new TextBox[8];
```



- Doppio clic su **button1** e associare il seguente codice:

```
vettore[0] = textBox1;      vettore[1] = textBox2;  
vettore[2] = textBox3;      vettore[3] = textBox4;  
vettore[4] = textBox5;      vettore[5] = textBox6;  
vettore[6] = textBox7;      vettore[7] = textBox8;
```

- Doppio clic su **button2** e associare il seguente codice:

```
for (int i = 0; i < 8; i++)  
    vettore[i].Text = "ciao";
```

- Doppio clic su **button3** e associare il seguente codice:

```
for (int i = 0; i < 8; i++)  
    vettore[i].Text = "casella" + i;
```

- Doppio clic su **button4** e associare il seguente codice:

```
for (int i = 0; i < 8; i++)  
    vettore[i].Visible = ! vettore[i].Visible;
```

- Provare il programma



ESERCIZI

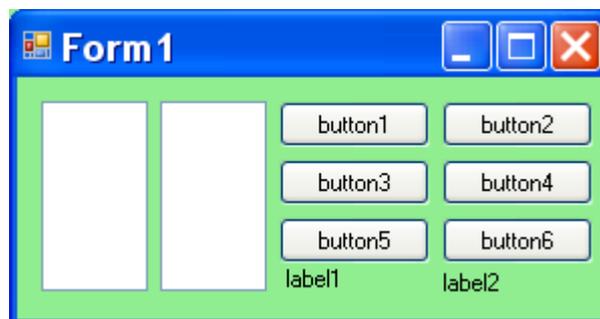
ESERCIZI SUI TIPI RIFERIMENTO

ESERCIZIO 1. ASSEGNAZIONE TRA VETTORI

➤ Dichiarare due vettori di interi.

Pulsante1 : genera 100 numeri casuali nel primo vettore; **assegna** il primo vettore al secondo; incrementa i contenuti delle celle del secondo di +1; infine visualizza i vettori nelle due ListBox

Pulsante2 : genera 100 numeri casuali nel primo vettore; **copia** il primo vettore nel secondo; incrementa i contenuti delle celle del secondo di +1; infine visualizza i vettori nelle due ListBox

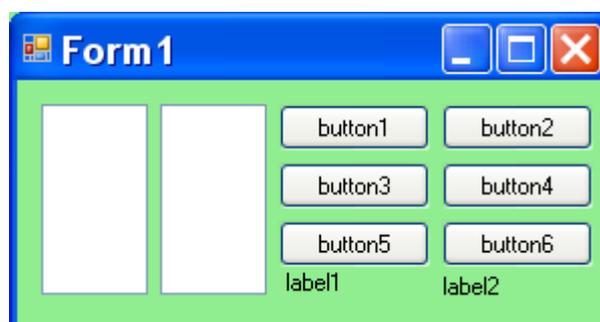


ESERCIZIO 2. ASSEGNAZIONE TRA VETTORI

➤ Dichiarare due vettori di stringhe.

Pulsante1 : genera stringhe casuali (usare i nomi di 4 colori preferiti) nel primo vettore; **assegna** il primo vettore al secondo; incrementa i contenuti delle celle del secondo di +1; infine visualizza i vettori nelle due ListBox

Pulsante2 : genera stringhe casuali (usare i nomi di 4 colori preferiti) nel primo vettore; **copia** il primo vettore nel secondo; incrementa i contenuti delle celle del secondo di +1; infine visualizza i vettori nelle due ListBox



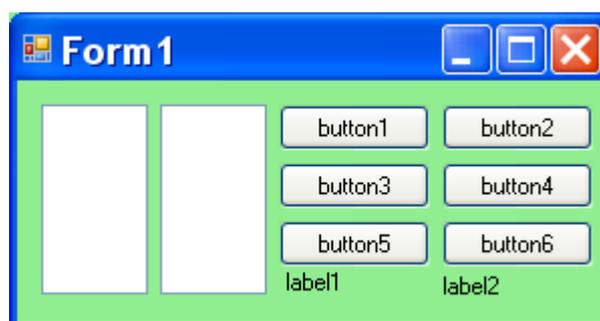
ESERCIZIO 3. ASSEGNAZIONE TRA VETTORI

➤ Dichiarare due vettori di interi.

Pulsante1 : genera 100 valori casuali nei due vettori e li visualizza nelle rispettive ListBox;

Pulsante2 : scambio (vedi algoritmo dello scambio) tra i due vettori **usando un vettore locale di appoggio**; verificare se funziona;

Pulsante3 : scambio (vedi algoritmo dello scambio) tra i due vettori **usando un intero locale di appoggio**; verificare se funziona;



**ESERCIZIO 4. ASSEGNAZIONE TRA VETTORI**

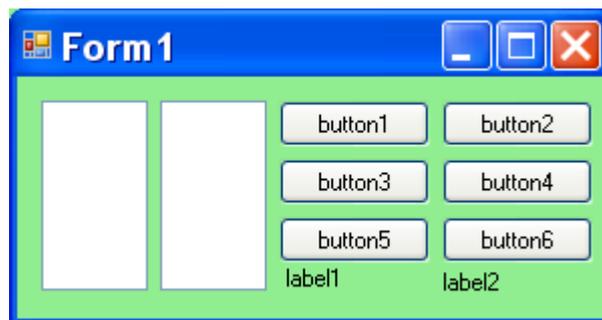
• Dichiarare un vettore di Button.

Pulsante1 : alloca memoria per 6 elementi;

Pulsante2 : assegna a ciascun elemento del vettore uno dei Button del Form1; usare sei assegnazioni;

Pulsante3 : con un ciclo for, modificare il titolo dei sei elementi del vettore, impostandolo con il numero del ciclo (usare la conversione da intero a stringa);

altri pulsanti: eseguire cicli sulle proprietà dei pulsanti





SOMMARIO

DIFFERENZE TRA I TIPI RIFERIMENTO E I TIPI VALORE	2
CONCETTO DI ISTANZA	2
Progetto guidato.....	2
Riflessioni sui tipi Riferimento ed sui tipi Valore	3
Esempi di Tipi valore.....	3
Esempi di Tipi riferimento.....	3
RIEPILOGO DI TIPI E AZIONI.....	3
Progetto guidato.....	3
RIFLESSIONE SUGLI SPAZI DELLE DICHIARAZIONI	4
Progetto guidato sulle dichiarazioni	4
Progetto guidato.....	5
ESERCIZI SUI TIPI RIFERIMENTO.....	6
Esercizio 1. Assegnazione tra vettori	6
Esercizio 2. Assegnazione tra vettori	6
Esercizio 3. Assegnazione tra vettori	6
Esercizio 4. Assegnazione tra vettori	7