

CHƯƠNG 5: ĐIỀU KHIỂN DÙNG ĐỂ XÂY DỰNG MENU

5.1. Điều khiển ImageList

ImageList là một kiểu collection đặc biệt chứa các hình có kích thước và độ sâu màu được xác định trước. Các điều khiển khác nếu có hỗ trợ dùng *ImageList* thì dùng các hình trong *ImageList* thông qua chỉ mục. Một số điều khiển hỗ trợ sử dụng *ImageList* như: *ListView*, *TreeView*, *ToolBar*, *Button*, ... *ImageList* nằm trong nhóm Components của cửa sổ Toolbox như hình 5.1.

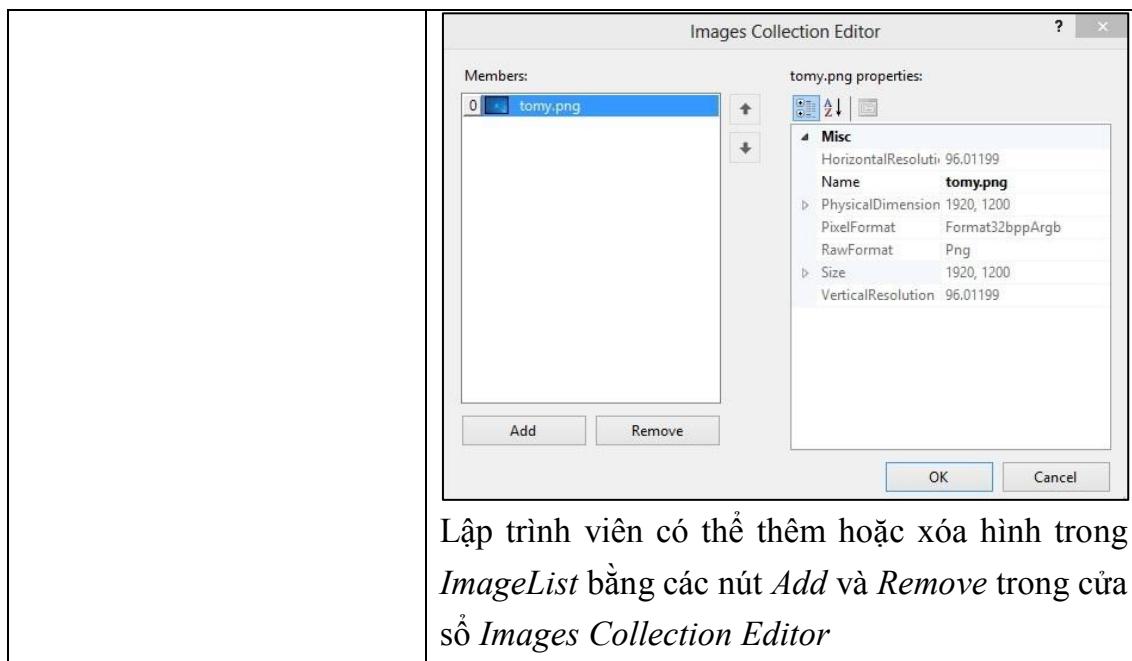


Hình 5.1: Điều khiển *ImageList* trong cửa sổ *Toolbox*

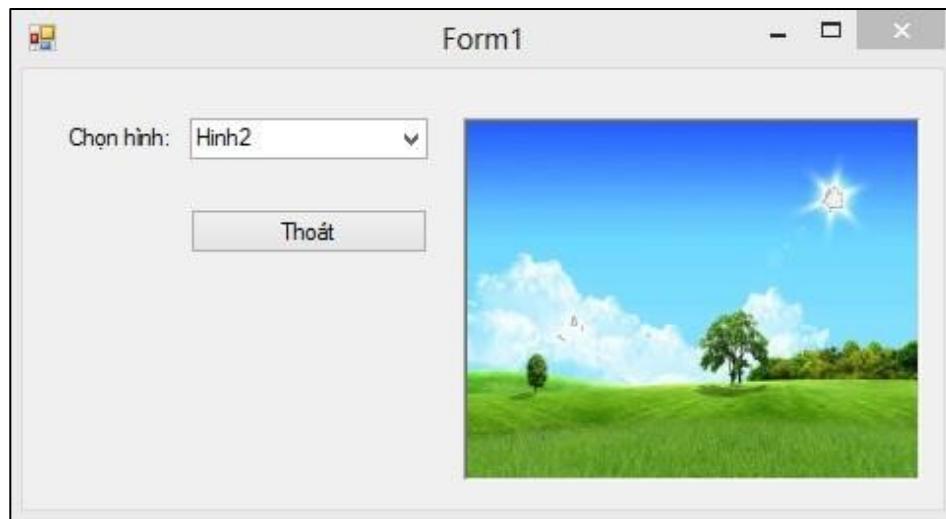
- Một số thuộc tính thường dùng của *ImageList*:

Bảng 5.1: Bảng mô tả các thuộc tính của *ImageList*

Thuộc tính	Mô tả
<i>ImageSize</i>	Kích thước của hình
<i>TransparentColor</i>	Định nghĩa độ trong suốt của màu
<i>ColorDepth</i>	Thiết lập độ sâu của hình được chứa trong <i>ImageList</i>
<i>Images</i>	Tập các hình chứa trong <i>ImageList</i> . Cửa sổ <i>Images Collection Editor</i> khi nhấp chuột chọn thuộc tính <i>Images</i> trong cửa sổ <i>Properties</i> :



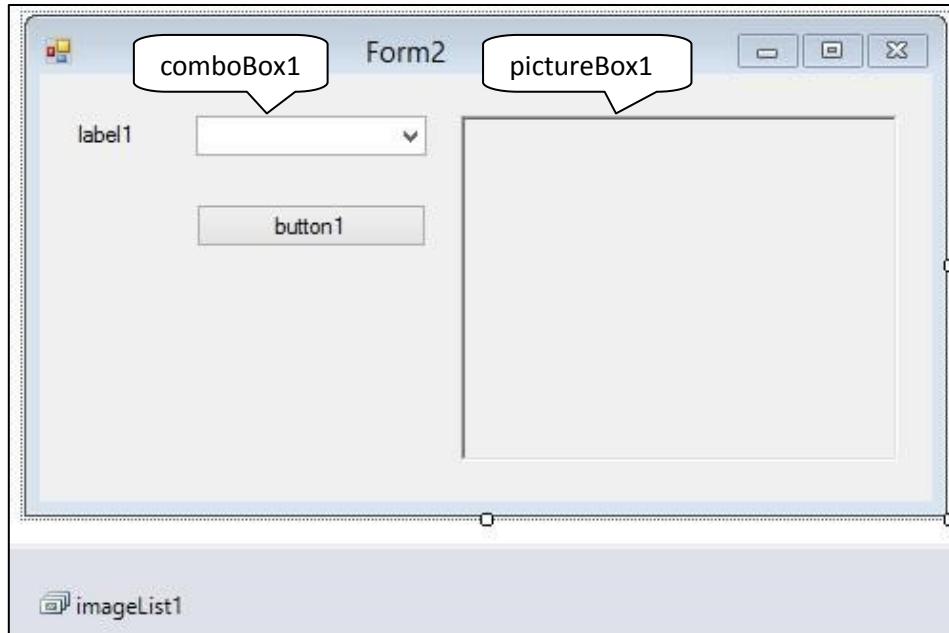
Ví dụ 5.1: Thiết kế giao diện chương trình như hình 5.2. Yêu cầu: Khi chọn hình muốn hiển thị trên ComboBox thì ở điều khiển PictureBox sẽ hiển thị hình tương ứng như đã chọn.



Hình 5.2: Giao diện form hiển thị hình ví dụ 5.1

Hướng dẫn:

Bước 1: Thiết kế giao diện chương trình ban đầu như hình 5.3



Hình 5.3: Giao diện ban đầu form hiển thị hình

Bước 2: Thiết lập giá trị thuộc tính cho các điều khiển trong cửa sổ Properties

- label1:

Thuộc tính *Text*: “Chọn hình:”

- comboBox1:

Thuộc tính *Name*: cboChonHinh

- button1:

Thuộc tính *Text*: “Thoát”

Thuộc tính *Name*: btnThoat

- pictureBox1: //PictureBox là điều khiển dùng để hiển thị hình ảnh trên Form

Thuộc tính *Name*: picHinh

Thuộc tính *Size*: 229, 181

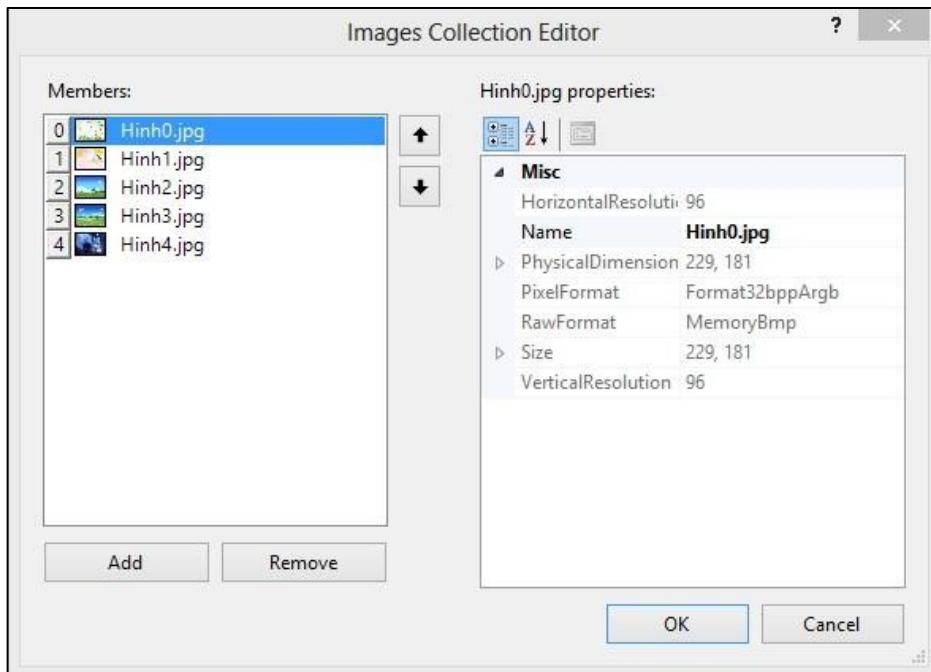
Thuộc tính *BorderStyle*: BorderStyle.FixedSingle

- imageList1:

Thuộc tính *Name*: MyImgList

Thuộc tính *ImageSize*: 229, 181

Thuộc tính *Images*: như hình 5.4



Hình 5.4: Thiết lập thuộc tính Images của ImageList1

Bước 3: Viết mã lệnh cho các điều khiển

- Sự kiện *Load* của Form1:

```
private void Form1_Load(object sender, EventArgs e)
{
    for (int i = 0; i < MyImgList.Images.Count; i++)
    {
        cboChonHinh.Items.Add("Hinh" + i);
    }
    picHinh.BorderStyle=BorderStyle.FixedSingle
}
```

- Sự kiện *SelectedIndexChanged* của nút cboHienThi:

```
private void cboChonHinh_SelectedIndexChanged(object sender,
                                         EventArgs e)
{
    //Thuộc tính Image của PictureBox là thuộc tính chỉ định
    //hình sẽ được hiển thị
    picHinh.Image=
        MyImgList.Images[cboChonHinh.SelectedIndex];
}
```

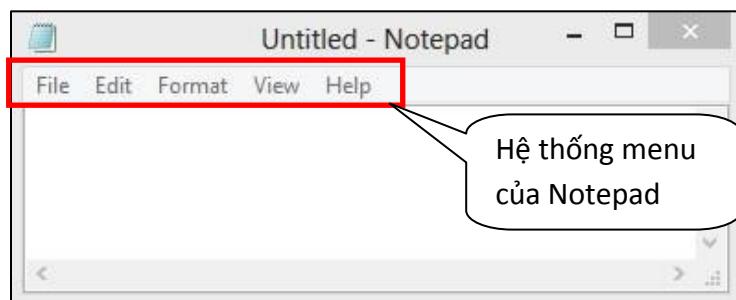
- Sự kiện *Click* của nút *btnThoat*:

```
private void btnThoat_Click(object sender, EventArgs e)
{
    Close();
}
```

5.2. Điều khiển ToolStrip

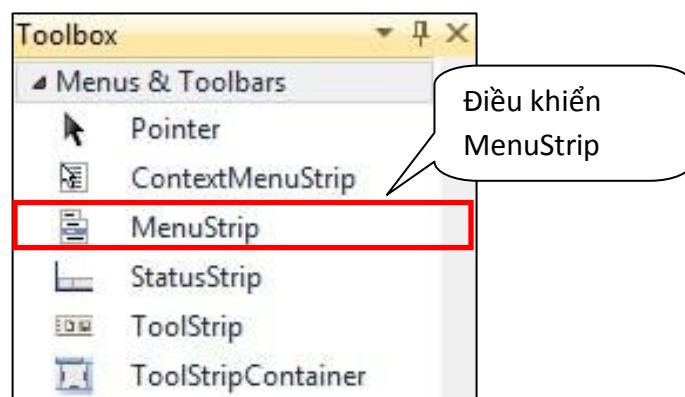
Điều khiển ToolStrip cho phép lập trình viên xây dựng hệ thống menu trên form. Menu có thể xây dựng ở dạng một cấp hoặc nhiều cấp. Ví dụ menu của Notepad như hình 5.5. ToolStrip cho phép xây dựng menu với các điều khiển:

- ToolStripSeparator
- ToolStripMenuItem (Menu con)
- ToolStripCombobox (Combobox)
- ToolStripTextbox (Textbox)



Hình 5.5: Menu của trình soạn thảo Notepad

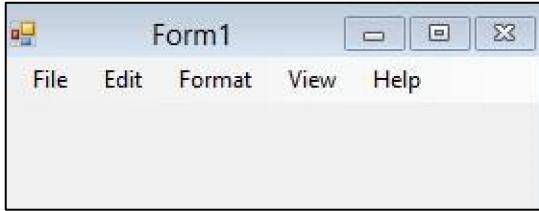
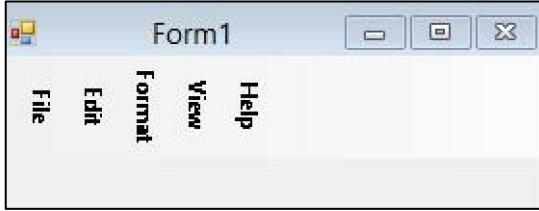
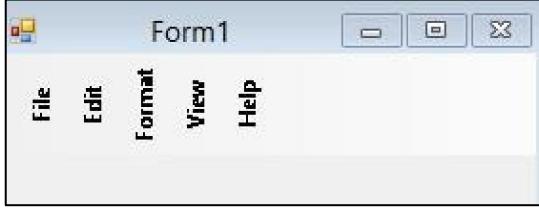
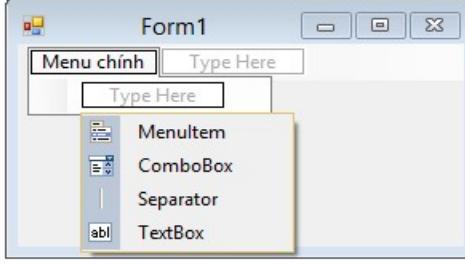
MenuStrip được đặt trong nhóm Menus & Toolbars của cửa sổ Toolbox như hình 5.6.

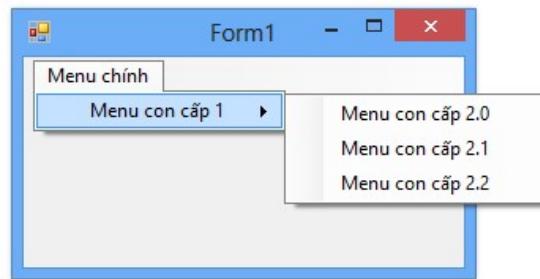


Hình 5.6: Điều khiển ToolStrip trong cửa sổ Toolbox

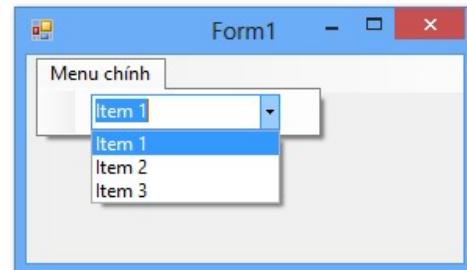
- Một số thuộc tính thường dùng của *MenuStrip*:

*Bảng 5.2: Bảng mô tả các thuộc tính của *MenuStrip**

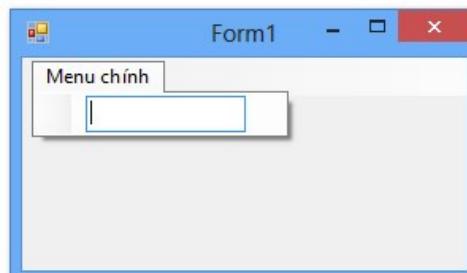
Thuộc tính	Mô tả
<i>TextDirection</i>	<p>Chọn hình thức trình bày menu.</p> <ul style="list-style-type: none"> - Hình thức Horizontal:  - Hình thức Vertical 90:  - Hình thức Vertical 270: 
<i>Items</i>	<p>Thêm các menu con. Kiểu menu có thể chọn một trong 4 dạng: MenuItem, ComboBox, Separator, TextBox.</p>  <ul style="list-style-type: none"> - Dạng MenuItem (menu con):



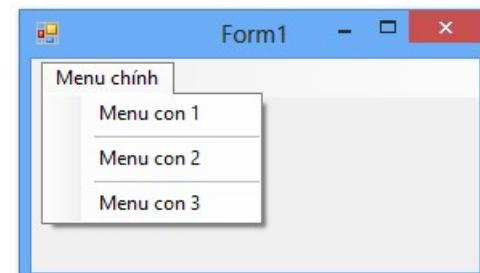
- Dạng Combobox:



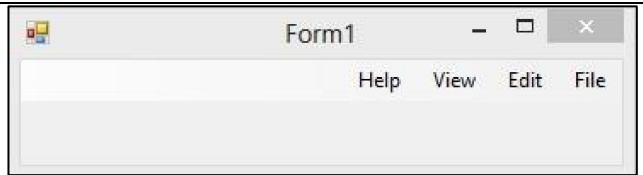
- Dạng TextBox:



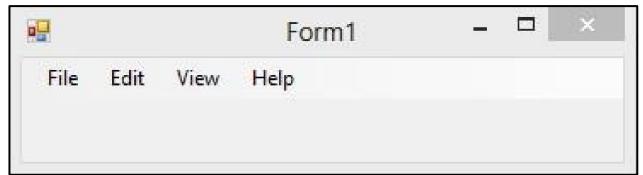
- Separator (Gạch phân cách):



<i>RightToLeft</i>	Mang giá trị Yes hoặc No. Nếu là Yes thì trình bày menu từ phải qua trái:
--------------------	--



Nếu là No thì trình bày menu từ trái qua phải:

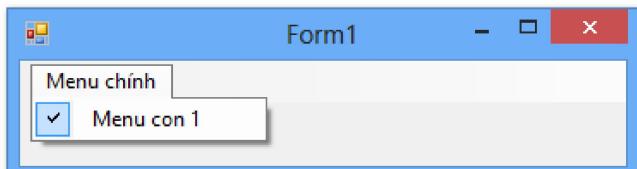
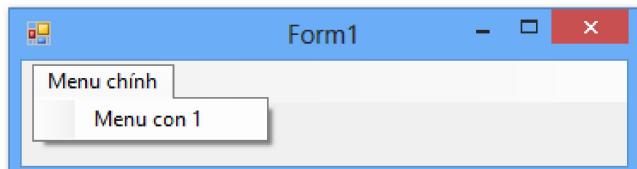


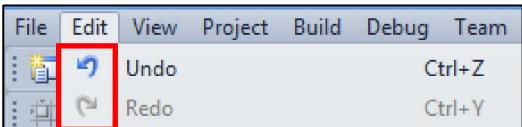
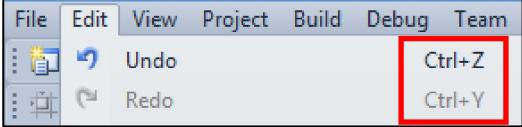
- Một số thuộc tính, phương thức và sự kiện của menu con trên *MenuStrip*:

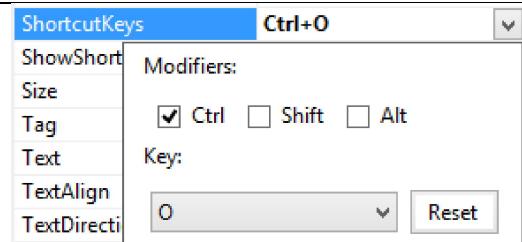
MenuStrip có 4 dạng menu con, mỗi menu con đều có thuộc tính, phương thức và sự kiện riêng tương ứng với mỗi dạng.

- Menu con dạng *MenuItem*:

Bảng 5.3: Bảng mô tả các thuộc tính thường dùng của MenuItem

Thuộc tính	Mô tả
<i>Checked</i>	<p>Mang giá trị True hoặc False.</p> <ul style="list-style-type: none"> - Nếu là True: Hiện biểu tượng CheckBox bên cạnh chuỗi Text  <ul style="list-style-type: none"> - Nếu là False: Không hiển thị CheckBox 
<i>CheckOnClick</i>	<p>Mang giá trị True hoặc False.</p> <ul style="list-style-type: none"> - Nếu là True: Biểu tượng CheckBox sẽ xuất hiện bên cạnh chuỗi Text của menu con khi người dùng nhấp chuột chọn. - Nếu là False: Thao tác nhấp chuột của người dùng sẽ không ảnh hưởng gì đến việc

	hiển thị hay không hiển thị của biểu tượng CheckBox
<i>CheckState</i>	Cho biết trạng thái của CheckBox trên menu con. Có 3 trạng thái: UnChecked, Checked, Indeterminate. Lưu ý: Trạng thái Indeterminate chỉ có hiệu lực khi thuộc tính Checked là True.
<i>DisplayStyle</i>	Hình thức trình bày của menu con. Có 4 kiểu hiển thị: <ul style="list-style-type: none"> - None: Không hiển thị gì trên menu con - Text: Cho phép hiển thị chuỗi mô tả - Image: Cho phép hiển thị hình hoặc biểu tượng bên cạnh Text. - ImageAndText: Cho phép hiển thị hình (biểu tượng) và chuỗi mô tả.
<i>Image</i>	Hình ảnh xuất hiện bên cạnh chuỗi Text 
<i>ImageScaling</i>	Kiểu trình bày của hình trong thuộc tính <i>Image</i> . Có thể thiết lập một trong hai giá trị: <ul style="list-style-type: none"> - <i>None</i>: Hiển thị bình thường - <i>SizeToFit</i>: Hiển thị đúng kích cỡ của hình hoặc biểu tượng
<i>ShortcutKeyDisplayString</i>	Chuỗi trình bày ứng với phím tắt mô tả cho menu đó. 
<i>ShorcutKeys</i>	Tổ hợp phím tắt ứng với menu

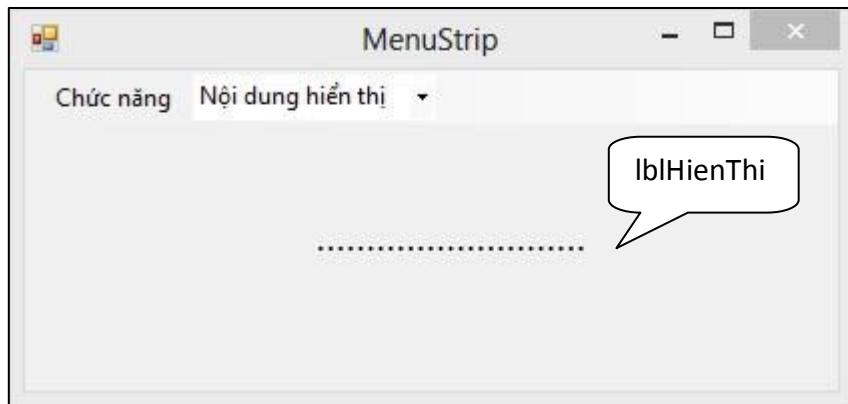
	
ShowShortcutKeys	Mang giá trị <i>True</i> hoặc <i>False</i> : <ul style="list-style-type: none"> - Nếu là <i>True</i>: Cho phép hiển thị giá trị trong thuộc tính <i>ShortcutKeyDisplayString</i>. - Nếu là <i>False</i>: giá trị trong thuộc tính <i>ShortcutKeyDisplayString</i> sẽ không hiển thị.
Text	Chuỗi ký tự hiển thị trên menu
ToolTipText	Chuỗi ký tự hiển thị khi rê chuột vào menu

Bảng 5.4: Bảng mô tả các sự kiện thường dùng của MenuItem

Thuộc tính	Mô tả
CheckedChange	Phát sinh khi trạng thái (<i>CheckState</i>) của <i>CheckBox</i> thay đổi
Click	Phát sinh khi người dùng nhấp chuột vào menu

- Menu con dạng *Separator*: Menu dạng này đơn thuần chỉ là một đường kẻ ngang giúp ngăn cách các menu với nhau, giúp hệ thống menu hiển thị một cách rõ ràng hơn.
- Menu con dạng *ComboBox*: Các thuộc tính, phương thức và sự kiện giống với điều khiển *ComboBox* tại mục 3.4 của chương 3.
- Menu con dạng *TextBox*: Các thuộc tính, phương thức và sự kiện giống với điều khiển *TextBox* tại mục 3.3 của chương 3.

Ví dụ 5.2: Thiết kế giao diện chương trình hiển thị thời gian như hình 5.7.



Hình 5.7: Giao diện chương trình hiển thị thời gian ví dụ 5.2

Yêu cầu:

Menu **Chức năng**: Chức mục **Thoát** dạng MenuItem. Khi người dùng nhấn chuột trái vào **Thoát** hoặc nhấn tổ hợp phím **Ctrl + T** sẽ thoát chương trình.

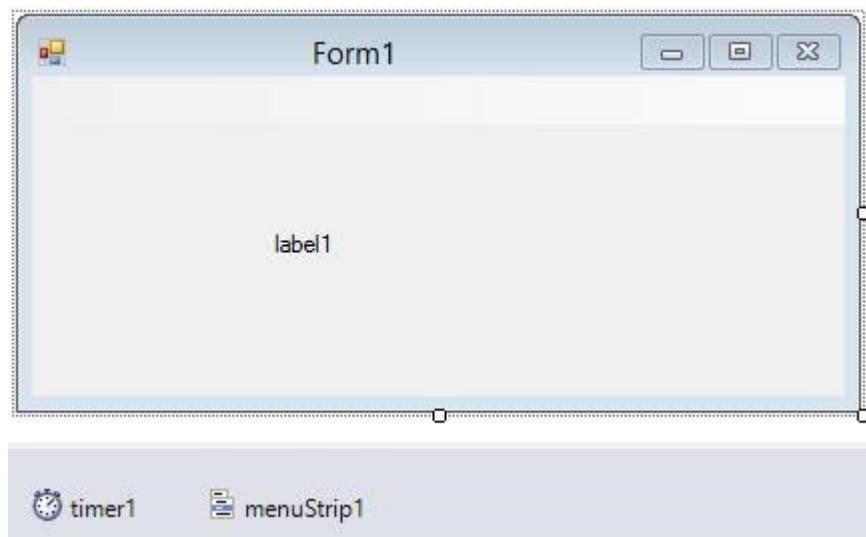
Menu **Nội dung hiển thị**: Dạng ComboBox. ComboBox chứa hai mục chọn:

Hiển thị thời gian: Giúp hiển thị giờ phút giây trên *label* **lblHienThi**

Hiển thị ngày tháng: Giúp hiển thị ngày tháng năm trên *label* **lblHienThi**

Hướng dẫn:

Bước 1: Thiết kế giao diện form ban đầu: Kéo các điều khiển từ cửa sổ Toolbox và form1 như hình 5.8.



Hình 5.8: Giao diện ban đầu form, hiển thị thời gian ví dụ 5.2

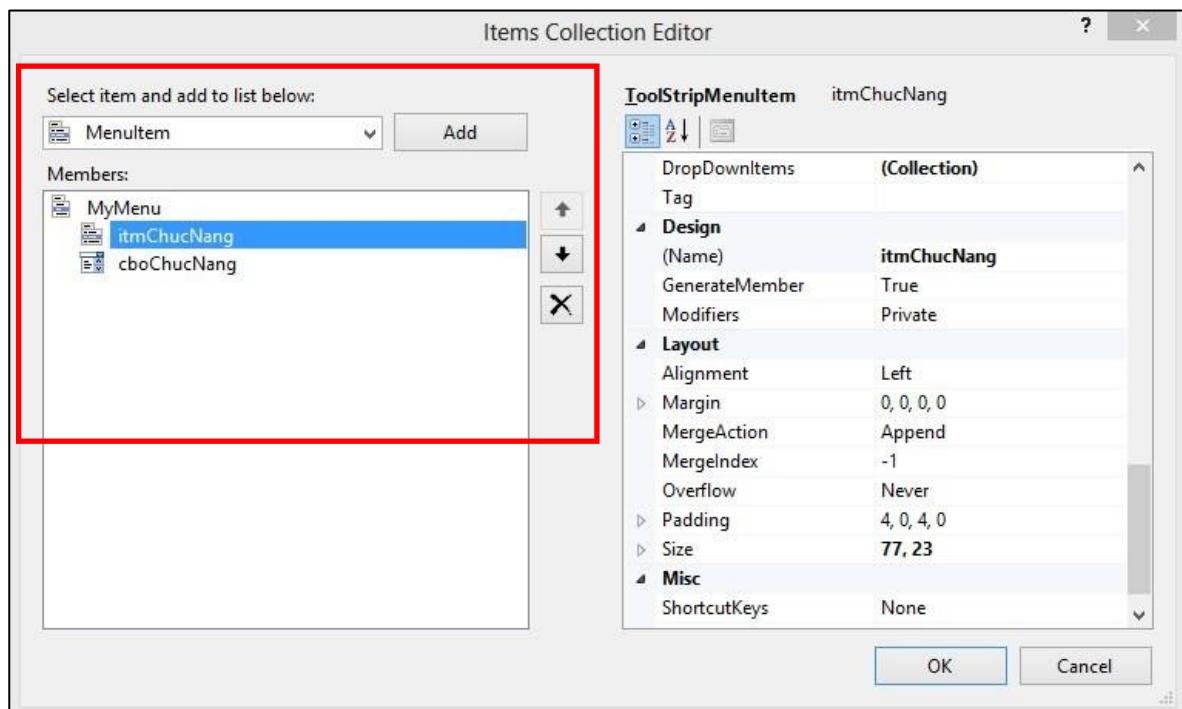
Bước 2: Thiết lập các giá trị cho điều khiển trong cửa sổ Properties

- form1:

Thuộc tính *Text*: “MenuStrip”

- label1:
Thuộc tính *Text*: “.....”
Thuộc tính *Size*: 14
Thuộc tính *Name*: lblHienThi
- timer:
Thuộc tính *Name*: MyTime
Thuộc tính *Enable*: True
Thuộc tính *Interval*: 1000
- menuStrip1:
Thuộc tính *Name*: MyMenu

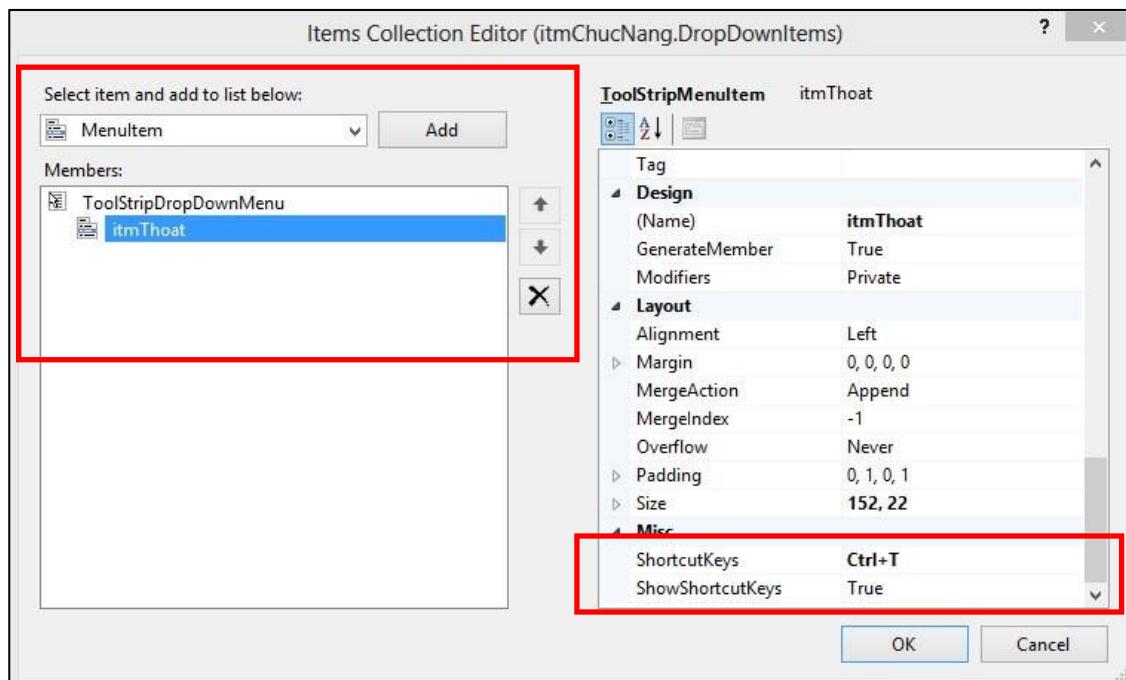
Thuộc tính *DropDownItems* của **MyMenu**: Chọn thuộc tính *DropDownItems* trong cửa sổ Properties sẽ hiển thị cửa sổ Items Collection Editor của **MyMenu**. Tại cửa sổ này thêm hai menu có tên: **itmChucNang** (dạng MenuItem) và **cboChucNang** (dạng ComboBox) như hình 5.9



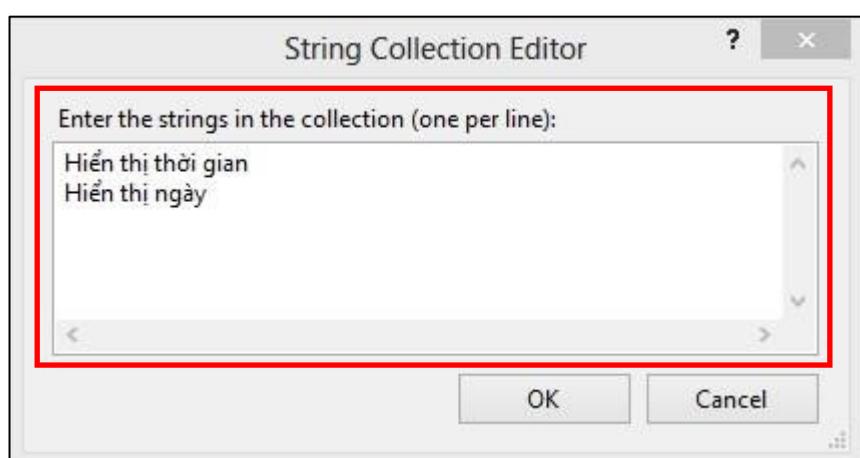
Hình 5.9: Cửa sổ Items Collection Editor

- Nhấn chuột chọn **itmChucNang** để thiết lập thuộc tính: Trên các thuộc tính của **itmChucNang**, chọn thuộc tính *DropDownItems* sẽ hiển thị cửa sổ Items Collection Editor của **itmChucNang**. Trên cửa sổ này thêm một menu dạng MenuItem có tên **itmThoat** như hình 5.10.

- Thiết lập thuộc tính *ShortCutKeys* của **itmThoat**: Ctrl + T
 Thiết lập thuộc tính *ShortCutKeyDisplayString* của **itmThoat**: Ctrl + T
 Thiết lập thuộc tính *ShowShortCutKeys* của **itmThoat**: True
- Nhấn chuột **cboChucNang** để thiết lập thuộc tính: Trên các thuộc tính của **cboChucNang**, chọn thuộc tính *Items* và thêm hai mục chọn: “Hiển thị thời gian”, “Hiển thị ngày tháng” như hình 5.11.



Hình 5.10: Cửa sổ Items Collection Editor của itmChucNang



Hình 5.11: Cửa sổ String Collection Editor của cboChucNang

Bước 3: Viết mã lệnh cho các điều khiển

- Khai báo biến:

```
int chon=3;  
DateTime dt = new DateTime();
```

- Sự kiện *Tick* của MyTimer:

```
private void MyTimer_Tick(object sender, EventArgs e)  
{  
    dt = DateTime.Now;  
    if (chon == 0)  
    {  
        lblHienThi.Text = dt.Hour + ":" + dt.Minute + ":"  
                         + dt.Second;  
    }  
}
```

- Sự kiện *Click* của itmThoat:

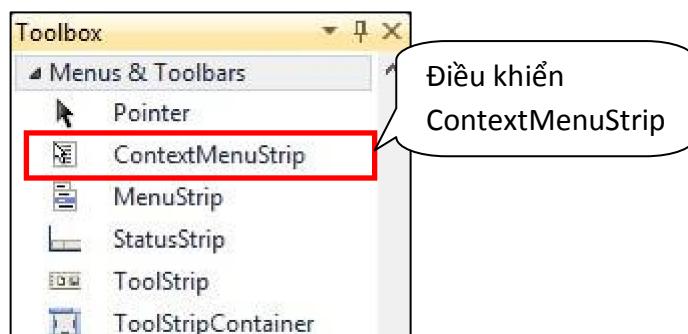
```
private void itmThoat_Click(object sender, EventArgs e)  
{  
    Close();  
}
```

- Sự kiện *SelectedIndexChanged* của cboChucNang:

```
private void cboChucNang_SelectedIndexChanged(object sender,  
                                         EventArgs e)  
{  
    chon = cboChucNang.SelectedIndex;  
    dt = DateTime.Now;  
    if (chon == 1)  
    {  
        lblHienThi.Text = dt.Date.ToString("dd/MM/yyyy");  
    }  
}
```

5.3. Điều khiển ContextMenuStrip

Điều khiển *ContextMenuStrip* dùng để thiết kế menu Popup (menu ngũ cành). Menu Popup là menu dạng như loại menu khi người dùng nhấn chuột phải vào màn hình desktop thì hiện lên một menu. Trong lập trình ứng dụng Windows Form, menu Popup sẽ xuất hiện khi người nhấn chuột phải vào các điều khiển như: *form*, *label*, *button*, *textbox*, ... Điều khiển *ContextMenuStrip* nằm trong nhóm Menus & Toolbars của cửa sổ Toolbox như hình 5.12



Hình 5.12: Điều khiển *ContextMenuStrip* trong cửa sổ *Toolbox*

Phần lớn các điều khiển trong cửa sổ *Toolbox* đều hỗ trợ thuộc tính *ContextMenuStrip*. Do đó muốn menu Popup xuất hiện trên điều khiển nào thì chỉ cần khai báo thuộc tính *ContextMenuStrip* của điều khiển đó là một điều khiển *ContextMenuStrip*.

Giống như điều khiển *MenuStrip*, điều khiển *ContextMenuStrip* cũng cho phép lập trình viên xây dựng menu theo 4 dạng: *MenuItem*, *ComboBox*, *TextBox*, *Separator*. Các thuộc tính, phương thức và sự kiện của 4 dạng menu trên điều khiển *ContextMenuStrip* cũng tương tự như trên điều khiển *MenuStrip* (xem mục 5.2 của chương).

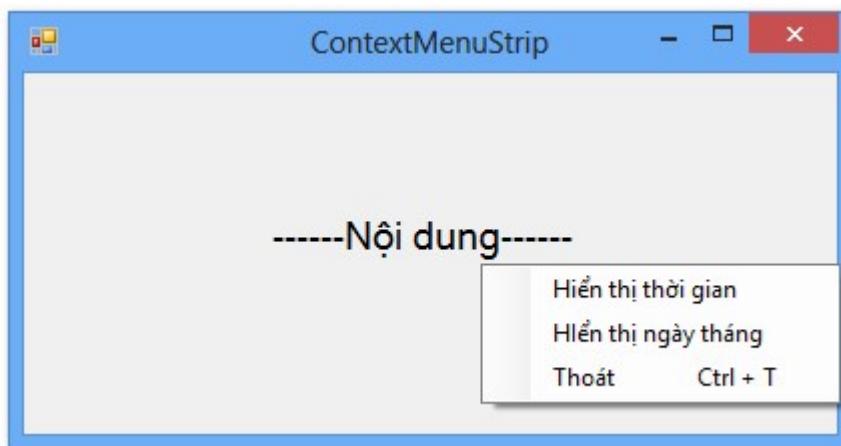
- Một số thuộc tính thường dùng của *ContextMenuStrip*:

Bảng 5.5: Bảng mô tả các thuộc tính thường dùng của *ContextMenuStrip*

Thuộc tính	Mô tả
<i>Items</i>	Thêm các menu con, kiểu menu con thuộc một trong bốn dạng: <i>Separator</i> , <i>MenuItem</i> , <i>ComboBox</i> , <i>TextBox</i> . Lập trình viên có thể thêm các menu trong cửa sổ <i>Items Collection Edition</i> . Cửa sổ <i>Items Collection Edition</i> có thể mở lên bằng cách nhấp chuột trái vào biểu tượng của thuộc tính <i>Items</i> trong cửa sổ <i>Properties</i> .

	<p>Select item and add to list below:</p> <p>MenuItem</p> <p>Add</p> <p>Members:</p> <p>contextMenuStrip1</p>	<p>Items Collection Editor</p> <p>ContextMenu contextMenuStrip1</p> <table border="1"> <tr><td>Text</td><td>False</td></tr> <tr><td>UseWaitCursor</td><td>False</td></tr> <tr><td>Behavior</td><td></td></tr> <tr><td>AllowDrop</td><td>False</td></tr> <tr><td>AllowMerge</td><td>True</td></tr> <tr><td>AutoClose</td><td>True</td></tr> <tr><td>Enabled</td><td>True</td></tr> <tr><td>ImeMode</td><td>NoControl</td></tr> <tr><td>ShowItemToolTips</td><td>True</td></tr> <tr><td>TabStop</td><td>False</td></tr> <tr><td>Data</td><td></td></tr> <tr><td>(ApplicationSettings)</td><td></td></tr> <tr><td>(DataBindings)</td><td></td></tr> <tr><td>Tag</td><td></td></tr> <tr><td>Design</td><td></td></tr> <tr><td>(Name)</td><td>contextMenuStrip1</td></tr> </table> <p>OK Cancel</p>	Text	False	UseWaitCursor	False	Behavior		AllowDrop	False	AllowMerge	True	AutoClose	True	Enabled	True	ImeMode	NoControl	ShowItemToolTips	True	TabStop	False	Data		(ApplicationSettings)		(DataBindings)		Tag		Design		(Name)	contextMenuStrip1
Text	False																																	
UseWaitCursor	False																																	
Behavior																																		
AllowDrop	False																																	
AllowMerge	True																																	
AutoClose	True																																	
Enabled	True																																	
ImeMode	NoControl																																	
ShowItemToolTips	True																																	
TabStop	False																																	
Data																																		
(ApplicationSettings)																																		
(DataBindings)																																		
Tag																																		
Design																																		
(Name)	contextMenuStrip1																																	
RightToLeft	<p>Mang giá trị True: Trình bày menu từ phải qua trái Mang giá trị False: Trình bày menu từ trái qua phải</p>																																	

Ví dụ 5.3: Viết chương trình hiển thị thời gian và tạo ContextMenuStrip như hình 5.13.



Hình 5.13: Giao diện chương trình hiển thị thời gian ví dụ 5.3

Yêu cầu:

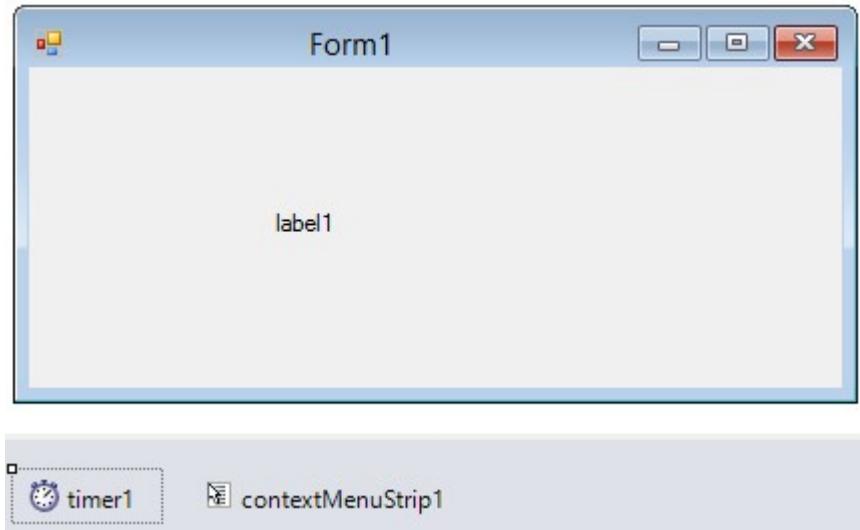
Khi chọn mục “Hiển thị thời gian”: Sẽ hiển thị giờ phút giây trên label lblNoiDung.

Khi chọn mục “Hiển thị ngày tháng”: Sẽ hiển thị ngày tháng năm trên label lblNoiDung.

Khi chọn mục Thoát hoặc nhấn tổ hợp phím Ctrl + T sẽ thoát chương trình.

Hướng dẫn:

Bước 1: Thiết kế giao diện ban đầu của chương trình như hình 5.14



Hình 5.14: Giao diện ban đầu chương trình hiển thị thời gian ví dụ 5.3

Bước 2: Thiết lập giá trị cho các điều khiển từ cửa sổ Properties

- form1:

Thuộc tính Text: “ContextMenuStrip”

- label1:

Thuộc tính Text: “-----Nội dung-----”

Thuộc tính Name: lblNoiDung

Thuộc tính Size: 14

- timer1:

Thuộc tính Name: MyTimer

- contextMenuStrip1:

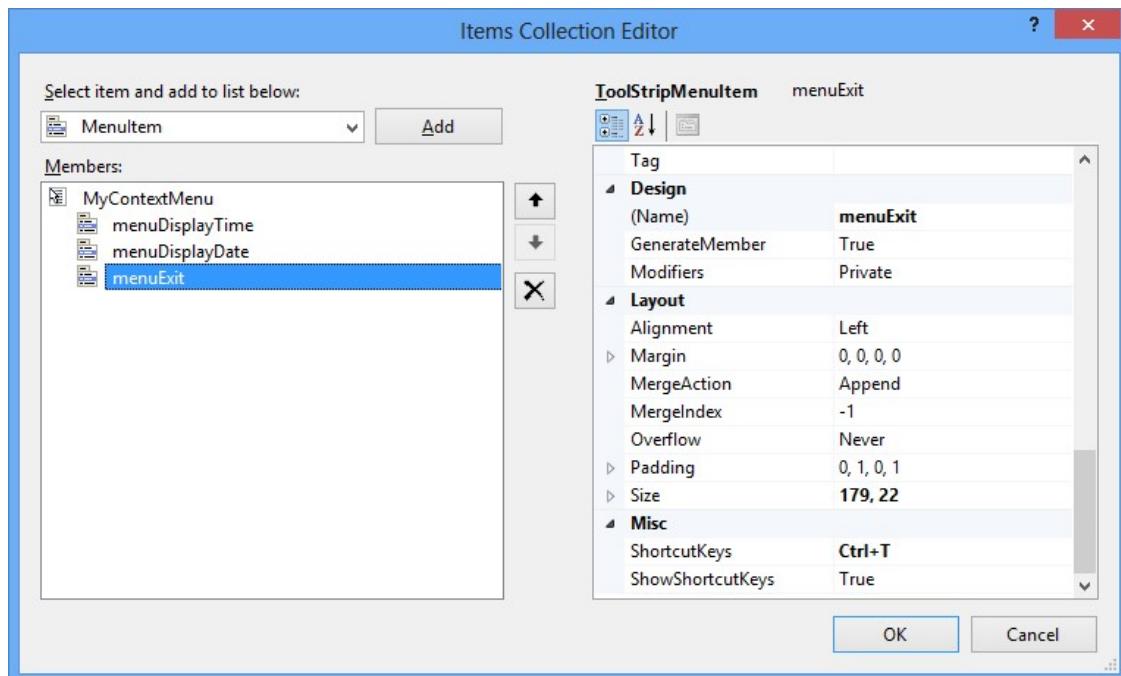
Thuộc tính Name: MyContextMenu

Thuộc tính Items: Thêm ba menu dạng MenuItem: menuDisplayTime, menuDisplayDate và menuExit trong cửa sổ Items Collection Editor như hình 5.15.

Thiết lập thuộc tính ShortCutKeys của menuExit: Ctrl + T

Thiết lập thuộc tính ShortCutKeyDisplayStyle của menuExit: Ctrl + T

Thiết lập thuộc tính ShowShortCutKeys của menuExit: True



Hình 5.15: Cửa sổ Items Collection Editor

Bước 3: Viết mã lệnh cho các điều khiển

- Khai báo biến:

```
int chon = 3;
DateTime dt = new DateTime();
```

- Sự kiện *Click* của menuDisplayTime:

```
private void menuDisplayTime_Click(object sender, EventArgs e)
{
    chon = 0;
}
```

- Sự kiện *Click* của menuDisplayDate:

```
private void menuDisplayDate_Click(object sender, EventArgs e)
{
    chon = 1;
    lblHienThi.Text = dt.Date.ToString("dd/MM/yyyy");
}
```

- Sự kiện *Click* của menuExit:

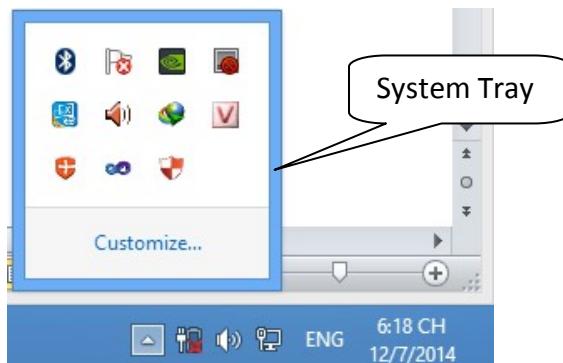
```
private void menuDisplayTime_Click(object sender, EventArgs e)
{
    chon = 0;
}
```

- Sự kiện *Tick* của MyTimer:

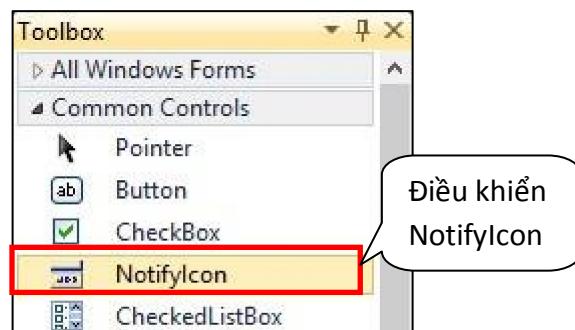
```
private void MyTimer_Tick(object sender, EventArgs e)
{
    dt = DateTime.Now;
    if (chon == 0)
    {
        lblHienThi.Text = dt.Hour + ":" + dt.Minute + ":" + dt.Second;
    }
}
```

5.4. Điều khiển NotifyIcon

Điều khiển *NotifyIcon* cho phép lập trình viên trình bày biểu tượng trên System Tray của màn hình Desktop như hình 5.16. Với việc sử dụng điều khiển *NotifyIcon*, khi người dùng di chuyển chuột đến biểu tượng thì chuỗi khai báo trong thuộc tính Text của điều khiển *NotifyIcon* sẽ hiện lên như một *Tooltip*. Điều khiển *NotifyIcon* nằm trong nhóm Common Controls của cửa sổ Toolbox như hình 5.17



Hình 5.16: System Tray trên màn hình Desktop



Hình 5.17: Điều khiển NotifyIcon

- Một số thuộc tính thường dùng của *NotifyIcon*:

Bảng 5.6: Bảng mô tả các thuộc tính thường dùng của NotifyIcon

Thuộc tính	Mô tả
<i>Icon</i>	Biểu tượng sẽ xuất hiện dưới khay hệ thống khi form hiển thị
<i>Text</i>	Chuỗi xuất hiện khi di chuột vào biểu tượng dưới khay hệ thống
<i>ContextMenuStrip</i>	Menu Popup sẽ xuất hiện khi người dùng nhấn chuột phải vào biểu tượng dưới khay hệ thống
<i>Visible</i>	Mang giá trị True: Hiển thị biểu tượng trên System Tray Mang giá trị False: Không hiển thị biểu tượng trên System Tray

Để hiển thị một *NotifyIcon* trên System Tray, lập trình viên bắt buộc phải thiết lập một biểu tượng nào đó cho thuộc tính *Icon* và thuộc tính *Visible* phải mang giá trị True.

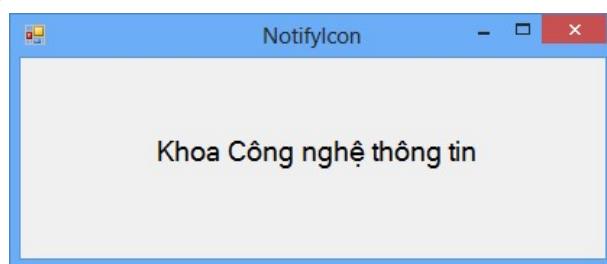
- Một số sự kiện thường dùng của *NotifyIcon*:

Bảng 5.7: Bảng mô tả các thuộc tính thường dùng của NotifyIcon

Sự kiện	Mô tả
<i>MouseClick</i>	Phát sinh khi người dùng nhấn chuột chọn vào biểu tượng được thiết lập trong thuộc tính <i>Icon</i>

Thông thường các biểu tượng trên System Tray được sử dụng bằng sự kiện *MouseClick* của chuột. Với sự kiện *MouseClick* khi nhấn chuột phải sẽ hiển thị một *ContextMenuStrip*, khi nhấn chuột trái sẽ mở chương trình được liên kết với *NotifyIcon*.

Ví dụ 5.4: Viết chương trình tạo form như hình 5.18.



Hình 5.18: Giao diện chương trình ví dụ 5.4

Yêu cầu:

Khi chạy chương trình sẽ hiển thị một biểu tượng trên System Tray.

Khi nhấp chuột phải vào biểu tượng sẽ hiển thị một *ContextMenu* như hình 5.19:



Hình 5.19: *ContextMenu* trên biểu tượng của *NotifyIcon*

Khi nhấp chuột trái vào biểu tượng sẽ hiển thị MessageBox với chuỗi hiển thị: “Khoa Công nghệ thông tin, Trường Đại học Lạc Hồng. Xin chào các bạn!”

Hướng dẫn:

Bước 1: Thiết kế giao diện form ban đầu như hình 5.20



Hình 5.20: Giao diện ban đầu ví dụ 5.4

Bước 2: Thiết lập giá trị thuộc tính cho các điều khiển ở cửa sổ Properties

- form1:

Thuộc tính *Text*: “NotifyIcon”

Thuộc tính *StartPosition*: FormStartPosition.CenterScreen;

- label1:

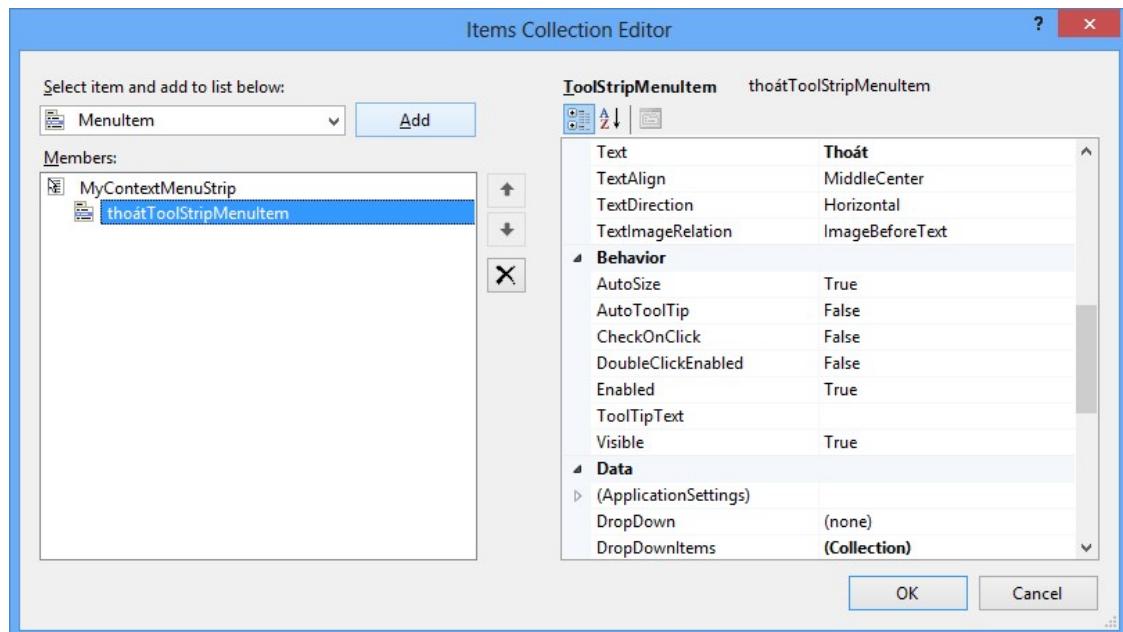
Thuộc tính *Text*: “Khoa Công nghệ thông tin”

Thuộc tính *Size*: 14

- contextMenuStrip1:

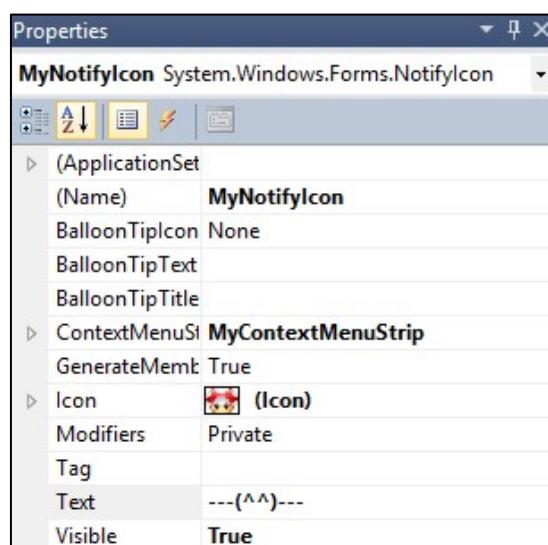
Thuộc tính *Name*: MyContextMenuStrip

Thuộc tính *Items*: Thêm một menu dạng MenuItem trong cửa sổ Items Collection Editor như hình 5.21.



Hình 5.21: Cửa sổ Item Collection Editor ví dụ 5.4

- notifyIcon1: Thiết lập giá trị thuộc tính của *NotifyIcon* như hình 5.22



Hình 5.22: Thuộc tính của *NotifyIcon*

Bước 3: Viết mã lệnh cho các điều khiển

- Sự kiện *Click* của *thoátToolStripMenuItem*:

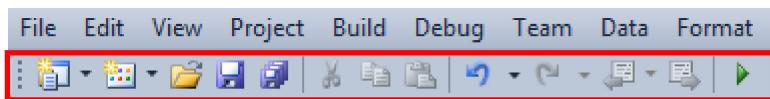
```
private void thoátToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Close();
}
```

- Sự kiện *MouseClick* của *MyNotifyIcon*:

```
private void MyNotifyIcon_MouseClick(object sender,
MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
        MessageBox.Show("Khoa Công nghệ thông tin, Trường
                        Đại học Lạc Hồng\n Xin chào các bạn");
}
```

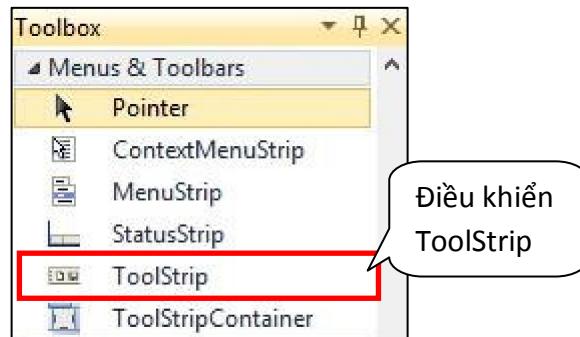
5.5. Điều khiển ToolStrip

ToolStrip là điều khiển cho phép tạo thanh công cụ trên form. Thông thường trong các ứng dụng Windows Forms, *ToolStrip* thường được bố trí phía dưới điều khiển *MenuStrip* như hình 5.23.



Hình 5.23: Vị trí của điều khiển *ToolStrip* trong *Visual Studio 2010*

Điều khiển *ToolStrip* nằm trong nhóm Menus & Toolbars của cửa sổ *Toolbox* như hình 5.24.

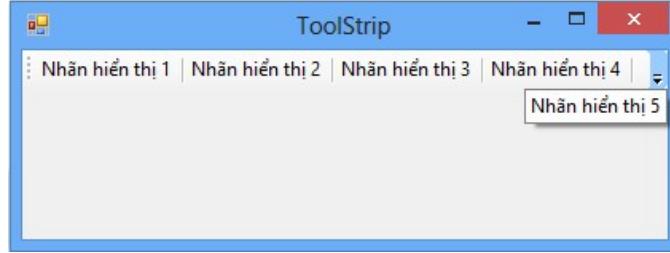
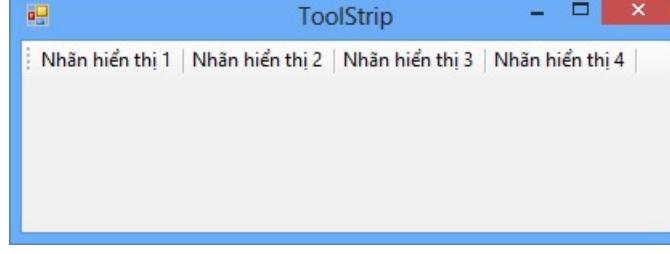


Hình 5.24: *ToolStrip* trong cửa sổ *Toolbox*

- Một số thuộc tính thường dùng của *ToolStrip*:

Bảng 5.8: Bảng mô tả các thuộc tính thường dùng của *ToolStrip*

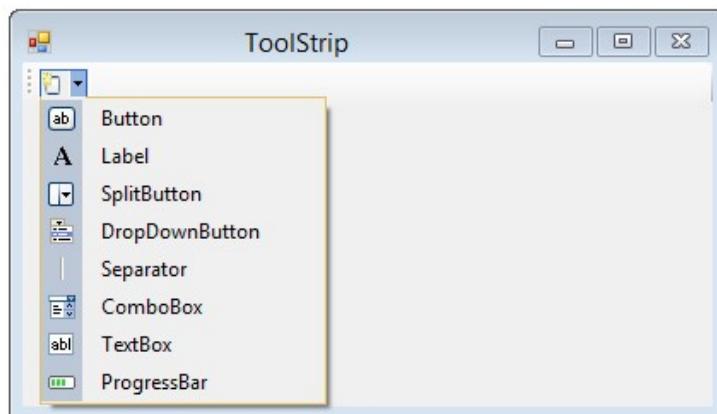
Thuộc tính	Mô tả
<i>Item</i>	Quản lý việc thêm xóa các điều khiển trên <i>ToolStrip</i>
<i>AllowItemReorder</i>	Mang giá trị True hoặc False. - Nếu là True: cho phép người dùng sắp xếp lại vị trí của các điều khiển trên <i>ToolStrip</i> ,

	<p>thay đổi vị trí bằng cách giữ phím Alt và nhấn chuột trái vào điều khiển và kéo đến vị trí mới trên ToolStrip.</p> <ul style="list-style-type: none"> Nếu là False: Các vị trí của điều khiển trên ToolStrip có định không thể thay đổi bởi người dùng.
<i>AllowMerge</i>	<p>Cho phép người dùng giữ phím Alt và giữ chuột trái vào điều khiển trên ToolStrip này và kéo thả vào một ToolStrip khác.</p> <p>Lưu ý: Thuộc tính này chỉ có hiệu lực khi thuộc tính <i>AllowItemReorder</i> là True</p>
<i>CanOverflow</i>	<p>Mang hai giá trị True và False.</p> <ul style="list-style-type: none"> Nếu là giá trị True: Khi số lượng điều khiển trong ToolStrip vượt ra khỏi phạm vi kích thước thì những điều khiển này sẽ được thu nhỏ trong biểu tượng  ở góc phải của ToolStrip.  <ul style="list-style-type: none"> Nếu là giá trị False: Những điều khiển nằm ngoài phạm vi kích thước sẽ không được thu nhỏ trong biểu tượng  ở góc phải của ToolStrip. Biểu tượng  sẽ không xuất hiện trên ToolStrip. 

<i>Dock</i>	Quy định vị trí hiển thị của ToolStrip trên form. 
<i>LayoutStyle</i>	Kiểu trình bày của ToolStrip
<i>ShowItemTooltips</i>	Mang hai giá trị True hoặc False. <ul style="list-style-type: none"> - Nếu là True: Cho phép hiển thị chuỗi khai báo trong thuộc tính ToolTipText của mỗi điều khiển chứa trong ToolStrip. - Nếu là False: Chuỗi khai báo trong ToolTipText của các điều khiển chứa trong ToolStrip không được hiển thị.

5.5.1. Các điều khiển chứa trong ToolStrip

Điểm đặc biệt của điều khiển *ToolStrip* là được cấu tạo dưới dạng container cho phép chứa các điều khiển khác như hình 5.25.



Hình 5.25: Các điều khiển ToolStrip chứa

Các điều khiển có thể tạo trên *ToolStrip* bao gồm:

- *ToolStripDropDownButton*: Điều khiển *ToolStripDropDownButton* cho phép lập trình viên tạo một menu dạng sổ xuống và sẽ hiển thị khi người dùng nhấp

chuột vào. Các menu được tạo trong điều khiển *ToolStripDropDownButton* là dạng *MenuItem*.

- *ToolStripButton*: Là điều khiển tương tự như điều khiển *Button*. Điều khiển này xuất hiện trên *ToolStrip* ở dạng một biểu tượng (*Icon*). Sự kiện thường dùng của *ToolStripButton* là sự kiện *Click*
- *ToolStripLabel*: Là điều khiển có chức năng như chức năng của điều khiển *Label* và điều khiển *LinkLabel*. Khi thuộc tính *IsLink* của *ToolStripLabel* là true thì điều khiển *ToolStripLabel* trở thành *LinkLabel*, khi thuộc tính *IsLink* là false *ToolStripLabel* như một điều khiển *Label* thông thường.
- *ToolStripProgressBar*: Tương tự như điều khiển *ProgressBar* và nằm trên *ToolStrip*.
- *ToolStripSeparator*: Là điều khiển dùng để hiển thị dấu gạch phân cách, giúp phân cách các điều khiển trong *ToolStrip* với nhau để hiển thị một cách dễ nhìn hơn.
- *ToolStripComboBox*: Điều khiển *ToolStripComboBox* có các thuộc tính, phương thức và sự kiện tương tự như điều khiển *ComboBox*, nhưng được đặt trên *MenuStrip*. Dạng hiển thị của *ToolStripComboBox* cũng có các dạng như: *Simple*, *DropDown* hoặc *DropDownList*.
- *ToolStripTextBox*: Tương tự như điều khiển *TextBox*, được đặt trên *MenuStrip*. Người dùng có thể gõ chuỗi ký tự vào ô văn bản, khi đó chuỗi ký tự đó sẽ được truyền vào cho thuộc tính *ToolStripTextBox.Text*. Điểm khác biệt chính của *ToolStripTextBox* và *TextBox* là điều khiển *ToolStripTextBox* không có thuộc tính *MultiLine*, do đó ô văn bản của *ToolStripTextBox* chỉ có một dòng.
- *ToolStripSplitButton*: Là điều khiển kết hợp chức năng của *ToolStripButton* và *ToolStripDropDownButton*. Điều khiển này hiển thị một nút mà người dùng có thể nhấn nút để thực thi mã lệnh, ngoài ra cũng cho phép hiển thị một menu theo dạng sổ xuống như của *ToolStripDropDownButton*. Lập trình viên có thể sử dụng sự kiện *ToolStripSplitButton.Click* để viết mã lệnh khi nhấn nút bấm hoặc có thể viết mã lệnh cho mỗi sự kiện *ToolStripMenuItem.Click*

➤ Hiển thị hình trên các điều khiển của *ToolStrip*

Các điều khiển *ToolStripButton*, *ToolStripSplitButton* và *ToolStripDropDownButton* có thể hiển thị hình, chuỗi mô tả hoặc cả hai. Bảng 5.9 mô tả các thuộc tính thường sử

dụng để biểu diễn hình và chuỗi mô tả trên các điều khiển *ToolStripButton*, *ToolStripSplitButton*, *ToolStripDropDownButton* của *ToolStrip*

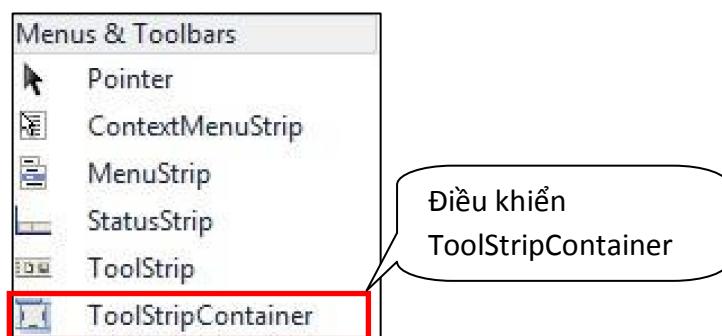
Bảng 5.9: Bảng mô tả thuộc tính của điều khiển trên *ToolStrip*

Thuộc tính	Mô tả
<i>DisplayStyle</i>	Gồm các giá trị: <i>None</i> , <i>Text</i> , <i>Image</i> , <i>ImageAndText</i> <ul style="list-style-type: none"> - <i>None</i>: Không hiển thị Image và Text trên điều khiển. - <i>Text</i>: Chỉ hiển thị chuỗi mô tả được thiết lập trong thuộc tính Text trên điều khiển (không hiển thị hình). - <i>Image</i>: Chỉ hiển thị hình được thiết lập trong thuộc tính Image trên điều khiển (không hiển thị chuỗi mô tả) - <i>ImageAndText</i>: Hiển thị cả hình và chuỗi mô tả trên điều khiển
<i>Image</i>	Chỉ định hình sẽ hiển thị trên điều khiển
<i>TextImageRelation</i>	Kiểu hiển thị của <i>Text</i> và <i>Image</i> trên điều khiển: <i>Overlay</i> , <i>ImageAboveText</i> , <i>TextAboveImage</i> , <i>TextBeforeImage</i> , <i>ImageBeforeText</i> <ul style="list-style-type: none"> - <i>ImageBeforeText</i>:  - <i>TextBeforeImage</i>:  - <i>ImageAboveText</i>:  - <i>TextAboveImage</i>:

	 <p>- Overlay:</p> 
<i>ImageScaling</i>	Kích thước của hình sẽ vừa với điều khiển hay không. <ul style="list-style-type: none"> - <i>None</i>: Hình sẽ hiển thị đúng với kích thước thật - <i>SizeToFit</i>: Hình sẽ hiển thị với kích thước bằng với kích thước của điều khiển
<i>ImageTransparentColor</i>	Làm trong suốt màu của hình

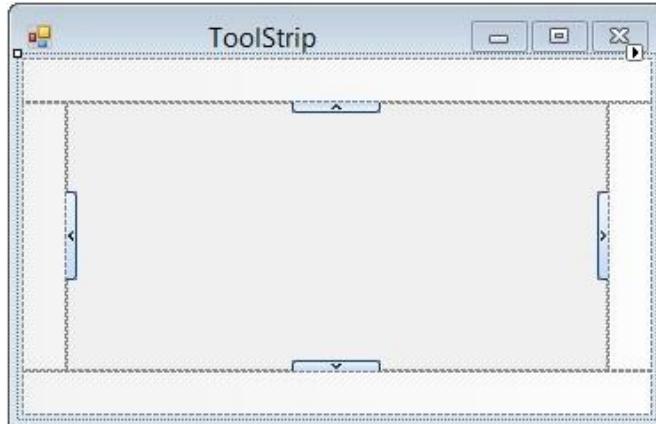
5.5.2. ToolStripContainer

ToolStripContainer là dạng điều khiển thiết kế, chứa các điều khiển *ToolStrip* bên trong. Điểm đặc biệt là khi sử dụng *ToolStripContainer*, người dùng có thể kéo và di chuyển các *ToolStrip* trên các cạnh của form (nếu trên cạnh form có sử dụng *ToolStripContainer*). *ToolStripContainer* nằm trong nhóm Menus & Toolbars của cửa sổ Toolbox như hình 5.26.



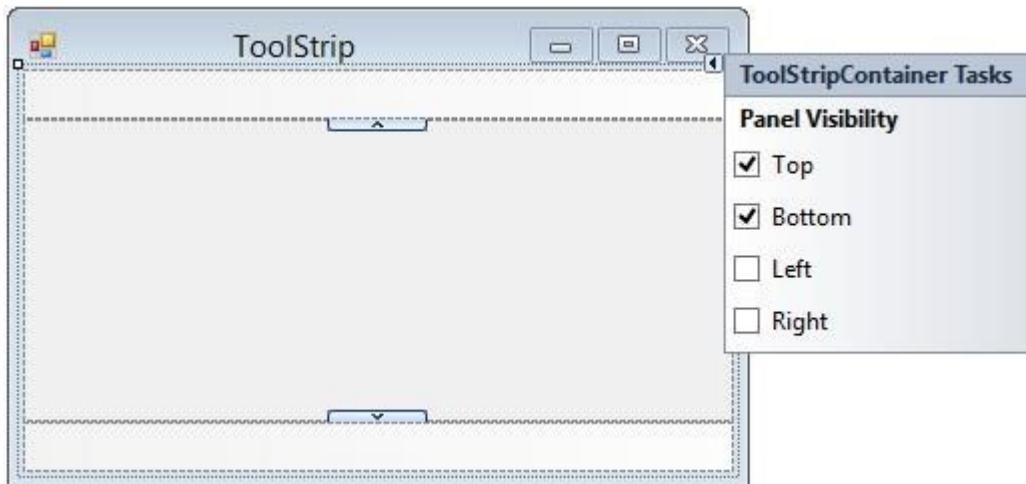
Hình 5.26: Điều khiển *ToolStripContainer*

ToolStripContainer được cấu tạo bởi 5 panel: 4 panel dạng *ToolStripPanel* được bố trí nằm trên 4 cạnh của form và 1 panel dạng *ContentPanel* nằm giữa form. Thông thường khi sử dụng *ToolStripContainer* trên form thì thuộc tính *Dock* của điều khiển này được thiết lập là *Fill* như hình 5.27.



Hình 5.27: ToolStripContainer có thuộc tính Dock thiết lập giá trị Fill

Lập trình viên có thể loại bỏ các panel dạng *ToolStripPanel* nếu không muốn panel bố trí trên một cạnh nào đó của form. Ví dụ nếu chỉ muốn sử dụng ToolStripContainer không có hai panel bên cạnh phải và cạnh trái của form có thể thiết lập như hình 5.28.



Hình 5.28: ToolStripContainer không có hai panel bên cạnh trái và cạnh phải của form

Có thể loại bỏ panel như hình 5.28 hoặc thiết lập giá trị các thuộc tính: *LeftToolStripPanelVisible*, *RightToolStripPanelVisible*, *TopToolStripPanelVisible*, *BottomToolStripPanelVisible*

- Một số thuộc tính thường dùng của ToolStripContainer:

Bảng 5.10: Bảng mô tả thuộc tính của ToolStripContainer

Thuộc tính	Mô tả
<i>Dock</i>	Kiểu hiển thị của ToolStripContainer. - <i>Fill</i> : Hiển thị bằng với kích thước form

	<ul style="list-style-type: none"> - <i>Top</i>: Hiển thị nằm sát cạnh trên của form - <i>Bottom</i>: Hiển thị nằm sát cạnh dưới của form - <i>Left</i>: Hiển thị sát cạnh trái của form - <i>Right</i>: Hiển thị sát cạnh phải của form <p>Lưu ý: Thường thì thuộc tính <i>Dock</i> được thiết lập giá trị <i>Fill</i></p>
<i>LeftToolStripPanelVisible</i>	Mang giá trị True hoặc False. <ul style="list-style-type: none"> - Nếu là True: Hiển thị panel bên cạnh trái của form - Nếu là False: Không hiển thị panel bên cạnh trái của form
<i>RightToolStripPanelVisible</i>	Mang giá trị True hoặc False. <ul style="list-style-type: none"> - Nếu là True: Hiển thị panel bên cạnh phải của form - Nếu là False: Không hiển thị panel bên cạnh phải của form
<i>TopToolStripPanelVisible</i>	Mang giá trị True hoặc False. <ul style="list-style-type: none"> - Nếu là True: Hiển thị panel cạnh trên của form - Nếu là False: Không hiển thị panel cạnh trên của form
<i>BottomToolStripPanelVisible</i>	Mang giá trị True hoặc False. <ul style="list-style-type: none"> - Nếu là True: Hiển thị panel cạnh dưới của form - Nếu là False: Không hiển thị panel cạnh dưới của form

Ví dụ 5.5: Thiết kế chương trình như hình 5.29.

Yêu cầu sau khi nhập địa chỉ website trong TextBox của ToolStrip và nhấn Enter thì sẽ hiển thị website có địa chỉ vừa nhập.

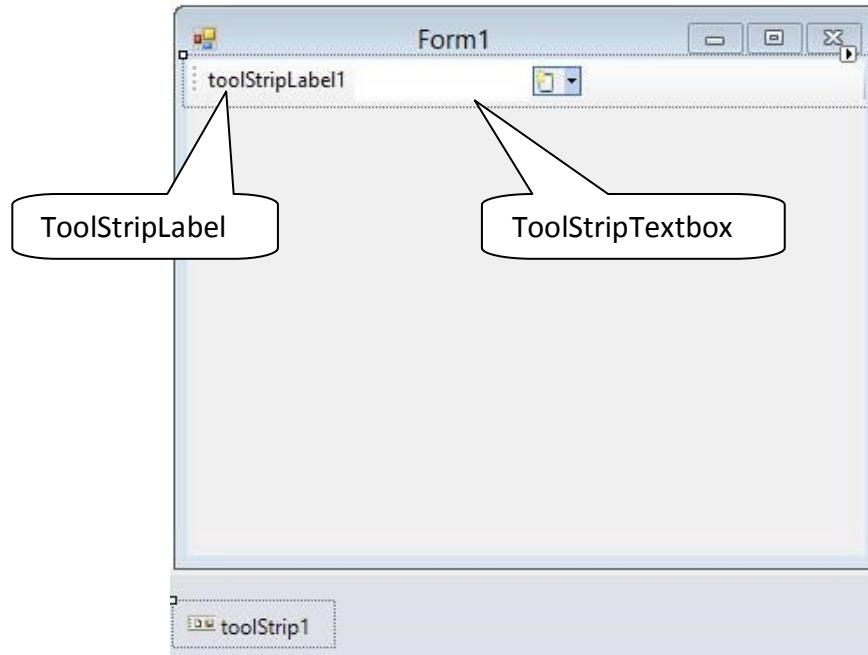


Hình 5.29: Giao diện form hiển thị Website ví dụ 5.5

Hướng dẫn:

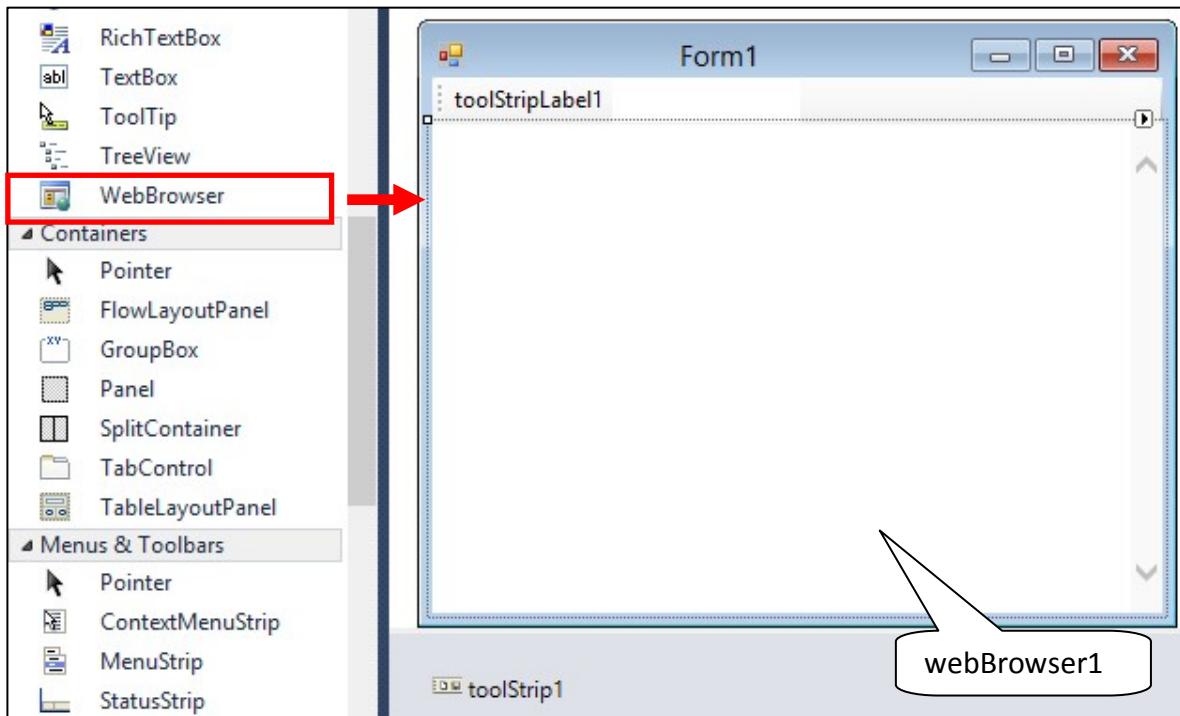
Bước 1: Thiết kế giao diện chương trình.

- Kéo ToolStrip từ cửa sổ Toolbox vào form. Sau đó thêm hai điều khiển ToolStripLabel và ToolStripTextBox trên ToolStrip như hình 5.30:



Hình 5.30: Giao diện form sau khi thêm ToolStrip

- Kéo điều khiển WebBrowser từ cửa sổ Toolbox và form như hình 5.31. WebBrowser là điều khiển cho phép hiển thị website trên form.



Hình 5.31: Giao diện form sau khi thêm điều khiển WebBrowser

Bước 2: Thiết lập các giá trị cho thuộc tính của điều khiển trong cửa sổ Properties

- form1:

Thuộc tính Text: "ToolStrip"

Thuộc tính Size: 405, 324

- toolStrip1:

Thuộc tính AutoSize: False

Thuộc tính Size: 389, 37

Thuộc tính BackColor: Color.DodgerBlue

- toolStripLabel1:

Thuộc tính Text: "Nhập địa chỉ website:"

- toolStripTextbox:

Thuộc tính Name: txtAddress

Thuộc tính AutoSize: False

Thuộc tính Size: 200, 25

- webBrowser1:

Thuộc tính Name: MyWebsite

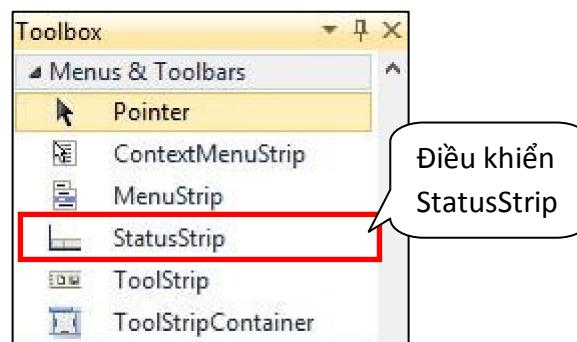
Bước 3: Viết mã lệnh cho các điều khiển trên form

- Sự kiện *KeyDown* của *txtAddress*:

```
private void txtAddress_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        MyWebsite.Navigate(txtAddress.Text);
}
```

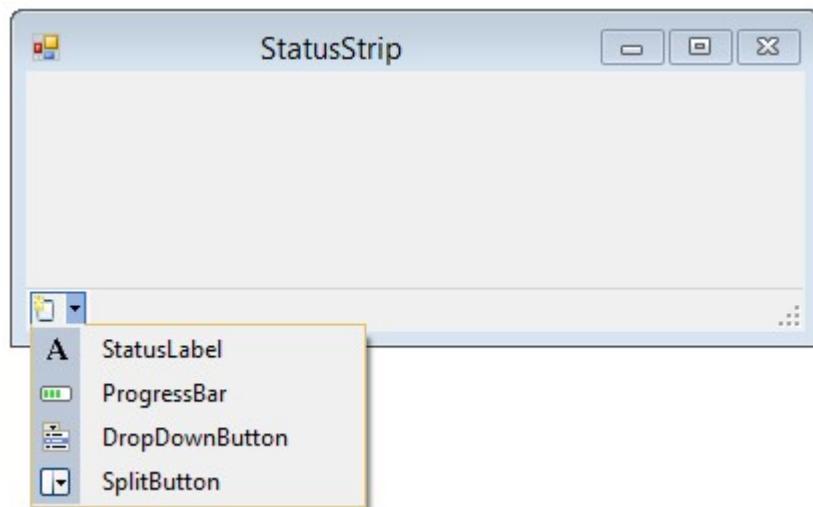
5.6. Điều khiển StatusStrip

StatusTrip sử dụng để hiển thị thông tin trạng thái của ứng dụng, *StatusStrip* nằm ở vị trí dưới cùng của form. *StatusStrip* nằm trong nhóm Menus & Toolbars của cửa sổ Toolbox như hình 5.32.



Hình 5.32: StatusStrip trong cửa sổ Toolbox

Cũng giống như *ToolStrip*, *StatusStrip* cũng có thể chứa các điều khiển khác như hình 5.33.



Hình 5.33: Các điều khiển có thể chứa trên StatusStrip

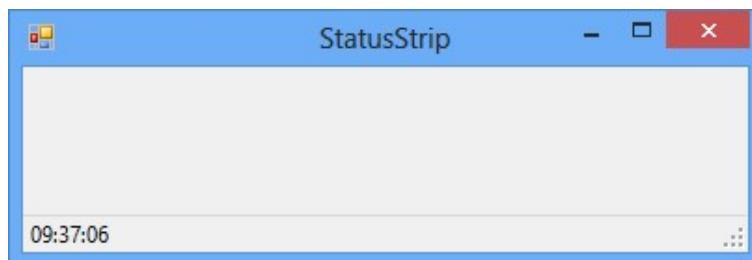
Các điều khiển: *StatusLabel*, *SplitButton*, *DropDownButton* hay *ProgressBar* trên *StatusStrip* có chức năng tương tự như trên *ToolStrip*. Trong các điều khiển trên, điều khiển thường sử dụng trên *StatusStrip* nhất là *StatusLabel*

- Một số thuộc tính thường dùng của *StatusStrip*:

Bảng 5.11: Bảng mô tả thuộc tính của *StatusStrip*

Thuộc tính	Mô tả
<i>Item</i>	Quản lý việc thêm, xóa các điều khiển trên <i>StatusStrip</i> như: <i>StatusLabel</i> , <i>SplitButton</i> , <i>DopDownButton</i> , <i>ProgressBar</i> .
<i>TextDirection</i>	Chiều hiển thị của chuỗi mô tả trong thuộc tính <i>Text</i> .
<i>LayoutStyle</i>	Vị trí hiển thị của <i>StatusStrip</i> .

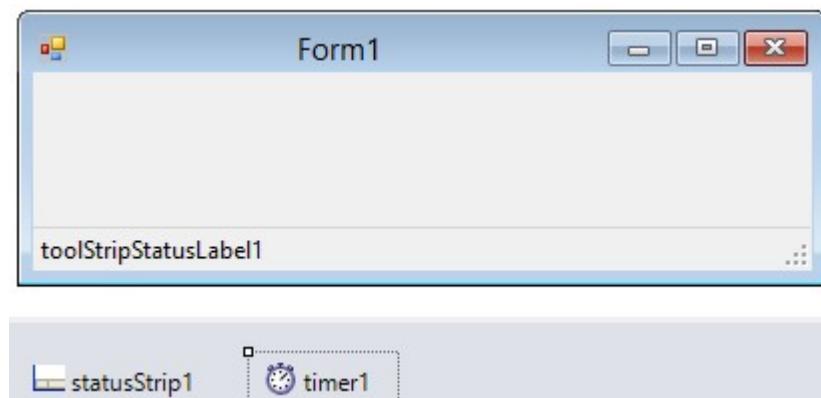
Ví dụ 5.6: Viết chương trình hiển thị thời gian trên *StatusStrip* của form như hình 5.34



Hình 5.34: Giao diện chương trình hiển thị thời gian trên *StatusStrip* ví dụ 5.6

Hướng dẫn:

Bước 1: Thiết kế giao diện chương trình: Kéo điều khiển *StatusStrip* và *Timer* từ cửa sổ *Toolbox* và *form*. Sau đó thêm điều khiển *StatusLabel* trên *StatusStrip* như hình 5.35.



Hình 5.35: Giao diện form sau khi thêm *StatusStrip*

Bước 2: Thiết lập giá trị thuộc tính cho điều khiển trong cửa sổ Properties

- form1:

Thuộc tính Text: “StatusStrip”

Thuộc tính Size: 405, 138

- timer1:

Thuộc tính Name: MyTimer

Thuộc tính Interval: 1000

Thuộc tính Enable: True

- toolStripStatusLabel1:

Thuộc tính Name: MyStatusLabel

Bước 3: Viết mã lệnh cho điều khiển

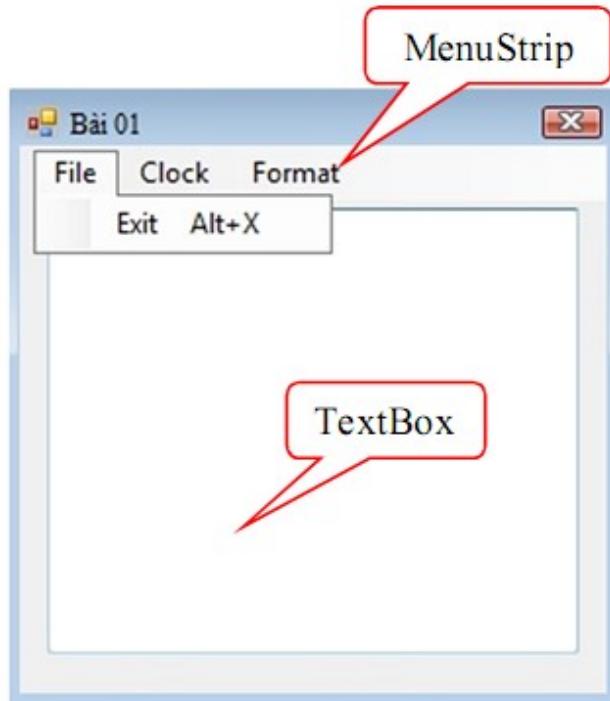
- Sự kiện *Tick* của MyTimer:

```
private void MyTimer_Tick(object sender, EventArgs e)
{
    MyStatusLabel.Text = DateTime.Now.ToString("hh:mm:ss");
}
```

5.7. Bài tập cuối chương

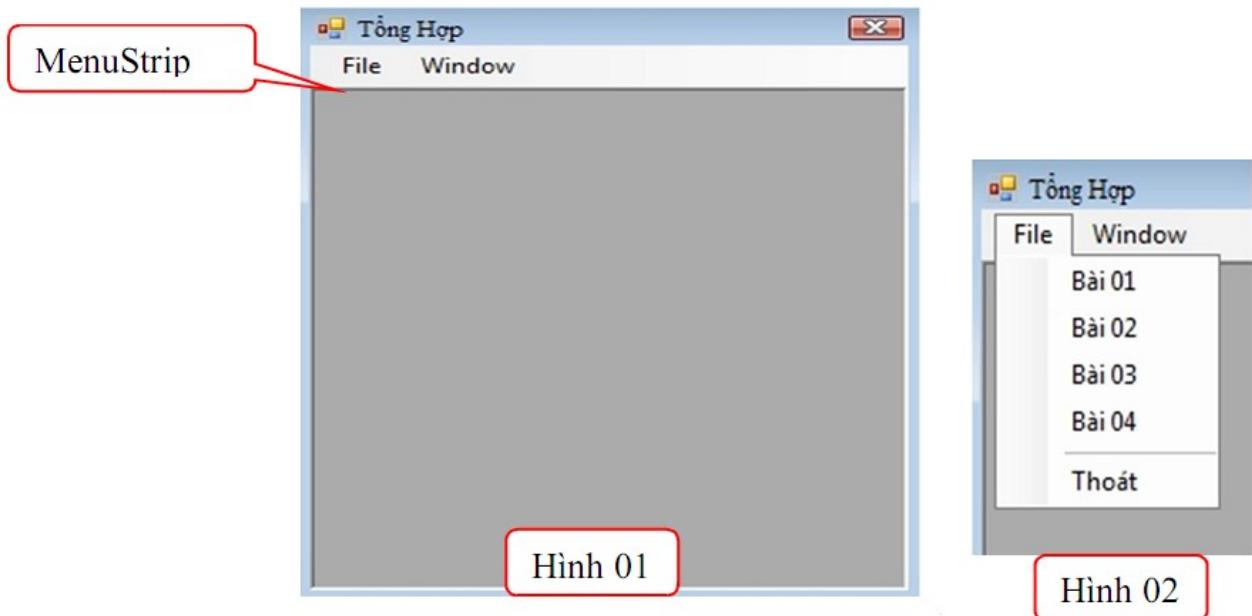
Câu 1: Viết chương trình soạn thảo văn bản có giao diện như hình 5.36. với các yêu cầu sau:

- Trên Form có một TextBox.
- Menu File có chứa mục Exit để thoát chương trình.
- Menu Clock có chứa mục:
 - o Date: cho xuất hiện ngày hiện tại trong TextBox.
 - o Time: cho hiện giờ hiện tại trong TextBox.
- Menu Format có chứa mục:
 - o Font: chọn font cho TextBox.
 - o Color: chọn màu cho văn bản trong TextBox.
 - o Align có chứa các mục:
 - o Left : canh lề trái cho TextBox.
 - o Right: canh lề phải cho TextBox.
 - o Center: canh giữa cho TextBox.



Hình 5.36: Giao diện chương trình soạn thảo văn bản

Câu 2: Viết chương trình gồm các bài tập có giao diện như hình 5.37.

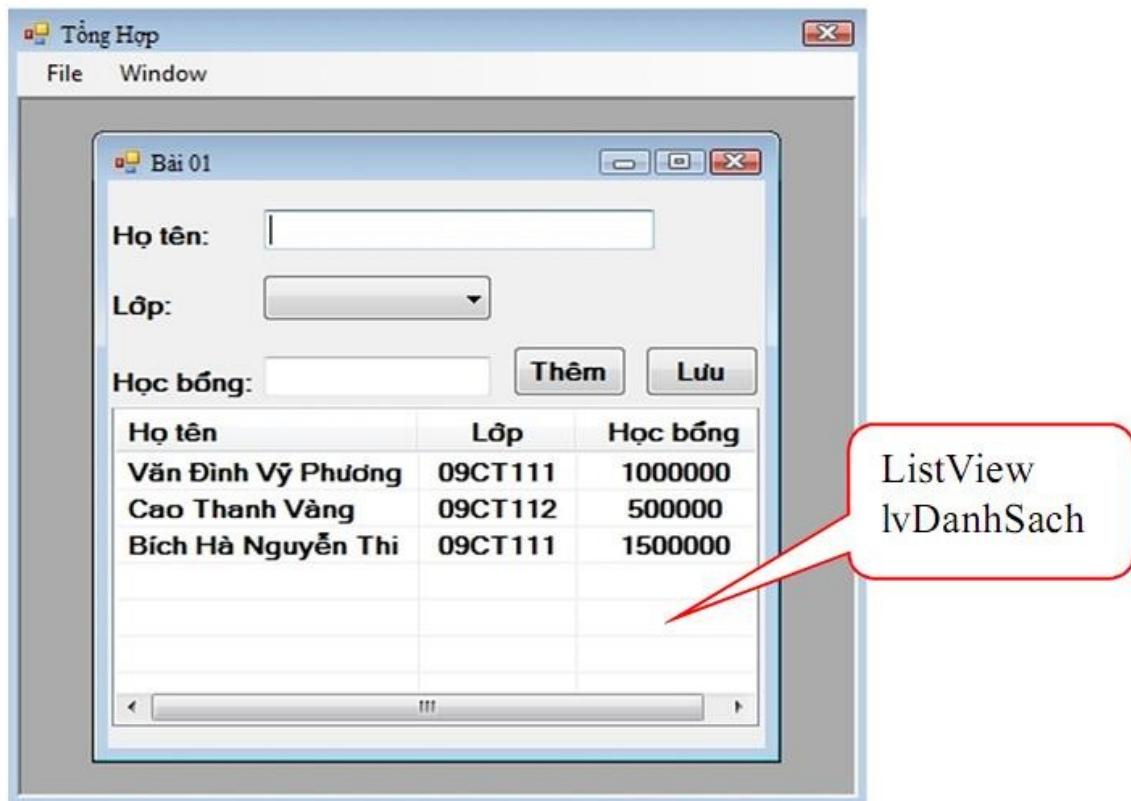


Hình 5.37: Giao diện form các bài tập

Yêu cầu:

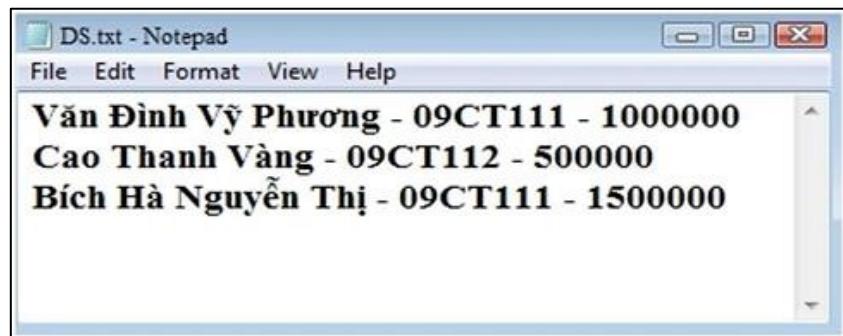
Khi nhấn F5 sẽ được form hiển thị như hình 01. Nhấp chuột chọn Menu sẽ được các MenuItem hiển thị như hình 02.

- Khi chọn MenuItem “Bài 01” sẽ hiển thị giao diện như hình 5.38.



Hình 5.38: Giao diện bài tập 01

- Cho người dùng nhập Họ tên, Chọn lớp, nhập học bỗng.
- Khi nhấn nút Thêm thì kiểm tra xem các thông tin được nhập (Họ tên, Học bỗng) và chọn (Lớp) chưa? Nếu chưa, thông báo cho người dùng nhập, chọn lại. Ngược lại thì thêm các thông tin vào ListView lvDanhSach
- Khi nhấn nút Lưu thì lưu danh sách các sinh viên cấp học bỗng vào một tập tin văn bản (*.txt) với dạng như hình 5.39.



Hình 5.39: Giao diện tập tin văn bản khi nhấn nút lưu

- Khi chọn MenuItem “Bài 02” sẽ hiển thị giao diện như hình 5.40.

Yêu cầu:

- Người dùng chọn sở thích và màu thích của mình.
- Khi nhấn nút Sở Thích Của Bạn thì xuất MessageBox các sở thích được chọn như hình 5.41.
- Khi nhấn nút Màu Bạn Thích thì xuất MessageBox màu được chọn như hình 5.42.



Hình 5.40: Giao diện bài tập 02



Hình 5.41: MessageBox sở thích



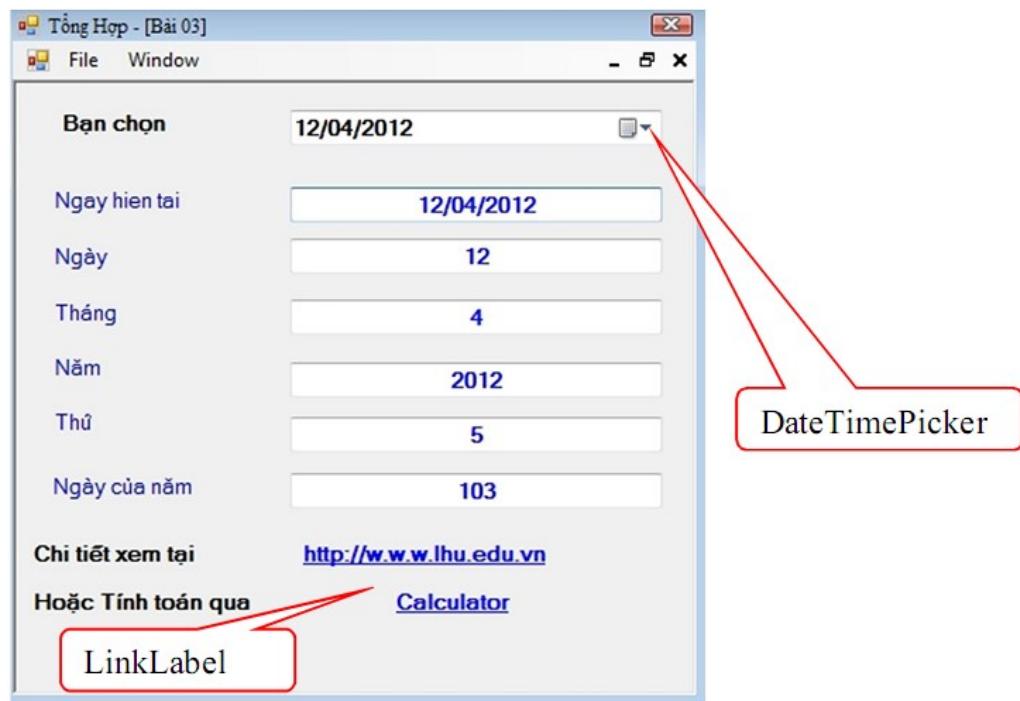
Hình 5.42: MessageBox màu đã chọn

- Khi chọn MenuItem “Bài 03” sẽ hiển thị giao diện như hình 5.43.

Yêu cầu:

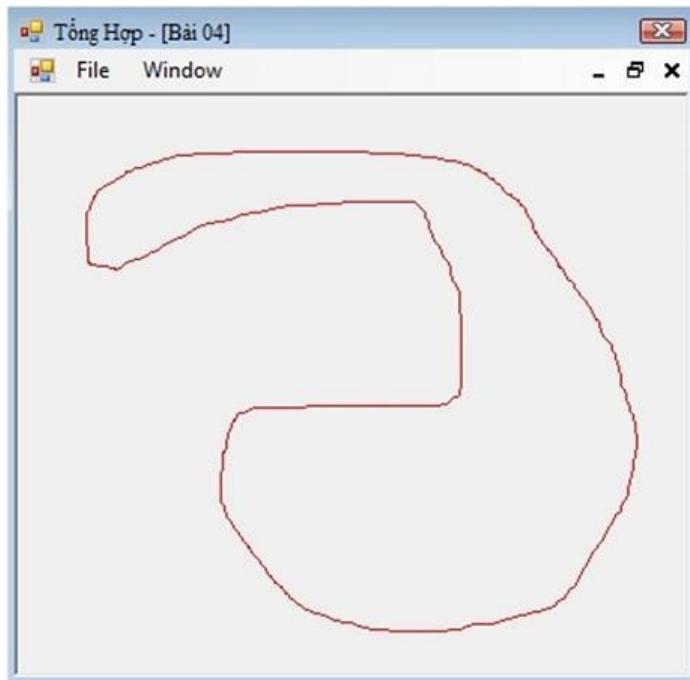
- Khi người dùng chọn giá trị trong DateTimePicker thì hiển thị các giá trị tương ứng vào các TextBox bên dưới.

- Khi người dùng nhấn LinkLabel sẽ mở ứng dụng tương ứng.



Hình 5.43: Giao diện bài tập 03

- Khi chọn MenuItem “Bài 04” sẽ hiển thị giao diện như hình 5.44



Hình 5.44: Giao diện bài tập 04

- Cho phép người dùng sử dụng chuột trái để vẽ trên form