

# **CORSO DI PROGRAMMAZIONE**

## **ALTRI TIPI DI DATO INTERESSANTI**

### **DISPENSA 01.05**

[01-05\\_Tipi\\_Ulteriori\\_\[ver\\_15\]](#)



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 DI DATO ULTERIORI

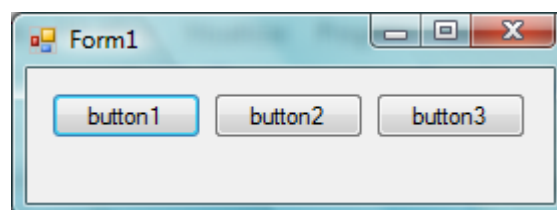
### ALCUNI TIPI DI DATO UTILI E DIVERTENTI

#### IL TIPO DEI COLORI

##### PROGETTO GUIDATO

- Prepara un form1 simile alla figura
- Associa al pulsante **button1** il seguente gestore di evento:

```
BackColor = Color.Coral;  
button1.BackColor = Color.Khaki;  
button2.BackColor = Color.LemonChiffon;  
button3.BackColor = Color.Plum;
```



- Associa al pulsante **button2** il seguente gestore di evento:

```
Color tmp = button1.BackColor;  
button1.BackColor = button2.BackColor;  
button2.BackColor = button3.BackColor;  
button3.BackColor = tmp;
```

- Associa al pulsante **button3** il seguente gestore di evento:

```
BackColor = Color.FromArgb(128, 111, 200);
```

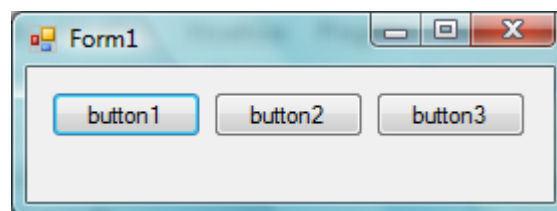
- Prova a eseguire il progetto

L'esercizio guidato usa i colori in modi diversi. Il primo pulsante assegna colori costanti ai controlli ed alla finestra; Il secondo pulsante scambia i colori dei pulsanti facendoli ruotare; Il terzo pulsante assegna un colore alla finestra usando i colori RGB (Red ; Green ; Blue ; ovvero Rosso ; Verde ; Blu ;).

##### PROGETTO GUIDATO

- Prepara un form1 simile alla figura
- Associa al pulsante **button1** il seguente gestore di evento:

```
ForeColor = Color.FromArgb(255, 1, 0);
```



- Associa al pulsante **button2** il seguente gestore di evento:

```
ForeColor = Color.Peru;  
button1.ForeColor = Color.DarkTurquoise;  
button2.ForeColor = Color.DeepPink;  
button3.ForeColor = Color.Aqua;
```

- Associa al pulsante **button3** il seguente gestore di evento:

```
Color tmp = button1.ForeColor;  
button1.ForeColor = button2.ForeColor;  
button2.ForeColor = button3.ForeColor;  
button3.ForeColor = tmp;
```



- Prova a eseguire il progetto

L'esercizio guidato propone l'uso dei colori anche per il testo dei controlli e della finestra.

## NUMERI CASUALI

### PROGETTO GUIDATO

- Prepara un form1 simile alla figura
- Tra le variabili globali dichiara la seguente:

```
Random dado = new Random();
```

- Adesso associa al pulsante **button1** il seguente gestore di evento:

```
double valore = dado.NextDouble();
label1.Text = Convert.ToString(valore);
```

- Poi associa al pulsante **button2** il seguente gestore di evento:

```
int valore = dado.Next();
label2.Text = Convert.ToString(valore);
```

- Associa al pulsante **button3** il seguente gestore di evento:

```
int limite = Convert.ToInt32(textBox1.Text);
int valore = dado.Next(limite);
label3.Text = Convert.ToString(valore);
```

- Associa al pulsante **button4** il seguente gestore di evento:

```
int limiteMin = Convert.ToInt32(textBox2.Text);
int limiteMax = Convert.ToInt32(textBox3.Text);
int valore = dado.Next(limiteMin, limiteMax);
label4.Text = Convert.ToString(valore);
```

- Prova a eseguire il progetto e usa molte volte i pulsanti;

### L'OGGETTO RANDOM: IL GENERATORE

Random in inglese significa Casuale. In Visual C# l'oggetto `Random` serve per generare numeri casuali. Random è una parola con doppio significato: è un tipo di dato, ma anche un costruttore di variabili.

Quando si scrive:

```
Random dado = new Random();
```

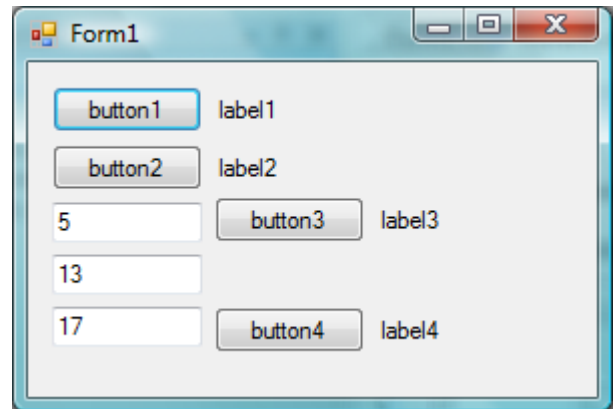
la prima occorrenza di Random è usata come tipo di dato; in effetti è possibile usarla anche più semplicemente (ma crea dei problemi successivamente, quindi è meglio usare anche la new):

```
Random dado ;
```

dove Random è usato come tipo di dato, alla stessa stregua di int oppure di string. D'altra parte la seconda occorrenza di Random, quella con le parentesi tonde, è un costruttore. Il costruttore, in Visual C#, è sempre usato insieme alla clausola new (nuovo).

In sintesi:

```
Random dado ;
```





significa: ho bisogno di usare una variabile di tipo Random che deve avere nome dado. Invece l'istruzione:


```
dado = new Random();
```

significa: chiedo memoria per creare l'oggetto di tipo Random che si collega alla variabile dado.

Questo concetto va tenuto bene a mente perché ci saranno **molte occasioni** in cui si useranno variabili che richiedono memoria mediante l'operatore new, per essere usate correttamente.



- La dichiarazione di dado ha solo creato lo spazio per il nome della variabile (ma di tipo `Random`).
- La allocazione di memoria, con `new Random()`, crea l'istanza di uno specifico oggetto `Random`.

L'oggetto `Random`  è un generatore di numeri casuali, come una macchina che fabbrica numeri.

Quindi la variabile identificata col nome «dado» non è un numero casuale: è un produttore di numeri. Questo significa anche che sarebbe scorretto tentare una istruzione come la seguente:

```
int numero = dado ; //ERRORE: TIPI INCOMPATIBILI
```

Per ottenere i numeri dal generatore è necessario chiederli. Per fare questo l'oggetto Random offre dei metodi.

#### IL METODO INTERO

Per ottenere numeri interi, il generatore offre un metodo chiamato **Next** (successivo) che estrae il **prossimo numero**. Il metodo si usa in modi diversi (anche se è sempre lo stesso metodo); un modo è il seguente:

```
int valore = dado.Next();
```

che significa la richiesta di estrarre il **prossimo numero** del generatore dado e di restituirlo per riporlo nella variabile `valore`. Il numero è un qualsiasi intero positivo a **32 bit**.

Un altro modo è il seguente:

```
int estremo = 100 ;
int valore = dado.Next( estremo );
```

che significa la richiesta di estrarre il prossimo numero **inferiore all'estremo indicato** del generatore dado e di restituirlo per riporlo nella variabile `valore`. Nell'esempio appena proposto, con l'estremo posto a 100, il metodo può restituire un qualsiasi numero intero positivo compreso tra 0 e 99.

Un altro modo di invocare Next è il seguente:

```
int minimo = 10 ;
int massimo = 100 ;
int valore = dado.Next( minimo , massimo );
```



che significa la richiesta di estrarre il prossimo numero **compreso tra gli estremi indicati, compreso il minimo ma escluso il massimo** del generatore `dado` e di restituirlo per riporlo nella variabile `valore`. Nell'esempio appena proposto, con gli estremi tra 10 e 100, il metodo può restituire un qualsiasi numero intero positivo compreso tra 10 e 99.

#### IL METODO DECIMALE

Per ottenere numeri con la virgola, il generatore offre un metodo chiamato **NextDouble** (decimale successivo) che estrae il **prossimo numero decimale**. Il metodo si usa in un solo modo; il seguente:

```
double reale = dado.NextDouble();
```

che significa la richiesta di estrarre il **prossimo numero decimale** del generatore `dado` e di restituirlo per riporlo nella variabile `reale`. Il numero è un numero con la virgola, compreso tra zero (incluso) e uno (escluso), ovvero tra 0 – 0.999999999.



# ESERCIZI

## ESERCIZI TEORICI

### ESERCIZIO 1. USARE LE RANDOM

Quali valori prenderanno (individuare il minimo ed il massimo possibili) le seguenti variabili?

```
double alce = dado.Next(10) + dado.NextDouble();
double bue = dado.Next(10) - dado.Next(10);
double cane = 1000 * dado.NextDouble();
int posinega = dado.Next(7) - dado.Next(7);
bool croce = (1 == dado.Next(2));
int roulette = (1 + dado.Next(36));
int zero = 1 / dado.Next();
double frazione = 1 / dado.Next();
double rifrazione = 1.0 / dado.Next();
```

### ESERCIZIO 2. ESTREMI DEI NUMERI CASUALI

Quali valori prenderanno (individuare il minimo ed il massimo possibili) le seguenti variabili?

```
int lotto = 1 + dado.Next(90);
int anni = 18 + dado.Next(130);
int età = 1 + dado.Next(255);
```

### ESERCIZIO 3. CASUALITÀ PREVEDIBILE

Quali valori prenderanno le seguenti variabili?

```
bool probabilità = (0 == dado.Next(10));
bool piove = (0 == dado.Next(5));
bool vincita = (1000000 == dado.Next(1000000));
bool possibile = (50 < dado.Next(100));
```

### ESERCIZIO 4. NUMERI PARI CASUALI

Come potresti generare numeri pari compresi tra 1 e 100, estremi inclusi?

### ESERCIZIO 5. NUMERI NEGATIVI CASUALI

Come potresti generare numeri negativi compresi tra -100 e 0, estremi inclusi?

### ESERCIZIO 6. NUMERI NEGATIVI PARI CASUALI

Come potresti generare numeri negativi pari compresi tra -100 e 0, estremi inclusi?

### ESERCIZIO 7. MULTIPLI DI TRE

Come potresti generare un multiplo di 3 compreso tra 1 e 100?

### ESERCIZIO 8. MULTIPLI DI SETTE

Come potresti generare un multiplo di 7 compreso tra 1 e 100?

### ESERCIZIO 9. LOGICA ASIMMETRICA

Come potresti generare True nel 25% dei casi e False nel 75% dei casi?

### ESERCIZIO 10. UNI POSITIVI E NEGATIVI

Come potresti generare +1 oppure -1 evitando lo zero? Con eque probabilità!





## ESERCIZI DI LABORATORIO

### ESERCIZIO 11. MOVIMENTI CASUALI DELLA FINESTRA

Scrivere un programma con un pulsante che sposta la finestra casualmente entro 10 pixel dalla posizione attuale, sia verso l'alto che verso il basso, sia verso destra che verso sinistra.

### ESERCIZIO 12. MOVIMENTI CASUALI DEL PULSANTE

Scrivere un programma con un pulsante che sposta il pulsante stesso casualmente entro 10 pixel dalla posizione attuale, sia verso l'alto che verso il basso, sia verso destra che verso sinistra.

### ESERCIZIO 13. INVISIBILITÀ CASUALE DEL PULSANTE

Scrivere un programma con un pulsante che agisce su una etichetta (label1) e la rende casualmente invisibile o visibile.

**SUGGERIMENTO:** è possibile assegnare ad una locazione booleana il risultato di un confronto tra un numero casuale e un numero costante, per esempio zero.

### ESERCIZIO 14. COLORI CASUALI DEL PULSANTE

Scrivere un programma con un pulsante che cambia il proprio colore casualmente.

**SUGGERIMENTO:** è possibile assegnare ad una locazione di tipo colore un valore numerico, ottenuto con un generatore di numeri casuali.

### ESERCIZIO 15. COLORI CASUALI DELLA FINESTRA

Scrivere un programma con un pulsante che cambia casualmente il colore della finestra.

### ESERCIZIO 16. POSIZIONE CASUALE DEL PULSANTE

Scrivere un programma con un pulsante che ubica il pulsante stesso casualmente dentro la finestra. Si deve evitare che possa uscire dalla finestra.

### ESERCIZIO 17. VALORI CASUALI INTERI NEGATIVI PARI

Scrivere un programma con un pulsante e una casella di testo. Il clic sul pulsante deve causare la comparsa nella casella di testo di un numero negativo pari, casuale, tra 0 e -99.

### ESERCIZIO 18. VALORI CASUALI NEGATIVI DECIMALI

Scrivere un programma con un pulsante e una casella di testo. Il clic sul pulsante deve far comparire nella casella di testo di un numero casuale negativo compreso tra 0 e -1.

### ESERCIZIO 19. VALORI CASUALI NEGATIVI DECIMALI

Scrivere un programma con un pulsante e una casella di testo. Il clic sul pulsante deve far comparire nella casella di testo di un numero casuale negativo compreso tra -1 e +1.

### ESERCIZIO 20. VALORI CASUALI NEGATIVI DECIMALI

Scrivere un programma con un pulsante e una casella di testo. Il clic sul pulsante deve far comparire nella casella di testo di un numero casuale negativo compreso tra -1 e +1 ma zero escluso.

### ESERCIZIO 21. TESTO CASUALE A SCELTA

Scrivere un programma con un pulsante che quando è cliccato mostra una frase diversa nel suo titolo: la frase deve essere compresa tra le seguenti: «Alce», «Bove», «Capra», «Daino».

### ESERCIZIO 22. TESTO CASUALE CRESCENTE

Scrivere un programma che modifica il suo testo aggiungendo una vocale casuale al testo precedente. Il testo di partenza del pulsante è «GH».



# SOMMARIO

<b>ALCUNI TIPI DI DATO UTILI E DIVERTENTI.....</b>	<b>2</b>
<b>IL TIPO DEI COLORI.....</b>	<b>2</b>
Progetto guidato.....	2
Progetto guidato.....	2
<b>NUMERI CASUALI .....</b>	<b>3</b>
Progetto guidato.....	3
L'oggetto Random: il Generatore .....	3
Il metodo intero.....	4
Il metodo decimale .....	5
<b>ESERCIZI TEORICI .....</b>	<b>6</b>
Esercizio 1. Usare le random .....	6
Esercizio 2. Estremi dei numeri casuali.....	6
Esercizio 3. Casualità prevedibile .....	6
Esercizio 4. Numeri pari casuali .....	6
Esercizio 5. Numeri negativi casuali .....	6
Esercizio 6. Numeri negativi pari casuali.....	6
Esercizio 7. Multipli di tre .....	6
Esercizio 8. Multipli di sette .....	6
Esercizio 9. Logica asimmetrica .....	6
Esercizio 10. Uni positivi e negativi.....	6
<b>ESERCIZI DI LABORATORIO.....</b>	<b>7</b>
Esercizio 11. Movimenti casuali della finestra .....	7
Esercizio 12. Movimenti casuali del pulsante .....	7
Esercizio 13. Invisibilità casuale del pulsante.....	7
Esercizio 14. Colori casuali del pulsante .....	7
Esercizio 15. Colori casuali della finestra .....	7
Esercizio 16. Posizione casuale del pulsante.....	7
Esercizio 17. Valori casuali interi negativi pari .....	7
Esercizio 18. Valori casuali negativi decimali .....	7
Esercizio 19. Valori casuali negativi decimali .....	7
Esercizio 20. Valori casuali negativi decimali .....	7
Esercizio 21. Testo casuale a scelta.....	7
Esercizio 22. Testo casuale crescente .....	7