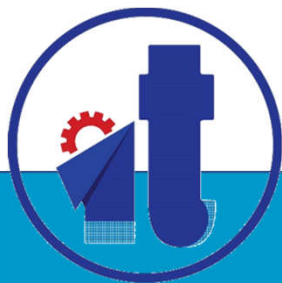




LẬP TRÌNH TRÊN MÔI TRƯỜNG WINDOWS





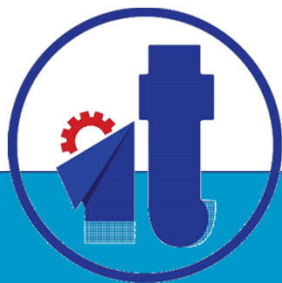
Chương 3: Lập Trình Windows Forms (tt)





Nội dung

- ☐ Xử lý sự kiện
- ☐ Lập trình ứng dụng Windows Forms
- ☐ Các controls cơ bản
- ☐ Các controls nâng cao
- ☐ Các sự kiện thông dụng cho chuột – bàn phím



Các controls nâng cao



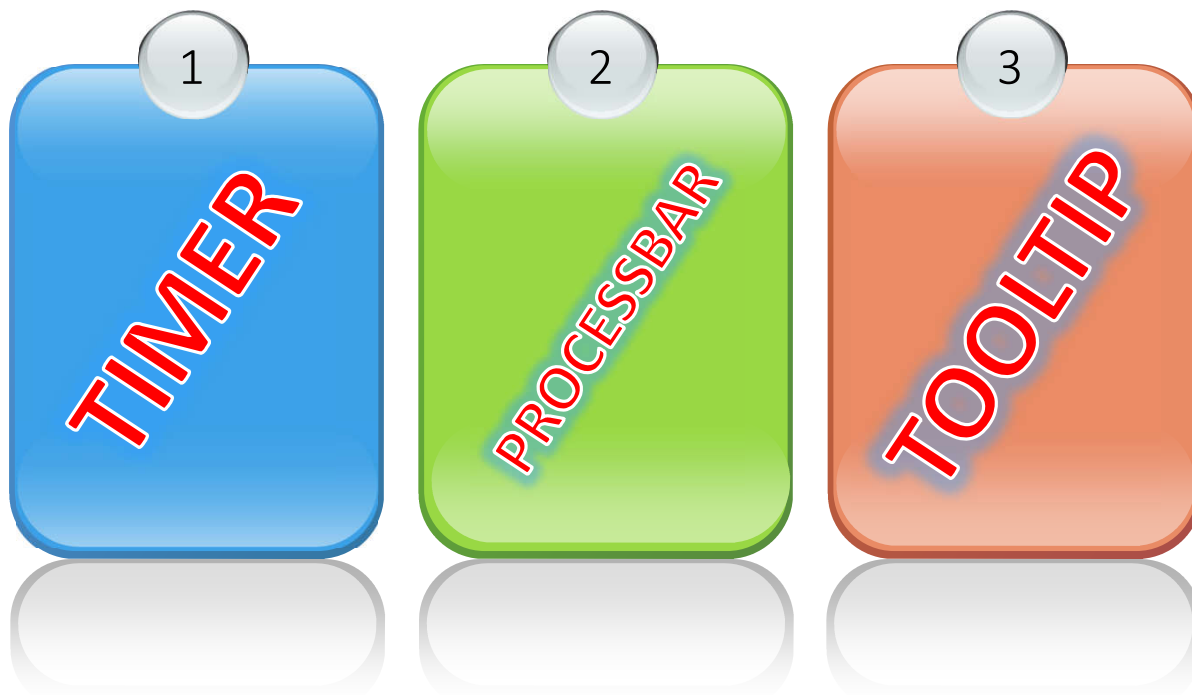


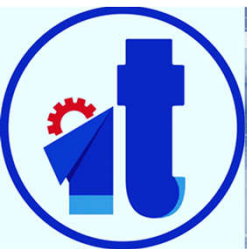
Các controls nâng cao

- ☐ Timer, ProcessBar, ToolTip
- ☐ Dialogs
- ☐ MDI (Form cha – con)
- ☐ MenuStrip, ContextMenuStrip



Timer, ProcessBar, ToolTip





Timer, ProgressBar, ToolTip

❑ Timer

- Bộ định thời gian, thiết lập một khoảng thời gian xác định (interval) và khi hết khoảng thời gian đó Timer sẽ phát sinh sự kiện tick.

Properties

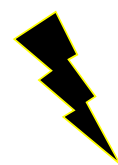
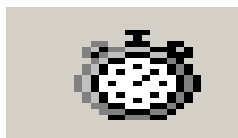
Enabled

Interval

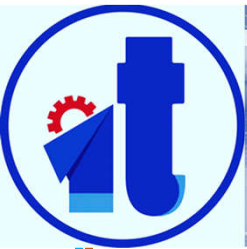
Methods

Start

Stop



Tick



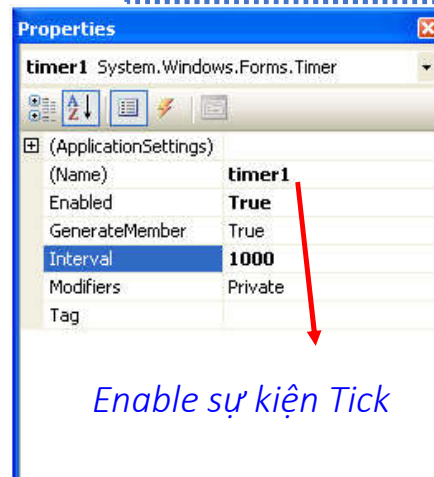
Timer, ProgressBar, ToolTip

□ Timer

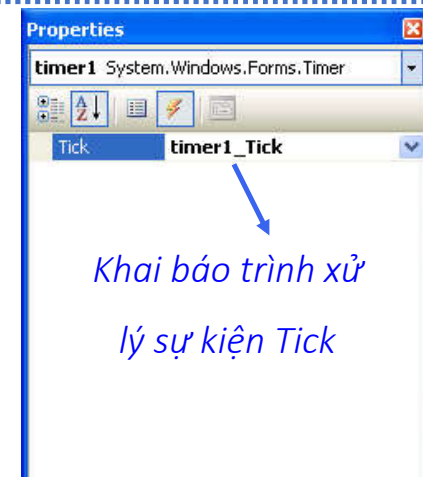
- Sự chính xác còn phụ thuộc vào đồng hồ hệ thống và các hoạt động hiện thời của hệ thống.
- Sự kiện Tick

```
private void timer1_Tick(object sender, EventArgs e)
{
    // Lấy thời gian hệ thống
    DateTime now = DateTime.Now;
    label1.Text = now.ToString();
}
```

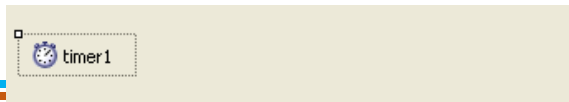
Hiển thị thời gian

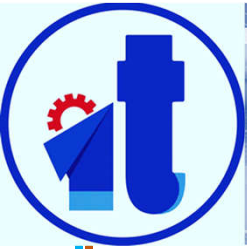


Enable sự kiện Tick



*Khai báo trình xử
lý sự kiện Tick*





Timer, ProgressBar, ToolTip

□ ProgressBar

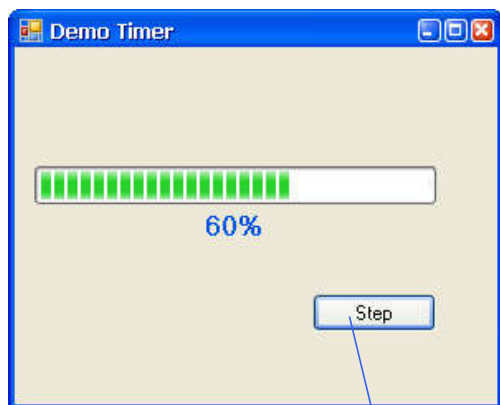
- Hiển thị tiến độ thực hiện của 1 công việc nào đó
- *Các thuộc tính:*
 - ✓ *Minimum/Maximum*: giá trị nhỏ/lớn nhất
 - ✓ *Step*: số bước tăng khi gọi hàm PerformStep
 - ✓ *Value*: giá trị hiện tại
 - ✓ *Style*: kiểu của progress bar
- *Phương thức:*
 - ✓ *PerformStep()*: tăng thêm step
 - ✓ *Increment(int value)*: tăng vị trí hiện tại của tiến độ với giá trị xác định



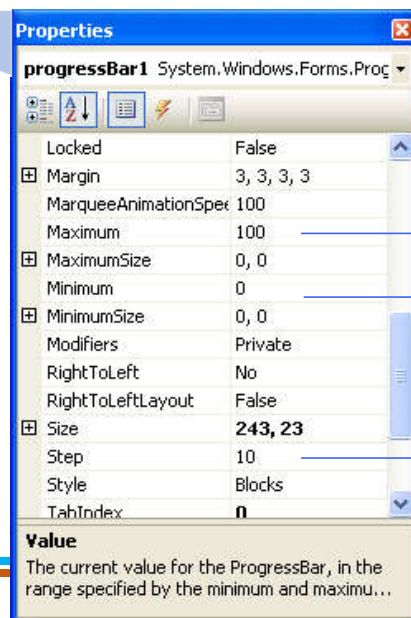
Timer, ProgressBar, ToolTip

□ ProgressBar

➤ Demo



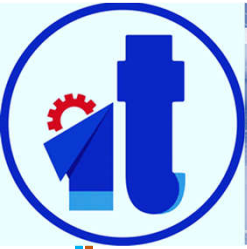
Tăng tiến độ theo step và cập nhật lại % hoàn thành lên label



Max = 100

Min = 0

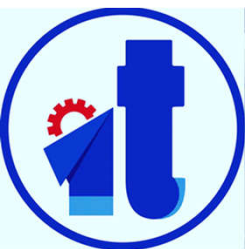
Step = 10



Timer, ProcessBar, ToolTip

❑ ToolTip

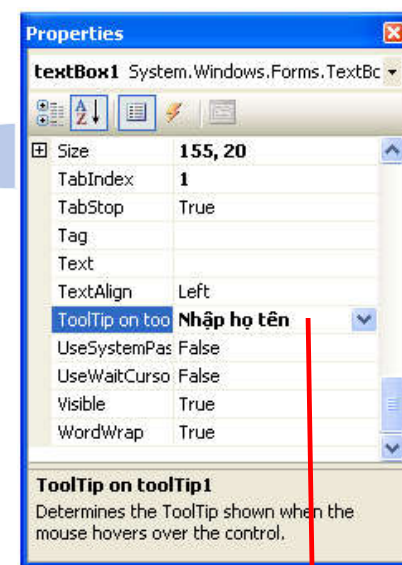
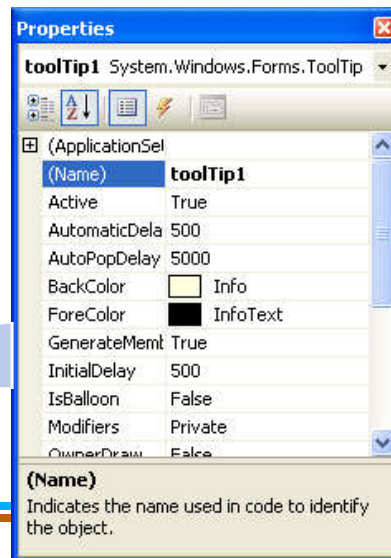
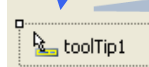
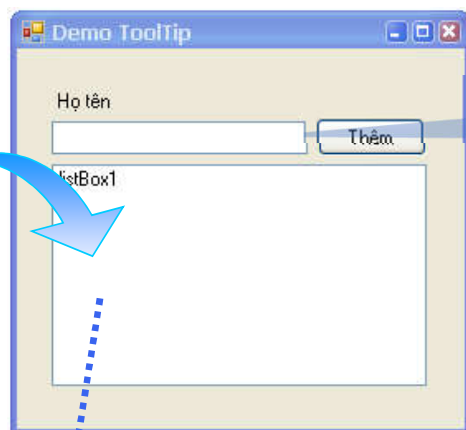
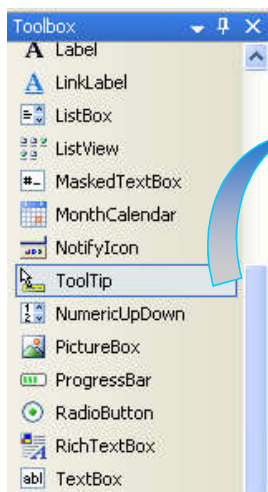
- Cung cấp chức năng hiển thị một khung text nhỏ khi user di chuyển chuột vào control bất kỳ
- Khung text chứa nội dung mô tả ý nghĩa của control
- Cách sử dụng:
 - ✓ Từ ToolBox kéo ToolTip thả vào form
 - ✓ Kích chọn control muốn thêm tooltip
 - ✓ Trong cửa sổ Properties của control sẽ có thuộc tính ToolTip. Thêm text vào thuộc tính này để hiển thị khi tooltip xuất hiện.



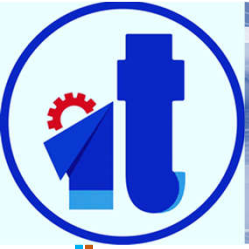
Timer, ProgressBar, ToolTip

❑ ToolTip

➤ Demo



Nội dung Tooltip



Dialogs

□ Common Dialog Boxes

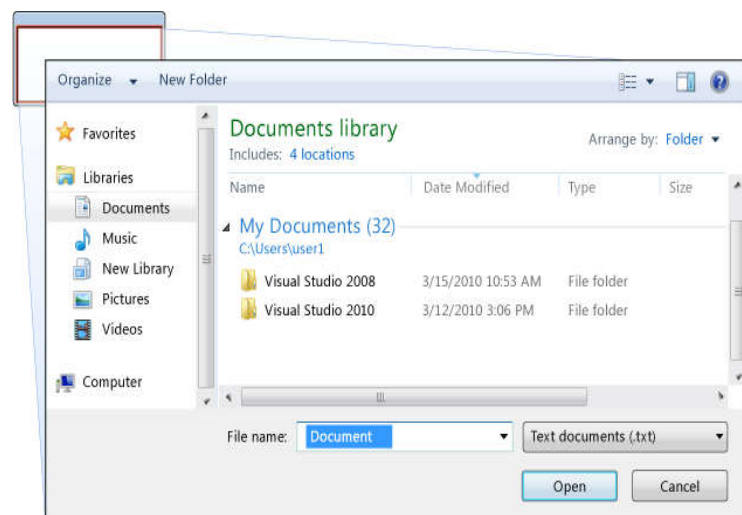
- Các Dialog Boxes thông dụng sử dụng nhiều trong các ứng dụng khác nhau như:
 - ✓ Open File Dialog
 - ✓ Save File Dialog Box
 - ✓ Print Dialog Box
 - ✓ Font and Color Dialog Boxes

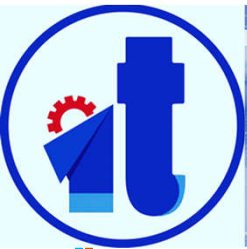


Dialogs

❑ Common Dialog Boxes

- **Open File Dialog:** hiển thị hộp thoại cho phép người dùng chỉ định tên tập tin để mở
 - ✓ **FileName:** đường dẫn đến tập tin được chọn
 - ✓ **FileNames:** mảng đường dẫn đến các tập tin được chọn
 - ✓ **SafeFileName:** tên tập tin được chọn
 - ✓ **SafeFileNames:** mảng tên các tập tin được chọn



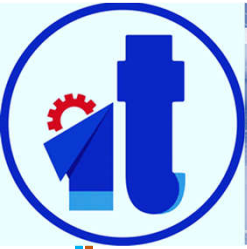


Dialogs

□ Common Dialog Boxes

➤ Open File Dialog

- ✓ ShowDialog(): hiển thị hộp thoại và trả về kết quả
 - OK: click vào nút Open
 - Cancel: click vào nút Cancel
- ✓ Multiselect: cho phép chọn nhiều tập tin (True)
- ✓ ShowReadOnly: cho phép hiển thị các tập tin ẩn (True)
- ✓ Title: tiêu đề của hộp thoại
- ✓ InitialDirectory: đường dẫn hiển thị các tập tin – thư mục trong hộp thoại
- ✓ Filter: lọc loại tập tin hiển thị trong hộp thoại



Dialogs

□ Common Dialog Boxes

➤ Open File Dialog

```
// Configure open file dialog box
Microsoft.Win32.OpenFileDialog dlg = new Microsoft.Win32.OpenFileDialog();
dlg.FileName = "Document"; // Default file name
dlg.DefaultExt = ".txt"; // Default file extension
dlg.Filter = "Text documents (*.txt)|*.txt"; // Filter files by extension

// Show open file dialog box
Nullable<bool> result = dlg.ShowDialog();

// Process open file dialog box results
if (result == true)
{
    // Open document
    string filename = dlg.FileName;
}
```

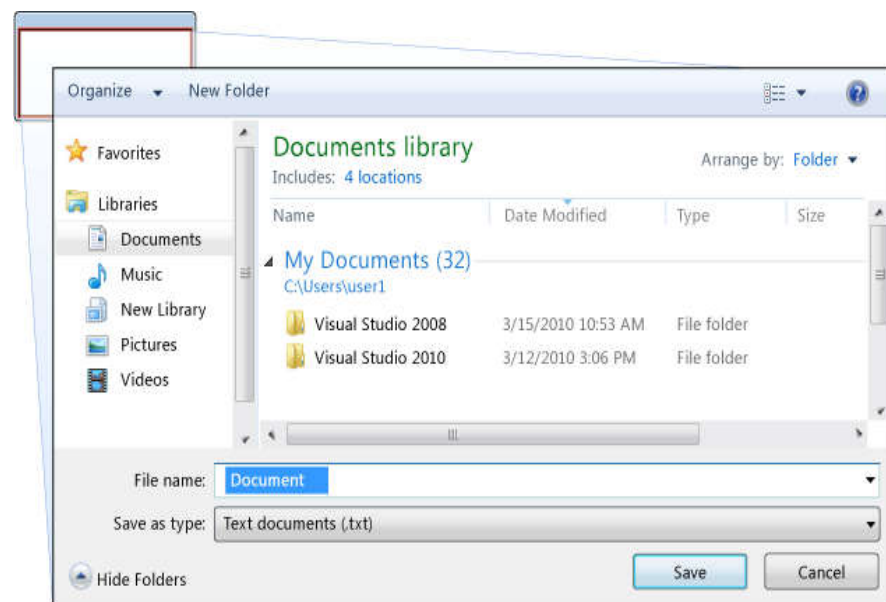


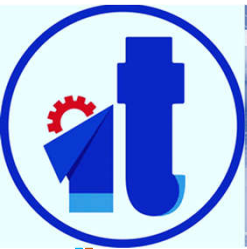
Dialogs

□ Common Dialog Boxes

➤ **Save File Dialog:** hiển thị hộp thoại cho phép người dùng chỉ định tên tập tin để lưu trữ

- ✓ **FileName:** đường dẫn đến tập tin được chọn
- ✓ **ShowDialog():** hiển thị hộp thoại và trả về kết quả
 - OK: click vào nút Save
 - Cancel: click vào nút Cancel



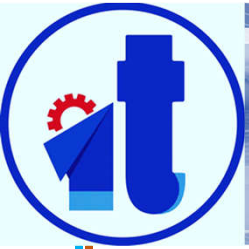


Dialogs

□ Common Dialog Boxes

➤ Save File Dialog

- ✓ **OverwritePrompt**: cho phép hiển thị hộp thoại thông báo ghi đè tập tin đã tồn tại (True)
- ✓ **Title**: tiêu đề của hộp thoại
- ✓ **InitialDirectory**: đường dẫn hiển thị các tập tin – thư mục trong hộp thoại
- ✓ **Filter**: lọc loại tập tin hiển thị trong hộp thoại



Dialogs

□ Common Dialog Boxes

➤ Save File Dialog

```
// Configure save file dialog box
Microsoft.Win32.SaveFileDialog dlg = new Microsoft.Win32.SaveFileDialog();
dlg.FileName = "Document"; // Default file name
dlg.DefaultExt = ".text"; // Default file extension
dlg.Filter = "Text documents (.txt)|*.txt"; // Filter files by extension

// Show save file dialog box
Nullable<bool> result = dlg.ShowDialog();

// Process save file dialog box results
if (result == true)
{
    // Save document
    string filename = dlg.FileName;
}
```



Dialogs

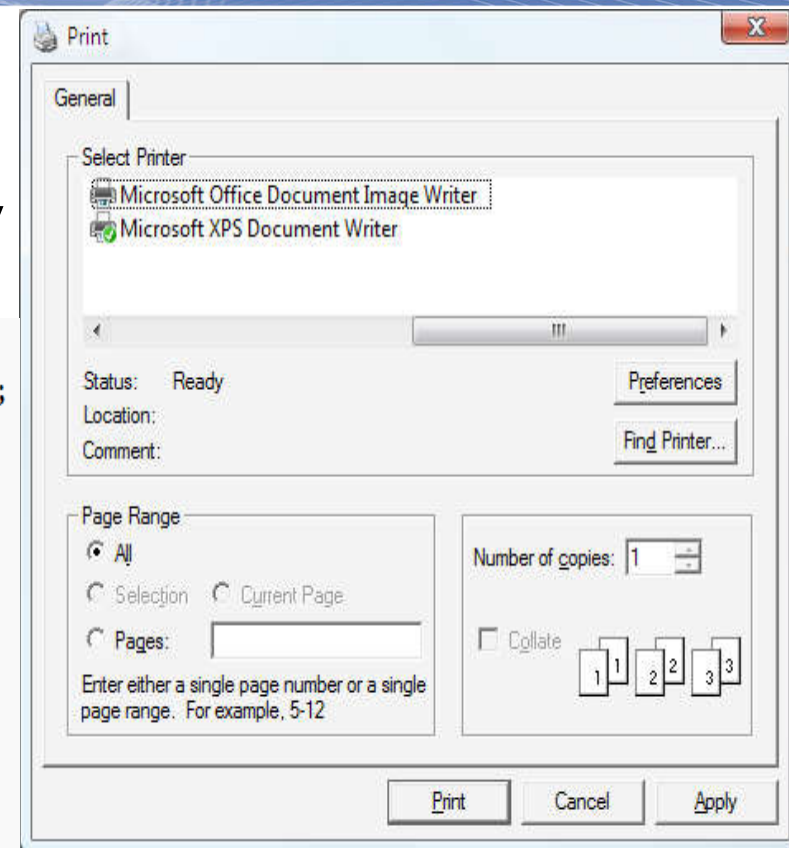
□ Common Dialog Boxes

- **Print Dialog:** hiển thị hộp thoại cho phép người dùng chọn và cấu hình máy in cho việc in ấn dữ liệu

```
// Configure printer dialog box
System.Windows.Controls.PrintDialog dlg = new System.Windows.Controls.PrintDialog();
dlg.PageRangeSelection = PageRangeSelection.AllPages;
dlg.UserPageRangeEnabled = true;

// Show save file dialog box
Nullable<bool> result = dlg.ShowDialog();

// Process save file dialog box results
if (result == true)
{
    // Print document
}
```



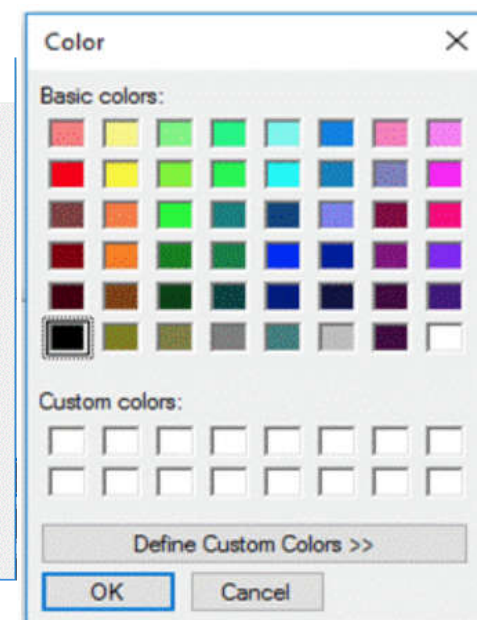
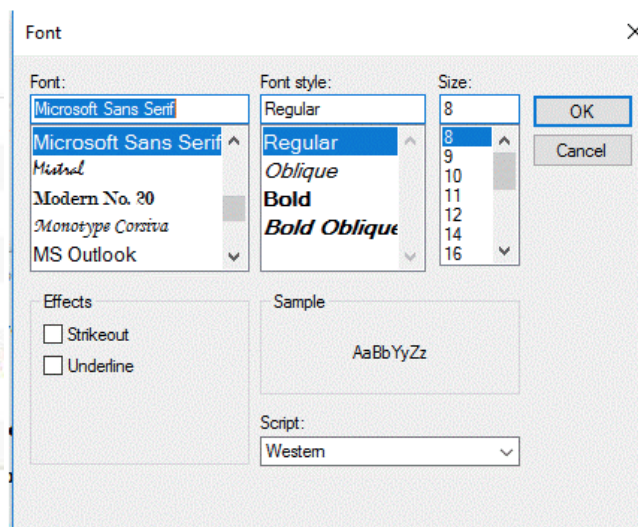


Dialogs

□ Common Dialog Boxes

- **Font - Color Dialog:** hiển thị hộp thoại cho phép người dùng chọn font chữ hay màu sắc chữ

```
//Font Dialog
FontDialog fdlg = new FontDialog();
fdlg.ShowDialog();
txtEx.Font = fdlg.Font;
//Color Dialog
ColorDialog cdlg = new ColorDialog();
cdlg.ShowDialog();
txtEx.ForeColor = cdlg.color;
```





Dialogs

❑ Custom Dialog Box

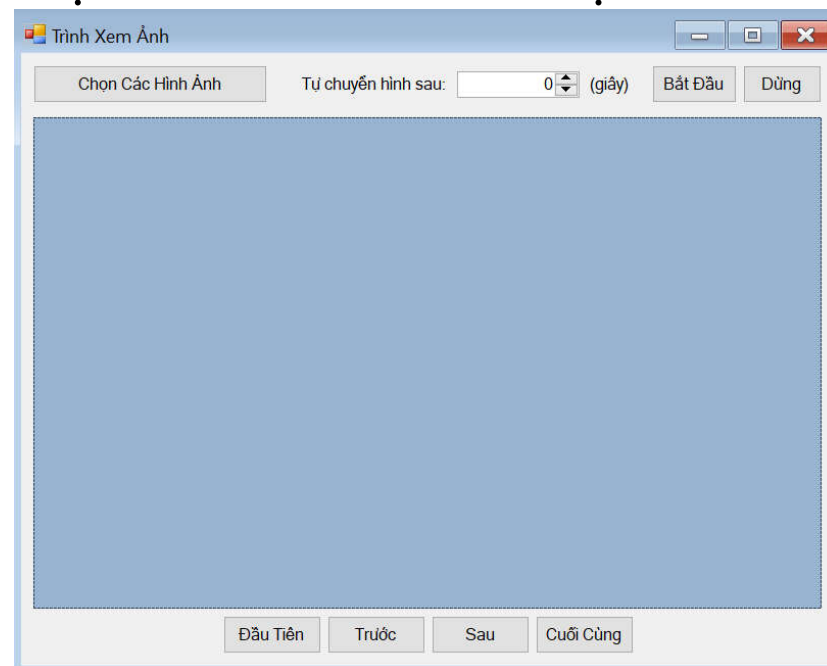
- Chương trình cần có những Dialog đặc thù riêng với những hành vi xử lý riêng biệt.
- Cách tạo
 - ✓ Thêm một form vào project
 - ✓ Vào thuộc tính form, thay đổi thuộc tính
`FormBorderStyle = FixedDialog`
 - ✓ Thiết kế giao diện Dialog thông qua form
 - ✓ Hiển thị form bằng hàm **ShowDialog()**

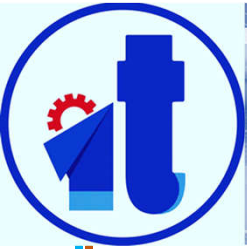


Bài tập

❑ Chọn các hình ảnh trên máy tính và sử dụng các Button (**Đầu Tiên, Trước, Sau, Cuối Cùng**) để chọn hình ảnh hiển thị trên PictureBox

➤ khi kích hoạt timer thì lần lượt hiển thị các hình ảnh lên PictureBox sau thời gian (?giây) định trước (vô hiệu hóa các button **Đầu Tiên, Trước, Sau, Cuối Cùng**)





MDI

- ❑ MDI (**M**ultiple **D**ocument **I**nterface): là giao diện lập trình cho phép người dùng làm việc với nhiều cửa sổ cùng lúc.
- ❑ Mỗi cửa sổ là một không gian riêng biệt, người dùng có thể làm việc với các cửa sổ khác nhau bằng việc di chuyển con trỏ từ cửa sổ này đến cửa sổ khác
- ❑ Ứng dụng MDI có duy nhất một cửa sổ cha (**MDI Parent**) và có thể có nhiều cửa sổ con (**MDI Child**).
- ❑ MDI thường kết hợp với menu danh mục để chuyển đổi các giao diện cửa sổ



MDI

❑ Các ưu điểm:

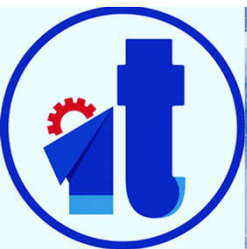
- Các cửa sổ con được quản lý bởi một cửa sổ cha duy nhất
- Hệ thống menu, thanh công cụ được sử dụng chung cho các cửa sổ con
- Đóng cửa sổ cha, các cửa sổ cha cũng được đóng



MDI

❑ Thuộc tính của MDI Parent

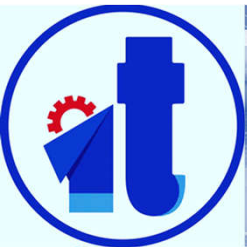
Thuộc tính	KDL	Mô tả
ActiveMdiChild	Form	Trả về MDI Child Form đang kích hoạt
IsMdiContainer	bool	Xác định form là MDI Parent hoặc không
MdiChildren	Form[]	Trả về mảng chứa các MDI Child Form



MDI

❑ Thuộc tính của MDI Child

Thuộc tính	KDL	Mô tả
IsMdiChild	bool	Xác định form là MDI Child hoặc không
MdiParent	Form	Trả về MDI Parent Form



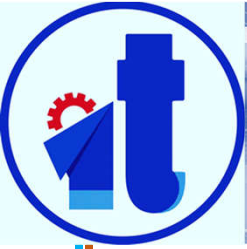
❑ Phương thức

➤ void LayoutMdi(**MdiLayout** value)

- ✓ Xác định cách hiển thị các MDI Child Form
- ✓ Một số giá trị của value:
 - **ArrangeIcons**: sắp xếp các biểu tượng của các Child Form
 - **Cascade**: sắp xếp các cửa sổ Child Form theo dạng hình mái ngói
 - **TileHorizontal**: sắp xếp các cửa sổ Child Form theo chiều ngang
 - **TileVertical**: sắp xếp các cửa sổ Child Form theo chiều dọc

❑ Sự kiện:

➤ **MdiChildActive**: xảy ra khi MDI Child Form được đóng lại hoặc được kích hoạt



MDI

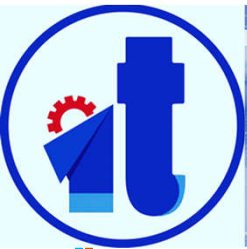
❑ Xét form cha – form con

- Form chính (form cha) sẽ được thiết lập thuộc tính

IsMdiContainer = True

- Form phụ (form con) đã khai báo trong form cha sẽ được gán thuộc tính MdiParent sang form cha

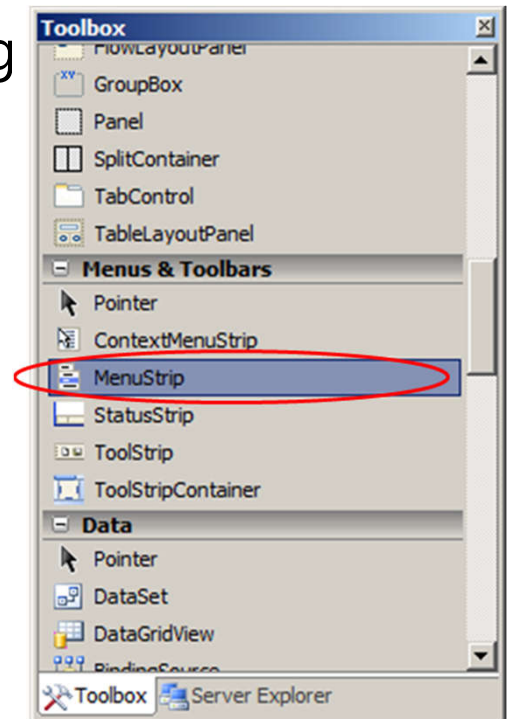
frm.MdiParent = this;

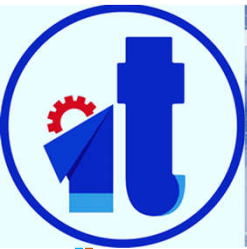


MenuStrip, ContextMenuStrip

❑ MenuStrip

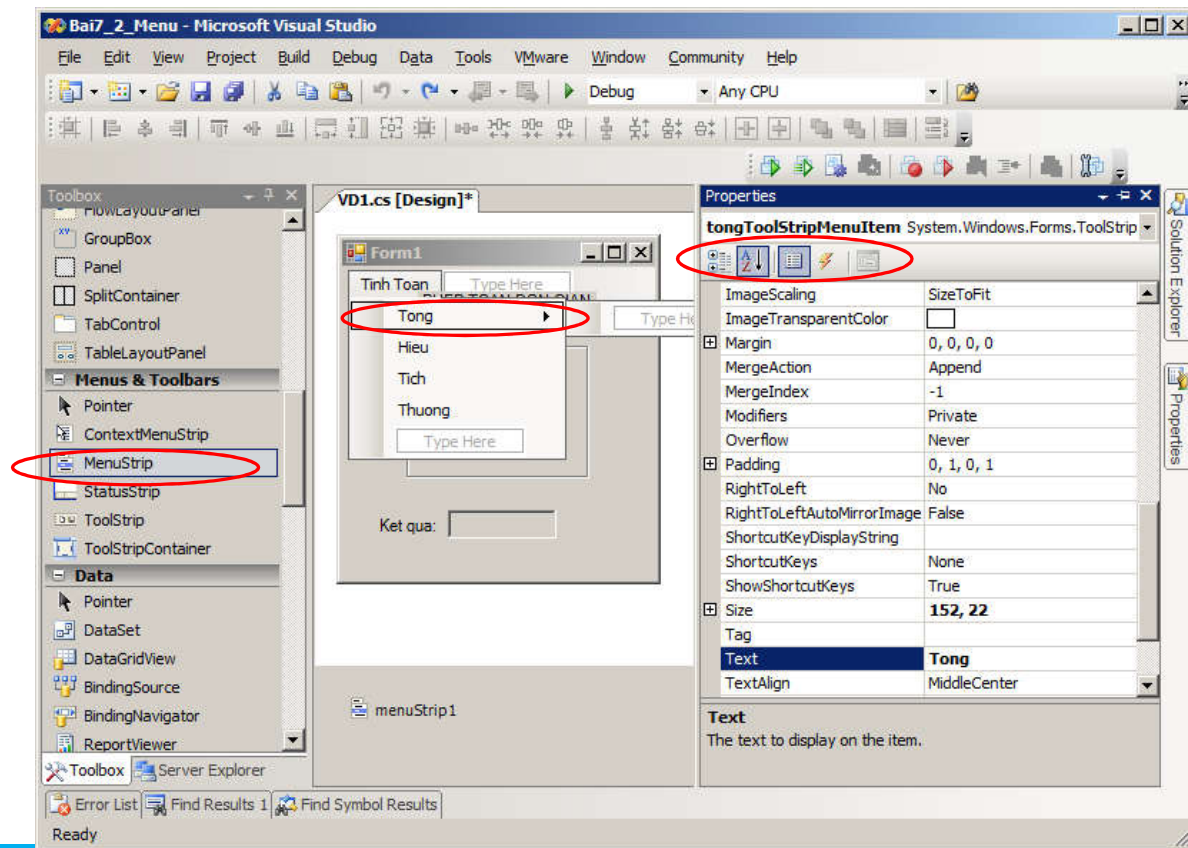
- MenuStrip control dùng để hiển thị các lệnh cũng như gom nhóm các chức năng.
- Thuộc tính quan trọng:
 - ✓ Text
 - ✓ Name
- Sự kiện: **Click**
- Mỗi phần tử trong menustrip:
 - ✓ ToolStripMenuItem
 - ✓ Sự kiện mỗi phần tử: **Click**





MenuStrip, ContextMenuStrip

❑ MenuStrip





MenuStrip, ContextMenuStrip

☐ ContextMenuStrip

- ContextMenuStrip cũng được biết như là popup menu xuất hiện khi click phải phải form hoặc control

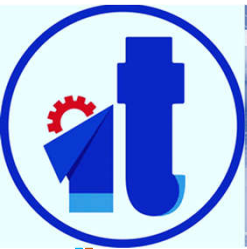
➤ Khởi tạo

```
ContextMenuStrip PopupMenu = new ContextMenuStrip();
```

➤ Thêm ContextMenuStrip vào form

```
<tên form>.ContextMenuStrip = PopupMenu;
```

```
PopupMenu.Show();
```

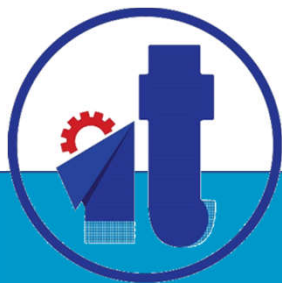


MenuStrip, ContextMenuStrip

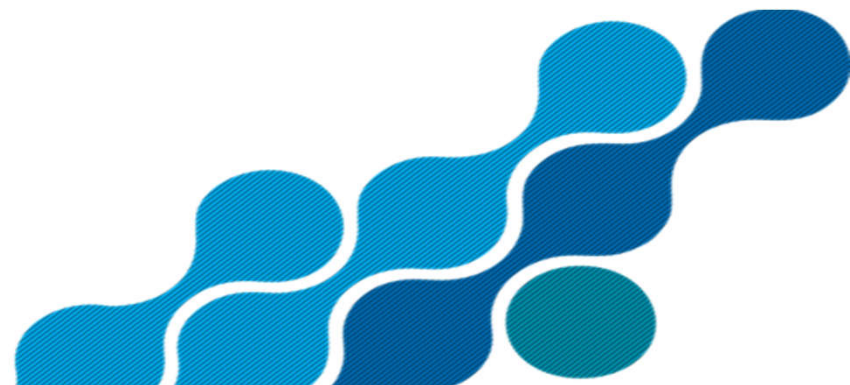
☐ ContextMenuStrip

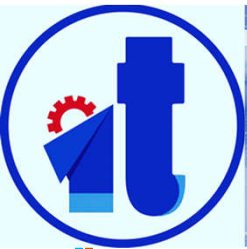
➤ ToolStripMenuItem: như một đối tượng menu

```
// Create a Menu Item
ToolStripMenuItem FileMenu = new ToolStripMenuItem("File");
FileMenu.BackColor = Color.OrangeRed;
FileMenu.ForeColor = Color.Black;
FileMenu.Text = "File Menu";
FileMenu.Font = new Font("Georgia", 16);
FileMenu.TextAlign = ContentAlignment.BottomRight;
FileMenu.ToolTipText = "Click Me";
PopupMenu.Items.Add(FileMenu);
```

Mouse - Keyboard Event





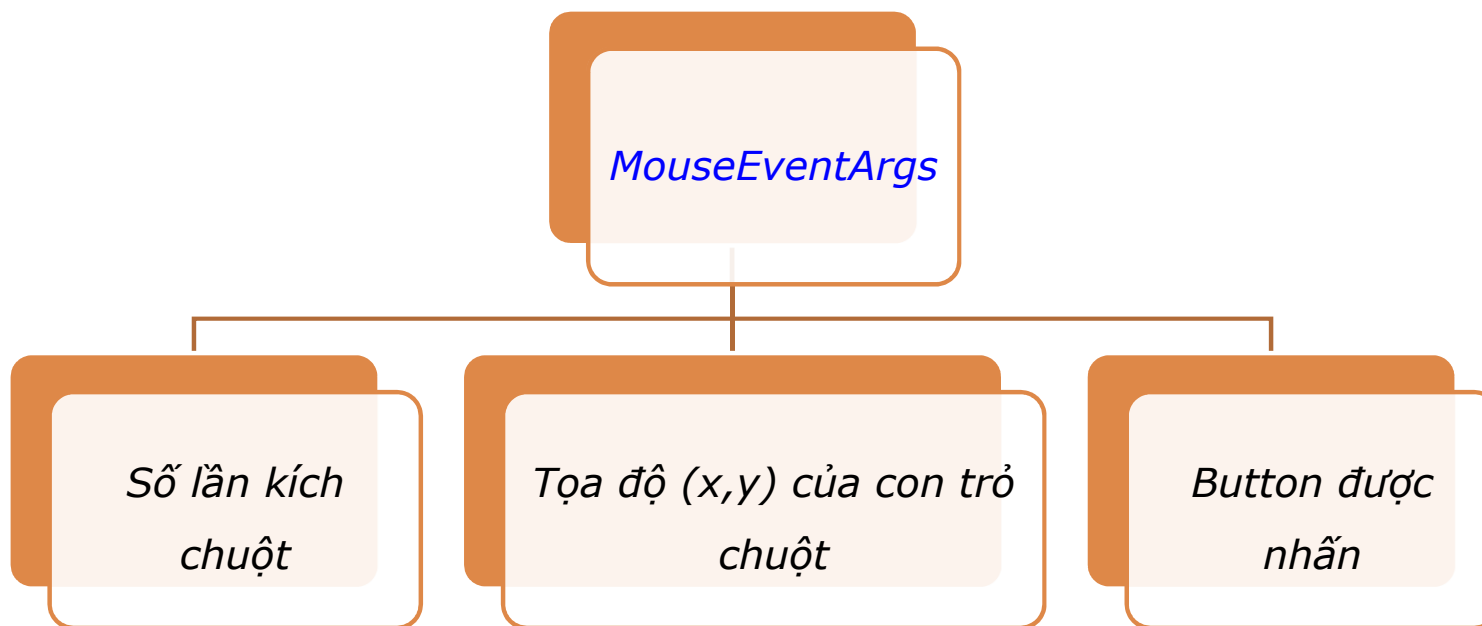
Mouse Event

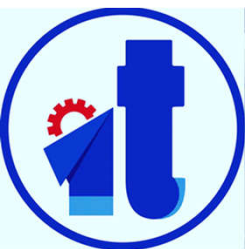
- ❑ Mouse là thiết bị tương tác thông dụng trên GUI
- ❑ Một số các thao tác phát sinh từ mouse
 - Di chuyển, kích chuột
- ❑ Ứng dụng cần xử lý sự kiện chuột nào sẽ khai báo trình xử lý tương ứng
- ❑ Lớp **MouseEventArgs** được sử dụng để chứa thông tin truyền vào cho trình xử lý sự kiện mouse.
- ❑ Mỗi trình xử lý sự kiện sẽ có tham số là đối tượng object và đối tượng **MouseEventArgs** (hoặc **EventArgs**)



Mouse Event

❑ Tham số cho sự kiện liên quan đến mouse





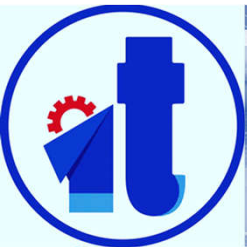
Mouse Event

Sự kiện chuột với tham số kiểu EventArgs

<i>MouseEnter</i>	Xuất hiện khi con trỏ chuột đi vào vùng biên của control
<i>MouseLeave</i>	Xuất hiện khi con trỏ chuột rời khỏi biên của control

Sự kiện chuột với tham số kiểu MouseEventArgs

<i>MouseDown/ MouseUp</i>	Xuất hiện khi button được nhấn/thả và con trỏ chuột đang ở trong vùng biên của control
<i>MouseMove</i>	Xuất hiện khi chuột di chuyển và con trỏ chuột ở trong vùng biên của control



Mouse Event

Thuộc tính của lớp MouseEventArgs

<i>Button</i>	Button được nhấn {Left, Right, Middle, none} có kiểu là MouseButton
<i>Clicks</i>	Số lần button được nhấn
<i>X</i>	Tọa độ x của con trỏ chuột trong control
<i>Y</i>	Tọa độ y của con trỏ chuột trong control



Keyboard Event

- ☐ Phát sinh khi một phím được nhấn hoặc thả
- ☐ Có 3 sự kiện
 - KeyPress
 - KeyUp
 - KeyDown
- ☐ KeyPress phát sinh kèm theo với mã ASCII của phím được nhấn
- ☐ KeyPress không cho biết trạng thái các phím bổ sung {Shift, Alt, Ctrl...}
- ☐ Sử dụng KeyUp & KeyDown để xác định trạng thái các phím bổ sung.



Keyboard Event

Sự kiện với tham số kiểu KeyEventArgs

<i>KeyDown</i>	Phát sinh khi phím được nhấn
<i>KeyUp</i>	Phát sinh khi phím được thả

Sự kiện với tham số kiểu KeyPressEventArgs

<i>KeyPress</i>	Khởi tạo khi phím được nhấn
-----------------	-----------------------------

Thuộc tính của lớp KeyPressEventArgs

<i>KeyChar</i>	Chứa ký tự ASCII của phím được nhấn
<i>Handled</i>	Cho biết sự kiện KeyPress có được xử lý chưa

Thuộc tính của lớp KeyEventArgs

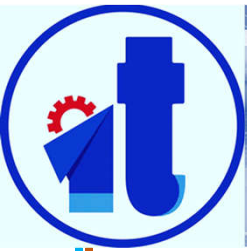
<i>Alt, Control, Shift</i>	Trạng thái các phím bổ sung
<i>Handled</i>	Cho biết sự kiện đã xử lý



Keyboard Event

Thuộc tính của lớp KeyEventArgs (tt)

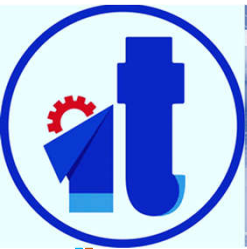
<i>KeyCode</i>	Trả về mã ký tự được định nghĩa trong Keys enumeration
<i>KeyData</i>	Chứa mã ký tự với thông tin phím bổ sung
<i>KeyValue</i>	Trả về số int, đây chính là mã Windows Virtual Key Code
<i>Modifier</i>	Trả về giá trị của phím bổ sung



Keyboard Event

□ Keys Enumeration

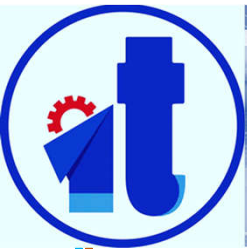
Key	Key Code	Description
A	65	The A key.
Add	107	The add key.
Alt	262144	The ALT modifier key.
Apps	93	The application key (Microsoft Natural Keyboard).
Attn	246	The ATTN key.
B	66	The B key.
Back	8	The BACKSPACE key.



Keyboard Event

□ Keys Enumeration

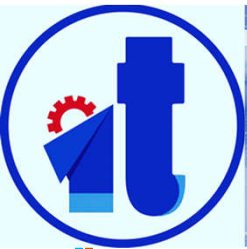
Key	Key Code	Description
BrowserBack	166	The browser back key (Windows 2000 or later).
BrowserFavorites	171	The browser favorites key (Windows 2000 or later).
BrowserForward	167	The browser forward key (Windows 2000 or later).
BrowserHome	172	The browser home key (Windows 2000 or later).
BrowserRefresh	168	The browser refresh key (Windows 2000 or later).
BrowserSearch	170	The browser search key (Windows 2000 or later).
BrowserStop	169	The browser stop key (Windows 2000 or later).



Keyboard Event

□ Keys Enumeration

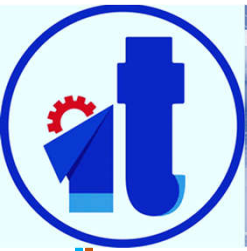
Key	Key Code	Description
C	67	The C key.
Cancel	3	The CANCEL key.
Capital	20	The CAPS LOCK key.
CapsLock	20	The CAPS LOCK key.
Clear	12	The CLEAR key.
Control	131072	The CTRL modifier key.
ControlKey	17	The CTRL key.



Keyboard Event

□ Keys Enumeration

Key	Key Code	Description
Crssel	247	The CRSEL key.
D	68	The D key.
D0	48	The 0 key.
D1	49	The 1 key.
D2	50	The 2 key.
D3	51	The 3 key.
D4	52	The 4 key.



Keyboard Event

□ Keys Enumeration

Key	Key Code	Description
D5	53	The 5 key.
D6	54	The 6 key.
D7	55	The 7 key.
D8	56	The 8 key.
D9	57	The 9 key.
Decimal	110	The decimal key.
Delete	46	The DEL key.

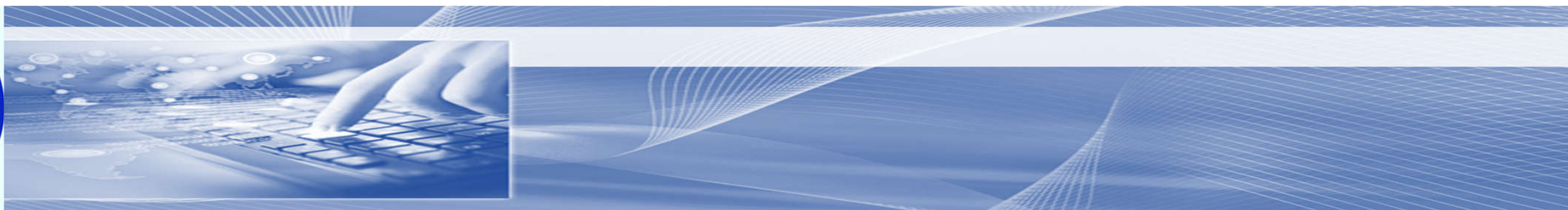
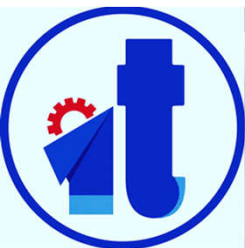


Keyboard Event

□ Keys Enumeration

Key	Key Code	Description
D5	53	The 5 key.
D6	54	The 6 key.
D7	55	The 7 key.
D8	56	The 8 key.
D9	57	The 9 key.
....		

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.keys?view=netframework-4.8>



Q&A

