

C# Programming

Các điều khiển cơ bản của C#

By Hoàng Hữu Việt

Email: viethh@vinhuni.edu.vn

Viện Kỹ thuật và Công nghệ, Đại học Vinh

Cao Thanh Sơn

Email: sonct@vinhuni.edu.vn

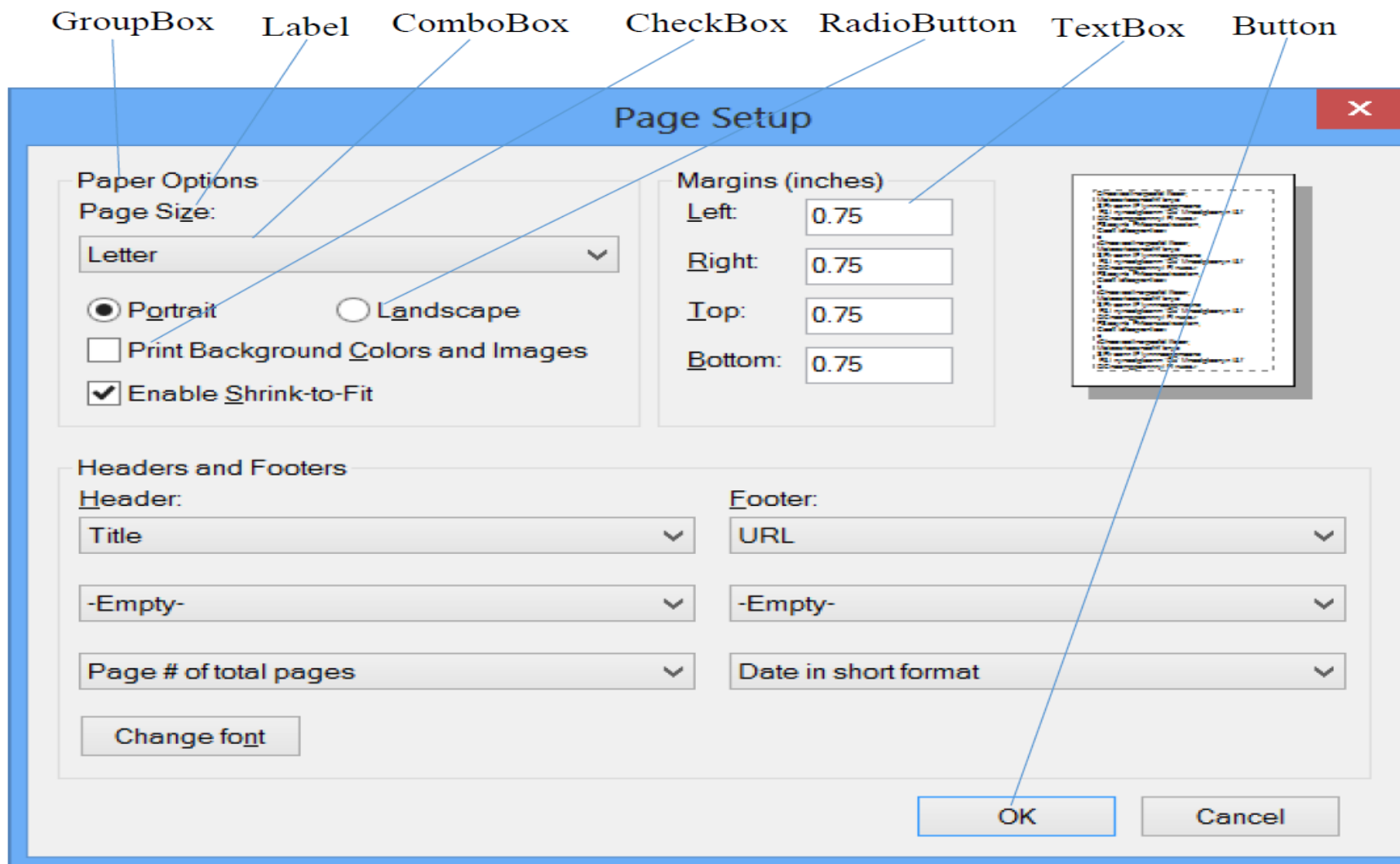
Vinh, 9/2020

Mục đích, chuẩn đầu ra và nội dung

- Mục đích
 - ❑ Giới thiệu về các điều khiển cơ bản để xây dựng các giao diện của ứng dụng.
- Chuẩn đầu ra
 - ❑ Lập trình các lớp điều khiển cơ bản để xây dựng các giao diện của các ứng dụng.
 - ❑ Thiết kế và lập trình các giao diện người dùng của ứng dụng.
 - ❑ Phát triển kỹ năng lập trình, kỹ năng tìm kiếm và đọc hiểu tài liệu.
- Nội dung
 - ❑ Điều khiển Form, Label, Textbox và Button
 - ❑ Điều khiển GroupBox và Panel
 - ❑ Điều khiển CheckBox và RadioButton
 - ❑ Điều khiển PictureBox
 - ❑ Xử lý các sự kiện con chuột và bàn phím

Điều khiển Form

- Để tạo giao diện đồ họa cho các ứng dụng



Điều khiển Form

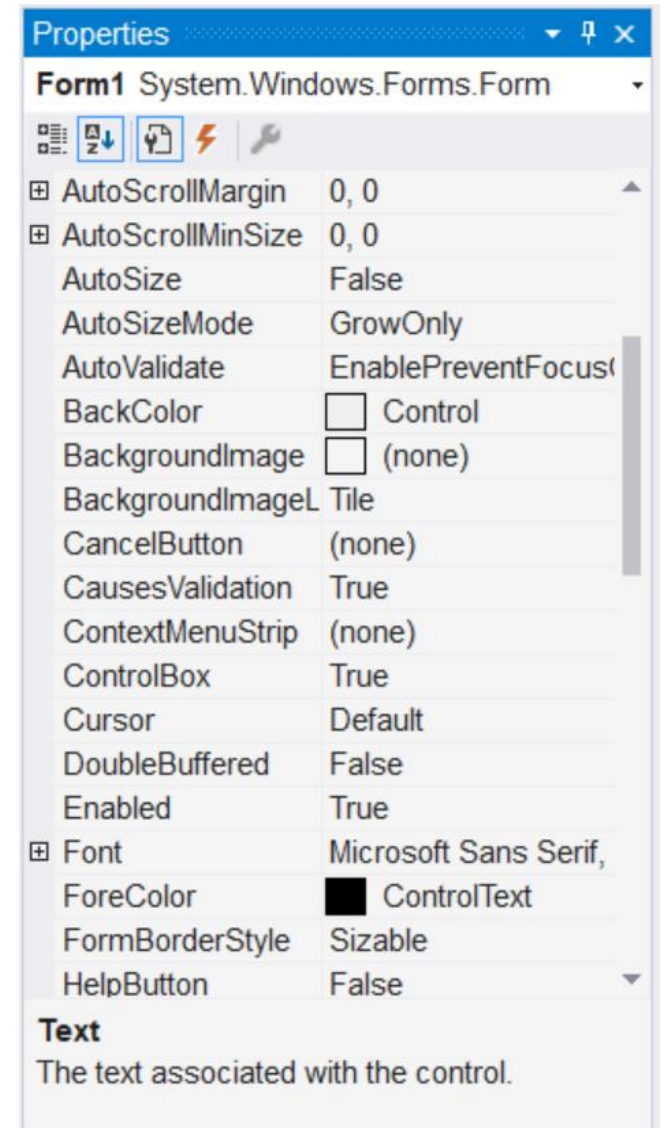
- Tạo một Form
 - Chọn thực đơn Project > Add Windows Form
 - Chọn Visual C# items và chọn Windows Form
 - Nhập tên Form > Add
- Form chạy đầu tiên trong ứng dụng được thiết lập bằng cách nhập tên Form trong tệp Program.cs

```
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Form1());
}
```

Điều khiển Form

■ Các thuộc tính thường dùng

- ❑ Font: Phong chữ ngầm định các đối tượng trên Form
- ❑ FormBorderStyle: Kiểu đường viền của Form
 - None: Form không có đường viền
 - Fix....: Cố định kích thước khi chạy Form
 - Sizeable: Có thể thay đổi kích thước Form
- ❑ ForeColor: Màu chữ ngầm định các đối tượng trên Form
- ❑ Text: Dòng văn bản hiển thị trên tiêu đề Form
- ❑ MaximizeBox: Có/không nút phóng to
- ❑ MinimizeBox: Có/không nút thu nhỏ
- ❑ StartPosition: Vị trí bắt đầu khi chạy Form
- ❑ WindowState: Xác định trạng thái ban đầu Form



Điều khiển Form

- Các phương thức thường dùng
 - ❑ Close(): Đóng Form và giải phóng bộ nhớ
 - ❑ Hide(): Ẩn Form, không giải phóng bộ nhớ
 - ❑ Show(): Hiện thị một Form đã ẩn
- Các sự kiện thường dùng
 - ❑ Load(): Được gọi khi chạy Form
 - ❑ FormClosing(): Được gọi khi đóng Form

Đối tượng = thuộc tính + phương thức + sự

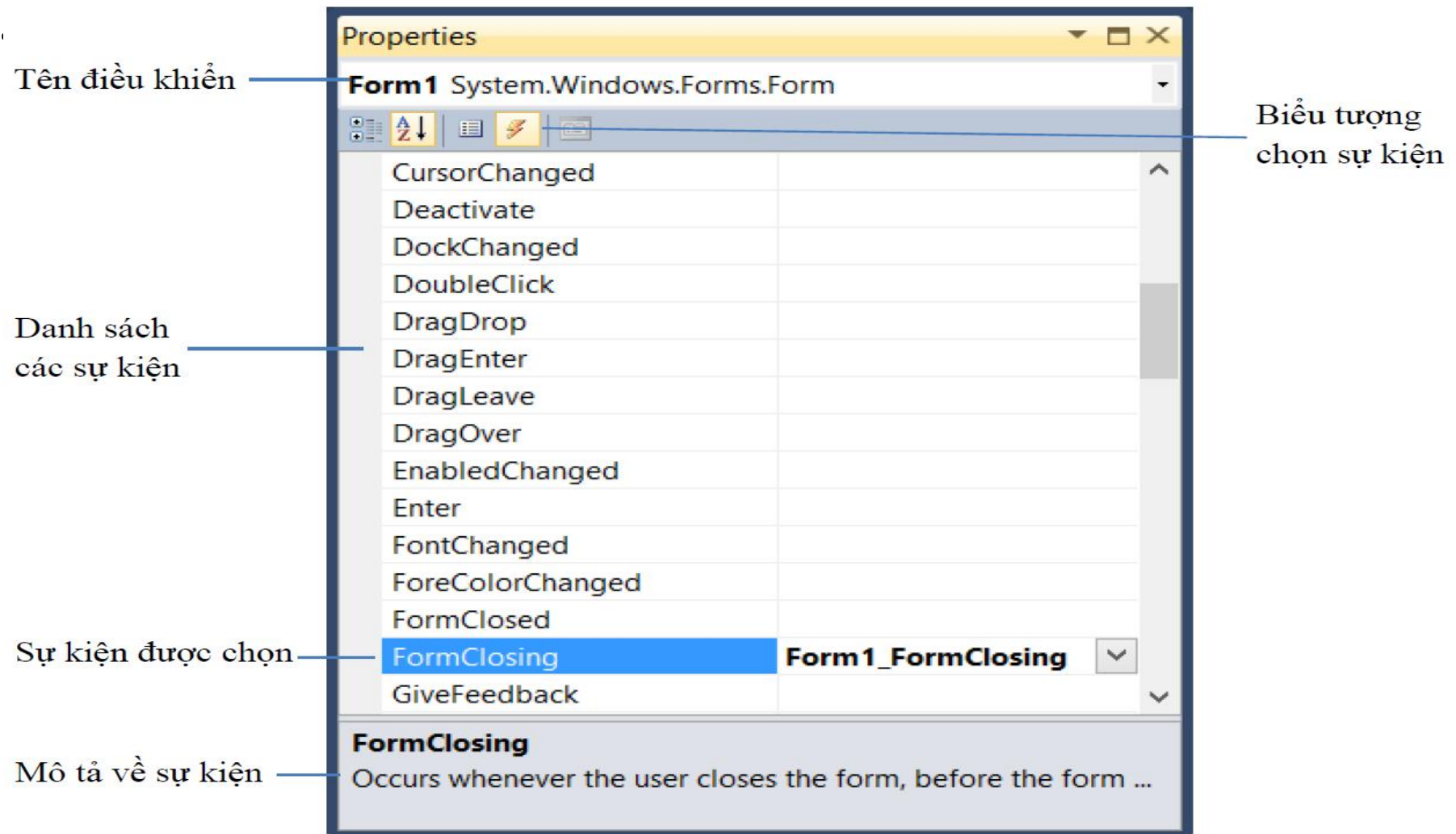
```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
    }

    private void Form1_FormClosing(object sender, FormClosingEventArgs e)
    {
    }
}
```

Mô hình điều khiển sự kiện

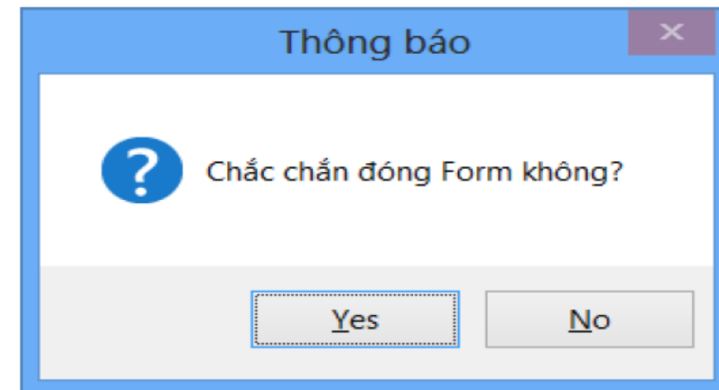
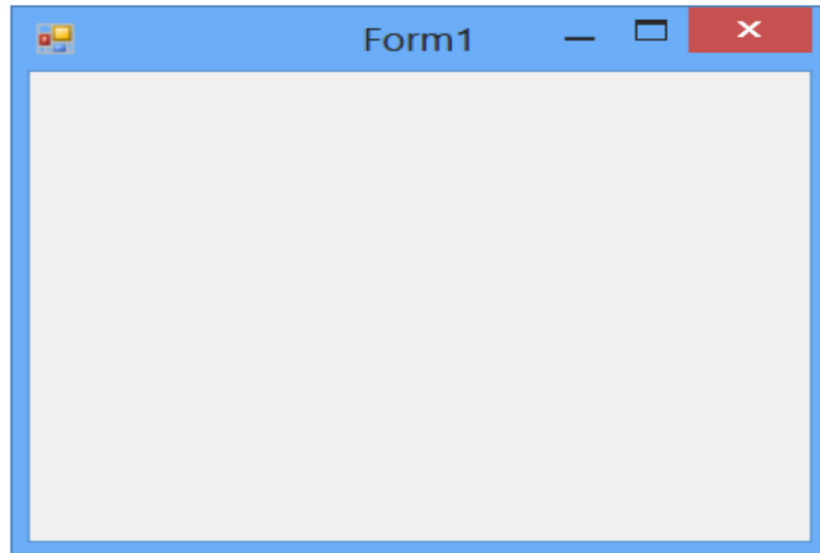
- Sự kiện (event) được sinh ra do người dùng điều khiển bàn phím hoặc con chuột.



Mô hình điều khiển sự kiện

■ Ví dụ sự kiện FormClosing khi đóng Form

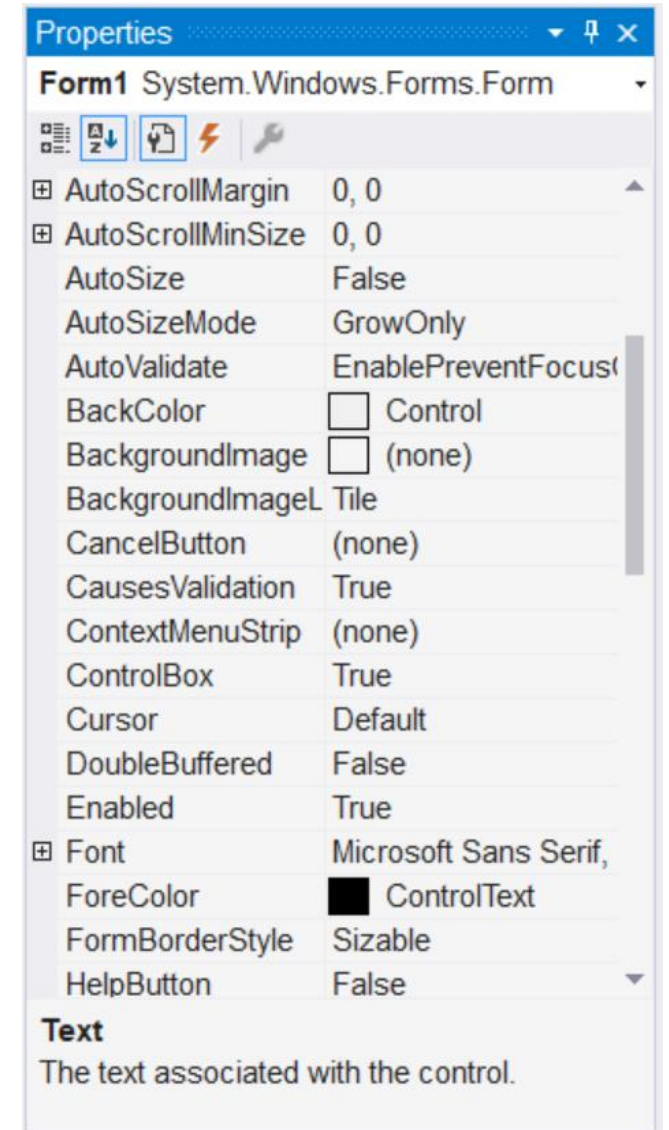
```
private void FormName_FormClosing(object sender,...)
{
    DialogResult dr;
    dr = MessageBox.Show("Chắc chắn đóng Form không?", "Thông báo",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (dr == DialogResult.No) e.Cancel = true;
}
```



Các thuộc tính chung

■ Thuộc tính thường dùng

- ❑ BackColor: Màu nền của đối tượng
- ❑ BackgroundImage: Ảnh nền của đối tượng
- ❑ Cursor: Hình dạng con trỏ chuột trên đối tượng
- ❑ Dock: Thay đổi kích thước theo các cạnh của đối tượng
- ❑ Enabled: Có/không cho phép tương tác với đối tượng
- ❑ Font: Phong chữ hiển thị của đối tượng
- ❑ ForeColor: Màu chữ của đối tượng
- ❑ TabIndex: Số thứ tự chuyển con trỏ khi ấn phím Tab
- ❑ Text: Dòng văn bản xuất hiện trên đối tượng
- ❑ TextAlign: Căn lề của văn bản xuất hiện trên đối tượng
- ❑ Visible: Có/không hiển thị đối tượng



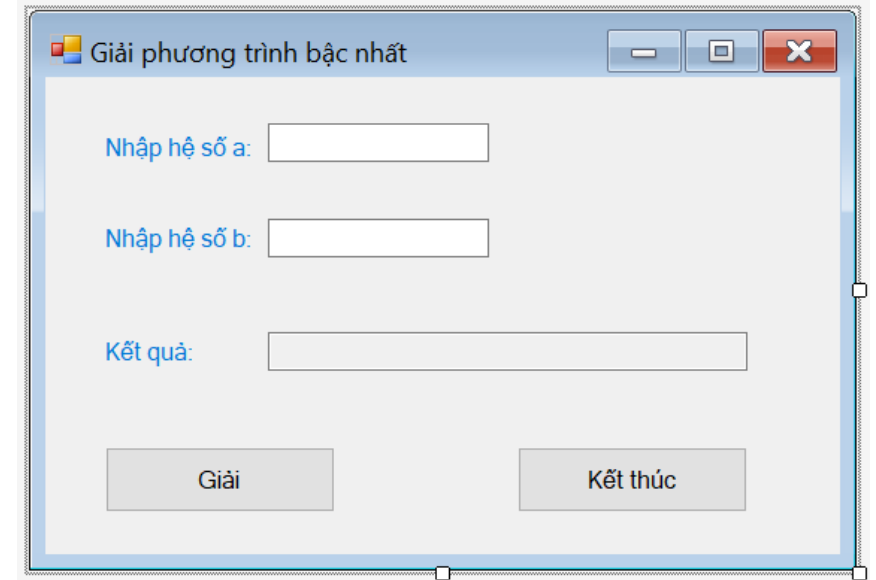
Các điều khiển Label, TextBox, Button

■ Label

- ❑ Tạo các đối tượng hiển thị dữ liệu/hình ảnh
- ❑ Text: Dòng văn bản hiển thị ở đối tượng
- ❑ Font: Phong chữ của đối tượng
- ❑ ForeColor: Màu chữ của dòng văn bản

■ TextBox

- ❑ Tạo các đối tượng nhập hoặc hiển thị dữ liệu
- ❑ Dữ liệu ở đối tượng lớp TextBox là xâu ký tự, nếu muốn dữ liệu là dạng số thì phải chuyển kiểu dữ liệu
- ❑ Text: Dòng dữ liệu hiển thị ở đối tượng
- ❑ Multiline: Có/không cho phép nhập/hiển thị nhiều dòng
- ❑ PasswordChar: Định nghĩa ký tự hiển thị khi nhập dữ liệu



Các điều khiển Label, TextBox, Button

■ Button

- ❑ Dùng để tạo đối tượng thực hiện chức năng trên Form
- ❑ Text: Dòng văn bản hiển thị ở đối tượng
- ❑ Click(): Sự kiện được gọi khi nháy chuột hoặc ấn phím Enter

■ Ví dụ xây dựng form giải phương trình bậc 1

The screenshot shows a Windows application window titled "Giải phương trình bậc nhất". 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 with blue labels: "Nhập hệ số a:", "Nhập hệ số b:", and "Kết quả:". Below these input fields are two buttons: "Giải" and "Kết thúc".

Các điều khiển Label, TextBox, Button

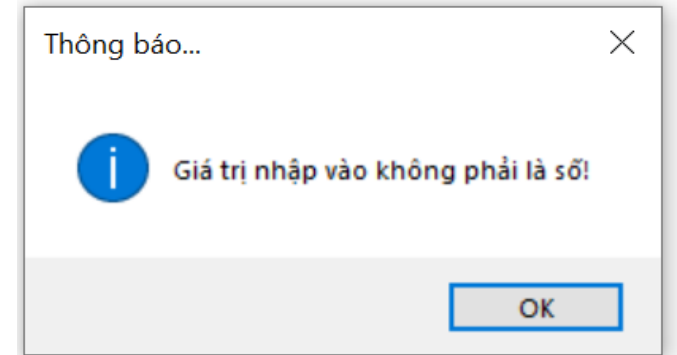
- Chuyển dữ liệu từ chuỗi ký tự sang số
 - Chuyển không kiểm tra dữ liệu: sử dụng hàm `Convert()`
 - `double num = Convert.ToDouble(str);`
 - Chuyển có kiểm tra dữ liệu: sử dụng hàm `TryParse()`
 - `bool IsNumber = double.TryParse(str, out num);`
 - `IsNumber = true` nếu đổi được
 - `IsNumber = false` nếu không đổi được
- Chuyển dữ liệu từ số sang chuỗi
 - Sử dụng hàm `Convert()`
 - `str = Convert.ToString(num);`
 - Sử dụng hàm `ToString()` của đối tượng
 - `obj.ToString();`

Các điều khiển Label, TextBox, Button

■ Hàm MessageBox

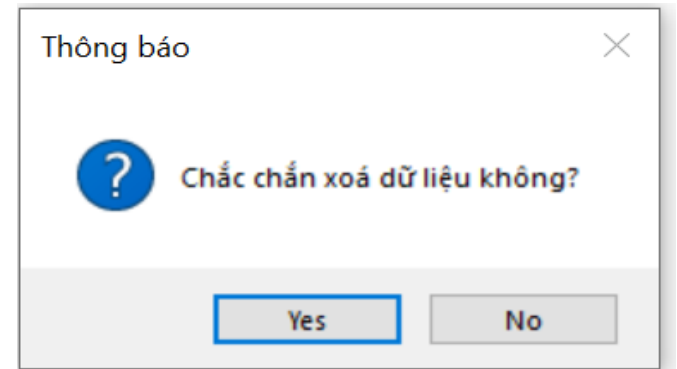
- Dùng để hiển thị một thông báo

```
MessageBox.Show("Giá trị nhập vào không phải là số!",  
                "Thông báo...", MessageBoxButtons.OK,  
                MessageBoxIcon.Information);
```



- Dùng để xác nhận một hành động

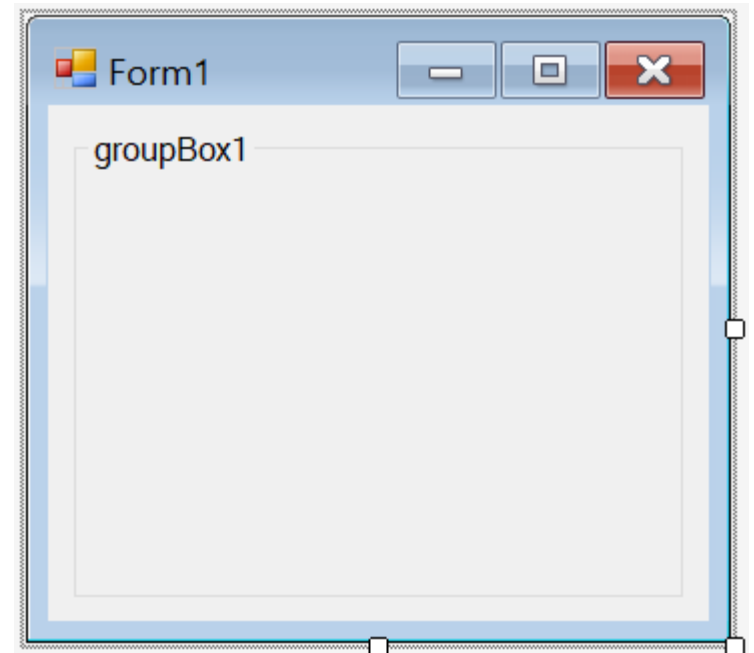
```
DialogResult dr;  
dr = MessageBox.Show("Chắc chắn xóa dữ liệu không?",  
                    "Thông báo", MessageBoxButtons.YesNo,  
                    MessageBoxIcon.Question);  
if (dr == DialogResult.No) return;
```



■ Viết mã lệnh giải phương trình bậc 1 ?

Điều khiển GroupBox

- Dùng để tạo các đối tượng chứa các đối tượng lớp điều khiển
- Khi kéo điều khiển từ cửa sổ Toolbox đặt vào đối tượng lớp GroupBox, đối tượng lớp GroupBox sẽ chứa đối tượng lớp điều khiển
- Khi di chuyển đối tượng lớp GroupBox, các đối tượng được chứa trong đối tượng lớp GroupBox sẽ di chuyển theo
- Các thuộc tính thường dùng
 - ❑ Controls: Các đối tượng được chứa trong GroupBox
 - ❑ Text: Dòng dữ liệu hiển thị trên tiêu đề của GroupBox

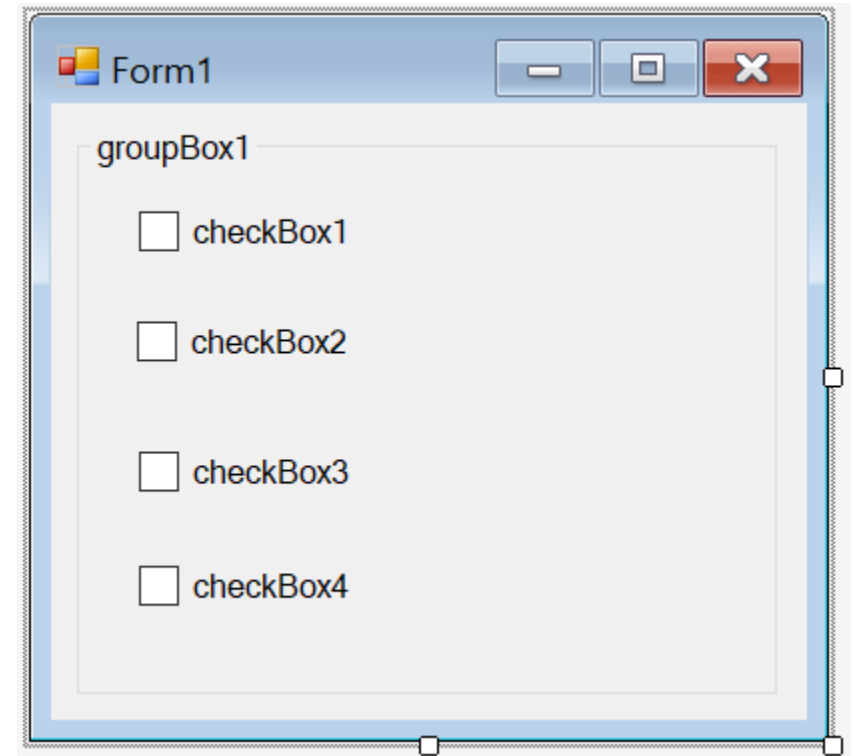


Điều khiển Panel

- Dùng để tạo các đối tượng chứa các đối tượng lớp điều khiển.
- Đối tượng lớp GroupBox có tiêu đề và đối tượng lớp Panel có các thanh cuộn.
- Các thuộc tính thường dùng:
 - ❑ AutoScroll: Nếu True, thanh cuộn tự động xuất hiện khi đối tượng lớp Panel không chứa hết các đối tượng khác.
 - ❑ BorderStyle: Định nghĩa kiểu đường viền của đối tượng.
 - ❑ Controls: Các đối tượng được chứa trong đối tượng lớp Panel.

Điều khiển CheckBox

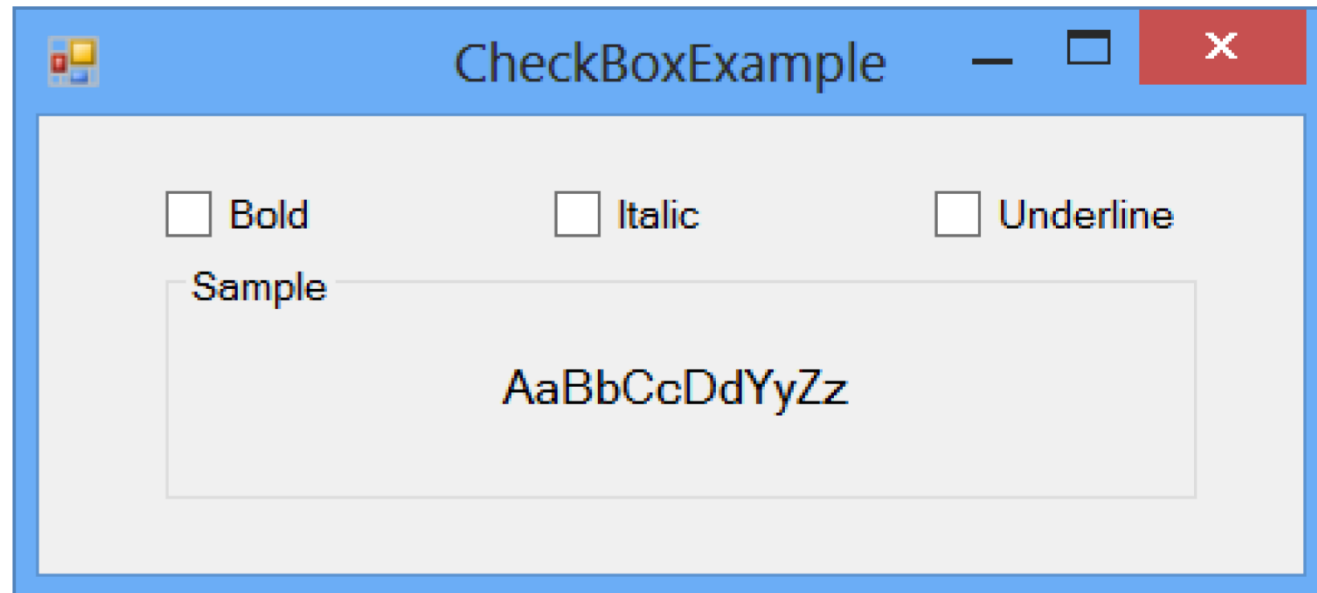
- Dùng để tạo các đối tượng cho phép chọn trạng thái đúng/sai (true/false) của thuộc tính dữ liệu.
- Có thể chọn nhiều đối tượng lớp CheckBox.
- Các thuộc tính thường dùng:
 - Text: Dòng dữ liệu hiển thị ở đối tượng.
 - Checked: Có/không chọn đối tượng.
- Sự kiện thường dùng:
 - CheckedChanged(): Được gọi khi nháy chuột để thay đổi trạng thái chọn/không chọn đối tượng.



Điều khiển CheckBox

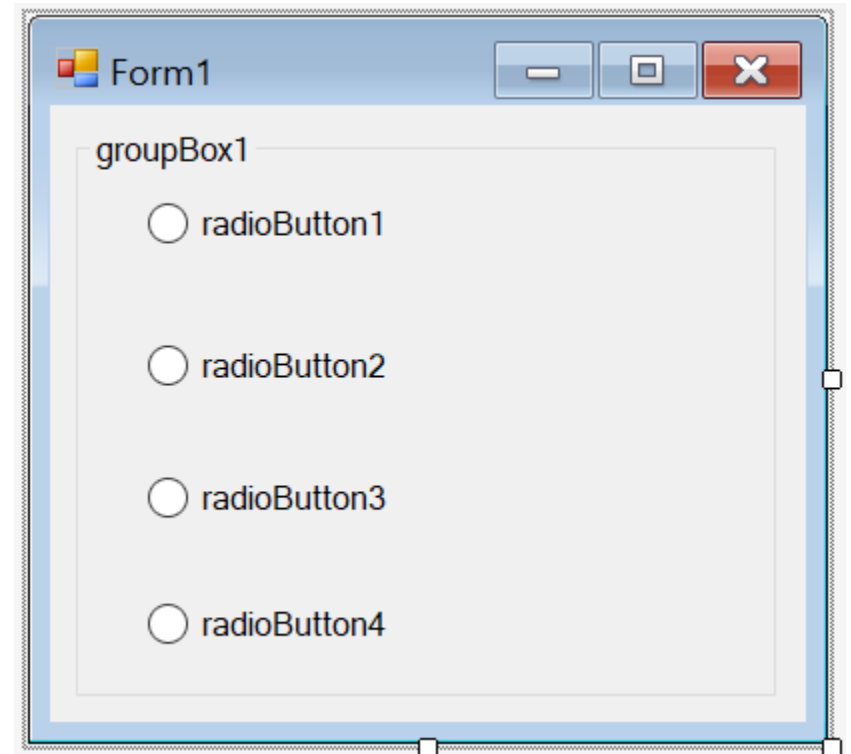
- Ví dụ xây dựng một form để thay đổi kiểu chữ của một đối tượng lớp Label:

```
private void chkBold_CheckedChanged(object sender, ...)
{
    lblSample.Font = new Font(lblSample.Font.Name,
    lblSample.Font.Size, lblSample.Font.Style ^ FontStyle.Bold);
}
```



Điều khiển RadioButton

- Dùng để tạo các đối tượng cho phép chọn trạng thái đúng/sai của thuộc tính dữ liệu.
- Chỉ cho phép chọn một đối tượng trong một nhóm các đối tượng.
- Các thuộc tính thường dùng:
 - ❑ Checked: Có/không chọn đối tượng.
 - ❑ Text: Dòng dữ liệu hiển thị ở đối tượng.
- Sự kiện thường dùng:
 - ❑ Click(): Được gọi khi đối tượng được chọn.
 - ❑ CheckedChanged(): Được gọi khi nháy chuột để chọn/không chọn đối tượng.



Điều khiển RadioButton

- Ví dụ xây dựng một form để thay đổi màu chữ, kiểu chữ và cỡ chữ của một đối tượng lớp Label:

The screenshot shows a Windows application window titled "RadioButtonExample". Inside the window, there are three columns of radio buttons for configuring a label's appearance:

- Color:** Three radio buttons labeled "Black", "Blue", and "Red". The "Black" button is selected.
- Style:** Three radio buttons labeled "Regular", "Italic", and "Bold". The "Regular" button is selected.
- Size:** Three radio buttons labeled "12 points", "16 points", and "20 points". The "12 points" button is selected.

Below these columns is a "Sample" label containing the text "AaBbCcDdYyZz". The text is displayed in a black, regular, 12-point font, matching the selected options.

Điều khiển RadioButton

- Mã lệnh thay đổi màu

```
private void rbBlack_CheckedChanged(object sender, ...)
{
    lbSample.ForeColor = Color.Black;
}
```

- Mã lệnh thay đổi kiểu chữ

```
private void rbRegular_CheckedChanged(object sender, ...)
{
    lbSample.Font = new Font(lbSample.Font.Name,
                             lbSample.Font.Size, FontStyle.Regular);
}
```

- Mã lệnh thay đổi cỡ chữ

```
private void rbSize12_CheckedChanged(object sender, ...)
{
    lbSample.Font = new Font(lbSample.Font.Name, 12, lbSample.Font.Style);
}
```

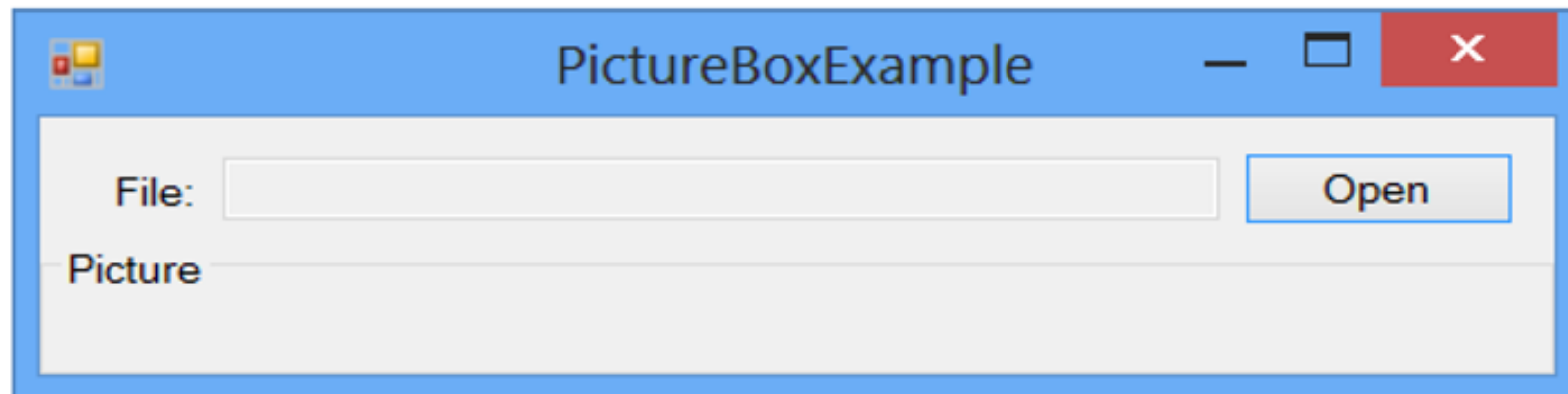
Điều khiển PictureBox

- Dùng để tạo các đối tượng hiển thị hình ảnh.
- Ảnh hiển thị ở đối tượng lớp PictureBox có thể một trong các dạng .bmp, .gif, .jpg, v.v.
- Các thuộc tính thường dùng
 - Image: Ảnh hiển thị trong đối tượng.
 - ImageLocation: Đường dẫn và tên tệp của ảnh hiển thị trong đối tượng.
 - SizeMode: Chế độ hiển thị ảnh trên đối tượng.
- Sự kiện thường dùng
 - Click(): Được gọi khi nhấn nút phải chuột trong khi con trỏ chuột ở trong đối tượng.

Điều khiển PictureBox

- Ví dụ xây dựng một form xem các tệp ảnh

```
private void btnOpen_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.InitialDirectory = "C:\\\\";
    openFileDialog.Filter = "Bitmap(*.bmp)|*.bmp|JPEG(*.jpg)|*.jpg";
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        txtFileName.Text = openFileDialog.FileName;
        pictureBox1.ImageLocation = txtFileName.Text;
    }
}
```

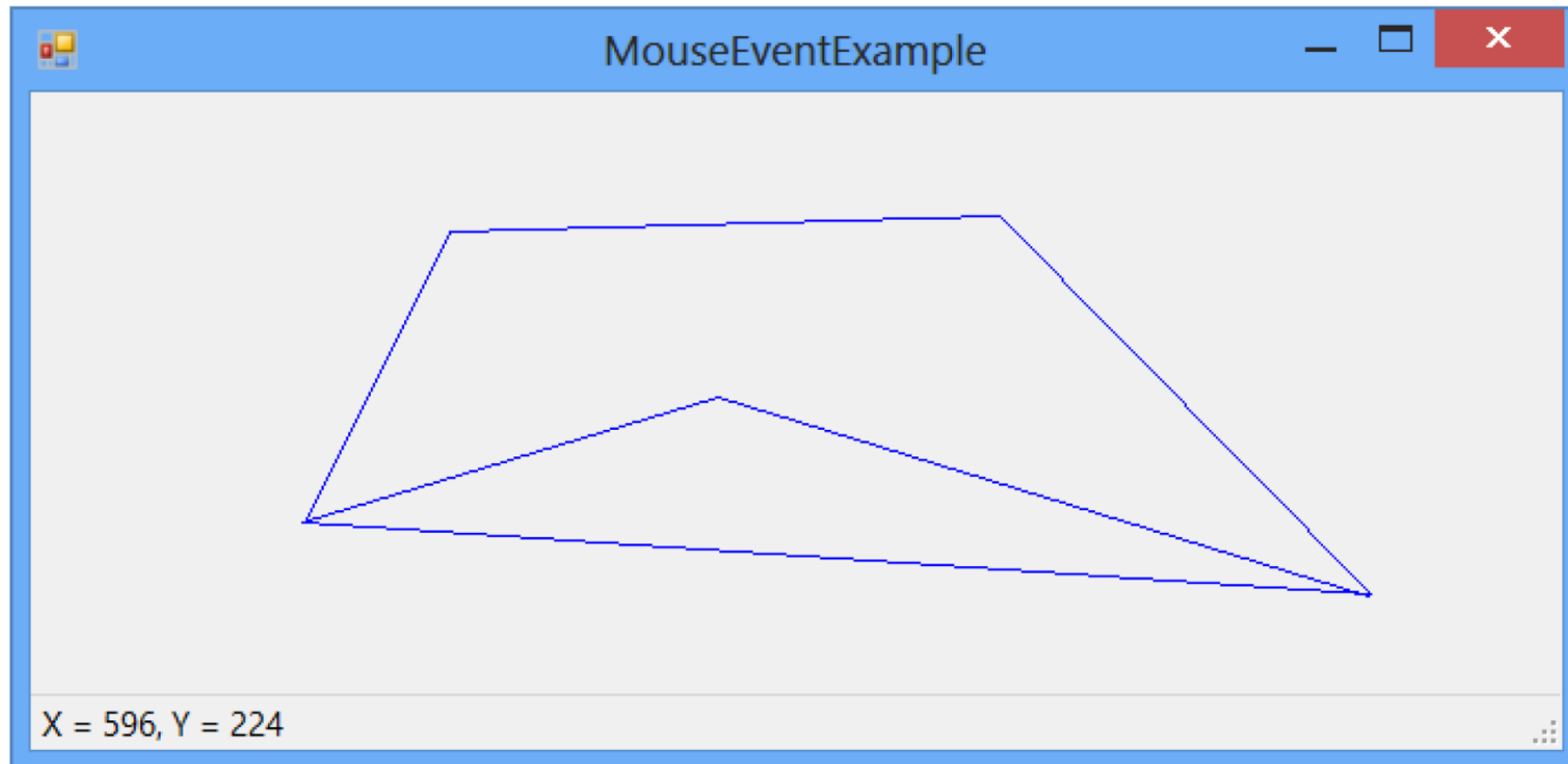


Xử lý các sự kiện của con chuột

- Các sự kiện chuột được sinh ra khi con trỏ chuột tương tác với một đối tượng.
- Một số sự kiện chuột thường dùng
 - ❑ `mouseenter()`: Được gọi khi con trỏ chuột đi vào vùng của đối tượng.
 - ❑ `mouseleave()`: Được gọi khi con trỏ chuột đi ra khỏi vùng của đối tượng.
 - ❑ `mousedown()`: Được gọi khi nút chuột được nhấn trong khi con trỏ chuột ở trong vùng của đối tượng.
 - ❑ `mousemove()`: Được gọi con trỏ chuột di chuyển trong vùng của đối tượng.
 - ❑ `mouseup()`: Được gọi khi nút chuột được thả trong khi con trỏ chuột ở trong vùng của đối tượng.

Xử lý các sự kiện của con chuột

- Ví dụ xây dựng một form để hiển thị tọa độ của con trỏ chuột khi di chuyển chuột và nối các điểm giữa các lần nháy chuột bằng các đoạn thẳng.



Xử lý các sự kiện của con chuột

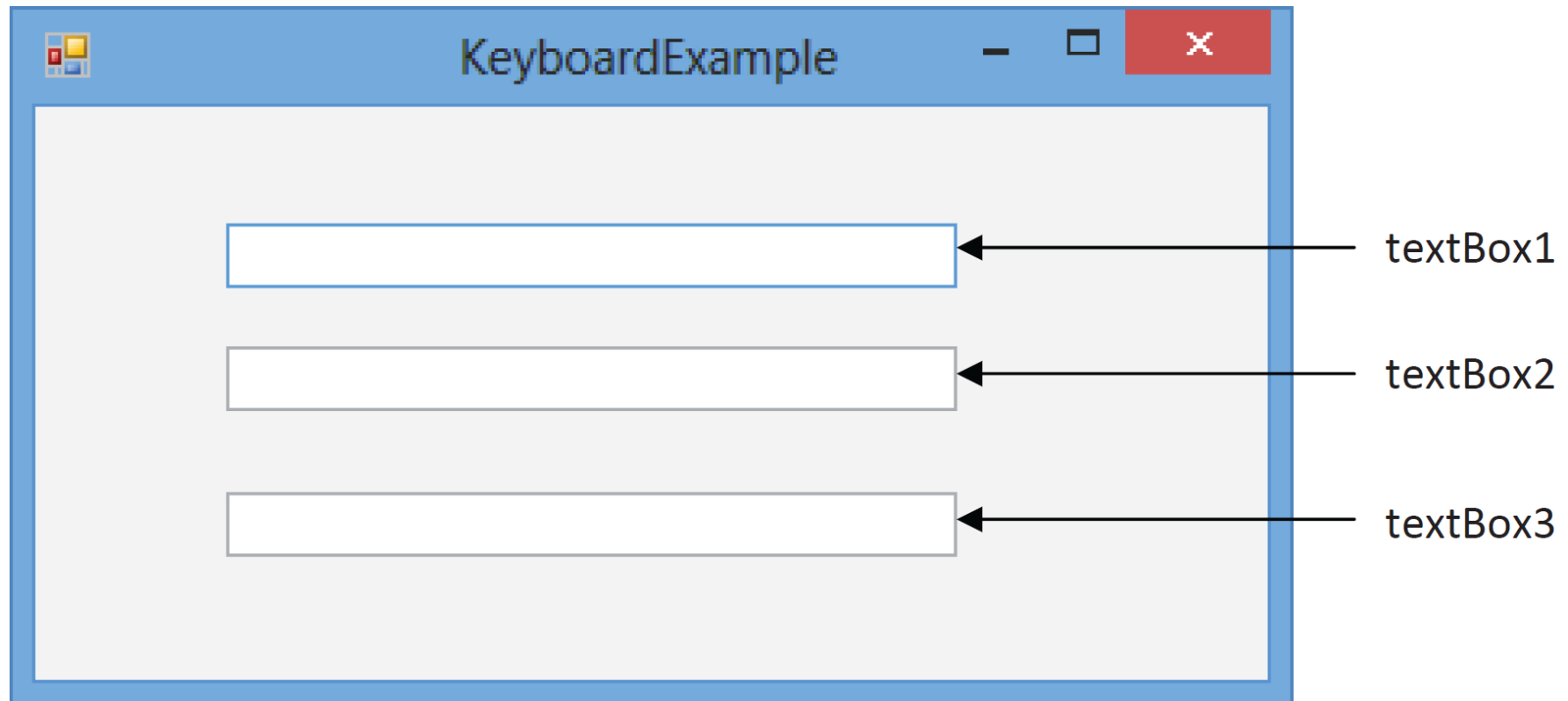
```
private Point lastPos;
private Graphics graphics;
private Pen pen;
private void Form1_Load(object sender, EventArgs e){
    lastPos.X = -1; lastPos.Y = -1;
    graphics = CreateGraphics();
    pen = new Pen(Color.Blue);
}
private void Form1_MouseMove(..., MouseEventArgs e){
    lbStatus.Text = "X =" + e.X.ToString()+"Y = " + e.Y.ToString();
}
private void Form1_MouseDown(..., MouseEventArgs e){
    if ((lastPos.X !=-1) && (lastPos.Y !=-1))
        graphics.DrawLine(pen, lastPos.X, lastPos.Y, e.X, e.Y);
    lastPos.X = e.X;
    lastPos.Y = e.Y;
}
```

Xử lý các sự kiện của bàn phím

- Các sự kiện của bàn phím được sinh ra khi các phím trên bàn phím được ấn.
- Ví dụ khi nhập dữ liệu vào một đối tượng lớp TextBox, sự kiện TextChanged() của đối tượng sẽ được gọi.
- Một số sự kiện chuột thường dùng
 - KeyDown(): Được gọi khi một phím được nhấn
 - KeyUp(): Được gọi khi một phím được nhả sau khi nhấn
 - KeyPress(): Được gọi khi một phím được nhấn
 - Sự khác nhau giữa sự kiện KeyDown và KeyPress() là tham số của sự kiện.

Xử lý các sự kiện của bàn phím

- Ví dụ xây dựng một ứng dụng cho phép gõ Enter, phím lên ↑, hoặc phím xuống ↓ để di chuyển con trỏ giữa các đối tượng lớp TextBox.



Xử lý các sự kiện của bàn phím

```
private void textBox1_KeyDown(object sender, KeyEventArgs e)
{
    if ((e.KeyValue == 13) || (e.KeyValue == 40))
        textBox2.Focus();
}
private void textBox2_KeyDown(object sender, KeyEventArgs e)
{
    if ((e.KeyValue == 13) || (e.KeyValue == 40))
        textBox3.Focus();
    if (e.KeyValue == 38)
        textBox1.Focus();
}
private void textBox3_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyValue == 38)
        textBox2.Focus();
}
```

Bài tập

1. Xây dựng ứng dụng tính diện tích và chu vi tam giác với giao diện như sau:

Tính diện tích và chu vi của tam giác ABC

Nhập dữ liệu...

Tọa độ điểm A (x1,y1):

Tọa độ điểm B (x2,y2):

Tọa độ điểm C (x3,y3):

Kết quả...

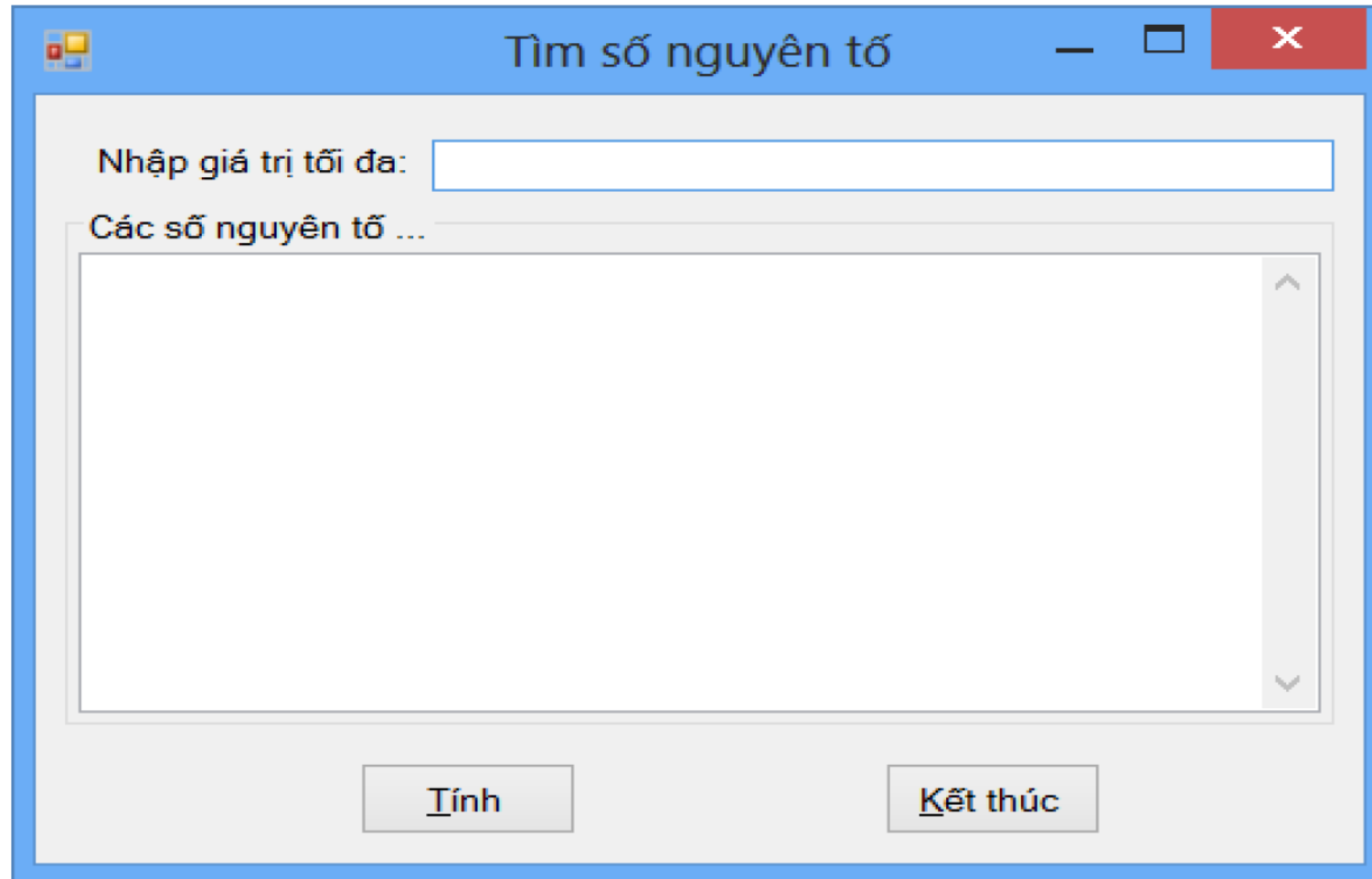
Chu vi tam giác ABC:

Diện tích tam giá ABC:

Tính toán...

Bài tập

2. Xây dựng ứng dụng tìm các số nguyên tố từ $1...n$ với giao diện được thiết kế như sau:



The image shows a Windows application window titled "Tìm số nguyên tố" (Find prime numbers). The window has a blue title bar with standard minimize, maximize, and close buttons. The main area contains a text input field labeled "Nhập giá trị tối đa:" (Enter maximum value:). Below this is a list box labeled "Các số nguyên tố ..." (Prime numbers ...). At the bottom of the window, there are two buttons: "Tính" (Calculate) and "Kết thúc" (End).

Bài tập

3. Xây dựng ứng dụng giải hệ phương trình bậc nhất 2 ẩn với giao diện được thiết kế như sau:

Giải hệ phương trình bậc nhất

Nhập các hệ số...

a1: b1: c1:

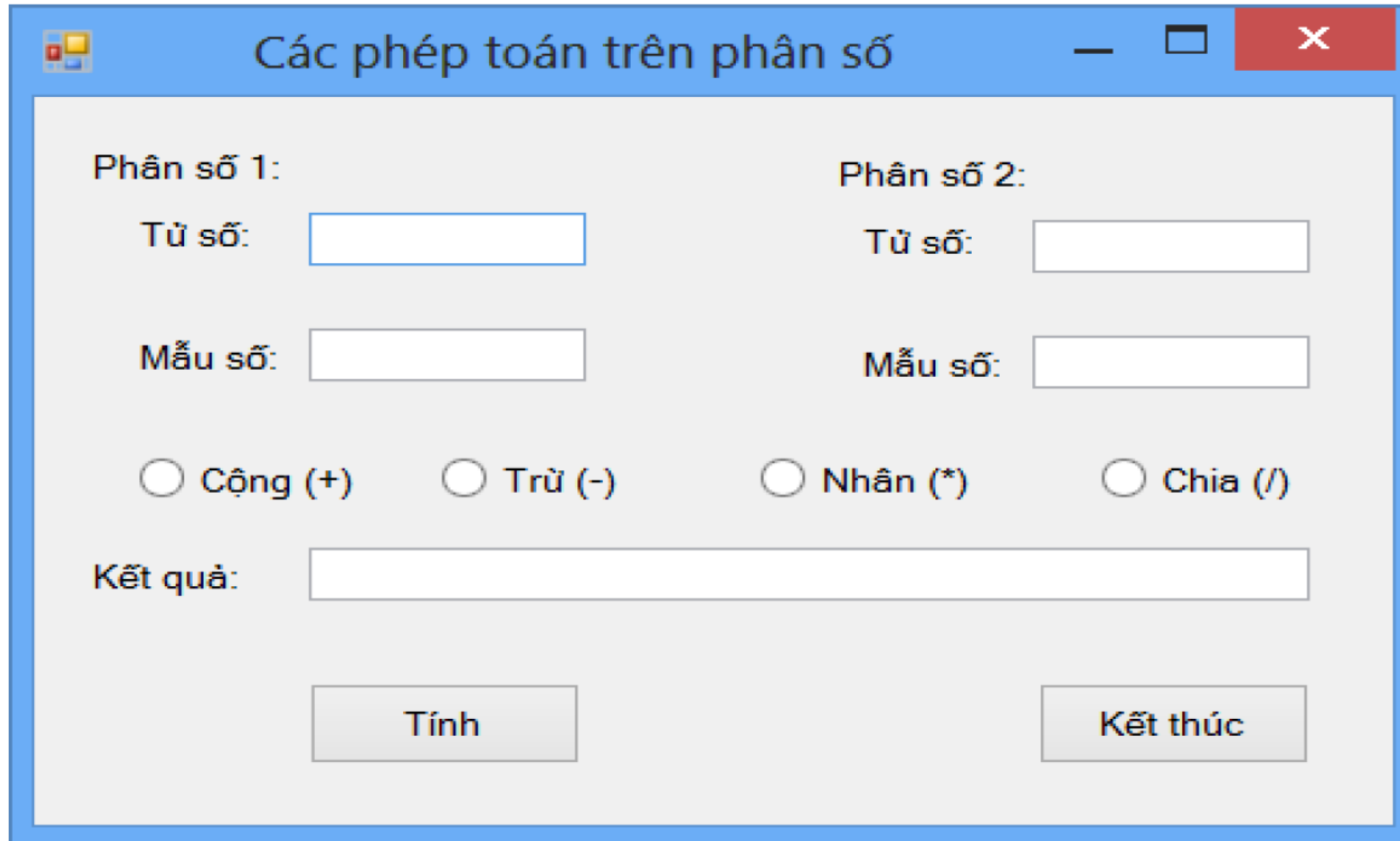
a2: b2: c2:

Kết quả...

Giải Kết thúc

Bài tập

4. Xây dựng ứng dụng thực hiện các phép toán $+$, $-$, $*$ và $/$ của 2 phân số với giao diện được thiết kế như sau:

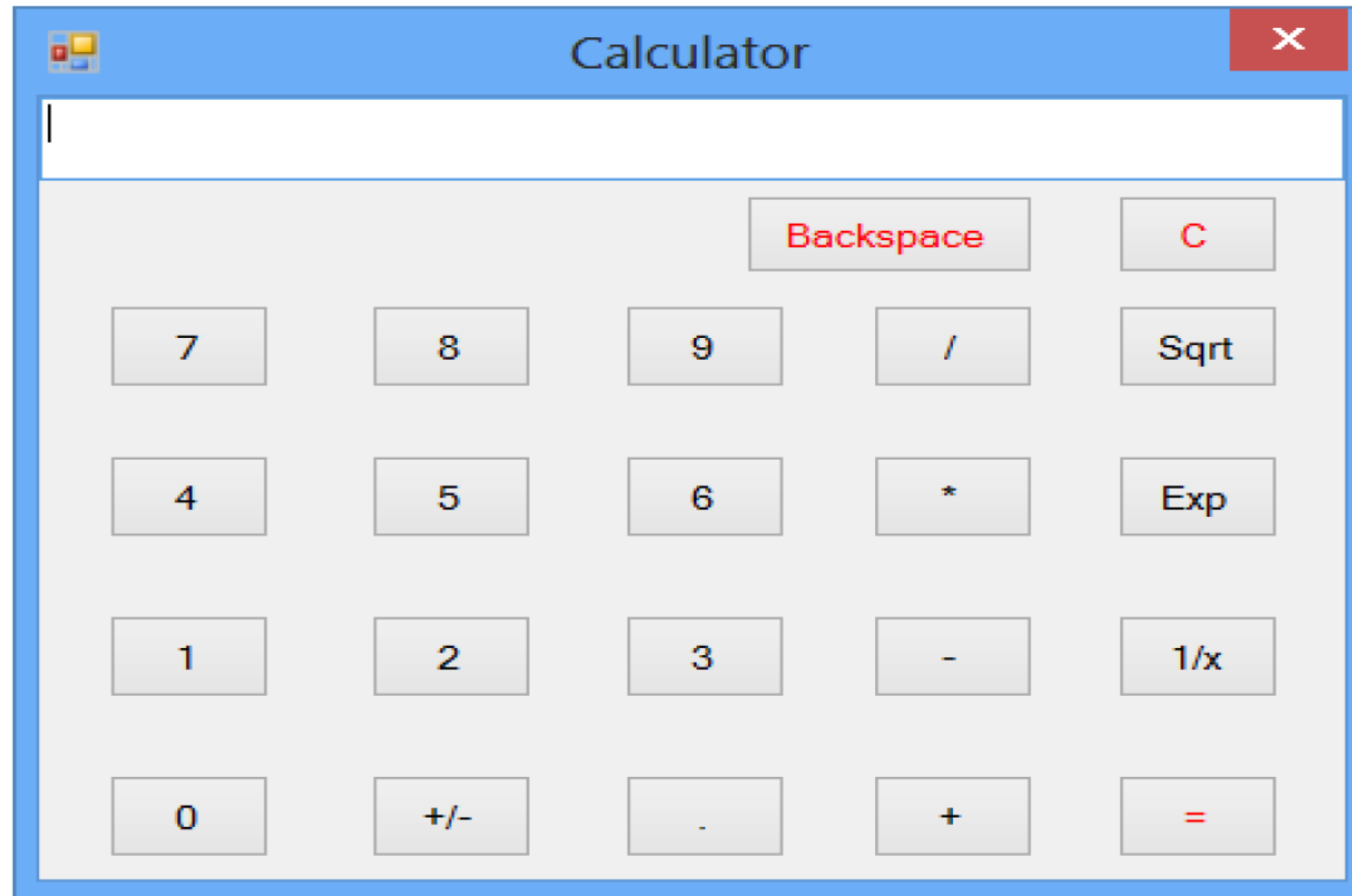


The application window is titled "Các phép toán trên phân số". It contains the following elements:

- Phân số 1:**
 - Tử số:
 - Mẫu số:
- Phân số 2:**
 - Tử số:
 - Mẫu số:
- Operations:** Four radio buttons are displayed: ☐ Cộng (+), ☐ Trừ (-), ☐ Nhân (*), and ☐ Chia (/).
- Kết quả:** A single-line text input field for the result.
- Buttons:** Two buttons at the bottom: "Tính" (Calculate) and "Kết thúc" (Exit).

Bài tập

5. Xây dựng ứng dụng Calculator với giao diện như sau:



Kết thúc



HỎI & TRẢ LỜI