

LẬP TRÌNH HƯỚNG SỰ KIỆN

Giảng viên: ThS. Phan Thanh Toàn

BÀI 6

LẬP TRÌNH ỨNG DỤNG WINDOW FORM VỚI C#

Giảng viên: ThS. Phan Thanh Toàn

MỤC TIÊU BÀI HỌC

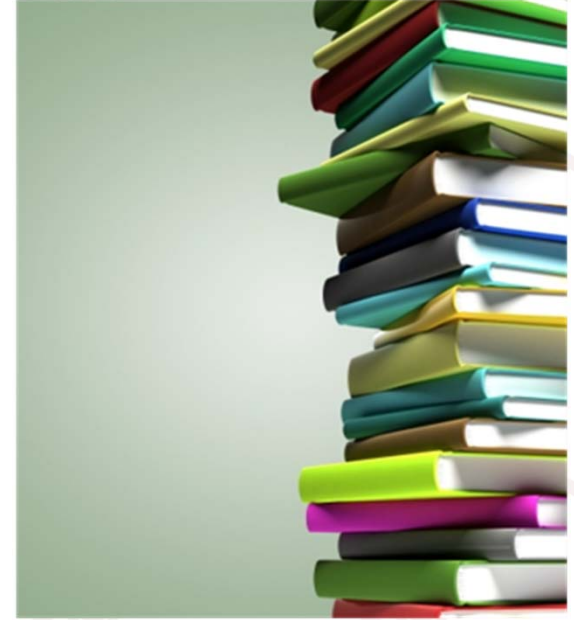
- Phân biệt được khái niệm sự kiện, lập trình hướng sự kiện.
- Liệt kê được các thuộc tính và phương thức cơ bản của đối tượng Window form.
- Liệt kê được các sự kiện cơ bản của đối tượng form.
- Liệt kê được các đối tượng cơ bản thường sử dụng để xây dựng ứng dụng Window form.



CÁC KIẾN THỨC CẦN CÓ

Để học được môn học này, sinh viên phải học xong các môn học:

- Cơ sở lập trình;
- Lập trình hướng đối tượng;
- Cơ sở dữ liệu;
- Hệ quản trị cơ sở dữ liệu SQL Server.



HƯỚNG DẪN HỌC

- Đọc tài liệu tham khảo.
- Thảo luận với giáo viên và các sinh viên khác về những vấn đề chưa hiểu rõ.
- Trả lời các câu hỏi của bài học.



CẤU TRÚC NỘI DUNG

6.1

Một số khái niệm cơ bản

6.2

Biểu mẫu (Form)

6.3

Các điều khiển cơ bản

6.1. MỘT SỐ KHÁI NIỆM CƠ BẢN

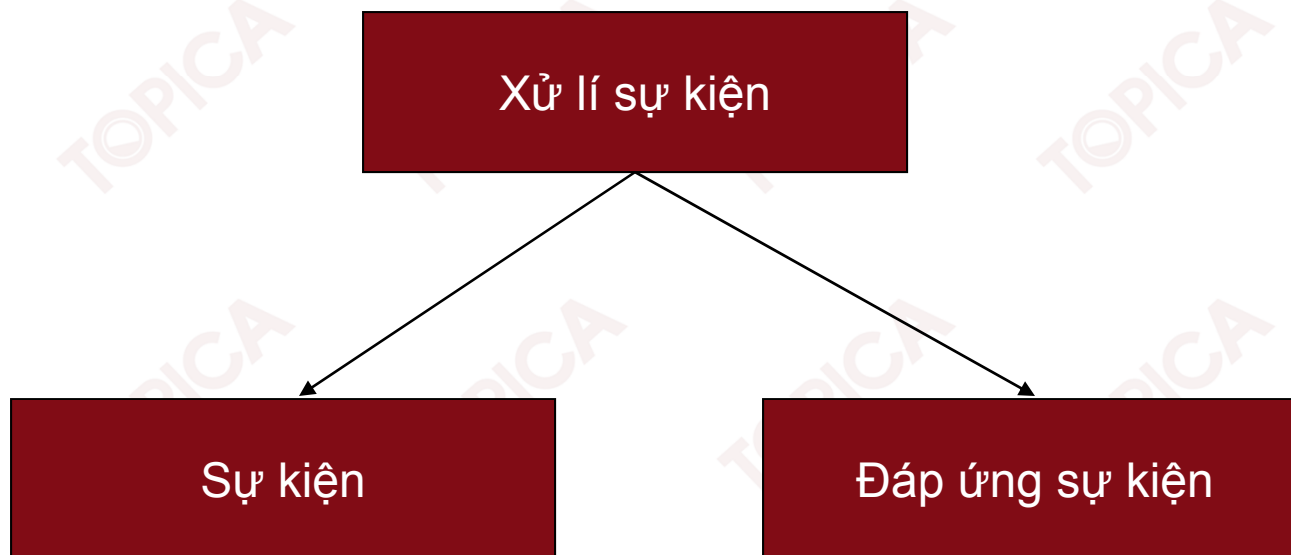
6.1.1. Sự kiện

6.1.2. Lập trình hướng
sự kiện

6.1.3. Các bước xây
dựng ứng dụng theo
phương pháp hướng
sự kiện

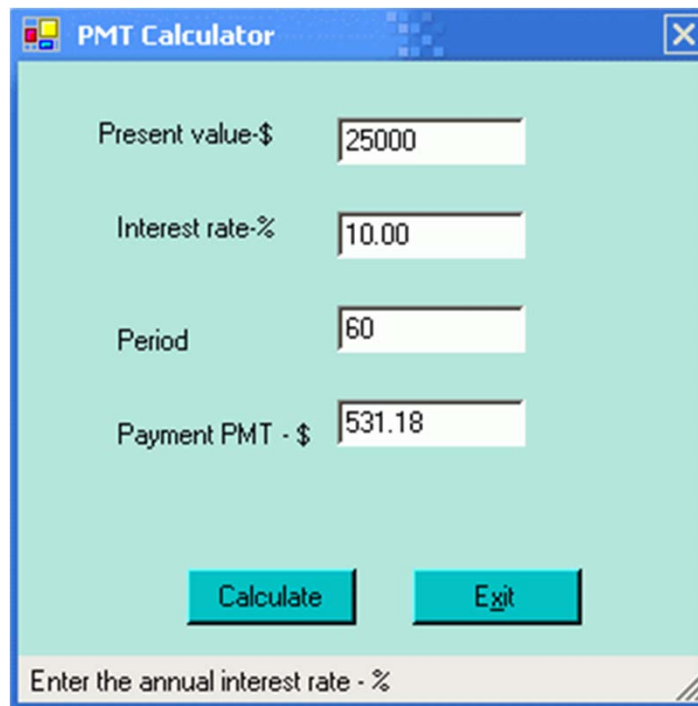
6.1.1. SỰ KIỆN

- Sự kiện là một hành động xác định xảy ra trên một đối tượng.
- Cơ chế gửi thông điệp giữa các lớp hay các đối tượng.
- Gửi thông báo cho lớp khác khi phát sinh ra một sự kiện.
- Ta có thể tương tác giữa các đối tượng khác nhau bên trong 1 ứng dụng, giữa 1 đối tượng và đối tượng bên ngoài thông qua sự kiện và đáp ứng sự kiện.



6.1.1. SỰ KIỆN (tiếp theo)

- Giao diện đồ họa GUI (Graphic User Interface): là giao diện tương tác với người dùng qua các đối tượng đồ họa.
- GUI được thiết kế dựa trên đối tượng cơ bản là form và sẽ đưa các đối tượng khác vào form cho phù hợp với ứng dụng.
- Điều khiển (Control): là các đối tượng trong form, mỗi đối tượng sẽ có các thuộc tính, phương thức và sự kiện.

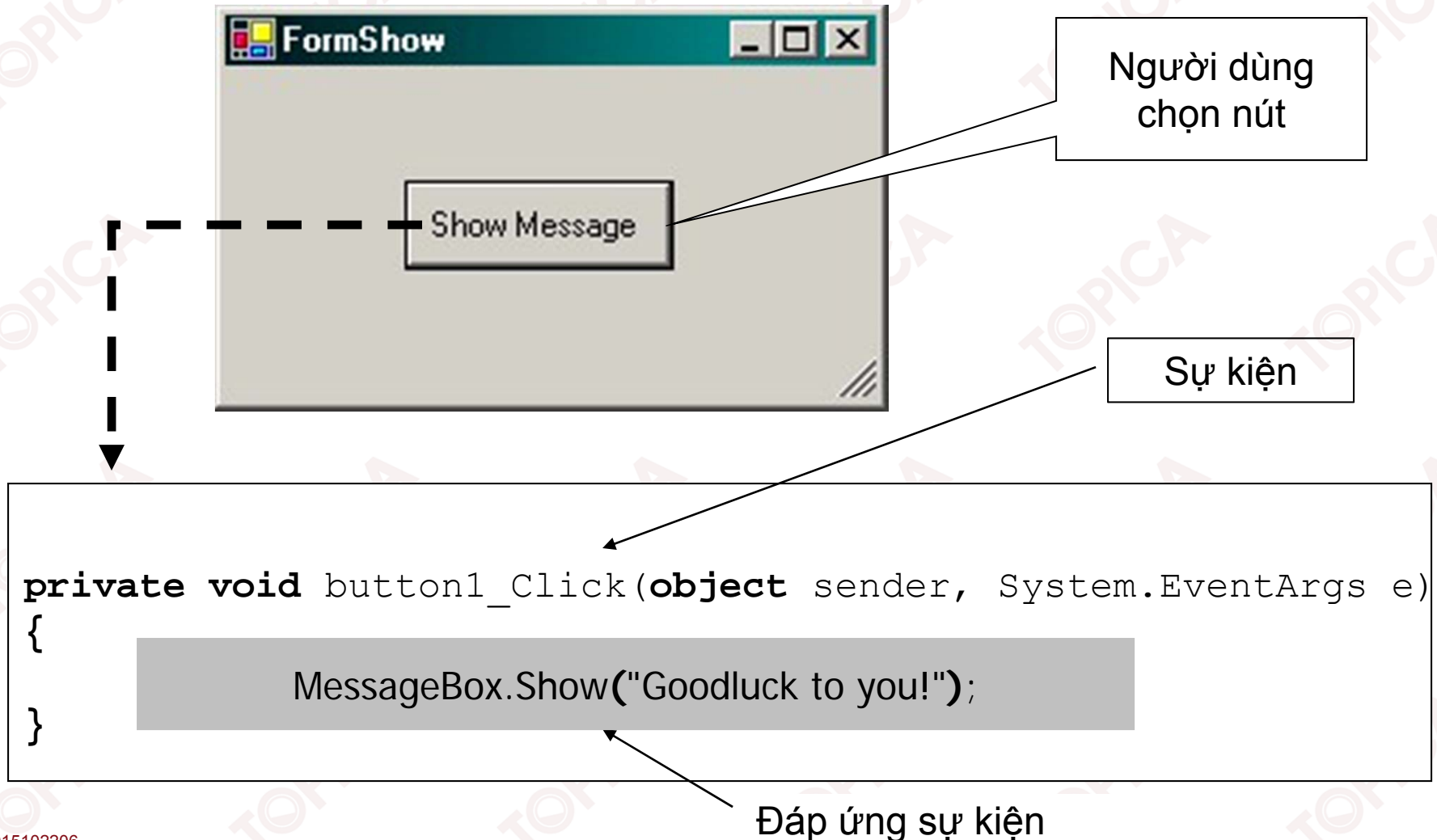


The image shows a screenshot of a software window titled "PMT Calculator". The window has a light blue background and a blue title bar. It contains four input fields with labels to their left: "Present value-\$" with the value "25000", "Interest rate-%" with the value "10.00", "Period" with the value "60", and "Payment PMT - \$" with the value "531.18". Below these fields are two buttons: "Calculate" and "Exit". At the bottom of the window, there is a status bar with the text "Enter the annual interest rate - %".

Field Label	Value
Present value-\$	25000
Interest rate-%	10.00
Period	60
Payment PMT - \$	531.18

6.1.2. LẬP TRÌNH HƯỚNG SỰ KIỆN

- Lập trình hướng sự kiện là phương pháp xây dựng hệ thống dựa trên việc phát triển giao diện tương tác đồ họa qua các đối tượng và định nghĩa các hành vi của hệ thống qua các sự kiện của các đối tượng.



6.1.3. CÁC BƯỚC XÂY DỰNG ỨNG DỤNG THEO PHƯƠNG PHÁP HƯỚNG SỰ KIỆN

- Tạo đối tượng form trên ứng dụng.
- Thiết kế giao diện tương tác người dùng bằng cách đưa các đối tượng vào form.
- Chọn các sự kiện cần tương tác với người dùng (Click, MouseOver...).
- Viết code xử lí cho các sự kiện.



Thiết kế form

```
private void gvDanhSach_Click(object sender, EventArgs e)
{
    txtMa.Text = gridView1.GetFocusedRowCellDisplayText("LH_Ma");
    txtTen.Text = gridView1.GetFocusedRowCellDisplayText("LH_Ten");
    txtKhoaHoc.Text = gridView1.GetFocusedRowCellDisplayText("KhoaHoc");
    txtNhanVien.Text = gridView1.GetFocusedRowCellDisplayText("NV_Ten");
}

private void ucDanhSachLopHoc_Load(object sender, EventArgs e)
{
    gvDanhSach.DataSource = frm.SelectAll();//load danh sach
    gridView1.GroupPanelText = "Kéo 1 cột vào đây để gom nhóm";
}
```

Xử lí các sự kiện

6.2. BIỂU MẪU - FORM

6.2.1. Thiết kế và sử dụng
biểu mẫu

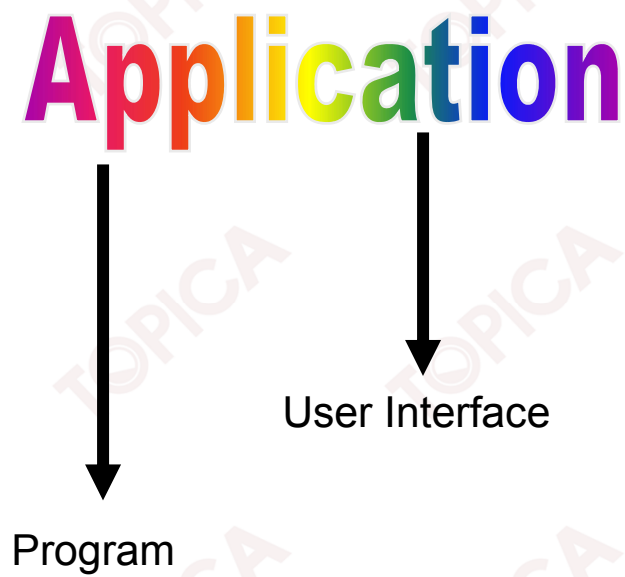
6.2.2. Các thuộc tính của
đối tượng form

6.2.3. Các phương thức của
đối tượng form

6.2.4. Các sự kiện của
đối tượng form

6.2.1. THIẾT KẾ VÀ SỬ DỤNG BIỂU MẪU

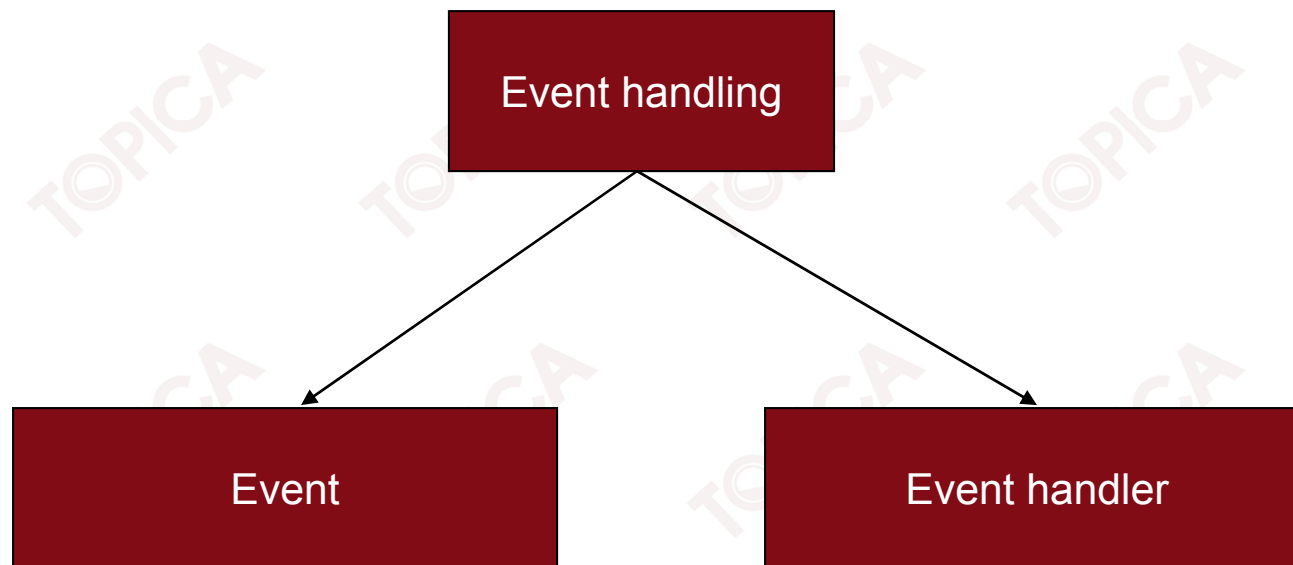
- Form là giao diện chính tương tác với người sử dụng.
- Trên form chứa các đối tượng khác gọi là các đối tượng điều khiển (control).
- Một ứng dụng gồm nhiều form.



A screenshot of a Windows-style window titled "Form1". Inside the window is a form titled "User Details Entry Form". The form has a dotted background and contains the following elements: a text label "Name" followed by a single-line text box; a text label "Address" followed by a multi-line text box; a text label "Sex" followed by two radio buttons labeled "Male" and "Female"; and at the bottom, two buttons labeled "Save" and "Print".

6.2.1. THIẾT KẾ VÀ SỬ DỤNG BIỂU MẪU (tiếp theo)

- Tương tác giữa các đối tượng với ứng dụng thông qua xử lý sự kiện.
- Sử dụng lớp System.Windows.Forms.
- Dựa trên nền tảng lập trình hướng đối tượng.
- Giao diện window được kế thừa từ lớp form.



6.2.2. CÁC THUỘC TÍNH CỦA ĐỐI TƯỢNG FORM

Thuộc tính	Ý nghĩa
AcceptButton	Lấy hoặc thiết lập nút bấm trên form để khi người dùng gõ Enter trên form sẽ gọi sự kiện Click của nút bấm.
Name	Lấy hoặc đặt tên cho form.
Width	Lấy hoặc thiết lập chiều rộng cho form.
Height	Lấy hoặc thiết lập chiều cao cho form.
StartPosition	Thiết lập vị trí cho form.
AutoSize	Cho phép thay đổi kích thước form, thuộc tính này thường sử dụng với AutoSizeMode.
AutoSizeMode	Thiết lập chế độ thay đổi kích thước của form.
BackColor	Thiết lập (lấy) màu nền của form.
CancelButton	Thiết lập nút bấm để khi bấm phím ESC sẽ triệu gọi sự kiện Click của nút bấm.
Controls	Trả về tập các đối tượng điều khiển trên form.

6.2.2. CÁC THUỘC TÍNH CỦA ĐỐI TƯỢNG FORM (tiếp theo)

Thuộc tính	Ý nghĩa
DialogResult	Nhận giá trị của hộp dialog cho form.
Enabled	Cho phép hoặc ngăn cản người sử dụng tương tác với các đối tượng trên form.
IsMdiChild	Cho biết một form có phải là form con không (sử dụng trong các ứng dụng MDI).
MdiParent	Thiết lập thuộc tính form cha cho một form con (sử dụng trong chế độ MDI).
Visible	Hiện thị hoặc ẩn form.
WindowState	Thiết lập chế độ hiển thị của form là minimized, maximized, hay normal.

6.2.3. CÁC PHƯƠNG THỨC CỦA ĐỐI TƯỢNG FORM

Phương thức	Ý nghĩa
Activate	Kích hoạt một form
Close	Đóng form
Dispose	Giải phóng form tất cả đối tượng trên form
Hide	Ẩn form

6.2.4. CÁC SỰ KIỆN CỦA ĐỐI TƯỢNG FORM

Sự kiện	Ý nghĩa
Activated	Xuất hiện khi một form được kích hoạt
BackColorChanged	Xuất hiện khi màu nền của form bị thay đổi
Click	Xuất hiện khi bấm chuột trên form
Closed	Xuất hiện khi đóng form
Deactivate	Xuất hiện khi một form không được kích hoạt
Disposed	Xuất hiện khi gọi phương thức Dispose
DoubleClick	Xuất hiện khi kích đúp chuột trên form
FormClosed	Xuất hiện sau khi form bị đóng
Move	Xuất hiện khi form bị di chuyển sang vị trí mới

6.3. CÁC ĐIỀU KHIỂN CƠ BẢN

6.3.1. Đối tượng Label

6.3.2. Đối tượng TextBox

6.3.3. Đối tượng Button

6.3.4. Đối tượng ListBox

6.3.5. Đối tượng
CheckedListBox

6.3.6. Đối tượng
ComboBox

6.3.7. Đối tượng
CheckBox và Radio

6.3.1. ĐỐI TƯỢNG LABEL

- Là đối tượng được sử dụng để hiển thị thông tin trên form.
- Các thuộc tính của Label:

Thuộc tính	Ý nghĩa
Name	Tên đối tượng
Text	Thông tin xuất hiện trên Label
Font	Thiết lập các tham số về font chữ cho đối tượng
ForeColor	Thiết lập màu cho thông tin hiển thị trên Label
Locked	Cho phép hoặc ngăn cản việc thay đổi thông tin trên Label
Visible	Hiển thị/ẩn Label

6.3.2. ĐỐI TƯỢNG TEXTBOX

- Cho phép nhập và hiển thị thông tin
- Các thuộc tính của TextBox:

Thuộc tính	Ý nghĩa
Name	Tên đối tượng TextBox.
Text	Thông tin xuất hiện trên TextBox.
Font	Thiết lập các tham số về font chữ cho đối tượng.
ForeColor	Thiết lập màu cho thông tin hiển thị trên TextBox.
Locked	Cho phép hoặc ngăn cản việc thay đổi thông tin trên TextBox.
Visible	Hiển thị/ẩn Label.
Multiline	Cho phép TextBox hiển thị ở chế độ nhiều dòng.
PasswordChar	Sử dụng TextBox ở chế độ hộp nhập mật khẩu (thường dùng kí tự *).
MaxLength	Thiết lập độ dài tối đa cho hộp nhập.
SelectedText	Lấy về dữ liệu được chọn trong hộp TextBox.
SelectionLength	Trả về số kí tự của đoạn văn bản được chọn trong hộp TextBox.

6.3.2. ĐỐI TƯỢNG TEXTBOX (tiếp theo)

- Các sự kiện của TextBox

Sự kiện	Ý nghĩa
Validated	Được sử dụng kiểm tra tính hợp lệ của dữ liệu.
TextChanged	Xuất hiện khi dữ liệu trong hộp TextBox bị thay đổi.

6.3.3. ĐỐI TƯỢNG BUTTON

- Sử dụng xử lý các dữ liệu trong form khi người dùng kích chuột lên button

Thuộc tính	Ý nghĩa
Name	Tên đối tượng Button.
Text	Thông tin xuất hiện trên Button.
Enable	Cho phép/cấm người sử dụng tương tác với Button.
DialogResult	Thiết lập giá trị nhằm tương tác với hộp thoại Dialog.

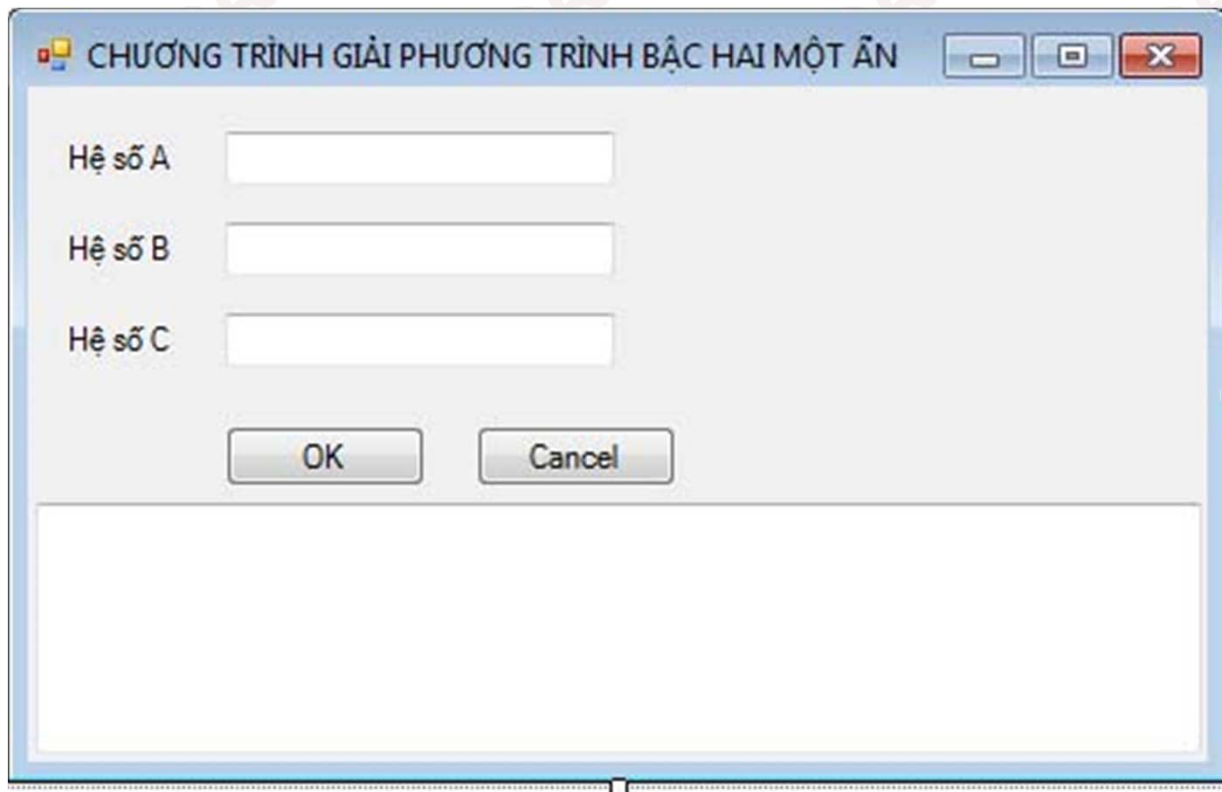
- Sự kiện:

Sự kiện	Ý nghĩa
Click	Xuất hiện khi kích chuột trên nút bấm.

6.3.3. ĐỐI TƯỢNG BUTTON (tiếp theo)

Ví dụ: Chương trình giải phương trình bậc hai $ax^2 + bx + c = 0$

- Bước 1: Sử dụng các đối tượng Label, TextBox, Button thiết kế giao diện như sau:



The image shows a screenshot of a Windows application window titled "CHƯƠNG TRÌNH GIẢI PHƯƠNG TRÌNH BẬC HAI MỘT ẨN". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is light gray and contains three input fields, each preceded by a label: "Hệ số A", "Hệ số B", and "Hệ số C". Below these input fields are two buttons labeled "OK" and "Cancel". The window is set against a white background with a large, faint "TOPICA" watermark.

6.3.3. ĐỐI TƯỢNG BUTTON (tiếp theo)

- Bước 2: Thiết lập các thuộc tính cho các đối tượng
 - Label: Các Label lần lượt là lblA, lblB, lblC.
 - TextBox:
 - Các TextBox cho nhập các hệ số A,B,C được đặt tên lần lượt là: txtA, txtB, txtC;
 - TextBox chứa kết quả đặt tên là txtResult và thiết lập thuộc tính Enabled là False.
 - Button:
 - Nút OK: Đặt tên là btnOK;
 - Nút Cancel: Đặt tên là btnCancel
- Bước 3: Viết lệnh xử lý sự kiện cho các nút bấm
 - Sự kiện Click cho nút Cancel: Khi bấm nút Cancel sẽ kết thúc chương trình

```
private void btnCancel_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```
 - Sự kiện Click cho nút OK: Khi bấm nút OK chương trình sẽ đọc dữ liệu từ các hộp TextBox và lưu vào các biến a, b, c sau đó sẽ thực hiện giải phương trình để tìm nghiệm và thông báo kết quả ra TextBox Result.

6.3.3. ĐỐI TƯỢNG BUTTON (tiếp theo)

```
if (delta > 0)
{
    private void btnOK_Click(object sender, EventArgs e)
    {
        x1 = (-b + Math.Sqrt(delta)) / (2 * a);
        x2 = (-b - Math.Sqrt(delta)) / (2 * a);
        msg = "PHƯƠNG TRÌNH CÓ 2 NGHIỆM PHÂN BIỆT";
        msg += "\r\nX1 = " + x1;
        msg += "\r\nX2 = " + x2;
    }
    //LAY DU LIEU TU CAC DIEU KHIEN TREN FORM
}
else if (delta == 0)
{
    a = float.Parse(txtA.Text);
    {
        b = float.Parse(txtB.Text);
        x1 = x2 = -b / (2 * a);
        msg = "PHƯƠNG TRÌNH CÓ NGHIỆM KÉP\r\n";
        msg += "x1 = x2 = " + x1;
    }
}
else msg = "PHƯƠNG TRÌNH VÔ NGHIỆM";
txtResult.Text = msg;
}
```

6.3.4. ĐỐI TƯỢNG LISTBOX (tiếp theo)

- Là đối tượng được sử dụng để hiển thị dữ liệu dưới dạng danh sách, cho phép người dùng chọn các mục dữ liệu trong danh sách.
- Các thuộc tính cơ bản của ListBox:

Thuộc tính	Ý nghĩa
Name	Tên đối tượng TextBox
Font	Thiết lập các tham số về font chữ cho đối tượng
ForeColor	Thiết lập màu cho thông tin hiển thị trên TextBox
Visible	Hiển thị/ẩn Label
DataSource	Thiết lập nguồn dữ liệu cho đối tượng ListBox, thường kết hợp với các đối tượng DataTable
DisplayMember	Xác định trường dữ liệu được hiển thị khi kết hợp với DataSource
Items	Danh sách các mục dữ liệu trong ListBox, cho phép thiết lập hoặc lấy ra các mục dữ liệu từ ListBox
SelectionMode	Thiết lập chế độ lựa chọn các mục trong ListBox, cho phép chọn một mục hay nhiều mục dữ liệu từ ListBox
ValueMember	Giá trị cho từng mục dữ liệu trong ListBox (không phải dữ liệu hiển thị trên ListBox)

6.3.4. ĐỐI TƯỢNG LISTBOX (tiếp theo)

- Các thuộc tính cơ bản của ListBox:

Thuộc tính	Ý nghĩa
SelectedIndex	Trả về chỉ số mục dữ liệu được chọn.
SelectedItem	Trả về mục dữ liệu được chọn trong ListBox.
SelectedItems	Trả về tập hợp các mục dữ liệu được chọn trong ListBox.
SelectedValue	Trả về giá trị của mục dữ liệu được chọn trong ListBox.
Text	Thông tin hiển trên ListBox.
Sorted	Cho phép sắp xếp dữ liệu trên ListBox.

- Các phương thức của ListBox:

Phương thức	Ý nghĩa
ClearSelected	Hủy bỏ việc lựa chọn đối với các mục dữ liệu đã được chọn.
GetSelected(index)	Trả về True nếu mục có chỉ số index được chọn ngược lại trả về giá trị False.
SetSelected	Thiết lập một mục dữ liệu trong ListBox thành mục được chọn.
Add	Thêm một mục dữ liệu vào ListBox.
Remove	Loại bỏ một mục dữ liệu ra khỏi ListBox.

6.3.4. ĐỐI TƯỢNG LISTBOX (tiếp theo)

- Các phương thức của ListBox:

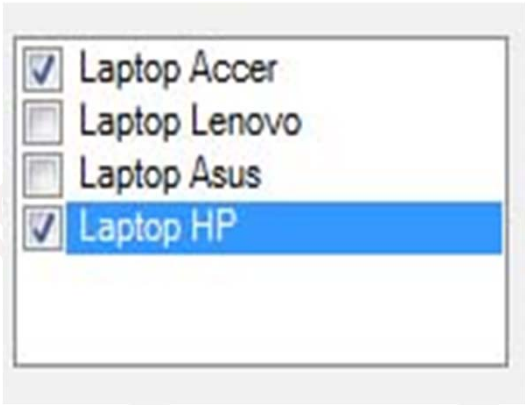
Phương thức	Ý nghĩa
ClearSelected	Hủy bỏ việc lựa chọn đối với các mục dữ liệu đã được chọn.
GetSelected(index)	Trả về True nếu mục có chỉ số index được chọn ngược lại trả về giá trị False.
SetSelected	Thiết lập một mục dữ liệu trong ListBox thành mục được chọn.
Add	Thêm một mục dữ liệu vào ListBox.
Remove	Loại bỏ một mục dữ liệu ra khỏi ListBox.

- Các sự kiện của ListBox:

Sự kiện	Ý nghĩa
SelectedIndexChanged	Xuất hiện khi chỉ số của các mục dữ liệu trong ListBox thay đổi, thường xuất hiện khi người sử dụng chọn các mục dữ liệu khác nhau trong ListBox.
SelectedValueChanged)	Xuất hiện khi giá trị trên các mục dữ liệu trong ListBox thay đổi.

6.3.5. ĐỐI TƯỢNG CHECKEDLISTBOX

- Tương tự như ListBox tuy nhiên các đối tượng hiển thị dưới dạng hộp kiểm.



- Các thuộc tính của CheckedListBox:

Thuộc tính	Ý nghĩa
CheckedIndices	Tập hợp các chỉ số của các mục dữ liệu được chọn.
CheckedItems	Tập hợp các mục dữ liệu được chọn.
Items	Tập các mục dữ liệu trong CheckedListBox.

6.3.5. ĐỐI TƯỢNG CHECKEDLISTBOX (tiếp theo)

- Các phương thức của CheckedListBox:

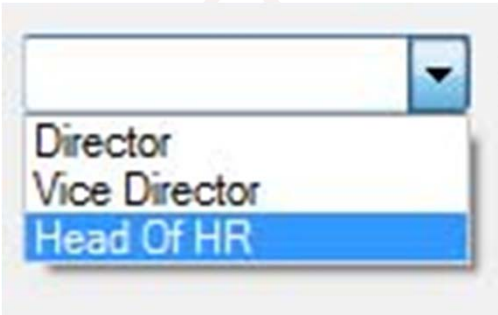
Phương thức	Ý nghĩa
SetItemChecked	Thiết lập hoặc hủy bỏ việc lựa chọn cho các mục dữ liệu
GetItemChecked(index)	Trả về True nếu mục index được lựa chọn, ngược lại trả về False
GetItemCheckState(index)	Trả về trạng thái của mục dữ liệu có chỉ số index
SetItemCheckState(index)	Thiết lập hoặc hủy bỏ trạng thái của mục dữ liệu có chỉ số index

- Sự kiện của CheckedListBox:

Sự kiện	Ý nghĩa
ItemCheck	Xuất hiện khi người sử dụng thực hiện tích chọn trên một mục dữ liệu.

6.3.6. ĐỐI TƯỢNG COMBOBOX

- Được sử dụng hiển thị dữ liệu dưới dạng danh sách các mục chọn trong hộp thả, cho phép người dùng chọn một mục dữ liệu từ danh sách.



- Các thuộc tính của ComboBox:

Thuộc tính	Ý nghĩa
DropDownStyle	Kiểu hiển thị của hộp Combo
MaxDropDownItems	Số mục dữ liệu tối đa được hiển thị trên hộp Combo
Items	Tập các mục dữ liệu trong hộp Combo

6.3.6. ĐỐI TƯỢNG COMBOBOX (tiếp theo)

- Các phương thức của ComboBox:

Phương thức	Ý nghĩa
Show	Hiển thị control
Dispose	Hủy đối tượng

- Các sự kiện của ComboBox:

Sự kiện	Ý nghĩa
DropDown	Xuất hiện khi người sử dụng bấm vào hộp thả để chọn các mục dữ liệu

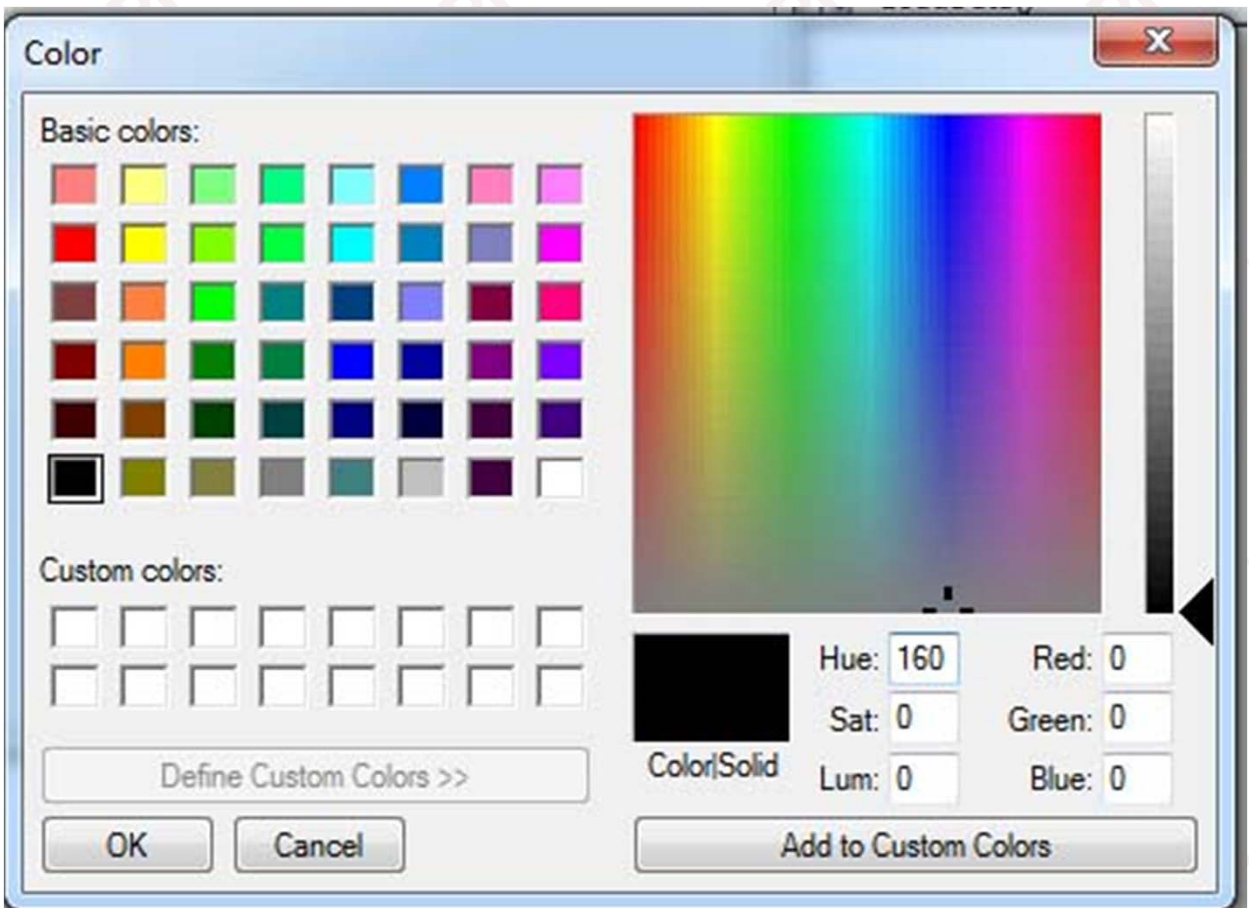
6.3.7. ĐỐI TƯỢNG CHECKBOX VÀ RADIO

Cho phép người sử dụng lựa chọn bằng cách tích chọn trên đối tượng.

Thuộc tính	Ý nghĩa
Checked	Nhận giá trị True nếu Checkbox/Radio được chọn, ngược lại False
Locked	Cho phép/không cho phép chọn trên đối tượng

6.4. CÁC HỘP THOẠI CƠ BẢN

- Hộp thoại được sử dụng để cung cấp thêm thông tin cho người dùng, giúp người sử dụng dễ dàng tương tác với hệ thống.
- Công cụ nhập dữ liệu cho hệ thống:



6.4. CÁC HỘP THOẠI CƠ BẢN (tiếp theo)

Types of Dialog (by definition)

Common dialog boxes



- OpenFileDialog
- PageSetupDialog
- FontDialog
- ColorDialog
- SaveFileDialog
- PrintPreviewDialog
- PrintDialog

6.4. CÁC HỘP THOẠI CƠ BẢN (tiếp theo)

- Hộp thoại OpenFileDialog: Cho phép chọn tệp tin cần mở trên hệ thống tệp của máy.

Thuộc tính	Ý nghĩa
CheckFileExists	Kiểm tra sự tồn tại của tệp tin được mở.
FileName	Trả về tên đầy đủ của tệp tin được mở.
Filter	Thiết lập bộ lọc cho phép hiển thị các kiểu tệp trên hộp thoại.
Multiselect	Thiết lập thuộc tính cho phép chọn nhiều tệp tin trong hộp thoại.

Phương thức	Ý nghĩa
ShowDialog	Hiển thị hộp thoại OpenFileDialog.
OpenFile	Mở tệp tin.
Dispose	Hủy bỏ đối tượng.

6.4. CÁC HỘP THOẠI CƠ BẢN (tiếp theo)

- Hộp thoại FontDialog: Cho phép chọn kiểu font chữ.

Thuộc tính	Ý nghĩa
ShowEffects	Hiển thị/ẩn các phần lựa chọn các hiệu ứng khác cho font chữ.
MinSize	Thiết lập cỡ chữ nhỏ nhất trên hộp thoại font.
ShowHelp	Hiển thị/ẩn nút trợ giúp.
ShowColor	Hiển thị/ẩn nút chọn màu chữ.
MaxSize	Thiết lập cỡ chữ lớn nhất trên hộp thoại font.
Color	Thiết lập hoặc lấy về màu sắc font chữ được chọn.
Font	Thiết lập hoặc lấy về kiểu font được chọn.

Phương thức	Ý nghĩa
ShowDialog	Hiển thị hộp thoại OpenFileDialog.
Dispose	Hủy bỏ đối tượng.

6.4. CÁC HỘP THOẠI CƠ BẢN (tiếp theo)

- Hộp thoại ColorDialog: Cho phép chọn màu sắc.

Thuộc tính	Ý nghĩa
AllowFullOpen	Hiển thị hộp chọn màu sắc một cách đầy đủ.
AnyColor	Cho phép hiển thị tất cả các màu.
Color	Trả về màu sắc được chọn bởi người sử dụng.
ShowHelp	Hiển thị/ẩn nút trợ giúp.

Phương thức	Ý nghĩa
ShowDialog	Hiển thị hộp thoại OpenFileDialog.
Dispose	Hủy bỏ đối tượng.

6.4. CÁC HỘP THOẠI CƠ BẢN (tiếp theo)

- Hộp thoại SaveFileDialog: Cho phép chọn hộp thoại thực hiện việc ghi tệp tin lên đĩa.

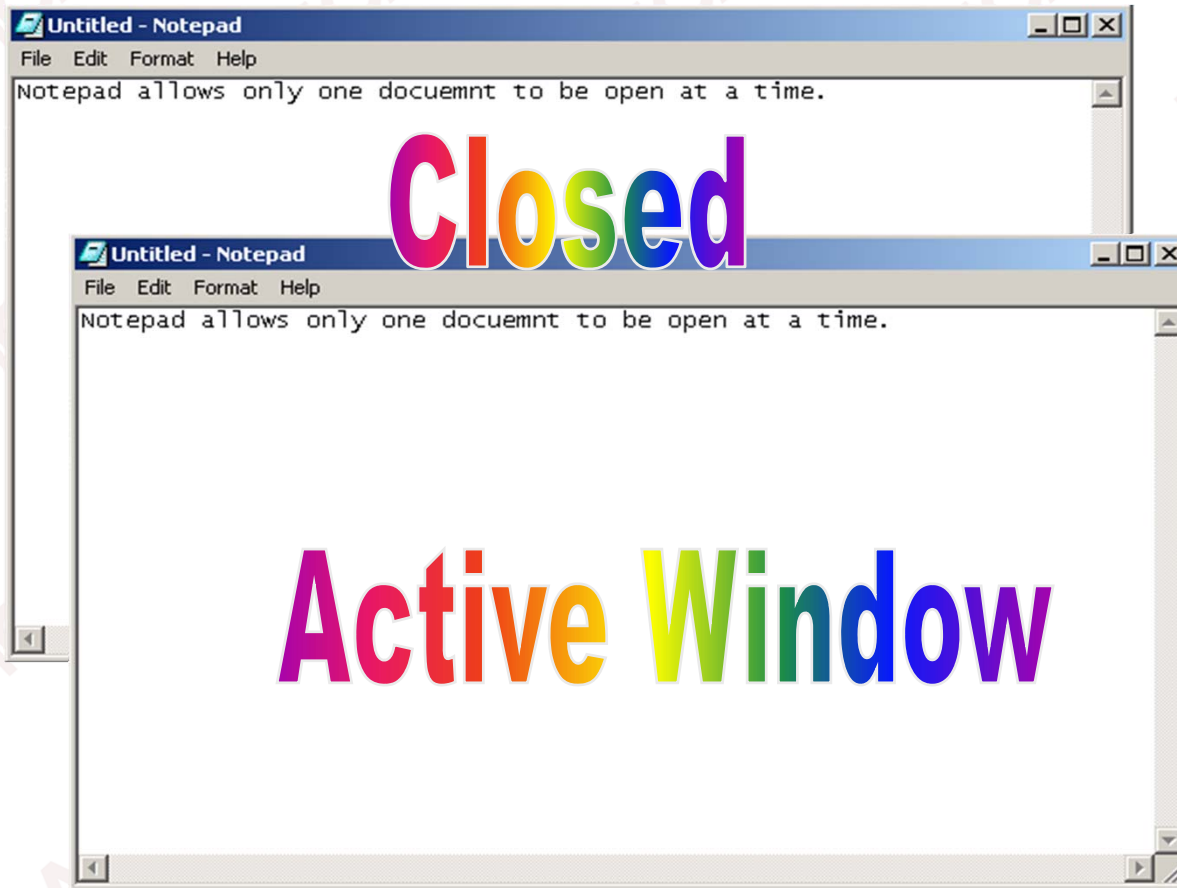
Thuộc tính	Ý nghĩa
CheckFileExists	Kiểm tra sự tồn tại của tệp tin được mở.
FileName	Trả về tên đầy đủ của tệp tin được mở.
Filter	Thiết lập bộ lọc cho phép hiển thị các kiểu tệp trên hộp thoại.
Multiselect	Thiết lập thuộc tính cho phép chọn nhiều tệp tin trong hộp thoại.

Phương thức	Ý nghĩa
ShowDialog	Hiển thị hộp thoại OpenFileDialog.
Dispose	Hủy bỏ đối tượng.

Sự kiện	Ý nghĩa
FileOk	Xuất hiện khi người sử dụng kích chuột vào nút Open hoặc nút Save.

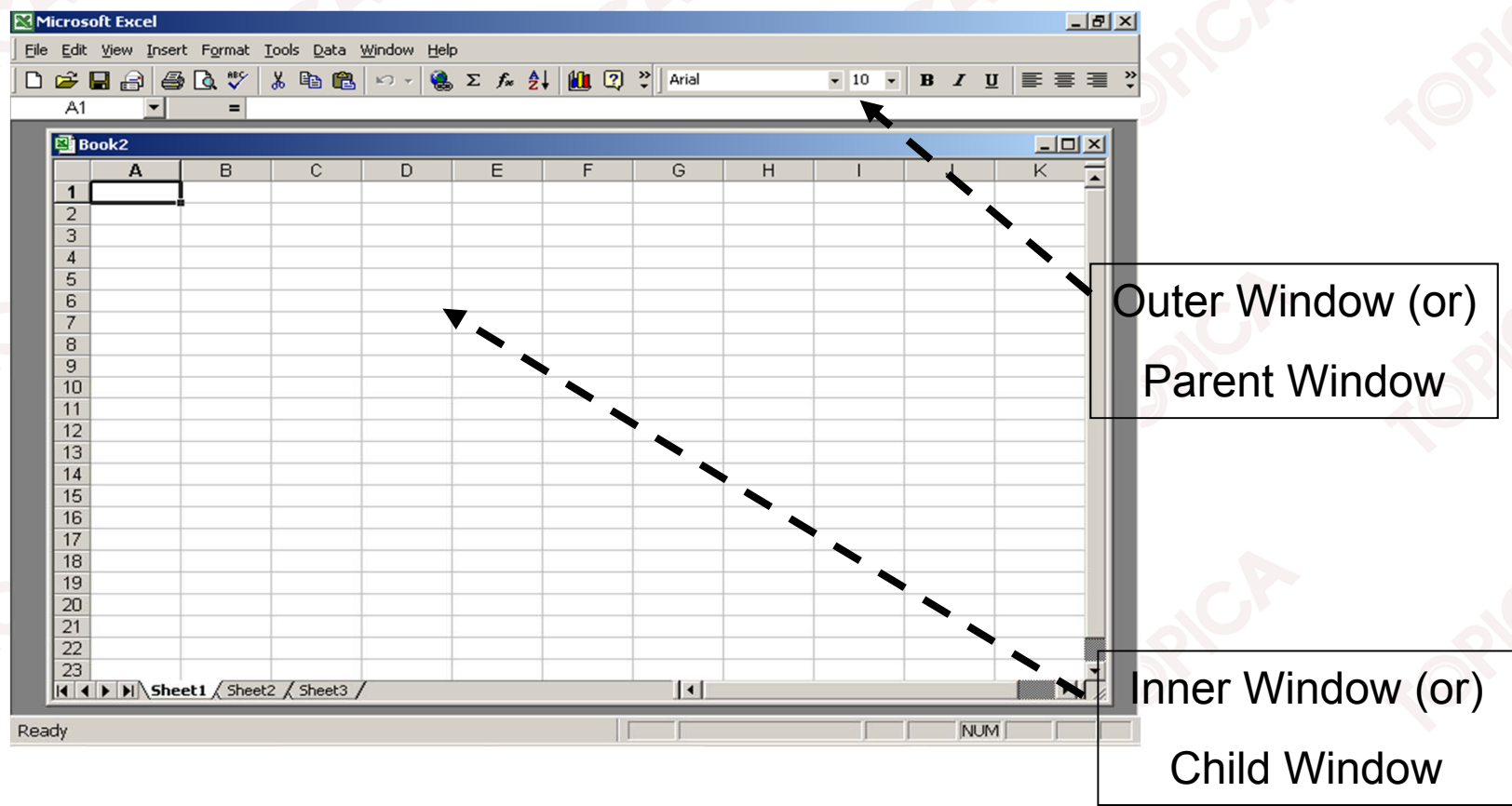
6.5. THỰC ĐƠN

- Ứng dụng SDI (Single Document Interface): Ứng dụng chỉ có một cửa sổ hoạt động tại một thời điểm.



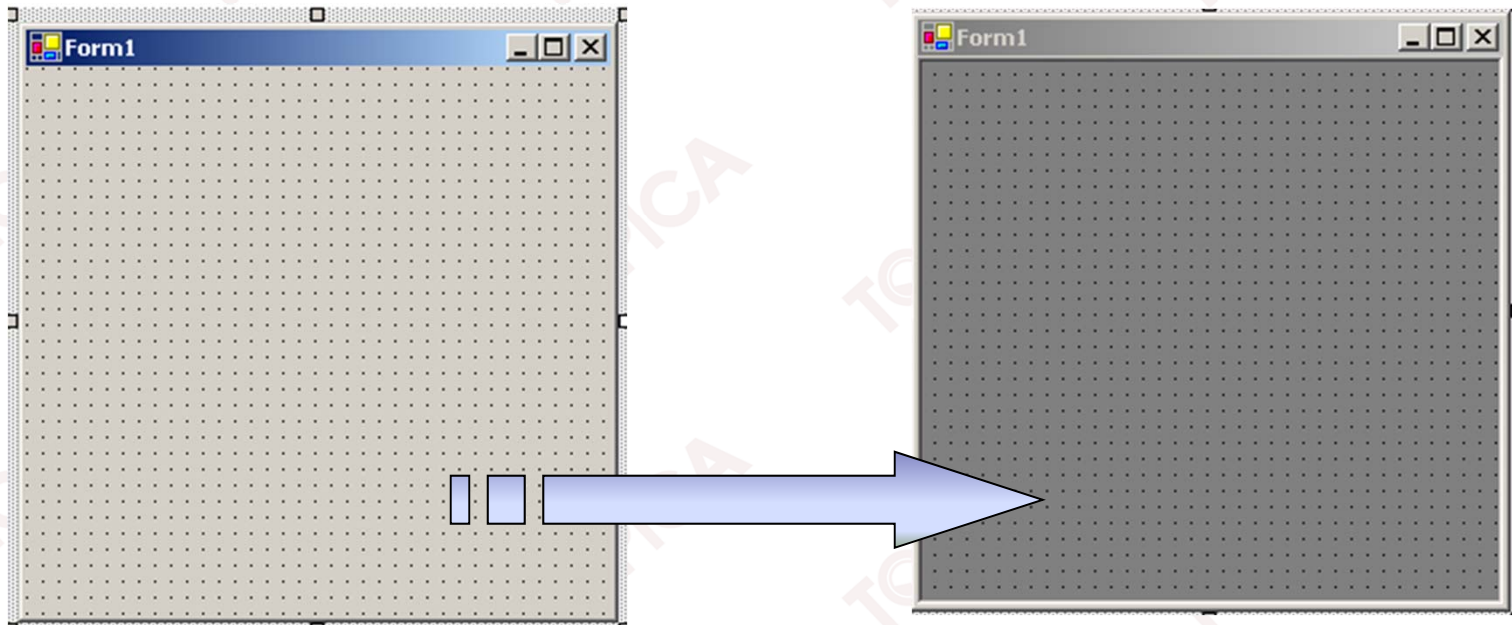
6.5. THỰC ĐƠN (tiếp theo)

- Ứng dụng MDI (Multiple Document Interface): Ứng dụng cho phép có nhiều cửa sổ hoạt động tại một thời điểm.



6.5. THỰC ĐƠN (tiếp theo)

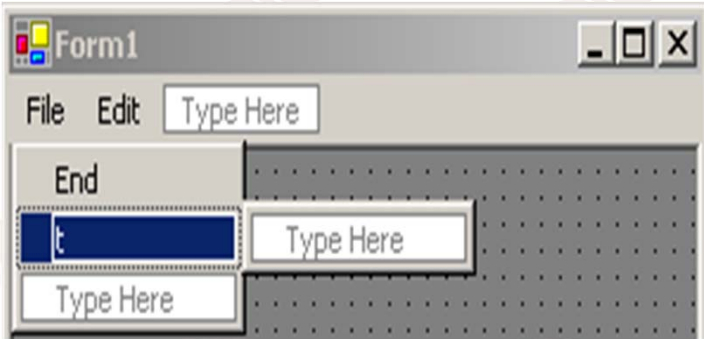
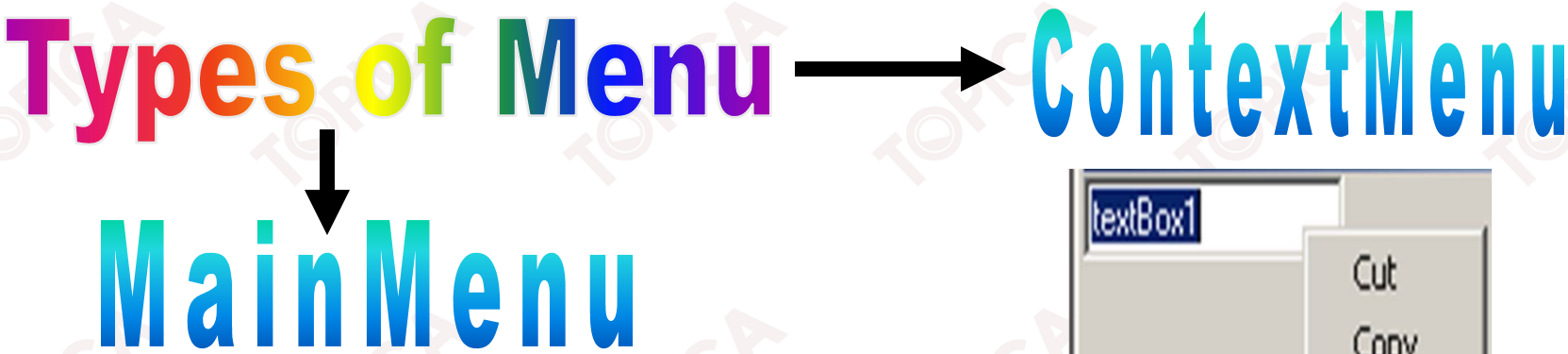
- Tạo ứng dụng MDI Application:
 - Tạo một Form: IsMdiContainer = True;
 - Tạo các Form con, thêm thực đơn cho Form chính.



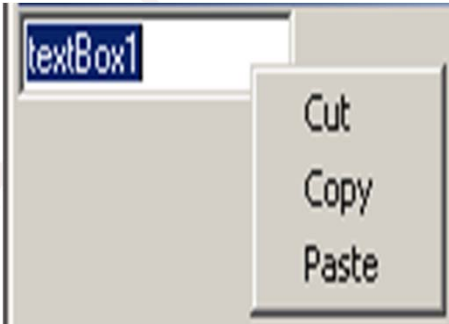
IsMdiContainer = true

6.5. THỰC ĐƠN (tiếp theo)

Các kiểu thực đơn trong ứng dụng: menu chính, menu ngữ cảnh.



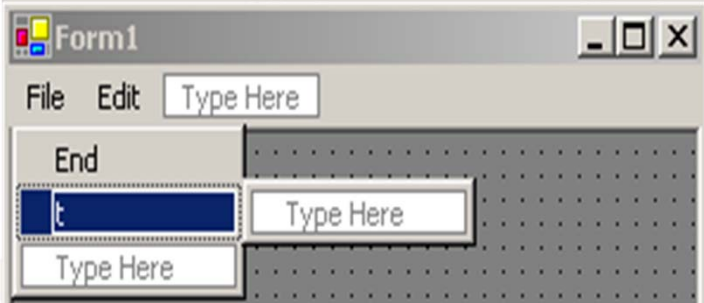
Thiết kế trong ứng dụng MDI



Xuất hiện chạy chương trình

6.5. THỰC ĐƠN (tiếp theo)

- Menu chính (Main Menu):
 - Xuất hiện ở đỉnh cửa sổ chính trong ứng dụng;
 - Tạo main menu bằng 2 cách: thiết kế, code.



Thiết kế giao diện bằng
công cụ thiết kế

Thiết kế menu bằng code

```
MainMenu myMainMenu = new MainMenu();
```

```
this.Menu = myMainMenu;
```

- Các thuộc tính của MenuItem:

Thuộc tính	Ý nghĩa
Checked	Thiết lập thuộc tính cho phép xuất hiện dấu tích trên mục menu.
Index	Thiết lập mục menu được chọn trong thực đơn.
IsParent	Trả về giá trị True nếu mục menu có chứa các mục menu con.

Trong bài này, chúng ta đã nghiên cứu các nội dung chính sau:

- Khái niệm về lập trình hướng sự kiện, sự kiện;
- Các thuộc tính, phương thức và sự kiện của lớp form;
- Một số đối tượng điều khiển cơ bản;
- Ứng dụng MDI và đối tượng thực đơn.