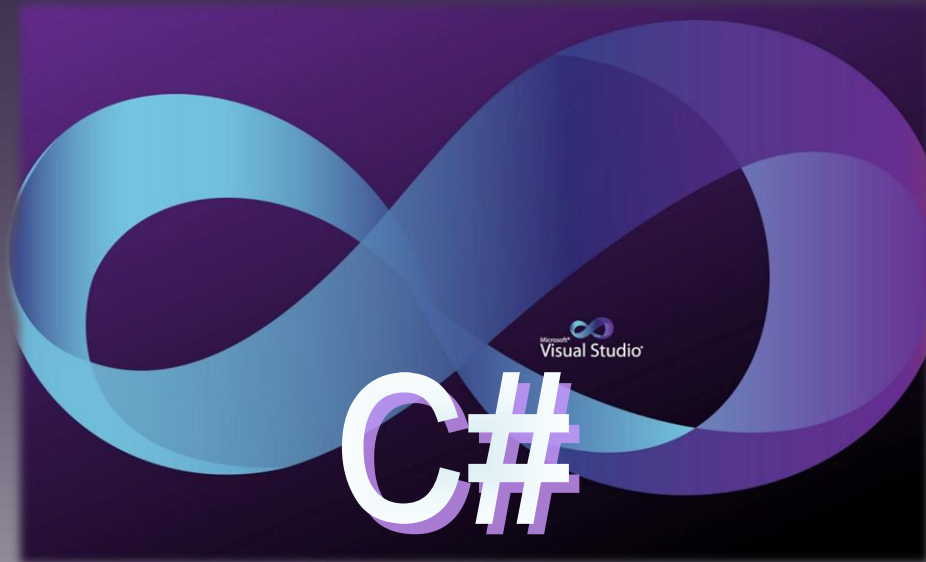


**ISTITUTO TECNICO INDUSTRIALE  
G. M. ANGIOY  
SASSARI**



# CORSO DI PROGRAMMAZIONE

## INTRODUZIONE A C#

### DISPENSA 01.01

01-01\_Introduzione\_a\_SiSciarp\_[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





# INTRODUZIONE AL CORSO

## IL PERCORSO DI INFORMATICA

### BENVENUTO

Se ti accingi a leggere queste righe della dispensa, allora è probabile che tu sia uno studente di informatica che deve iniziare il corso sui linguaggi di programmazione.

Ho preparato queste dispense per fornirti un aiuto e una guida lungo il percorso, che inizia adesso e proseguirà negli anni successivi.

L'obiettivo di queste dispense è di offrire una descrizione sintetica agli elementi di base della programmazione con Visual C# (che si legge Visual-Si-Sciarp).

### OBIETTIVI DIDATTICI

La disciplina di Informatica prepara principalmente all'uso di un ambiente di programmazione, con cui costruire programmi.

Se hai usato videogiochi, editori di videoscrittura, fogli di calcolo o strumenti per navigare sul web, allora hai usato dei programmi; ovviamente i programmi sono stati costruiti da qualcuno, che si definisce un «programmatore». In questo corso apprenderai gli elementi di base per iniziare un percorso da programmatore; in pratica imparerai come si programma, come si costruisce un programma e come lo si predispone per compiere i lavori desiderati.

Oltre a questo nel corso si studieranno anche altri elementi, come i diritti del copyright, la sicurezza sul lavoro, i linguaggi per il web e la gestione delle basi di dati.

### STRUMENTI DIDATTICI

Il corso richiede impegno, memoria, esercizio e attenzione. Inoltre, per esercitarti a casa con gli strumenti informatici, puoi richiedere alla scuola il software della Microsoft che il nostro Istituto ha ottenuto per distribuirlo con regolare licenza a tutti gli studenti della scuola; devi portare un DVD-ROM vergine con sopra scritto il tuo nome, la classe e la dicitura «Microsoft Visual Studio»; consegnalo al tuo insegnante di laboratorio e chiedi di masterizzarci i file di installazione.

Oltre a questo propongo alcuni suggerimenti:

<b>1. Frequenza delle lezioni</b>	La presenza regolare alle lezioni di teoria e di laboratorio ti può avvantaggiare notevolmente; in classe si risolvono esercizi, si confrontano diverse soluzioni e si spiegano i meccanismi per una progettazione del software. Se ti è possibile non saltare le lezioni.
<b>2. Quaderno di teoria</b>	Un quaderno è necessario per annotare le lezioni svolte alla lavagna e per prendere appunti su esercizi da svolgere a casa. Di ogni lezione annota la data e l'argomento, prima di scrivere quanto proposto dal docente e dai compagni.
<b>3. Pendrive dedicato</b>	Una pendrive è indispensabile per salvare queste dispense e per trasportare e custodire i progetti svolti a casa e a scuola. Nel pendrive puoi anche scrivere degli appunti con le lezioni di laboratorio.

Detto questo considererei conclusa l'introduzione e non mi resta che augurarti in bocca la lupo.

Buon corso.

Andrea Zoccheddu



## MICROSOFT VISUAL STUDIO

### ACRONIMI

#### DOT NET



Fonte: Wikipedia [http://it.wikipedia.org/wiki/Microsoft\\_.NET](http://it.wikipedia.org/wiki/Microsoft_.NET)

Microsoft ® ha da tempo creato una piattaforma di sviluppo software che si scrive **.NET** (un punto seguito dalla parola NET in maiuscole) e si pronuncia **DOTNET** (poiché in americano il punto si dice DOT).

Microsoft ha sviluppato .NET come risposta al linguaggio Java (che è open source) per accrescere la diffusione dei suoi software di applicazioni e di ambienti di sviluppo. La piattaforma .NET dovrebbe restare la base per le architetture client/server, internet ed intranet Microsoft nel prossimo decennio. Rispetto a Java, .NET è uno standard ISO riconosciuto (ISO 23270 e ISO 23271) e quindi non è possibile, da parte della casa madre, modificarne la sintassi (a meno di discostarsi dal proprio stesso standard).



#### DOT NET FRAMEWORK

Fonte

Wikipedia:

[http://it.wikipedia.org/wiki/.NET\\_Framework](http://it.wikipedia.org/wiki/.NET_Framework)

Il **.NET Framework** è la parte centrale della tecnologia .NET di Microsoft. È l'ambiente per la creazione, la distribuzione e l'esecuzione di tutti gli applicativi che supportano .NET siano essi Servizi Web o altre applicazioni. .NET si compone di:

- Compilatori per i principali linguaggi supportati da Microsoft
- Ambiente di esecuzione Common Language Runtime o CLR
- Libreria di Classi

#### MSDN

Fonte: <http://it.answers.yahoo.com/question/index?qid=20060929011433AAbMUSq>

MSDN è il **Microsoft Developer Network** ossia una divisione di Microsoft che intrattiene i rapporti con i programmatori esterni che sviluppano tool (pacchetti di applicazioni per computer) e programmi per i sistemi operativi e le applicazioni Microsoft.

Sul sito web MSDN di Microsoft vengono offerti diversi servizi tra cui guide e documentazione online, tutorial, tools, esempi di codice, forum e mailing-list.

Se la consultazione via web non piace o è troppo limitata è anche possibile abbonarsi (a pagamento) e ricevere cd/dvd con quintali di documentazione, nuovi tools di sviluppo, anteprime...

Di solito, quando si parla di MSDN, la gente associa automaticamente l'idea a questi cd e a Visual Studio.

#### RAD



Fonte: Wikipedia [http://it.wikipedia.org/wiki/Rapid\\_Application\\_Development](http://it.wikipedia.org/wiki/Rapid_Application_Development)

Il Rapid Application Development (indicato anche con l'acronimo RAD, letteralmente "sviluppo rapido di applicazioni") è una metodologia di sviluppo del software introdotta inizialmente da James Martin negli anni ottanta. Questa metodologia coinvolge modelli di sviluppo iterativi, la costruzione di prototipi e l'utilizzo di strumenti CASE. Solitamente questo approccio allo sviluppo comporta compromessi tra usabilità, funzionalità e velocità d'esecuzione.

L'ambiente di sviluppo di Visual C# è un RAD di proprietà di Microsoft ®. L'ambiente consente di realizzare velocemente applicazioni per i sistemi Windows di notevole potenza.



Da tempo ormai, la progettazione con linguaggi di programmazione è svolta utilizzando sistemi di sviluppo semplificati. Gli ambienti RAD sono numerosi e sono spesso specializzati in particolari linguaggi o sistemi operativi.

I RAD aiutano il progettista mostrando varie finestre visuali ciascuna con compiti precisi, agevolano la scrittura delle istruzioni suggerendo le parole corrette e integrando frammenti di codice predefiniti. Nonostante questo aiuto, la progettazione resta in qualche modo un lavoro artigianale e quasi artistico, che richiede competenze, intuito, immedesimazione nell'utente e esperienza di sviluppo.

### MICROSOFT VISUAL STUDIO

Basandosi sulla tecnologia del .NET Framework, Microsoft ha creato Visual Studio.

Microsoft Visual Studio è il RAD della Microsoft che fornisce ambienti integrati di progettazione con diversi linguaggi, tra cui Visual Basic e `Visual C#`.

In queste dispense vedremo solo l'uso per C#.



Fonte: Wikipedia <http://bios1993.blogspot.it/2008/01/cos-visual-studio.html>

Visual Studio .NET è un ambiente di sviluppo integrato (Integrated Development Environment o IDE) sviluppato da Microsoft, che supporta diversi tipi di linguaggio, quali C++, C#, J#, Visual Basic .Net e ASP .Net, e che permette la realizzazione di applicazioni, siti web, applicazioni web e servizi web.

È inoltre un RAD (Rapid Application Development), ovvero una applicazione atta ad aumentare la produttività aiutando il programmatore con mezzi come l'IntelliSense o un designer visuale delle forms.

Visual Studio .NET è inoltre multiplatforma: con esso è possibile realizzare programmi per server, workstation, pocket PC, smartphone e per il web.



### MSDN GUIDA PER PROGRAMMATORI C#



Fonte: Wikipedia [http://msdn.microsoft.com/it-it/library/67ef8sbd\(v=vs.80\).aspx](http://msdn.microsoft.com/it-it/library/67ef8sbd(v=vs.80).aspx)

Nella sezione ONLINE del sito Microsoft sono fornite informazioni dettagliate sulle funzionalità principali del linguaggio C# e sulle funzionalità accessibili a C# tramite .NET Framework.

La maggior parte delle informazioni descritte in questo testo provengono dalle pagine di questa guida on line. Se lo studente ha intenzione di approfondire lo studio del linguaggio dovrebbe fare riferimento a questa guida che è in continua evoluzione.





# L'AMBIENTE DI SVILUPPO

## COS'È UN RAD?

### IL RAD

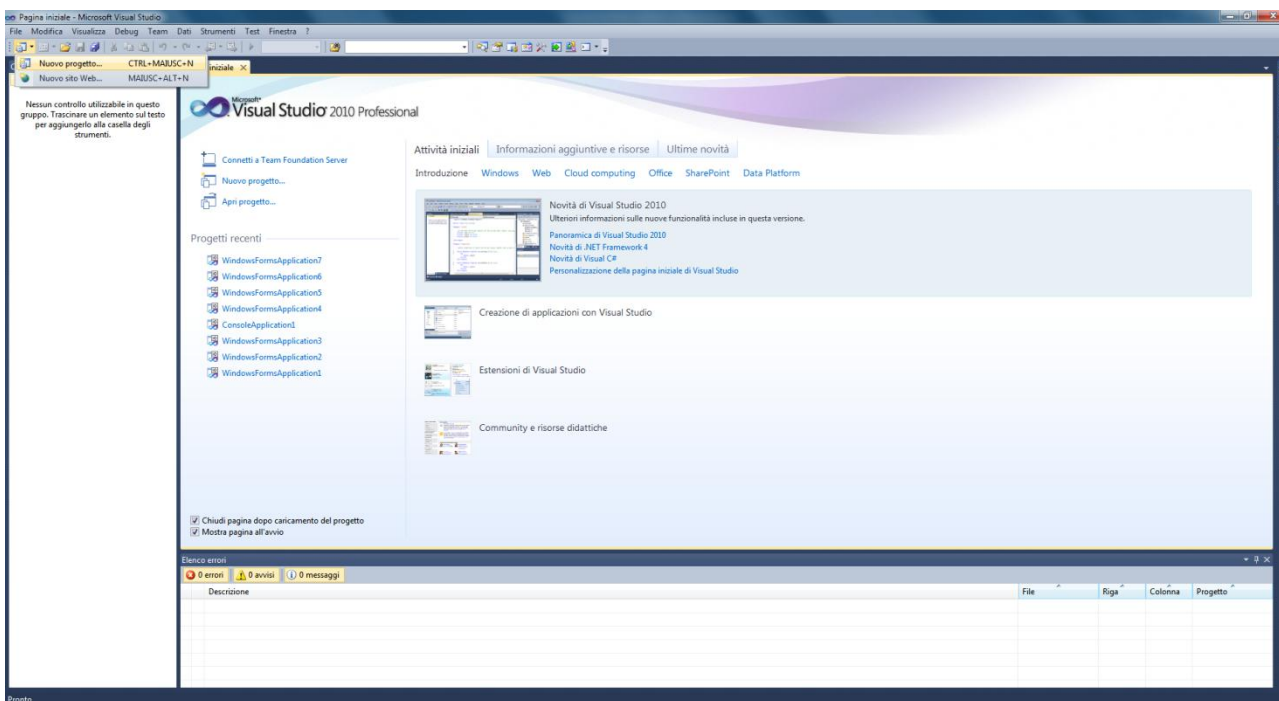
Vediamo come Microsoft ha disegnato l'ambiente RAD di Visual Studio.

L'ambiente di sviluppo di Visual C# è detto RAD (Rapid Application Development = Sviluppo Rapido di Applicazioni). L'ambiente permette di realizzare velocemente applicazioni Windows di notevole potenza.

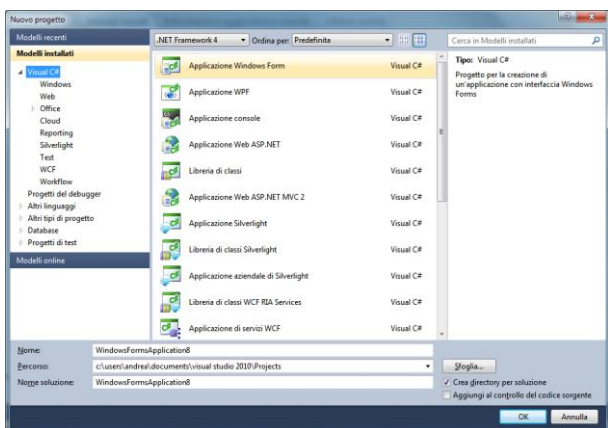
Con queste dispense si cerca di descrivere un percorso semplice ma potente di creazione di programmi.

Intanto supponiamo di aver installato l'ambiente di sviluppo. Se vuoi installarlo sul tuo computer CHIEDI AL TUO INSEGNANTE come fare per ottenere una copia con regolare licenza.

Dal menu principale del sistema operativo (Windows) avviamo Visual Studio. Compare un ambiente con diverse finestre simile alla seguente figura:



Dal menu File scegli Nuovo Progetto.

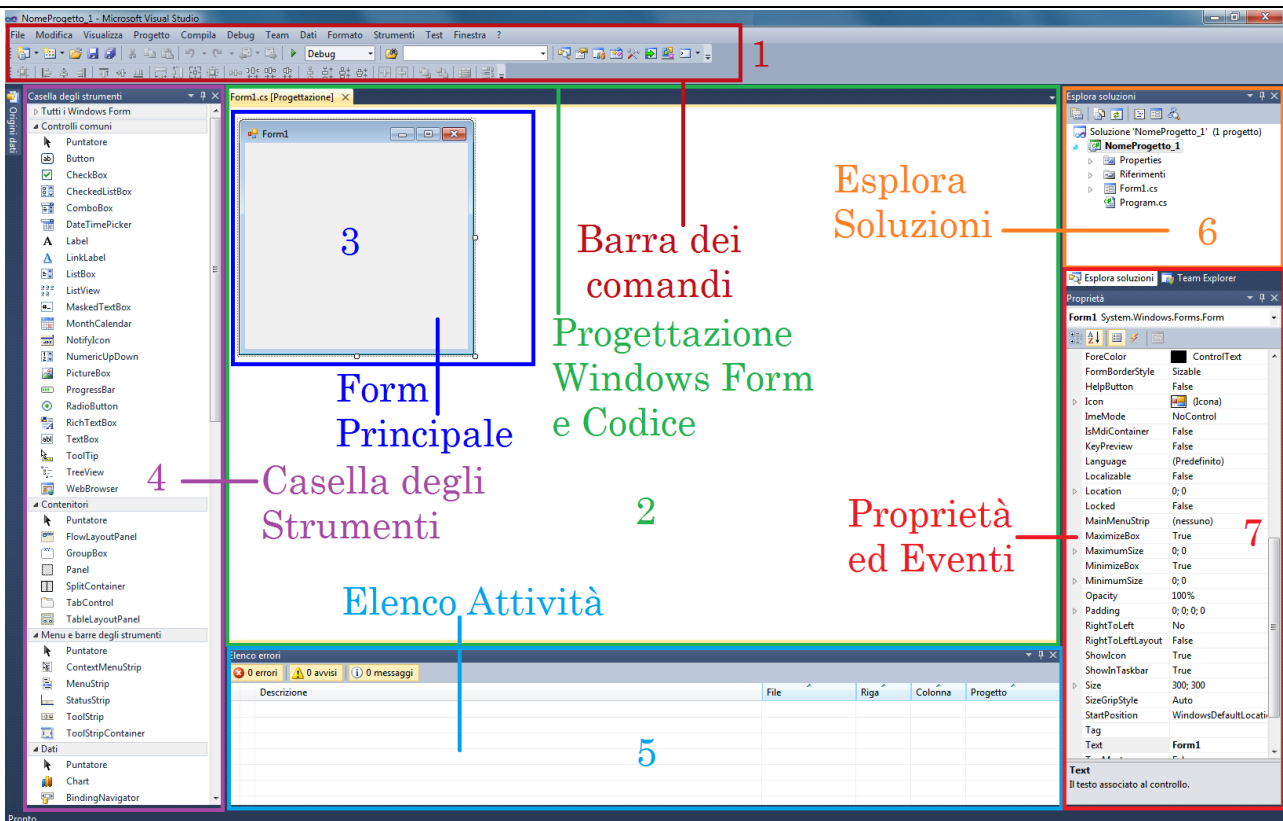


Può darsi che l'aspetto delle finestre possa essere differente dalle immagini proposte qui. Non preoccuparti. Un primo motivo può essere la versione usata (per esempio 2008 o 2010 sono leggermente diverse) ma è anche possibile che le impostazioni dell'ambiente siano state modificate da altri programmatori in precedenza. Inoltre Visual Studio consente di usare altri linguaggi oltre a C# e l'ambiente «ricorda» le scelte precedenti. In caso di difficoltà chiedi a un docente oppure a un compagno.



## FINESTRE DELL'AMBIENTE RAD

Quando si sceglie crea Nuova Applicazione e si sceglie Applicazione Windows, il RAD inizia a creare una applicazione elementare visuale (ovvero con finestre) con una finestra pronta e funzionante. L'ambiente di sviluppo consiste di diverse finestre.



Le finestre più interessanti ai fini di questo corso sono le seguenti:

<b>1 Barra dei comandi (menu e pulsanti)</b>	contiene i pulsanti e i menu di sviluppo tra cui quello per l'esecuzione del programma
<b>2 Area di Progettazione</b>	ospita il Form ma anche il codice delle varie sezioni che costituiscono l'applicazione
<b>3 Form (modulo della applicazione)</b>	è la finestra che andremo a costruire, inserendo dentro i controlli
<b>4 Casella degli Strumenti</b>	sono i controlli che possiamo mettere nel Form e sono raggruppati per categorie
<b>5 Elenco delle Attività</b>	qui compaiono i messaggi d'errore se si sbaglia a programmare, ma anche altre informazioni
<b>6 Esplora Soluzioni</b>	qui si riepiloga come è costruito il programma (più file)
<b>7 Proprietà ed Eventi</b>	elenca le caratteristiche dei controlli che stiamo progettando

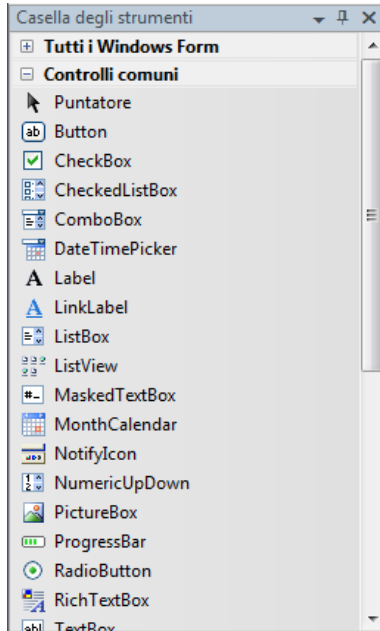


## PRIMO PROGETTO

### INSERIMENTO DI CONTROLLI

#### PROGETTO GUIDATO

- ➔ Crea un progetto visuale con una finestra vuota; puoi anche usare quello appena creato;



Sul lato sinistro dello schermo dovrebbe apparire un elenco di controlli, come nella figura qui a lato.

I controlli sono strumenti visuali (alcuni anche non visuali) che servono per far interagire l'applicazione con l'utente.

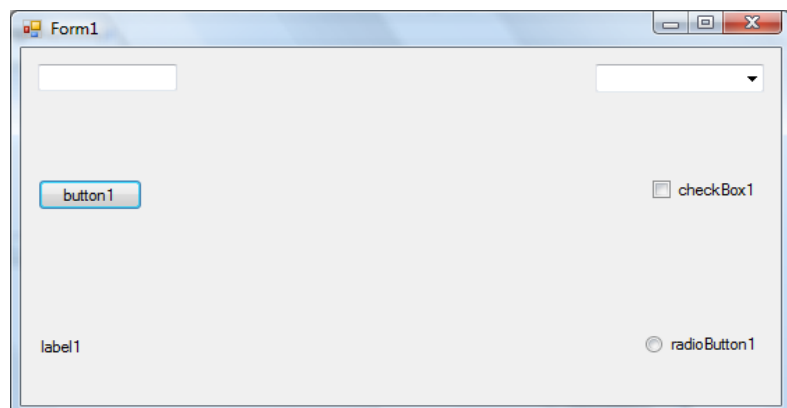
Per inserire un controllo nella finestra (detta form) è sufficiente fare un singolo clic sul controllo (MA NON DOPPIO CLIC) e poi un clic sul form1 (NON DOPPIO). Il controllo appare nella finestra.

I controlli che andremo a conoscere e usare sono:

- Button: pulsante per ricevere clic del mouse
- Label: etichetta per mostrare scritte a video
- TextBox: casella di testo per ricevere input utente
- CheckBox: casella di spunta per selezionare una scelta

Ogni controllo solitamente serve per una funzione di interazione utente specifica, come descritto nell'elenco qui sopra.

I pulsanti (**button**) servono per ricevere i clic del mouse e eseguire un compito specifico. Impareremo ad agganciare un algoritmo a ciascun pulsante. Le etichette (**label**) servono per mostrare a video delle frasi per aiutare l'utente a comprendere lo scopo degli altri controlli o il risultato di un'elaborazione. Le caselle di testo (**textBox**) servono per ricevere frasi che l'utente scrive (come il proprio nome, la password o un numero su cui eseguire calcoli).



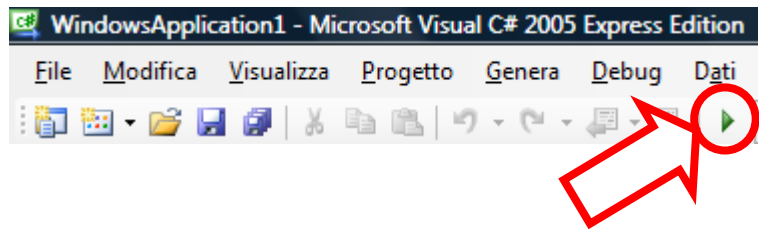
- ➔ Prova a costruire un form1 come quello della seguente figura; se non riesci a trovare i controlli chiedi aiuto al docente. Quando hai finito manda in esecuzione il programma, usa i controlli e poi terminalo.



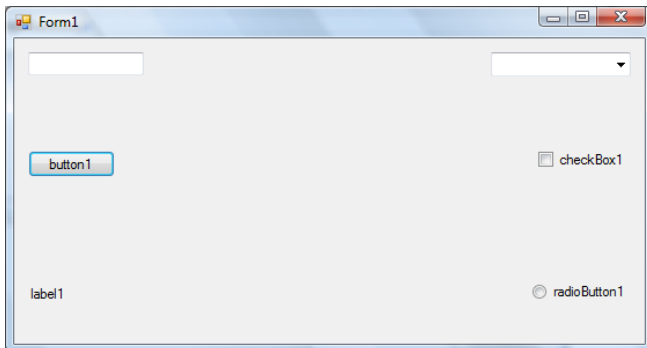
### PROGETTAZIONE ED ESECUZIONE

Quando il RAD mostra tutte le finestre già descritte ed è pronto per creare l'applicazione si dice che ci si trova in modalità di Editing o Progettazione; in questa fase è possibile modificare l'applicazione, prepararla a nostro piacimento, arricchirla di controlli come Menu, Pulsanti e caselle di testo.

Nella finestra della Barra principale ci sono diversi pulsanti; uno in particolare (RUN) è una freccetta verde che consente di eseguire l'applicazione. È sufficiente fare un singolo clic sulla freccia per avviare l'applicazione.



Quando si «lancia» l'applicazione, ovvero la si manda in esecuzione il programma compila (lo traduce in linguaggio eseguibile dalla macchina) il programma e lo avvia (esegue) in modo debug (sotto controllo del RAD che registra la sua attività e gli eventuali errori. Quando il programma è in esecuzione appare la finestra e si può interagire con essa (clic su pulsanti, scrittura su caselle di testo, usare i menu, ecc.).



Quando il programma è in esecuzione NON è possibile modificare il programma che si è scritto. Fino a quando l'applicazione è attiva la si può usare ma NON modificare.

Per poterla aggiustare o arricchire è necessario PRIMA TERMINARE L'ESECUZIONE, ovvero chiudere la finestra. È possibile chiudere la finestra come al solito (pulsante di chiusura) oppure dal RAD scegliere **Menu | Debug | Termina Debug**.

### TIPI E ISTANZE

Prima di proseguire è utile approfondire un concetto molto importante.

Ogni controllo che si inserisce nel form1 è di un certo tipo ed ha un certo nome; per esempio:

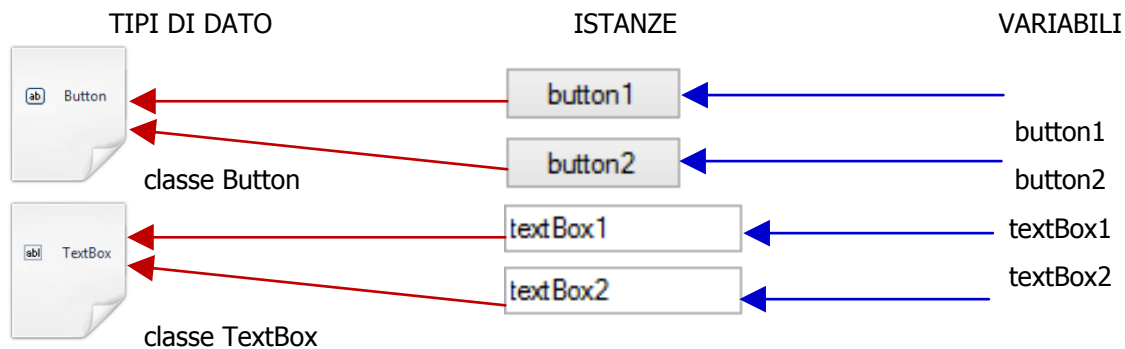
Controllo	Tipo
BUTTON1, BUTTON2, BUTTON3, . . .	Button
label1, label2, label3, . . .	Label
textBox1, textBox2, textBox3, . . .	TextBox
checkBox1, checkBox2, checkBox3, . . .	CheckBox

Il tipo di un controllo è la forma che il controllo ha.





Per comprendere meglio si pensi al progetto di una casa, scritto su un foglio di carta: il progetto descrive una casa nei dettagli, ampiezza, numero di stanze, porte, finestre; tuttavia il progetto NON è una casa, ma solo un disegno; quindi non ci posso abitare. Da quel progetto però posso costruire tante case, tutte secondo lo stesso progetto: quindi ho molte case (distinte) dello stesso progetto (tipo).

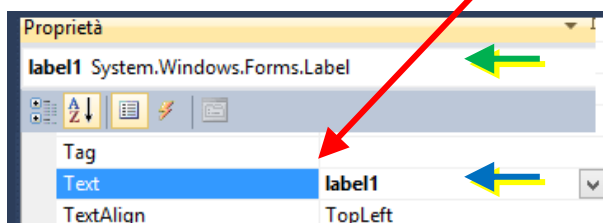
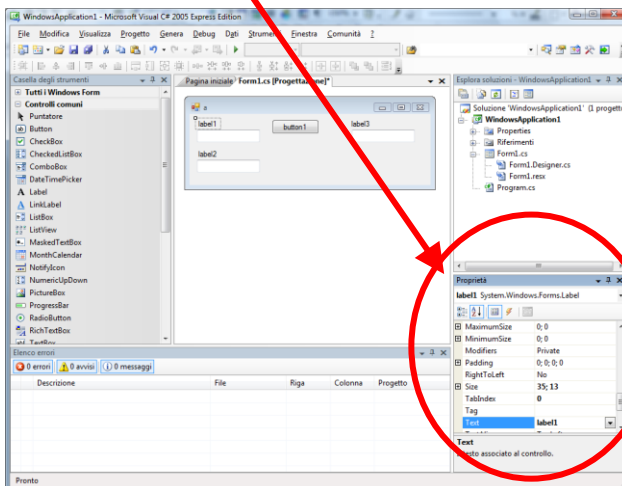
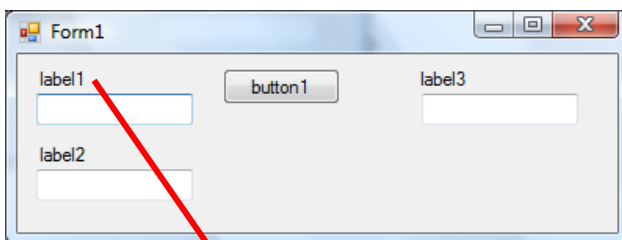


Il tipo determina come il controllo è fatto: l'aspetto, le capacità, il modo di interagire, ecc. ma ciascun controllo è distinto dagli altri, sebbene si somiglino.

## PROPRIETÀ DEI CONTROLLI

### PROGETTO GUIDATO

Crea un progetto visuale e predisponi i controlli come nella figura seguente:



- Nel Form ci sono 3 label, 3 textBox e 1 solo pulsante
- Seleziona la **Label1**
- Digita un tasto qualsiasi per visualizzare la finestra delle proprietà
- Osserva che nell'elenco di proprietà stiamo per modificare **Text**
- Quando si seleziona un controllo, la finestra delle Proprietà (vedi la figura qui a lato, dove è cerchiata in rosso) si adatta al controllo selezionato; in alto nella finestra compare il nome del controllo selezionato, nel nostro caso label1
- La finestra Proprietà mostra tutte le caratteristiche del controllo selezionato
- Se si seleziona il controllo textBox1, la finestra Proprietà cambia e mostra le caratteristiche della textBox1
- Se si seleziona il controllo button1, la finestra Proprietà cambia e mostra le caratteristiche della button1

Nella figura a lato è ingrandita la finestra Proprietà, con l'indicazione che il controllo selezionato è label1 (freccia verde) e che la proprietà in edit è Text (freccia blu).

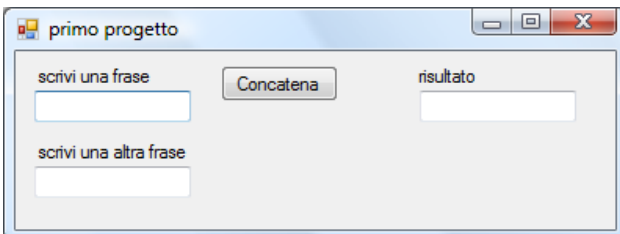


Alcune di queste proprietà sono comuni a molti controlli; vediamo le principali:

Proprietà comuni a controlli	Descrizione
<b>Text</b>	Testo visualizzato sul controllo
<b>BackColor</b>	Colore di sfondo del controllo
<b>ForeColor</b>	Colore di primo piano del controllo, di solito il colore del testo visualizzato
<b>Visible</b>	Indica se il controllo è visibile in esecuzione
<b>Enabled</b>	Indica se il controllo reagisce ai comandi dell'utente

Per osservare come usare le proprietà a tempo di progettazione è sufficiente selezionare un controllo, cercare la proprietà desiderata e cambiarne il valore; se il valore immesso non è corretto è scartato. Prova a procedere in questo modo:

- seleziona la **label1** e cerca la sua Proprietà **Text**; infine modifica il suo valore scrivendo:  
*scrivi una frase*  
Premi Invio quando hai finito
- seleziona la **label2** e cerca la sua Proprietà **Text**; infine modifica il suo valore scrivendo:  
*scrivi una altra frase*  
Premi Invio quando hai finito
- seleziona la **label3** e cerca la sua Proprietà **Text**; infine modifica il suo valore scrivendo:  
*risultato*  
Premi Invio quando hai finito
- seleziona la **button1** e cerca la sua Proprietà **Text**; infine cambia il suo valore scrivendo:  
*concatena*  
Premi Invio quando hai finito
- adesso con un clic sulla finestra (evita i controlli che ospita), selezioni l'intero **form1**; cerca la sua Proprietà **Text**; infine modifica il suo valore scrivendo:  
*primo progetto*  
Premi Invio quando hai finito



- Verifica se alla fine il progetto si presenta simile alla figura qui a fianco
- Provalo (manda in esecuzione ed usa i controlli)
- Infine termina l'applicazione; ritorniamo in fase di progettazione e cambiamo altri valori

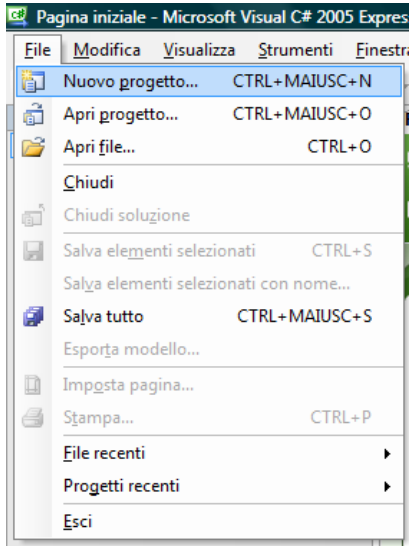
- seleziona la **label1** e cerca la sua Proprietà **Visible**; infine modifica il suo valore scrivendo:  
*false*  
Premi Invio quando hai finito
- seleziona il **button1** e cerca la sua Proprietà **Visible**; infine modifica il suo valore scrivendo:  
*false*  
Premi Invio quando hai finito
- Manda in esecuzione ed usa i controlli: hai notato che non si vedono né la etichetta né il pulsante?
- Infine termina l'applicazione; ritorniamo in fase di progettazione e cambiamo altri valori
- seleziona la **label1** e cerca la sua Proprietà **Visible**; infine modifica il suo valore scrivendo:  
*true*  
Premi Invio quando hai finito
- seleziona il **button1** e cerca la sua Proprietà **Visible**; infine modifica il suo valore scrivendo:  
*true*  
Premi Invio quando hai finito
- seleziona la **label1** e cerca la sua Proprietà **Enabled**; infine modifica il suo valore scrivendo:  
*false*  
Premi Invio quando hai finito
- seleziona il **button1** e cerca la sua Proprietà **Enabled**; infine modifica il suo valore scrivendo:  
*false*  
Premi Invio quando hai finito



## RIFLESSIONI

Quando si crea un progetto è possibile cambiare l'aspetto dei controlli mediante le loro Proprietà; queste modifiche si notano quando il programma va in esecuzione, ma non sempre in modalità di progettazione. Per cambiare il testo di un controllo è opportuno cambiare il valore della proprietà **Text**; per impedire ad un utente di usarlo è possibile impostare la proprietà **Enabled** a **false**; se lo si vuole rendere invisibile usare la proprietà **Visibile**.

## SALVATAGGIO E RIAPERTURA

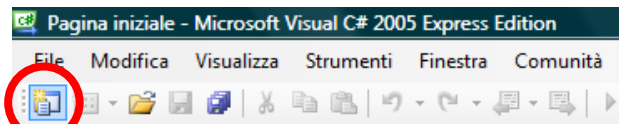


Quando si deve costruire un nuovo progetto occorre chiedere al sistema di predisporre uno.

Il comando da menu è :

**Menu | File | Nuovo Progetto . . .**

Oppure è possibile usare il pulsante **Nuovo Progetto**:



oppure si può usare la combinazione di tasti **Ctrl+Maiusc+N**

da tastiera.

Compare una finestra simile a quella qui di seguito, dove occorre indicare:

### Tipo progetto:

è il tipo di progetto desiderato; ai fini del nostro corso useremo sempre un modello C# ovvero:

#### Visual C#

### Modelli:

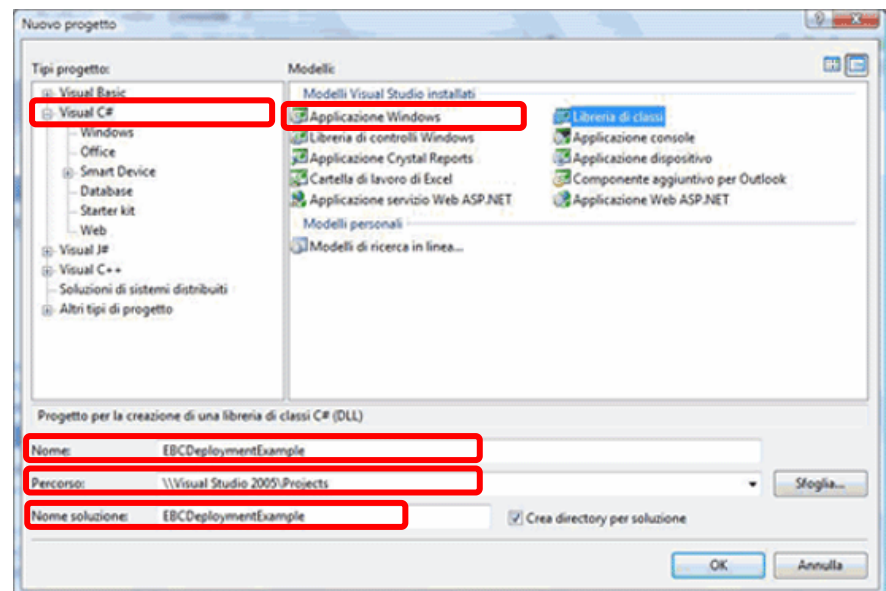
è il modello che predispose il nuovo progetto; per il momento sceglieremo una:

#### Applicazione Windows

che significa una applicazione visuale, con una finestra e i controlli da usare col mouse

### Nome:

qui puoi scrivere il nome del progetto che si desidera creare. Nella posizione indicata dal campo **Percorso** sarà creata una cartella con lo stesso **Nome** del progetto. Scegli un nome, possibilmente senza usare spazi e senza usare caratteri speciali (come per esempio: !"£\$%&/()=?^\*è\$°ç\_;;,-òàù+èì).





Resta sul semplice e compatto come per esempio:

[MioProgetto\\_01](#)

oppure

[ProgettoScambioColori](#)

### **Percorso:**

È la posizione nel file system dove salverà il progetto. Possibilmente scegli una cartella facile da trovare, su cui godi di ampi diritti, dove si creeranno ulteriori cartelle. Scrivi qui il percorso in cui si desidera creare il progetto oppure scegline uno dall'elenco. Se hai difficoltà a scriverlo, usa il pulsante **Sfoggia . . .** che consentesi esplorare il sistema e di individuare. Per esempio puoi usare:

[C:\Users\Scuola\Documents\Progetti\Equazione2grado](#)

### **Nome soluzione**

È possibile lasciare il valore predefinito, che suggerisce lo stesso nome digitato nel campo Nome.

La Soluzione è un concetto più ampio di Progetto; una soluzione può includere più progetti; in questo corso non affronteremo questa questione. Se, tuttavia, si desidera modificarla, si può scrivere qui il nome della nuova soluzione dentro la quale si desidera creare il progetto.

### **Crea directory per soluzione**

Questa casella di spunta permette di l'opzione di creare una nuova cartella (directory) con il nome scelto per il progetto e la soluzione. Conviene spuntare la casella e far costruire detta directory al sistema.

### **CHE SUCCUDE?**

Quando si scelgono il nome del Progetto e il suo Percorso, fondamentalmente hai scelto il luogo fisico su disco dove salvare tutti i file del progetto e il nome della cartella che li conterrà. È MOLTO importante che ricordi dove hai salvato il progetto, per poterlo riaprire e poterlo copiare.

Salva progetto

Nome: MioProgetto\_01

Percorso: C:\Users\Public\Documents\Prova\Prova\_01 Sfoggia...

Nome soluzione: MioProgetto\_01  Crea directory per soluzione

Salva Annulla

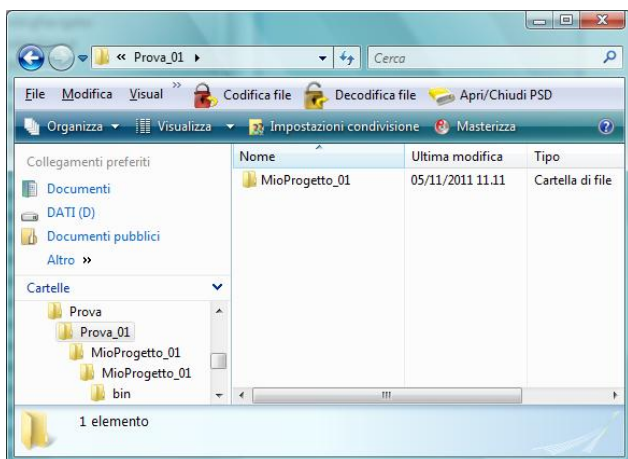
nell'esempio qui sopra il progetto si chiama:

**MioProgetto\_01**

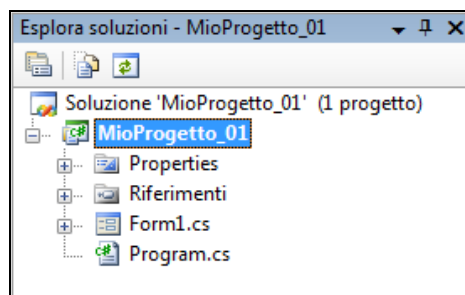
e si trova nella cartella:

**C:\Users\Public\Documents\Prova\Prova\_01**

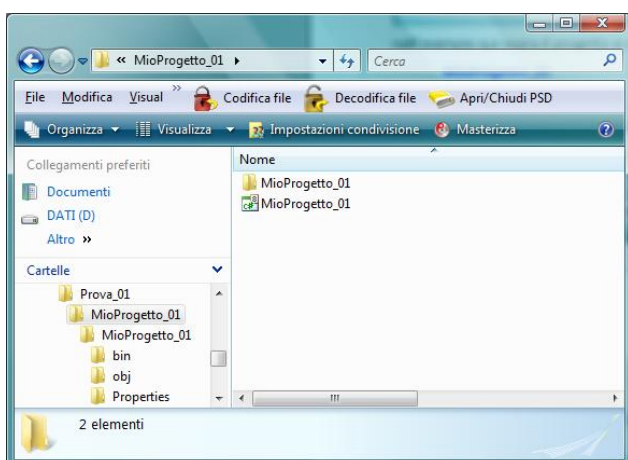




Nella finestra del RAD «**Esplora Soluzioni**» si può notare che si è creata una Soluzione che contiene un Progetto che contiene altri elementi.



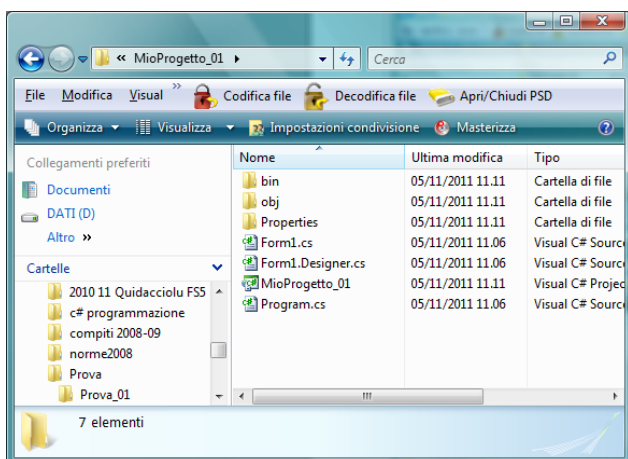
Se si cerca la cartella **Prova\_01** nel file system, con esplora risorse, si trova una nuova cartella creata dal sistema che si chiama **MioProgetto\_01**.



Se si vuole copiare completamente il progetto, è opportuno copiare questa cartella e tutto il suo contenuto.

Se si entra nella cartella **MioProgetto\_01**, si trova un'ulteriore cartella chiamata **MioProgetto\_01** ed un file omonimo di tipo «Soluzione Visual Studio» (un file di riferimenti).

Se si vuole aprire il progetto, è possibile fare un doppio clic sul file del progetto.



Scendendo di un altro livello si trovano molte cartelle:

<b>bin</b>	conterrà i file eseguibili
<b>obj</b>	conterrà i file degli oggetti
<b>Properties</b>	conterrà i file degli oggetti

E molti altri file:

<b>Form1.cs</b>	codice sorgente della finestra
<b>Form1.Designer.cs</b>	codice sorgente ulteriore
<b>MioProgetto_01</b>	file principale di progetto
<b>Program.cs</b>	codice sorgente del programma



## RIAPERTURA DEL PROGETTO

Se si desidera chiudere la soluzione (che include il progetto) si può scegliere il Menu **Chiudi Soluzione**.

Per riaprire un Progetto chiuso da poco, un progetto recente, è possibile sceglierlo dal Menu **File | Progetti Recenti**.

The screenshot shows the 'File' menu of Microsoft Visual Studio 2005 Express. The 'Progetti recenti' (Recent Projects) submenu is open, displaying a list of recent project files:

- 1 C:\...\Prova\Prova\_01\MioProgetto\_01\MioProgetto\_01.sln
- 2 C:\...\Projects\WindowsApplication2\WindowsApplication2.sln
- 3 C:\...\Visual Studio 2005\Projects\SaveToFile1\SaveToFile1.sln
- 4 C:\...\Projects\WindowsApplication1\WindowsApplication1.sln
- 5 C:\...\Projects\ConsoleApplication2\ConsoleApplication2.sln
- 6 C:\...\Projects\ConsoleApplication1\ConsoleApplication1.sln

Il Menu propone un certo numero di progetti chiusi da poco e li apre dalla posizione in cui erano salvati.

Un'altra possibilità è scegliere il Menu **File | Apri Progetto**. . . che, nel caso che si siano chiuse tutele soluzioni, riapre la soluzione con il progetto dentro.

Per i fini del corso, che non è rivolto a esperti programmatori, ma solo a novizi della programmazione che devono apprendere le basi dei linguaggi, forse il primo linguaggio appreso formalmente, non approfondiremo i significati di Soluzione, Progetti e File. In seguito si fornirà qualche precisazione su Codice Sorgente e Codice Eseguitabile ma sempre in modo intuitivo e non formale.

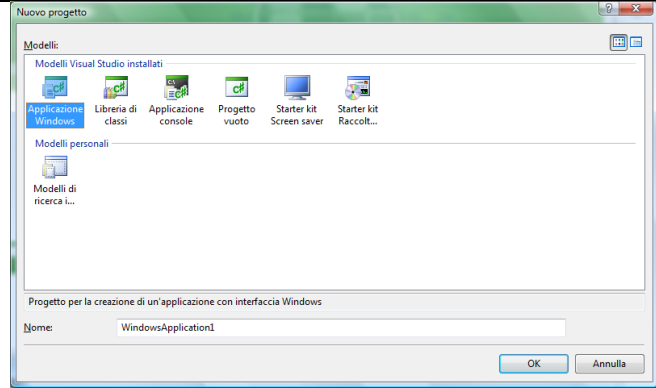
## RIEPILOGO

- 1) quando si crea una nuova applicazione, si crea una Soluzione
- 2) una Soluzione può contenere MOLTI Progetti; noi ci limiteremo a Soluzione che contengono un solo Progetto, ma è importante ricordare che non è sempre così e che il sistema si prepara come se potesse accoglierne molti
- 3) un Progetto è composto SEMPRE da molti file e cartelle
- 4) per copiare una Soluzione è necessario copiare l'intera cartella con tutto il suo contenuto
- 5) quando farai Verifiche Pratiche di Laboratorio, il docente potrebbe chiederti di consegnare l'intera Soluzione: ricorda dove hai salvato la Soluzione e ricorda di copiare l'intera sua cartella !!!

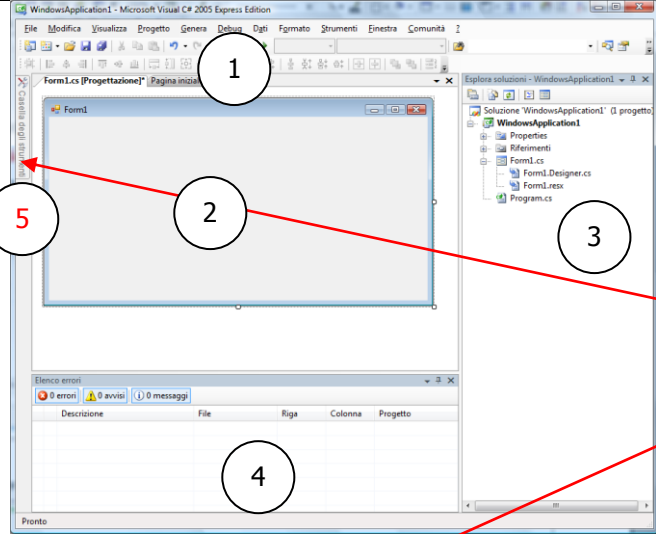


# ESERCIZI

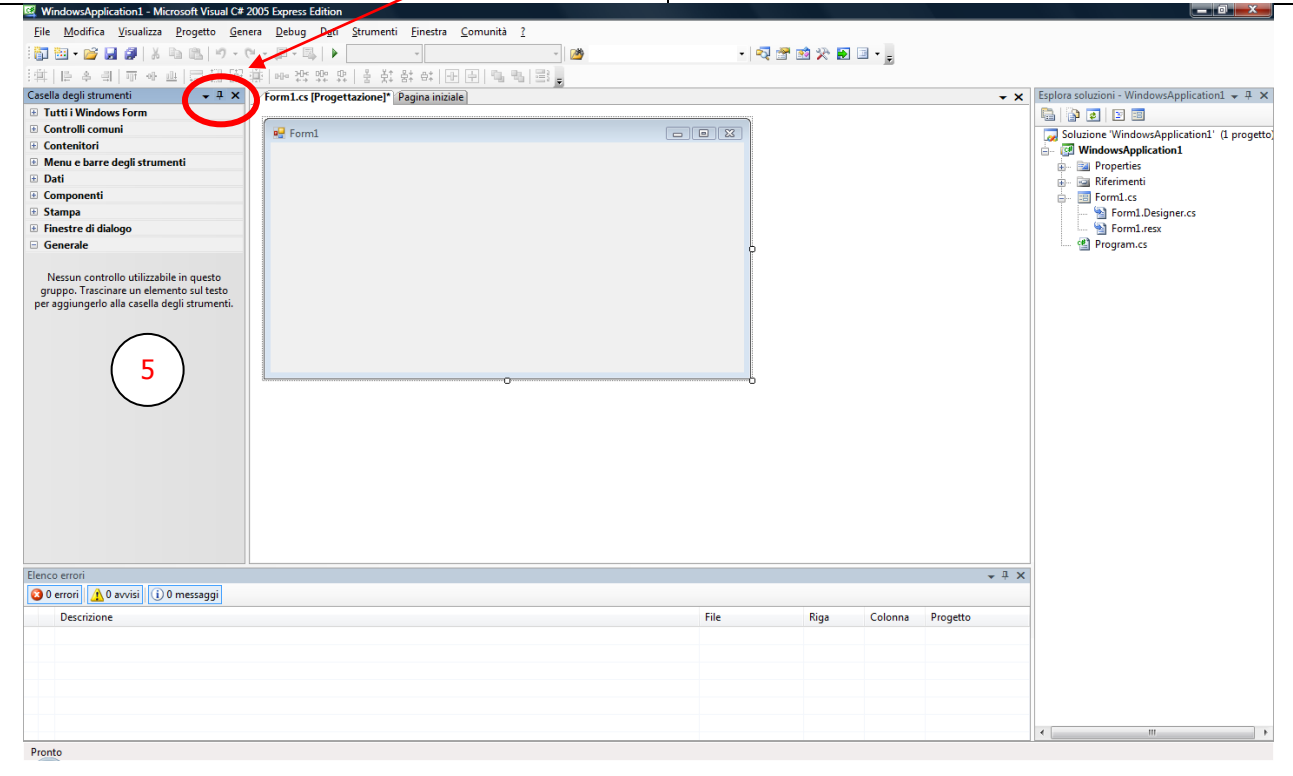
## ESERCIZIO 1. GUIDATO DI RIEPILOGO



- Crea un nuovo progetto ma utilizza la scelta Applicazione Windows invece che l'applicazione Console, come nella figura a lato:
- puoi lasciare il nome applicazione proposto
- appare una grande finestra che al suo interno ne contiene altre

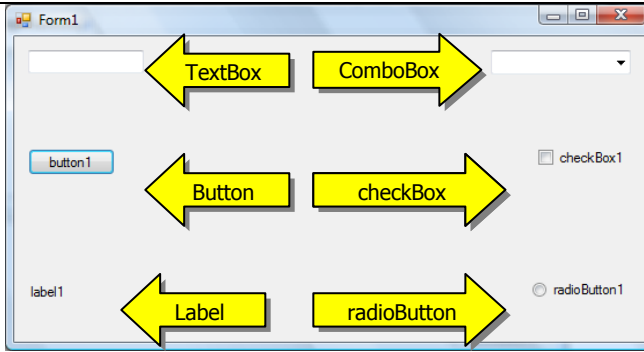


- le finestre interne interessanti sono:
  1. Barra principale (menu e pulsanti)
  2. Form (modulo della applicazione)
  3. file della soluzione
  4. messaggi d'errore
  5. finestra dei controlli
- la finestra dei controlli potrebbe (come nella figura) essere ristretta; conviene esplicitarla
- clic sulla paletta per esibire la finestra e clic sulla puntina per fissare (se no scompare)





- nella barra dei controlli è possibile mostrare dei sotto elenchi;
- inizialmente scegli i controlli comuni;



- in questa fase useremo i seguenti controlli:

- TextBox (casella di testo)
- Button (pulsante)
- Label (etichetta)

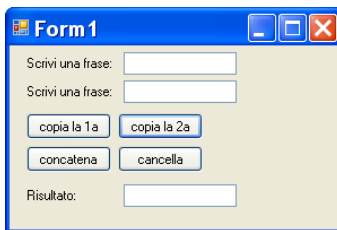
- Altri controlli che useremo in futuro saranno:

- ComboBox (casella combinata)
- CheckBox (casella di spunta)
- RadioButton (scelta alternativa)

- Prova a fare clic sul controllo e poi a fare clic sul Form per inserirlo

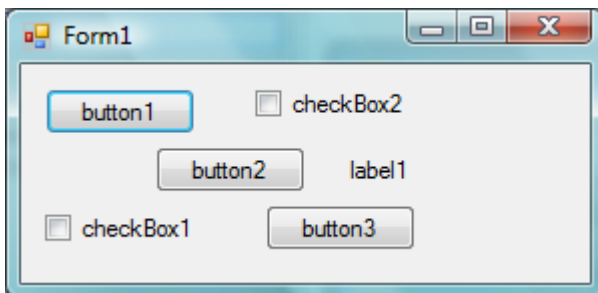
- Quando hai finito lancia l'esecuzione con RUN

### ESERCIZIO 3. CONTROLLI ELEMENTARI



- Crea un progetto visuale e predisponi i controlli come nella figura
- Ci sono tre Label, tre TextBox, quattro Button
- Modifica (se ci riesci) i colori dei controlli
- Modifica (se ci riesci) il colore della finestra

### ESERCIZIO 2. CONTROLLI BASE



- Crea un progetto visuale e predisponi i controlli come nella figura
- Modifica posizione e dimensioni come suggerito
- Modifica (se ci riesci) i colori dei controlli
- Modifica (se ci riesci) il colore della finestra

### ESERCIZIO 4. CONTROLLI GENERALI



- Crea un progetto visuale e predisponi i controlli come nella figura
- Modifica il titolo della finestra
- Modifica il testo dei controlli
- Modifica i colori dei controlli e dei testi
- Modifica il colore della finestra in bianco





# SOMMARIO

<b>IL PERCORSO DI INFORMATICA .....</b>	<b>2</b>
<b>BENVENUTO .....</b>	<b>2</b>
<b>OBIETTIVI DIDATTICI .....</b>	<b>2</b>
<b>STRUMENTI DIDATTICI.....</b>	<b>2</b>
<b>MICROSOFT VISUAL STUDIO .....</b>	<b>3</b>
<b>ACRONIMI .....</b>	<b>3</b>
DOT NET.....	3
DOT NET Framework .....	3
MSDN .....	3
RAD 3 .....	
Microsoft Visual Studio .....	4
Msdn Guida per programmatori C# .....	4
<b>L'AMBIENTE DI SVILUPPO .....</b>	<b>5</b>
<b>Cos'è un RAD?.....</b>	<b>5</b>
il RAD .....	5
Finestre dell'Ambiente RAD .....	6
<b>PRIMO PROGETTO.....</b>	<b>7</b>
<b>INSERIMENTO DI CONTROLLI.....</b>	<b>7</b>
Progetto guidato.....	7
Progettazione ed esecuzione .....	8
Tipi e istanze .....	8
<b>PROPRIETÀ DEI CONTROLLI.....</b>	<b>9</b>
Progetto guidato.....	9
Riflessioni.....	11
<b>SALVATAGGIO E RIAPERTURA .....</b>	<b>11</b>
Che succede? .....	12
Riapertura del Progetto .....	14
Riepilogo.....	14
Esercizio 1. Guidato di riepilogo .....	15
Esercizio 3. Controlli elementari .....	16
Esercizio 2. Controlli base.....	16
Esercizio 4. Controlli generali .....	16