

Công nghệ .NET

Ứng dụng Windows Form

TS. Phạm Ngọc Hưng – Khoa Công nghệ Thông tin



Giới thiệu thiệu Windows Form trong # .Net



Sử dụng Form và các điều khiển trên form



Bài tập luyện tập

Giới thiệu Windows Form trong # .Net

Tài liệu tham khảo: Chương 12, *Giáo trình Cơ sở kỹ thuật lập trình với C#*, tác giả:
TS. Phạm Ngọc Hưng (Chủ biên), Phạm Quốc Hùng – Ngô Thanh Huyền

- Các ứng dụng trong thực tế đa phần được lập trình bằng giao diện cửa sổ. Đây là giao diện đồ họa đẹp không phải được thực hiện tương tác ở dạng dòng lệnh như ứng dụng dạng *Console*.
- Mỗi ứng dụng sẽ có ít nhất một cửa sổ. Trên đó có thể có các hộp nhập dữ liệu (*TextBox*), các nút lệnh (*Button*) để người dùng ra lệnh thực hiện tác vụ nào đó. Dưới đây là một ví dụ chương trình giải phương trình bậc 2 (GPTB2) được lập trình bằng giao diện Windows Form.

Giới thiệu Windows Form

Giải phương trình bậc 2

Hệ số a: 4

Hệ số b: 7

Hệ số c: 1

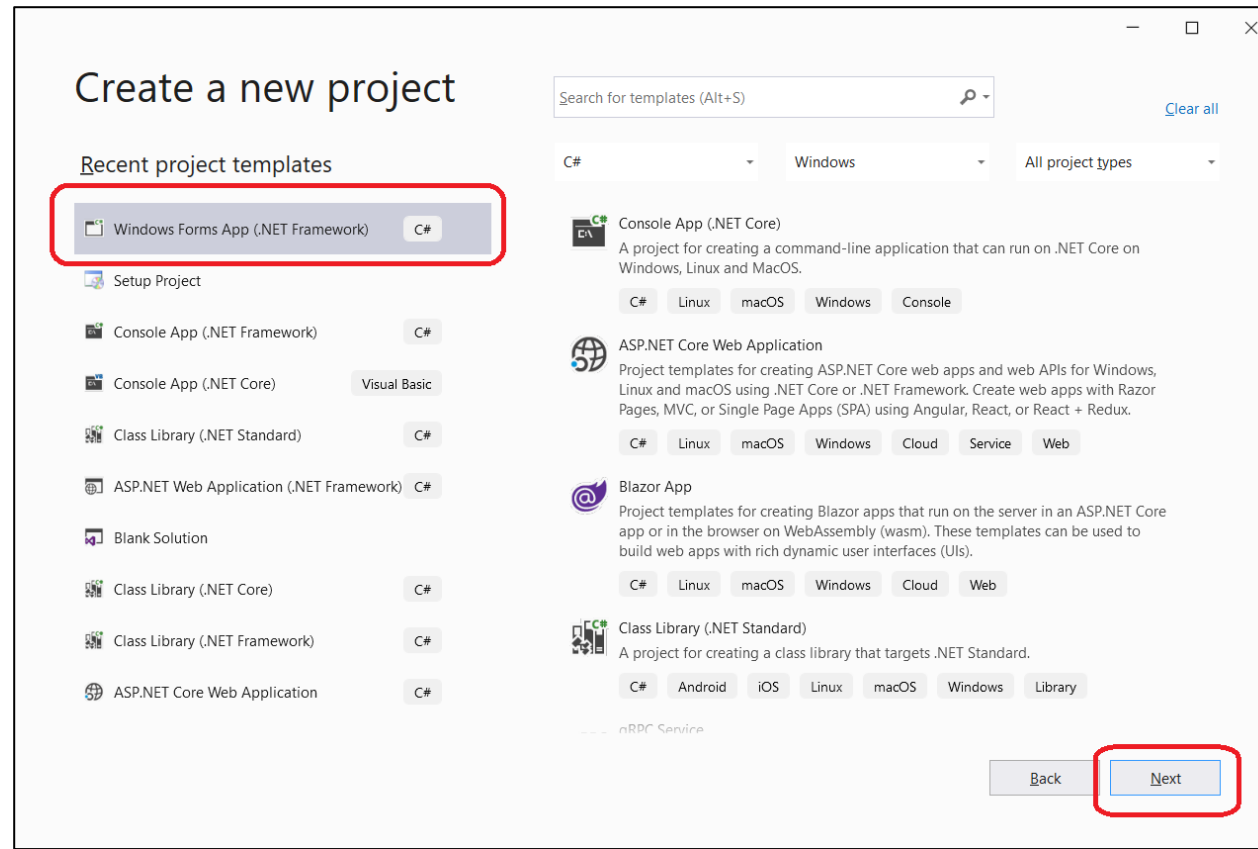
Giải phương trình

Kết quả:

Phương trình có 2 nghiệm phân biệt:
x1=-1.59307
x2=-0.1569297

Các bước để tạo ứng dụng Windows Form

- Để tạo một ứng dụng giao diện *Windows Form*, ngay sau khi khởi động *Visual Studio*, tại giao diện tạo dự án (*project*) mới, chọn kiểu ứng dụng là *Windows Form* như minh họa dưới đây.



- Trên cửa sổ tạo dự án mới: chọn mục “*Windows Forms App (.NET Framework)*” sau đó nhấn nút “*Next*”. Lưu ý, phiên bản *Visual Studio* khác nhau thì tùy chọn này có thể được hiện lên có đôi chút khác biệt. Tuy nhiên từ khóa để nhận biết đúng chính là “*Windows Forms*”.
- Ở cửa sổ tiếp theo, nhập vào tên dự án muốn tạo ở ô nhập liệu “*Project name*”. Ví dụ, nhập *GPTB2* cho dự án “*Giải phương trình bậc 2*” như ở hình sau:

Các bước để tạo ứng dụng Windows Form

Configure your new project

Windows Forms App (.NET Framework) C# Windows Desktop

Project name

GPTB2

Location

D:\C# App\

Solution name ⓘ

GPTB2

☐ Place solution and project in the same directory

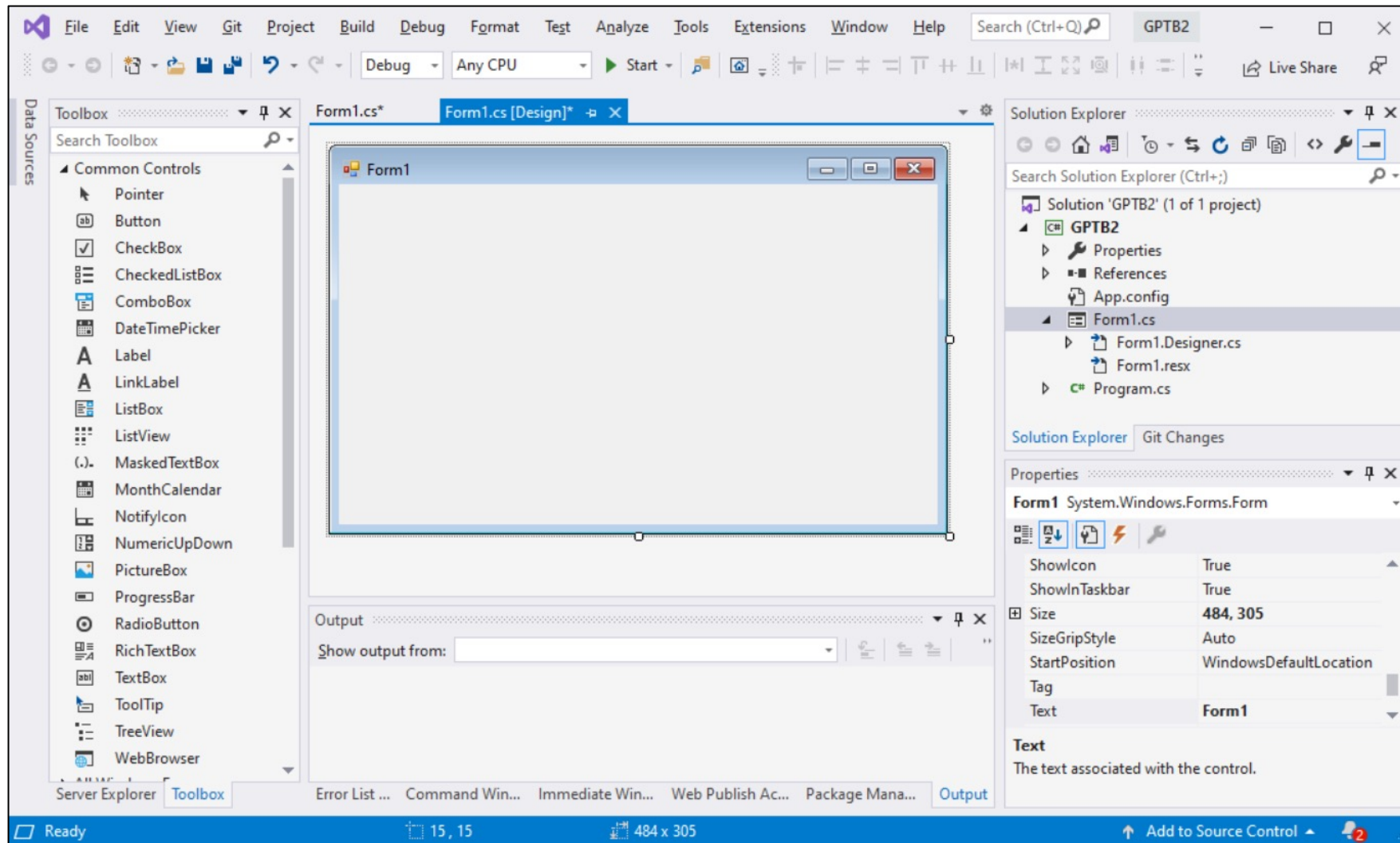
Framework

.NET Framework 4.7.2

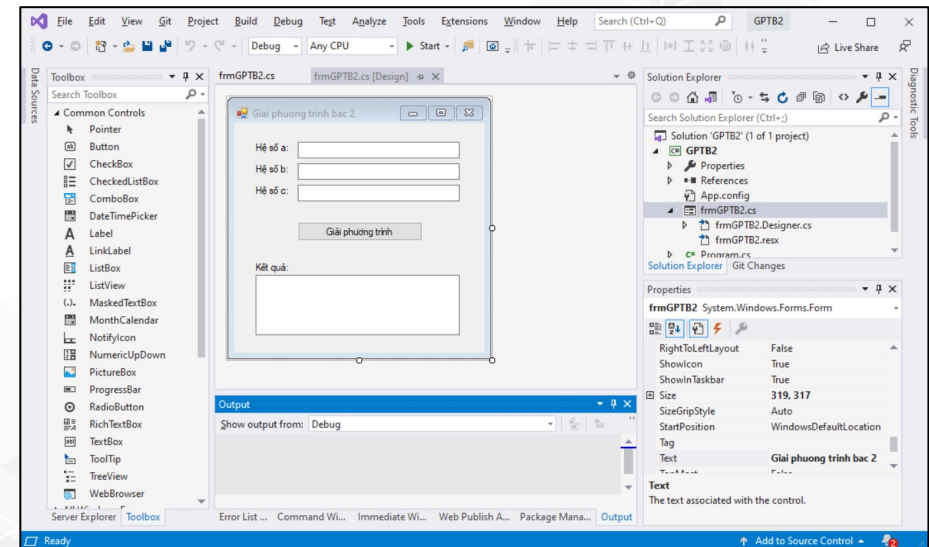
Back Create

- Chú ý, chọn nơi lưu trữ dự án ở mục “*Location*” và ghi nhớ để có thể tìm và mở lại dự án sau này khi cần.
- Tiếp theo sẽ xuất hiện giao diện làm việc chính, ở đó có sẵn một cửa sổ (*Windows Form*) với tên mặc định là *Form1* như hình sau.

Các bước để tạo ứng dụng Windows Form

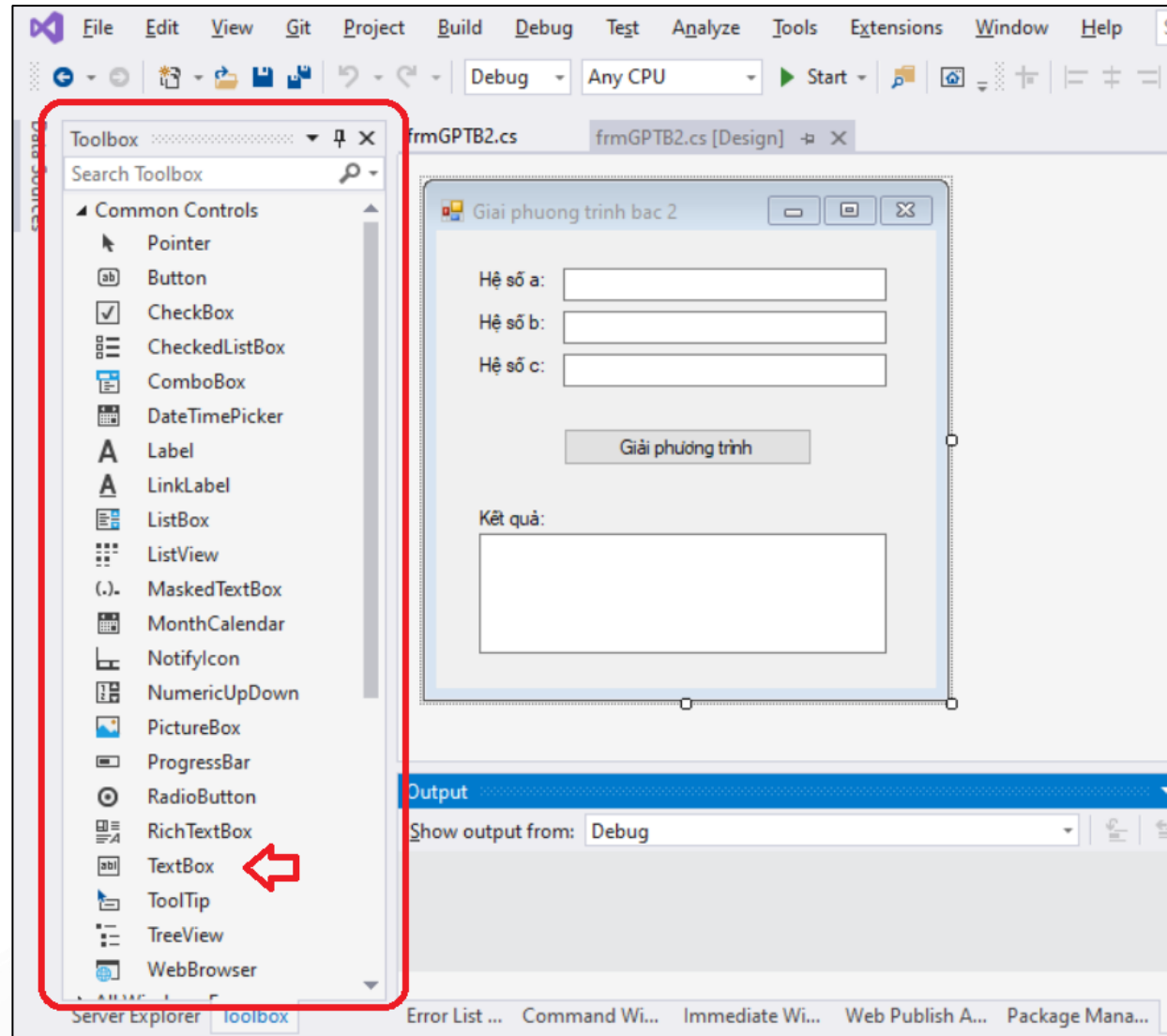


- Nhiệm vụ tiếp theo là thiết kế chi tiết các hộp nhập liệu, nút lệnh để thực hiện các nhiệm vụ của bài toán giải phương trình bậc 2.
- Để thực thi (chạy) chương trình, người dùng chỉ cần nhấn phím chức năng **F5** hoặc nhấn vào nút “**Start**” có biểu tượng hình tam giác màu xanh lá cây trên thanh công cụ.



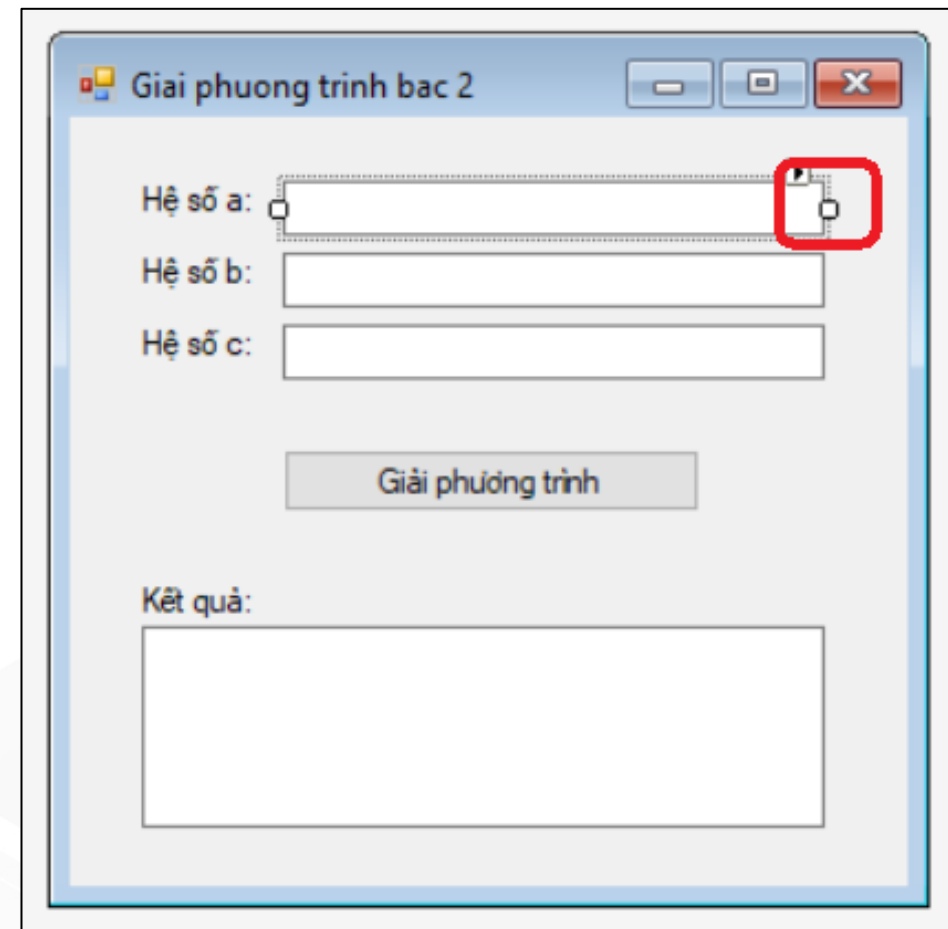
- Việc nhập dữ liệu trên giao diện ứng dụng được viết theo kiểu *Windows Form* có khác biệt đáng kể so với cách nhập liệu thường thấy ở giao diện *Console*.
- Có nhiều cách để nhập liệu khác nhau trên Windows Form, tuy nhiên cách thường dùng là sử dụng hộp nhập liệu văn bản (TextBox) đã được chuẩn bị sẵn trong bộ công cụ lập trình Visual Studio.
- Hình sau cho thấy thành công cụ này:

Nhập/xuất trên Windows Form

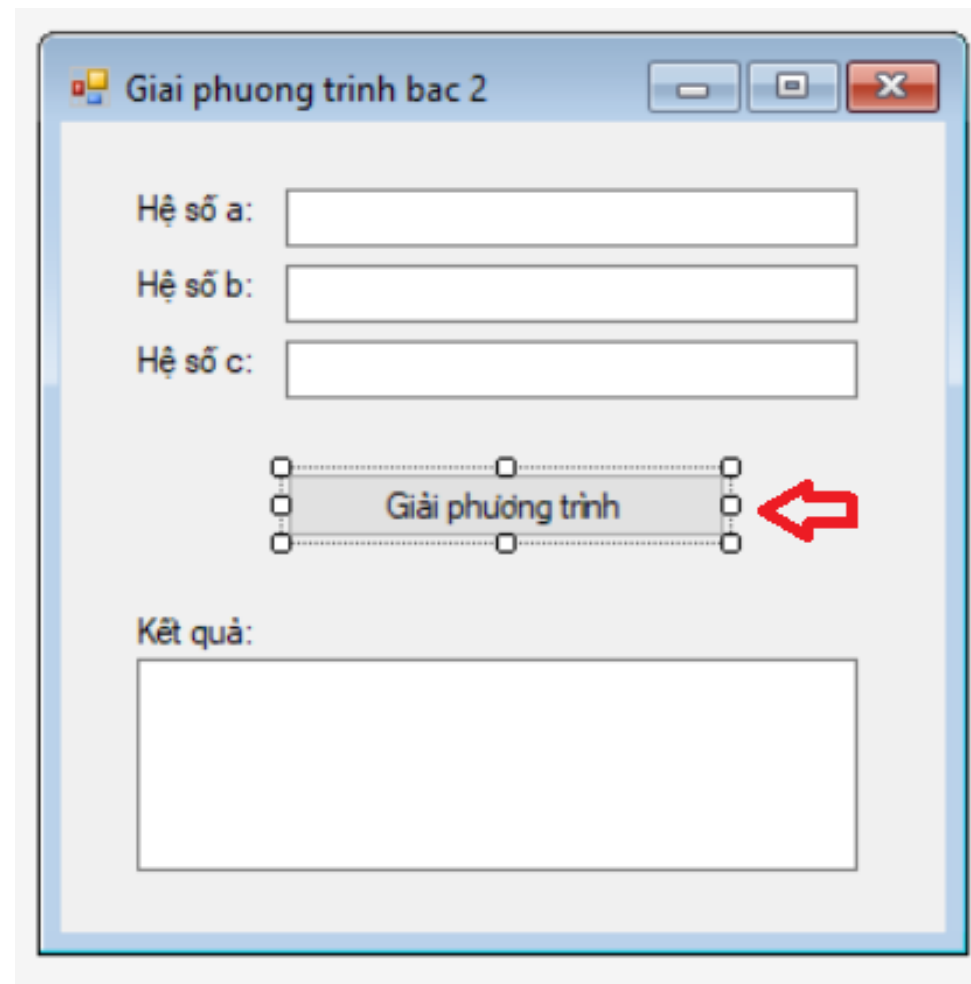


- Để có được hộp nhập liệu cho ứng dụng (nhập các hệ số a , b , c như ở GPTB2), hãy nhấn chuột và kéo biểu tượng của hộp nhập liệu `TextBox` trên thanh công cụ thả vào vùng trống bất kỳ trên cửa sổ (Form) của ứng dụng.
- Có thể lặp lại nhiều lần các thao tác này để tạo ra nhiều các hộp nhập dữ liệu `TextBox` trên cửa sổ của ứng dụng. Như trong GPTB2, sẽ cần lặp lại thao tác đó 4 lần để tạo ra 3 hộp nhập hệ số a , b , c tương ứng và một hộp hiển thị kết quả giải phương trình.

- Có thể dùng chuột “gấp/drag” (đưa con trỏ đến hộp *TextBox*, nhấn giữ phím chuột trái và di chuyển chuột”) và di chuyển, sắp xếp lại vị trí các hộp nhập liệu trên *Form* theo ý muốn.
- Để thay đổi kích thước của hộp nhập liệu, chúng ta có thể nhấn chuột chọn đối tượng (*TextBox*), khi đó sẽ xuất hiện hình vuông nhỏ ở 2 cạnh bên của đối tượng (gọi là điểm neo). Lúc này chỉ cần drag chấm hình vuông đó đi khỏi vị trí là sẽ làm thay đổi kích thước của đối tượng.



- Với nút lệnh (như nút “Giải phương trình bậc 2”), khi nhấn chọn, sẽ có 8 điểm neo xuất hiện ở 4 góc và 4 cạnh. Ta có thể kéo bất cứ điểm neo nào để làm thay đổi kích thước, thậm chí là vị trí của nút lệnh.



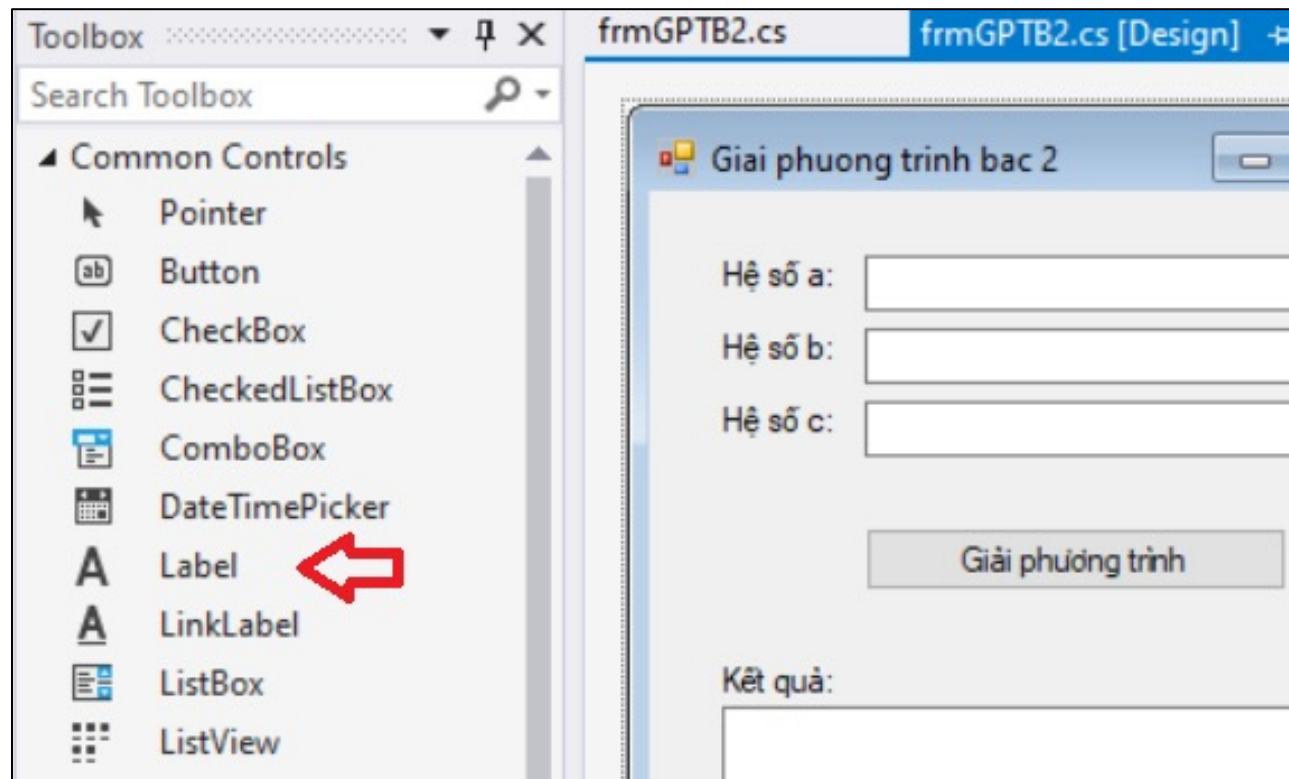
- **Kết xuất dữ liệu trên Windows Form**

- Việc kết xuất dữ liệu trên Windows Form cũng có nhiều cách khác nhau có thể áp dụng. Thông thường các dữ liệu (hiển thị dưới dạng văn bản) sẽ được đưa ra màn hình thông qua đối tượng TextBox, Label hoặc một hộp thoại (xuất hiện cửa sổ mới để hiển thị thông báo khi sử dụng MessageBox).
- Như trong ví dụ GPTB2, để hiển thị kết quả phương trình vô nghiệm, dòng thông báo được gán cho thuộc tính Text của đối tượng TextBox có tên là txtKetQua .

```
2  if (delta<0)
3  {
4      txtKetQua.Text = "Phương trình vô nghiệm";
5  }
```

- Ngoài *TextBox*, *Label* là đối tượng thường được dùng để hiển thị dữ liệu dạng văn bản.
- Như trong ví dụ GPTB2, *Label* được dùng để đưa ra các dòng chữ cố định “*He so a:*”, “*He so b:*”, ... bên cạnh các hộp nhập liệu tương ứng.
- Điểm khác biệt quan trọng giữa *Label* và *TextBox* là *Label* chỉ hiển thị và không cho phép người dùng sửa đổi nội dung hiển thị như đối với *TextBox*.
- *Label* chỉ dùng để hiển thị dữ liệu ra màn hình (*Form*) chứ không thể dụng cho nhập liệu.

- Để đưa *Label* vào *Form*, hãy thực hiện thao tác tương tự như khi đưa *TextBox* từ hộp điều khiển vào *Form*. Điểm khác ở đây là thay vì kéo đối tượng *TextBox* trên hộp điều khiển sẽ là kéo đối tượng *Label*.

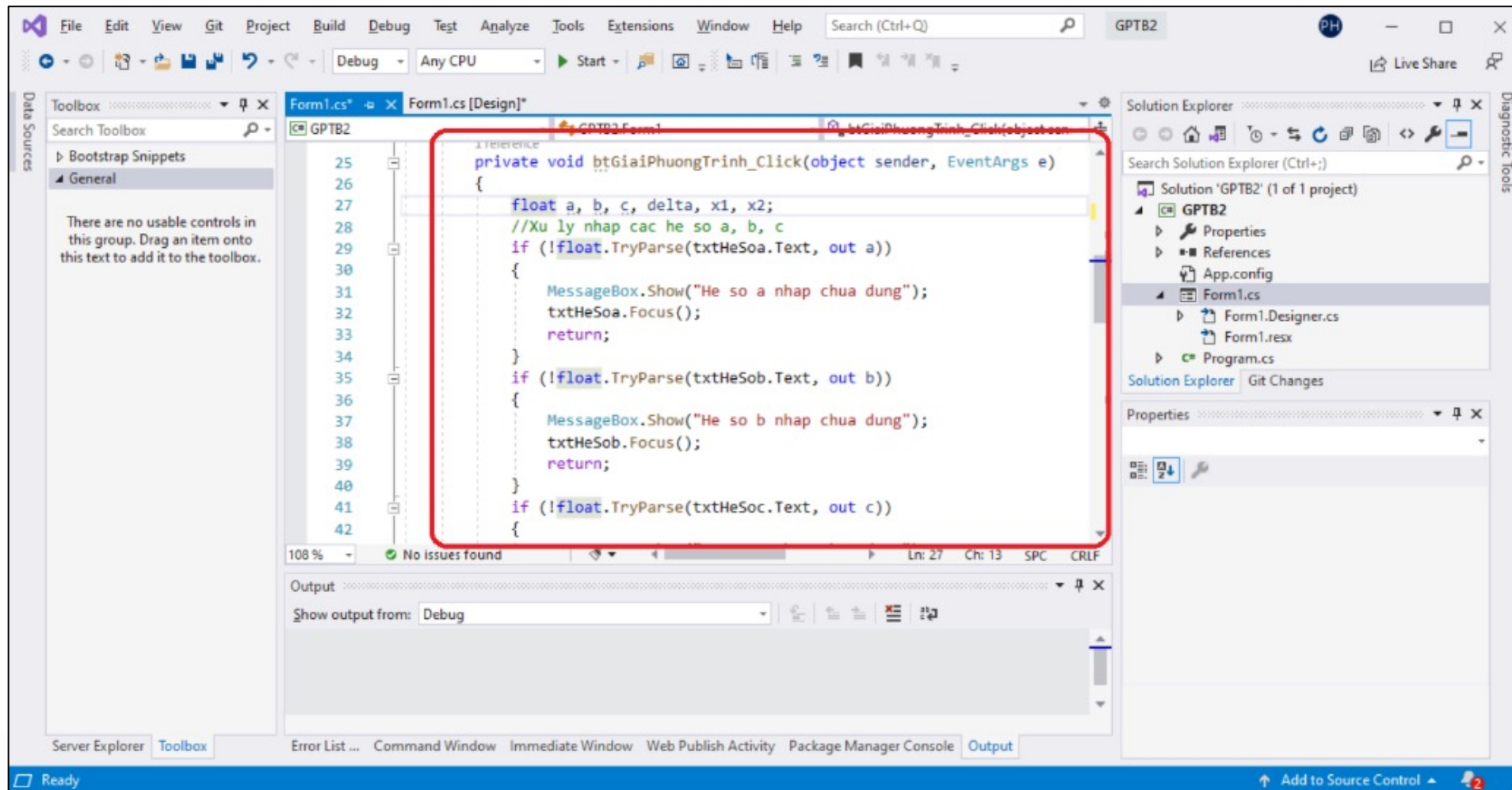


- *MessageBox* cũng là một công cụ thường được dùng để hiển thị ra màn hình. Khi dùng *MessageBox*, một cửa sổ mới xuất hiện kèm theo thông báo muốn được đưa ra. Người dùng sẽ phải nhấn nút điều khiển trên cửa sổ đó (thường là nút “OK”) để đóng cửa sổ thông báo để quay trở lại ứng dụng. Dưới đây là ví dụ sử dụng *MessageBox* đưa ra thông báo khi người dùng nhập hệ số $a=0$ (tương ứng với tình huống không phải là một phương trình bậc 2):

```
3  if (a==0)
4  {
5      MessageBox.Show("Day khong phai phuong trinh bac 2");
6      txtHeSoa.Focus();
7      return;
8  }
```

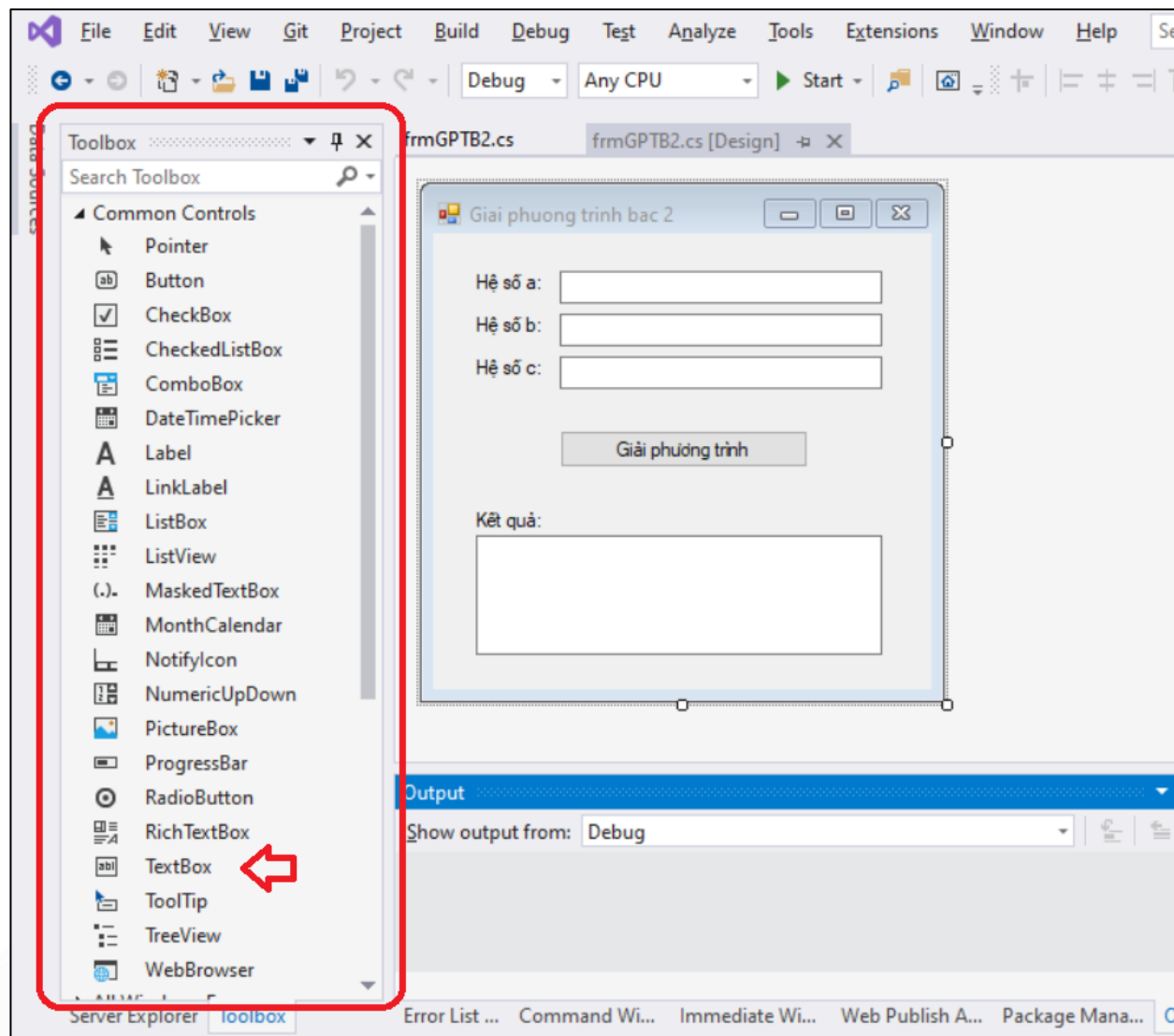
- **Sử dụng Button trên Windows Form**
- *Button* (nút lệnh) là điều khiển thường được dùng để tạo giao diện tương tác người dùng trên ứng dụng Windows Form.
- Đối tượng này hiển thị một thông tin (văn bản) cho biết chức năng của nó.
- Khi người dùng muốn thực hiện chức năng đó thì nhấn chuột vào nút lệnh.
- Để viết chương trình khi người dùng nhấn nút xử lý công việc nào đó hãy nháy kép chuột vào nút lệnh, khi đó xuất hiện giao diện để viết lệnh ứng với sự kiện click của nút lệnh.

- Giao diện viết lệnh



Sử dụng form và các điều khiển trên form

Sử dụng form và các điều khiển trên form



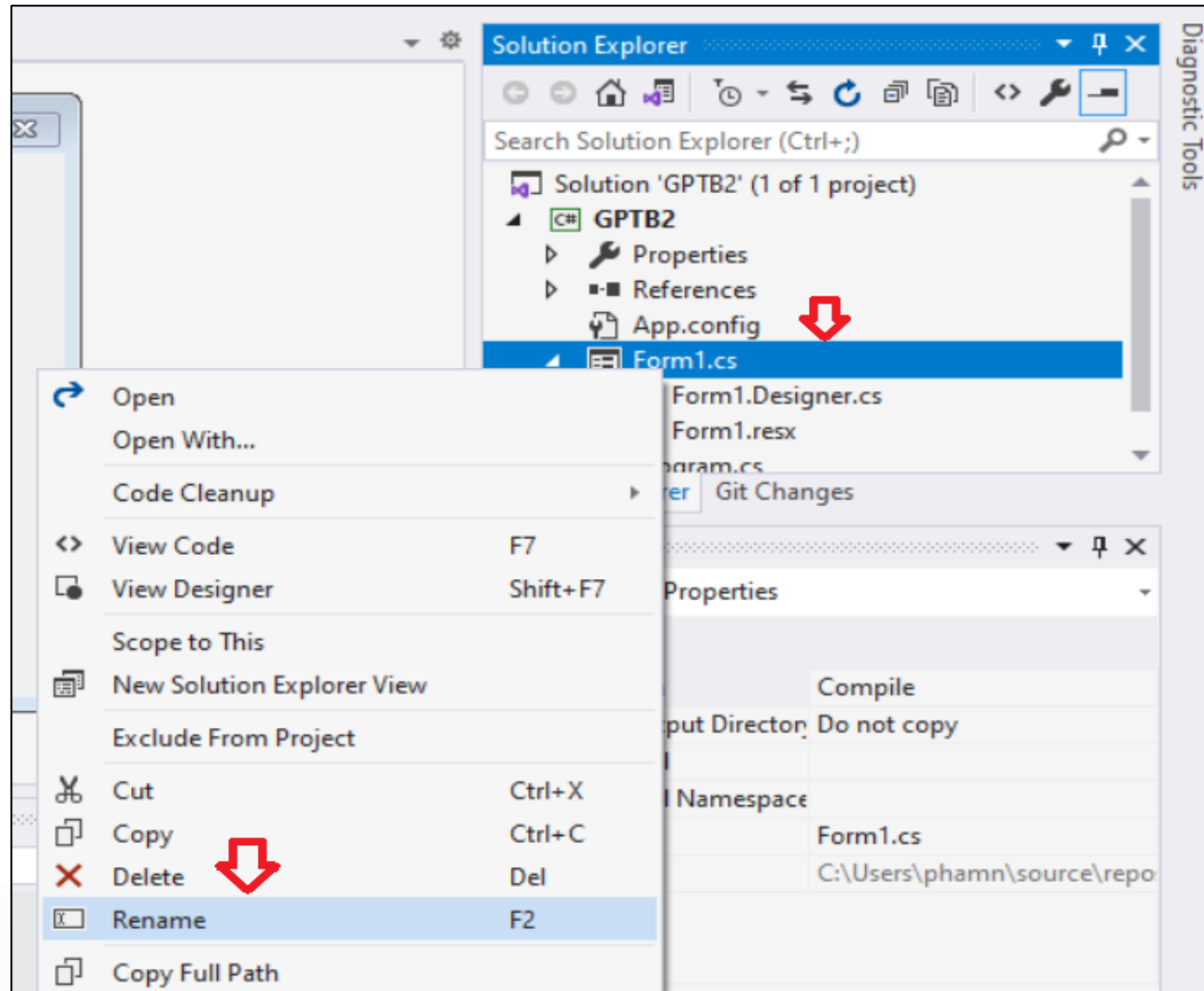
- **Form:** tạo cửa sổ tương tác với người dùng. Trên form có thể sắp đặt các điều khiển với các chức năng khác nhau tùy yêu cầu bài toán.
- **Textbox:** hiển thị và nhập dữ liệu dạng văn bản
- **Label:** hiển thị văn bản
- **Button:** tạo tương tác người dùng bằng click chuột, nhấn phím để ra lệnh
- **Checkbox:** nhập dữ liệu dưới dạng đánh dấu mục chọn (đa lựa chọn)
- **RadioButton:** nhập dữ liệu dưới dạng đánh dấu mục chọn (đơn lựa chọn)
- **Listbox, Combobox:** nhập liệu dưới dạng chọn trên danh sách có sẵn

Bài tập luyện tập

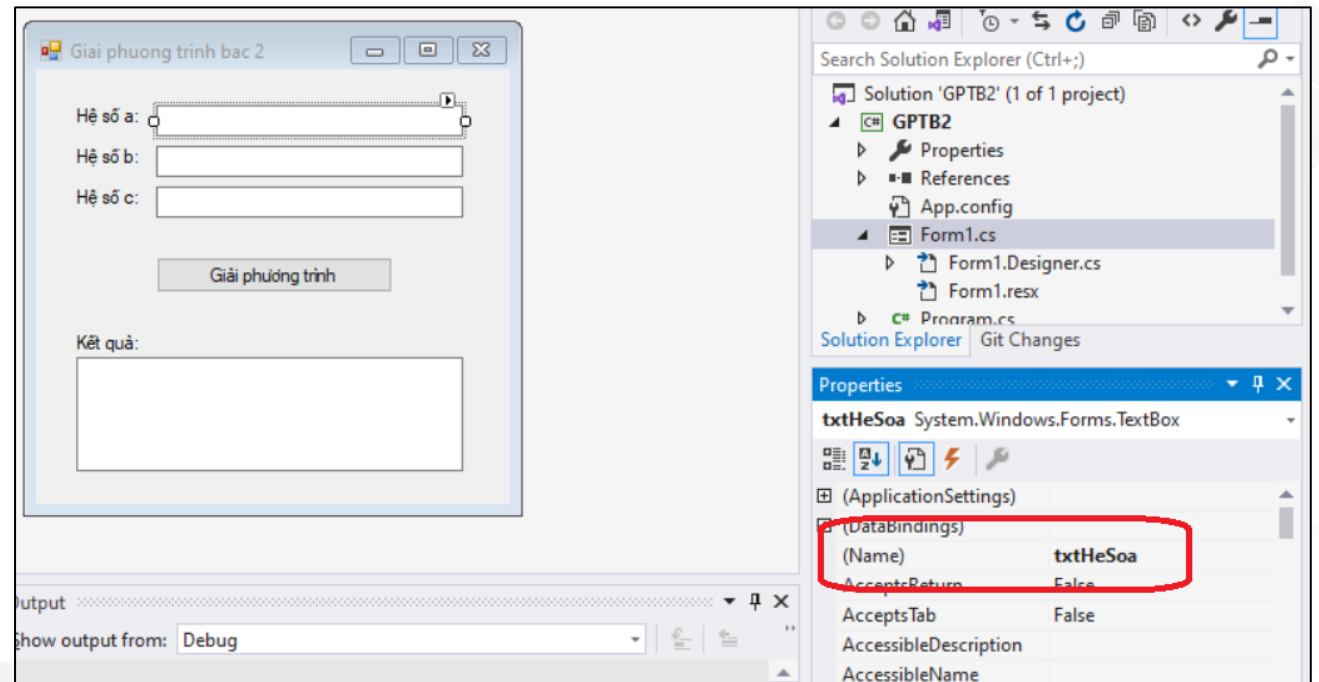
- **Lập trình chương trình giải phương trình bậc 2**

- Thiết kế giao diện và lập trình chương trình giải phương trình bậc 2 với giao diện cửa sổ (windows form) trên C#.
- Ứng dụng có một cửa sổ làm việc, trên cửa sổ có các label và các textbox để thể hiện thông tin và cho phép người dùng nhập các hệ số của phương trình.
- Nút lệnh (command button) với nội dung "*Giải phương trình*". Khi người dùng nhấn nút "Giải phương trình" tiến hành tính delta, xét nghiệm của phương trình và thông báo trên các hộp text tương ứng kết quả.

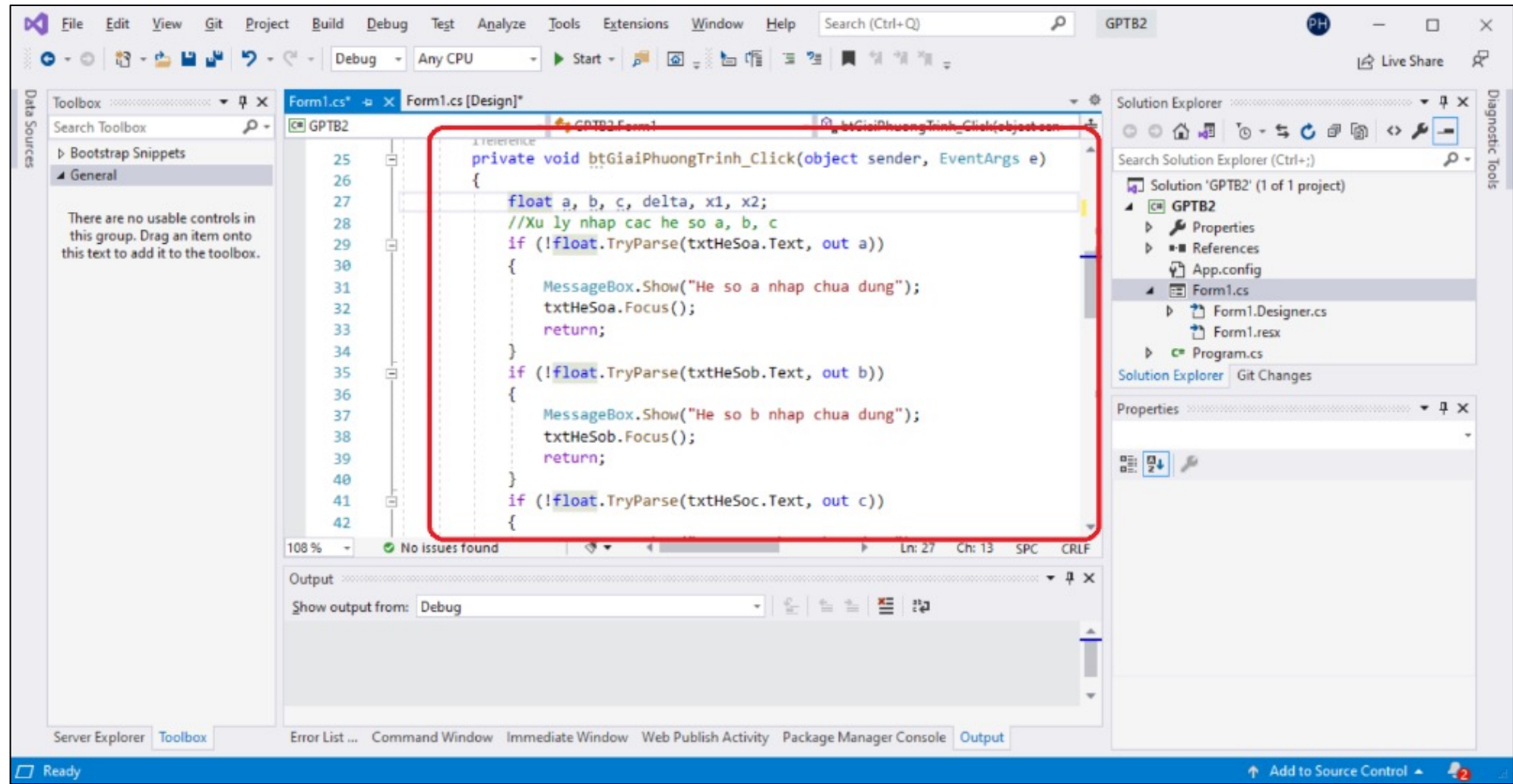
- ***Hướng dẫn Lập trình chương trình giải phương trình bậc 2:***
- Tạo dự án mới, lựa chọn kiểu dự án là “*Windows Forms App (.NET Framework)*”.
- Nhập tên cho dự án là GPTB2 đồng thời chọn nơi lưu trữ dự án này và kết thúc bước tạo dự án mới.
- Nhấn chuột phải vào tên *Form1* ở cửa sổ “*Solution Explorer*”, chọn Rename trên menu xuất hiện, rồi nhập tên mới cho *Form* là “*frmGPTB2*”, xác nhận để hoàn thành thao tác này.



- Tiếp theo tiến hành gấp, thả các điều khiển *TextBox*, *Label* và *Button* vào *Form*. Điều chỉnh kích thước, sắp xếp vị trí của các điều khiển sao cho có được giao diện ưng ý.
- Nhấn chuột chọn từng điều khiển (*TextBox*) và nút lệnh, sửa lại thuộc tính *Name* trên cửa sổ *Properties* để đặt tên cho chúng.



- Lặp lại các thao tác đổi tên để thiết lập tên cho các hộp *TextBox* lần lượt là *txtHeSoa*, *txtHeSob*, *txtHeSoc* và hộp *TextBox* cuối cùng là *txtKetQua*. Nút lệnh là *btGiaiPhuongTrinh*.
- Đến đây, việc thiết kế giao diện cho chương trình GPTB2 đã hoàn tất. Công việc còn lại là viết lệnh để xử lý việc nhập dữ liệu (các hệ số a , b , c), tính delta và các nghiệm x_1 , x_2 rồi đưa thông báo kết quả ra màn hình.
- Việc giải phương trình sẽ được thực hiện khi người dùng nhấn nút lệnh “*Giải phương trình*”, do vậy các lệnh cần được viết ứng với thao tác đó của người dùng. Hãy nhấp kép chuột lên nút lệnh để chuyển sang giao diện lập trình, viết mã lệnh xử lý giải phương trình bậc 2. Chương được viết ở đây sẽ được kích hoạt khi người dùng nhấn nút “Giải phương trình”. Giao diện lập trình sẽ xuất hiện thay cho phần thiết kế giao diện như ở hình sau:



- Chương trình:

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
namespace GPTB2  
{
```

- Chương trình:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
    private void frmGPTB2_Load(object sender,
EventArgs e)
    {
        //Các lệnh thực hiện khi form được load
    }
}
```

- Chương trình:

```
private void btGiaiPhuongTrinh_Click(object
sender, EventArgs e)
{
    float a, b, c, delta, x1, x2;
    //Xu ly nhap cac he so a, b, c
    if (!float.TryParse(txtHeSoa.Text, out a))
    {
        MessageBox.Show("He so a nhap chua dung");
        txtHeSoa.Focus();
        return;
    }
    if (!float.TryParse(txtHeSob.Text, out b))
    {
        MessageBox.Show("He so b nhap chua dung");
        txtHeSob.Focus();
        return;
    }
}
```

- Chương trình:

```
if (!float.TryParse(txtHeSoc.Text, out c))  
{  
    MessageBox.Show("He so c nhap chua dung");  
    txtHeSoc.Focus();  
    return;  
}
```

```
//Xet dieu kien ton tai phuong trinh bac 2  
if (a == 0)  
{  
    MessageBox.Show("Day khong phai phuong  
trinh bac 2");  
    txtHeSoa.Focus();  
    return;  
}
```

- Chương trình:

```
//Tính delta
delta = b * b - 4 * a * c;
//Xet cac nghiem cua phuong trinhf
if (delta < 0)
{
    txtKetQua.Text = "Phuong trinh vo nghiem";
}
else if (delta == 0)
{
    x1 = x2 = (-b) / (2 * a);
    txtKetQua.Text = "Phuong trinh co nghiem
    kep x1=x2= " + x1.ToString();
}
```

- Chương trình:

```
else
{
    x1 = (float)((-b) - Math.Sqrt(delta)) / (2
* a);
    x2 = (float)((-b) + Math.Sqrt(delta)) / (2
* a);
    txtKetQua.Text = "Phuong trinh co 2 nghiem
phan biet: \r\nx1=" + x1.ToString() +
"\r\nx2=" + x2.ToString();
}
}
}
}
```




Thank you !