

CHƯƠNG 9: ĐIỀU KHIỂN NGƯỜI DÙNG TỰ TẠO

9.1. User Control

User Control là điều khiển do lập trình viên tạo ra theo nhu cầu sử dụng riêng. Trong nhiều trường hợp, lập trình viên đã thiết kế xong một điều khiển hay một giao diện chương trình gồm nhiều điều khiển, và nếu có nhu cầu sử dụng nhiều lần hoặc mang sang một dự án khác sử dụng thì lập trình viên sẽ chuyển những điều khiển hay nhóm điều khiển đó về dạng User Control.

Một cách cụ thể hơn User Control là điều khiển được tạo ra trên cơ sở các điều khiển mà Windows Form hỗ trợ sẵn như: TextBox, Label, Button, ...

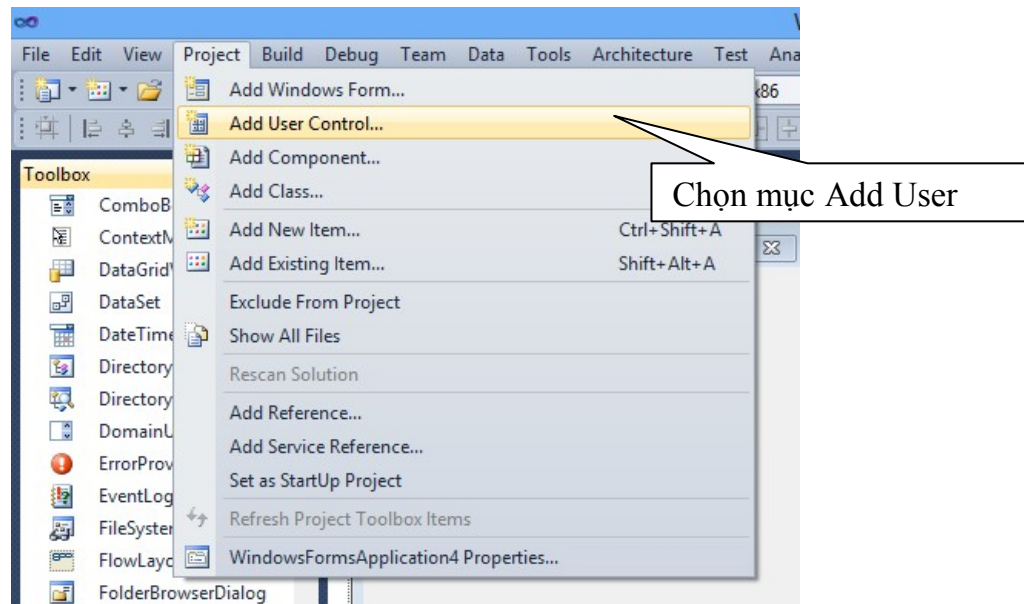
Việc sử dụng User Control như vậy sẽ giúp tiết kiệm chi phí thời gian khi thiết kế giao diện chương trình. Lập trình viên có thể tạo ra User Control bằng các cách như sau:

- Thêm mới User Control ngay trong dự án.
- Tạo lớp và khai báo thừa kế từ lớp Control.
- Tạo dự án mới bằng cách chọn loại dự án là Windows Control Library.

9.2. Xây dựng User Control

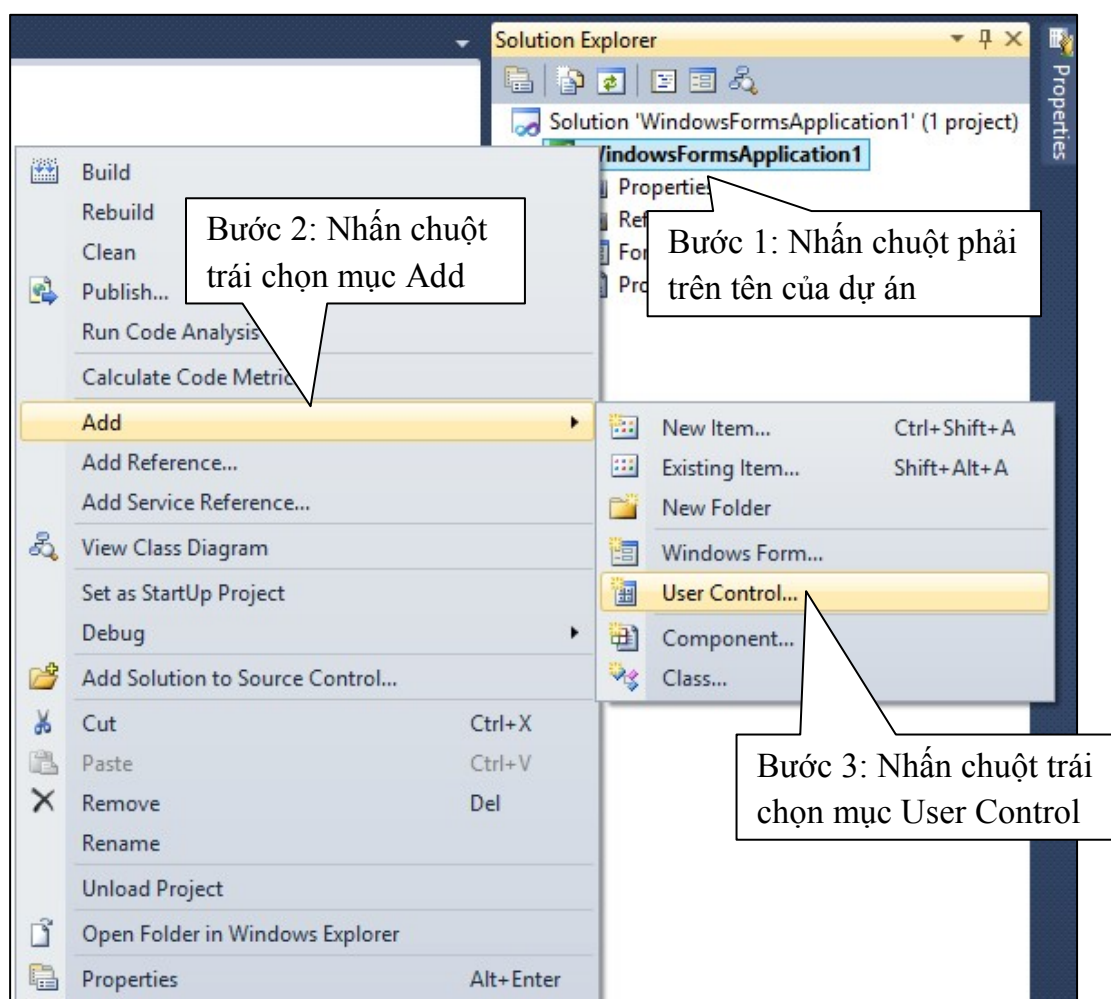
9.2.1. Thêm mới User Control ngay trong dự án

Lập trình viên thêm mới User Control trên thanh thực đơn như hình 9.1.



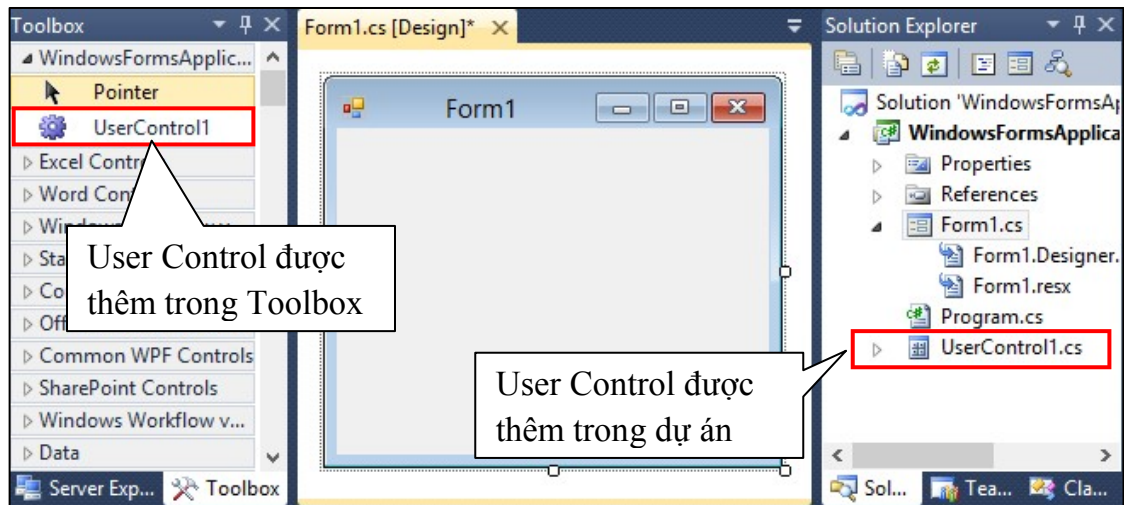
Hình 9.1: Thêm User Control từ thực đơn Project

Hoặc thêm mới User Control từ cửa sổ Solution Explorer như hình 9.2.



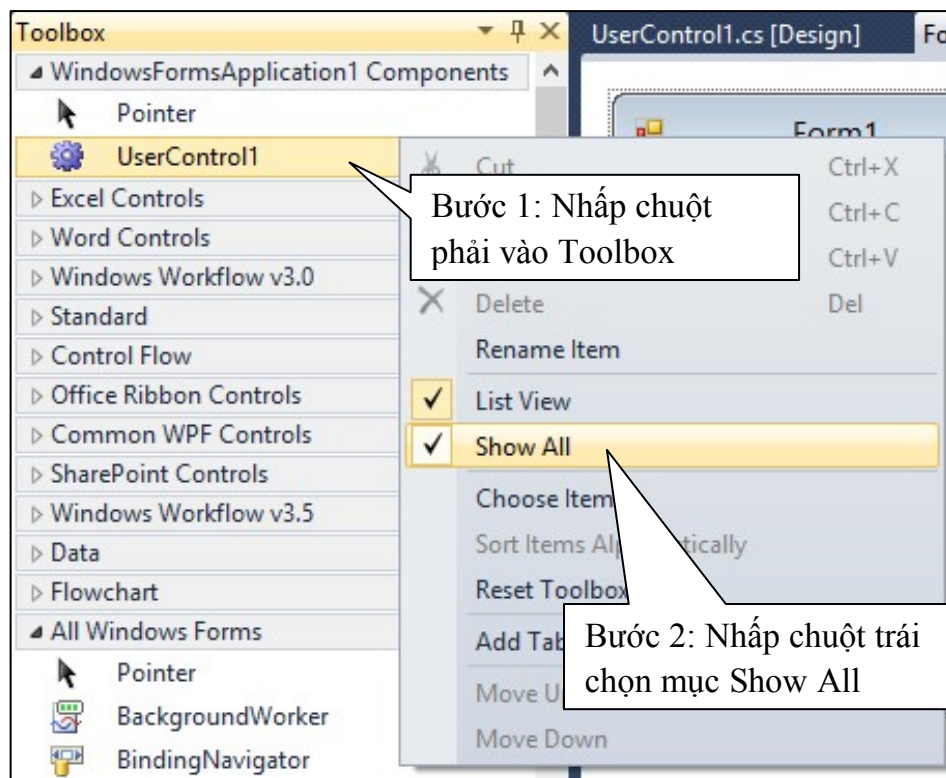
Hình 9.2: Thêm mới User Control từ cửa sổ Solution Explorer

Sau khi lập trình viên thêm mới một User Control vào dự án thì bên cửa sổ Toolbox sẽ hiển thị thêm một điều khiển là User Control vừa thêm trong nhóm UserControlInProject Components như hình 9.3.



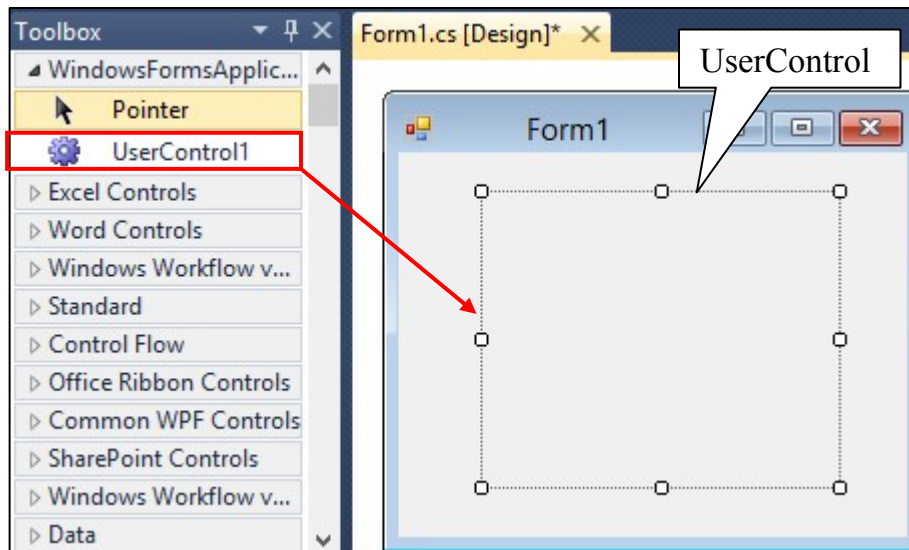
Hình 9.3: User Control mới được hiển thị trên cửa sổ Toolbox

Nếu sau khi thêm mới User Control mà không thấy hiển thị trên cửa sổ Toolbox có thể xử lý bằng cách nhấp phải vào Toolbox và chọn mục Show All như hình 9.4.



Hình 9.4: Chọn chức năng hiển thị tất cả điều khiển trên Toolbox

Sau khi đã thêm User Control, lập trình viên có thể sử dụng User Control bằng cách kéo User Control trên Toolbox vào form như hình 9.5.

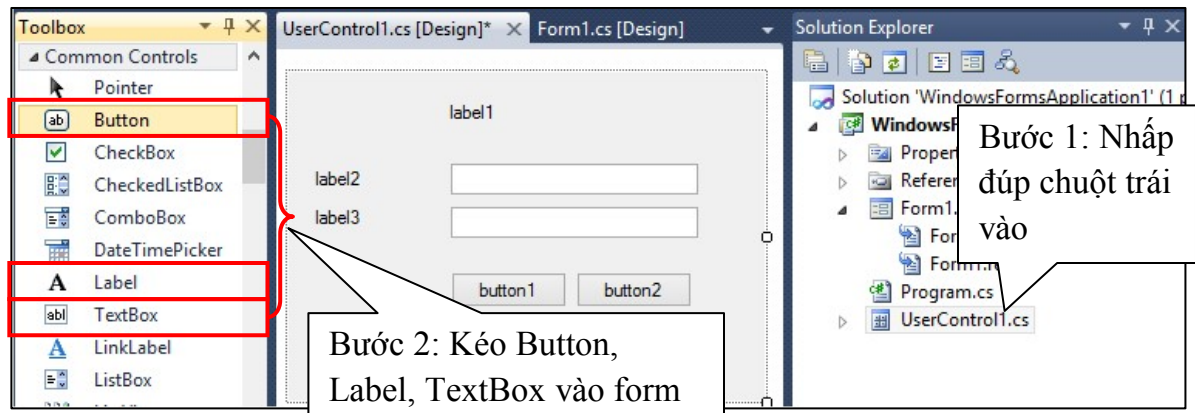


Hình 9.5: Giao diện form sau khi thả User Control vào

Ví dụ: Tạo UserControl Đăng nhập như hình 9.6. Sau đó kéo thả UserControl vào form.

Hình 9.6: Form đăng nhập

Bước 1: Thêm User Control có tên UserControl1 vào form. Sau đó thêm các điều khiển Label, TextBox và Button vào UserControl1 như hình 9.7.

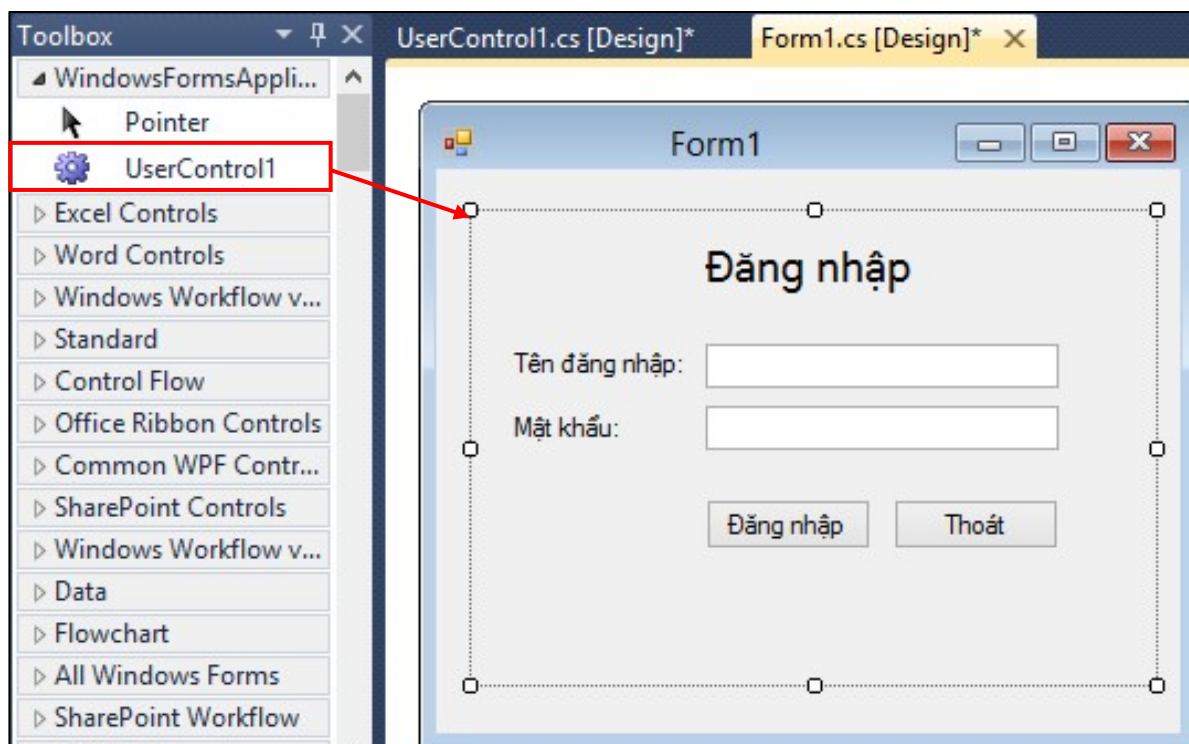


Hình 9.7: Giao diện UserControl1 sau khi thêm điều khiển.

Bước 2: Thiết lập giá trị thuộc tính cho điều khiển trên UserControl1 trong cửa sổ Properties.

- label1:
Thuộc tính *Text*: “Đăng nhập”
Thuộc tính *Font Size*: 14
- label2:
Thuộc tính *Text*: “Tên đăng nhập:”
- label3:
Thuộc tính *Text*: “Mật khẩu:”
- textBox1:
Thuộc tính *Name*: txtTen
- textBox2:
Thuộc tính *Name*: txtMatKhau
- button1:
Thuộc tính *Name*: btnDangNhap
Thuộc tính *Text*: “Đăng nhập”
- button2:
Thuộc tính *Name*: btnThoat
Thuộc tính *Text*: “Thoát”

Bước 3: Khi đã thiết kế xong UserControl1, tiến hành biên dịch chương trình để UserControl1 cập nhật các thay đổi. Sau đó lập trình viên kéo UserControl1 từ cửa sổ Toolbox vào form sẽ được giao diện đăng nhập như hình 9.8



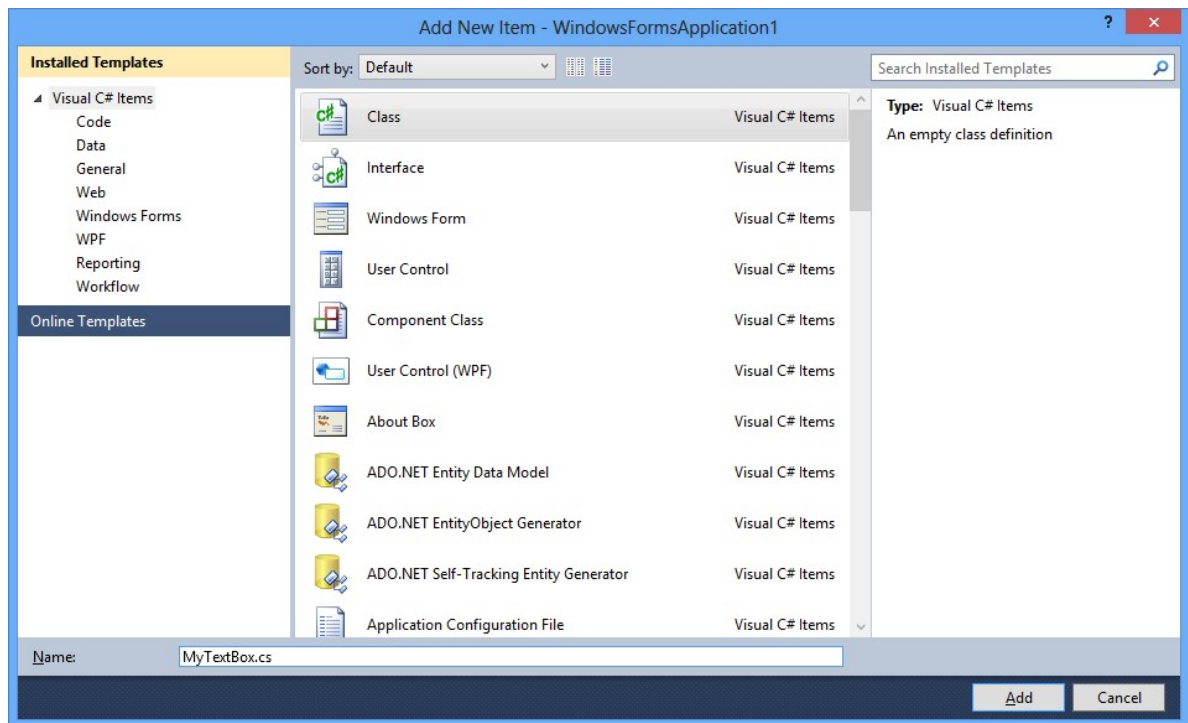
Hình 9.8: Giao diện form sau khi kéo điều khiển UserControl1 thả vào form

9.2.2. Tạo lớp và khai báo thừa kế từ lớp Control

Lập trình viên có thể tạo ra Controls riêng bằng cách tạo ra một lớp mới và chỉ định thừa kế đến một lớp cụ thể. Lớp cụ thể này có thể là lớp: TextBox, Label, ComboBox, ... Ví dụ 9.2: Tạo ra điều khiển MyTextBox. Điều khiển MyTextBox thừa kế từ lớp TextBox và có thêm thuộc tính mới là NewValue.

Hướng dẫn:

Bước 1: Tạo lớp mới bằng cách chọn Project > Add Class để hiển thị cửa sổ Add New Item và đặt tên lớp là MyTextBox như hình 9.9.



Hình 9.9: Giao diện cửa sổ Add New Item

Bước 2: Viết mã lệnh cho lớp MyTextBox

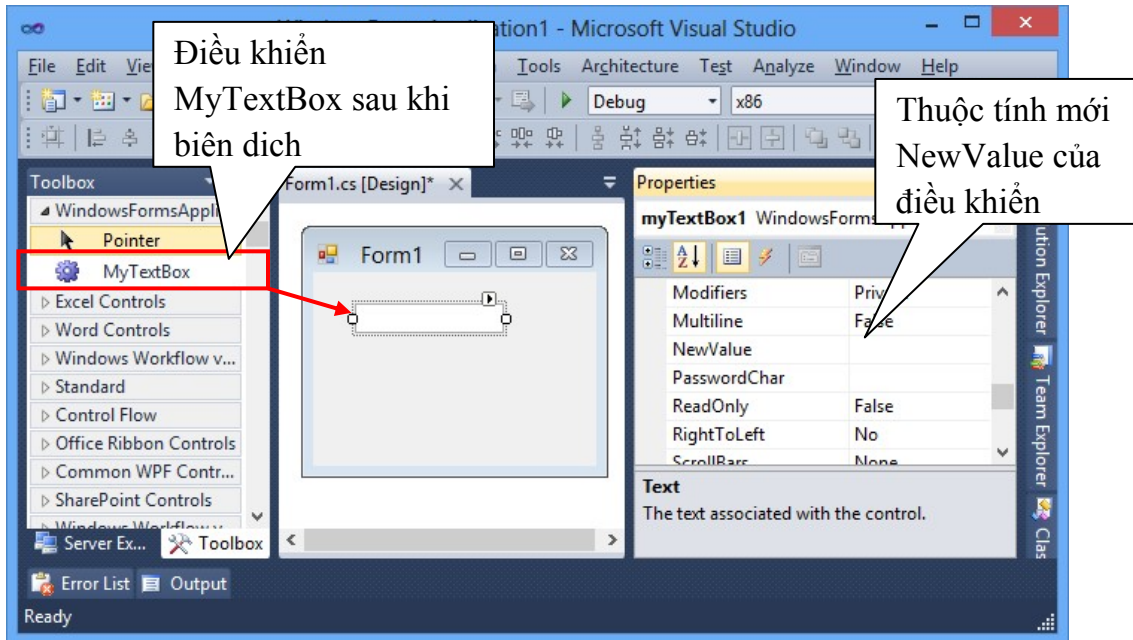
- Khai báo thừa kế đến lớp TextBox và thêm thuộc tính NewValue:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
    class MyTextBox:TextBox
    {
        string strNewValue;
        public string NewValue
        {
            get
            {
                return strNewValue;
            }
            set
            {
                strNewValue = value;
            }
        }
    }
}
```

Khai báo thêm
không gian tên
Forms

Thừa kế đến lớp
TextBox

Bước 3: Biên dịch chương trình, điều khiển MyTextBox sẽ xuất hiện trên cửa sổ Toolbox như hình 9.10.



Hình 9.10: Giao diện form sau khi thêm điều khiển MyTextBox

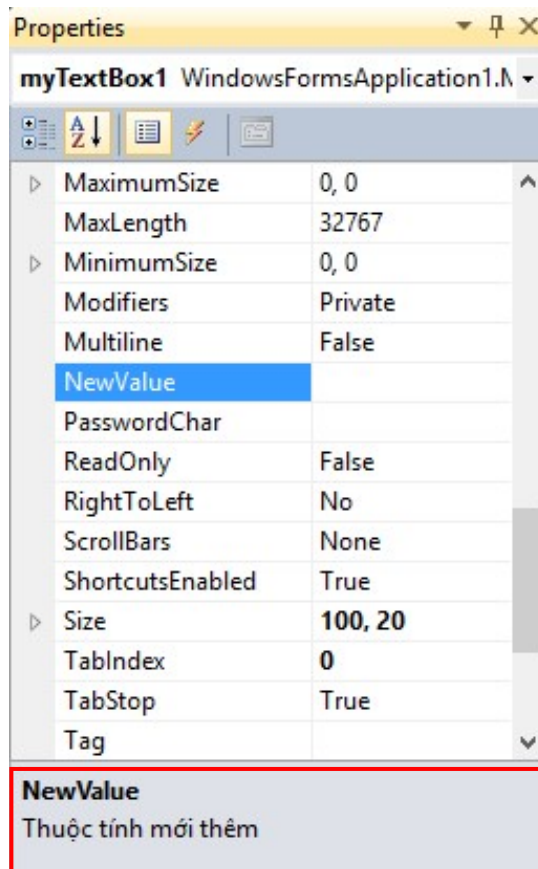
Lập trình viên cũng có thể khai báo chuỗi diễn giải để mô tả chức năng của thuộc tính vừa thêm. Để làm được việc này cần khai báo không gian tên System.ComponentModel và thiết lập thuộc tính Description.

Ví dụ: Khai báo diễn giải cho thuộc tính NewValue cho điều khiển MyTextBox ở ví dụ 9.2.

- Viết mã lệnh trong lớp MyTextBox như sau:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.ComponentModel;
namespace WindowsFormsApplication1
{
    class MyTextBox:TextBox
    {
        string strNewValue;
        [Description("Thuộc tính mới thêm")]
        public string NewValue
        {
            get{ return strNewValue; }
            set{ strNewValue = value; }
        }
    }
}
```


- Giao diện chương trình khi thêm mô tả thuộc tính như hình 9.11:

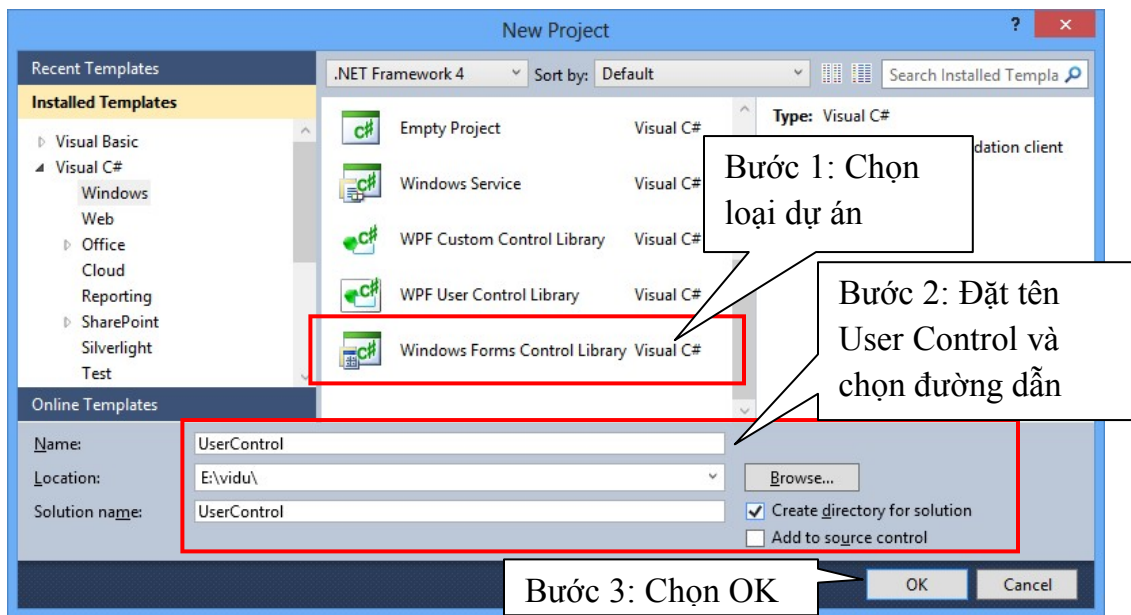


Hình 9.11: Mô tả thuộc tính *NewValue*

9.2.3. Tạo dự án mới bằng cách chọn loại dự án là Windows Control Library

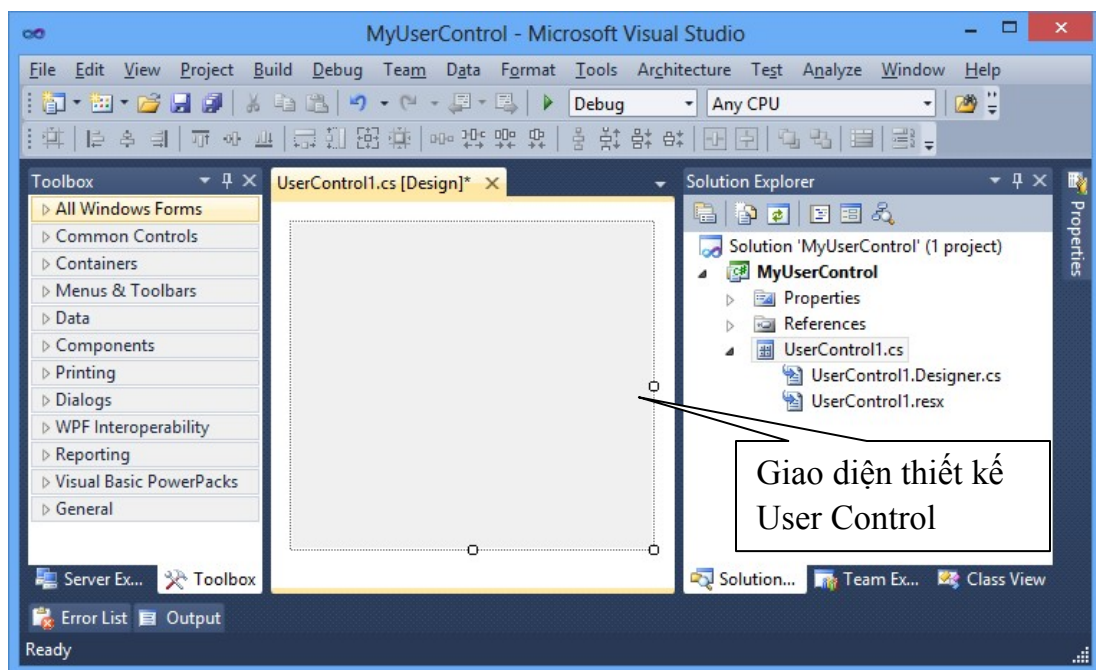
Trong trường hợp muốn tạo User Control để sử dụng cho các dự án khác nhau, lập trình viên có thể tạo User Control và đưa về dạng tập tin .DLL.

Để tạo được User Control ở dạng .DLL, cần tạo mới dự án ở dạng Windows Forms Control Library: Chọn File > New Project hiển thị cửa sổ New Project như hình 9.12.



Hình 9.12: Cửa sổ New Project tạo User Control dạng DLL.

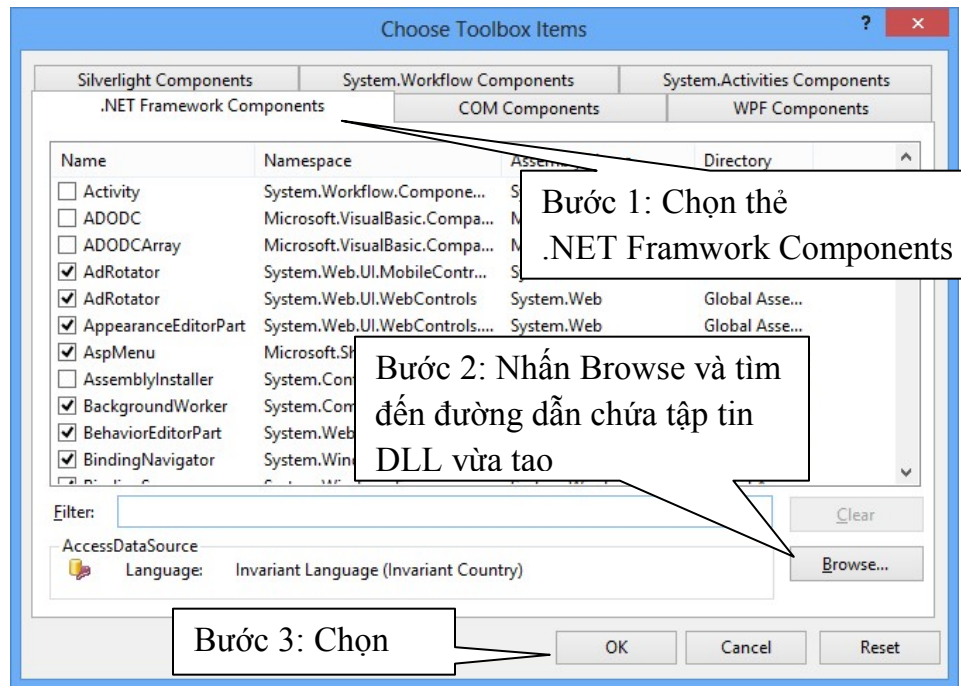
Sau khi chọn OK sẽ được giao diện User Control như hình 9.13:



Hình 9.13: Giao diện User Control khi thêm

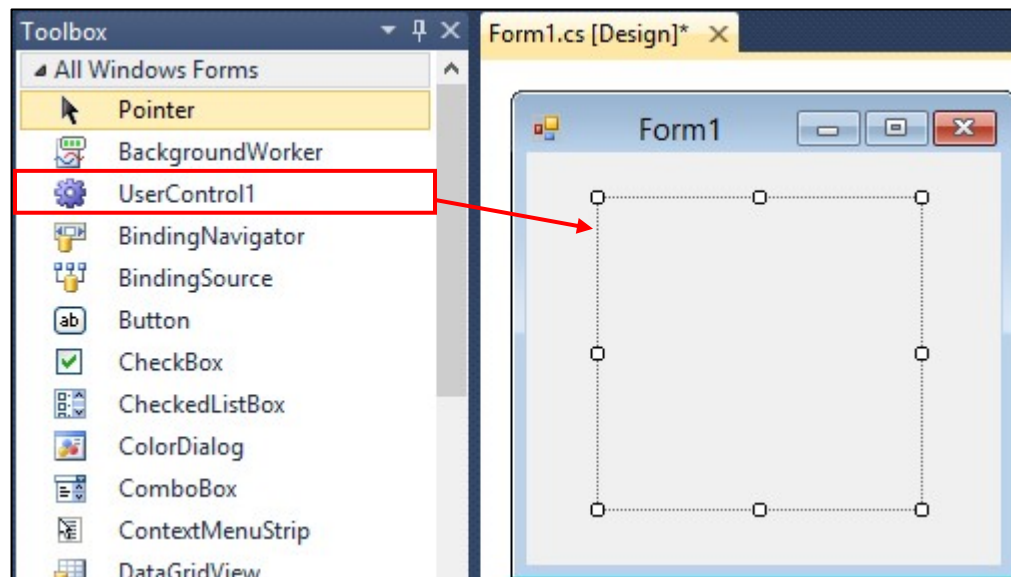
Khi thiết kế xong User Control, lập trình viên cần biên dịch dự án để tạo User Control ở dạng tập tin DLL. Tập tin DLL được tạo trong thư mục Debug. Khi muốn sử dụng User Control trong trong án nào chỉ cần thêm DLL đó vào Toolbox như sau:

- Trên dự án muốn thêm, nhấp phải chuột trên Toolbox chọn Choose Items, cửa sổ Choose Toolbox Items được hiển thị như hình 9.14:



Hình 9.14: Cửa sổ Choose Toolbox Item

User Control được hiển thị trên cửa sổ Toolbox, lập trình viên sử dụng bằng cách kéo User Control vừa thêm vào form như hình 9.15.



Hình 9.15: Thêm User Control vào form từ cửa sổ Toolbox

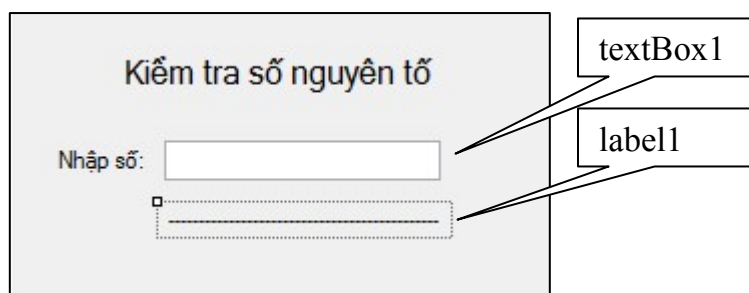
9.3. Sử dụng User Control

Sau khi thêm User Control từ cửa sổ Toolbox vào form, lập trình viên có thể truy xuất từng điều khiển trên User Control thông qua thuộc tính Controls.

Ví dụ muốn gán giá trị “Xin chào các bạn” cho thuộc tính Text cho điều khiển Label có tên lblNhan của UserControl1 cần thực hiện như sau:

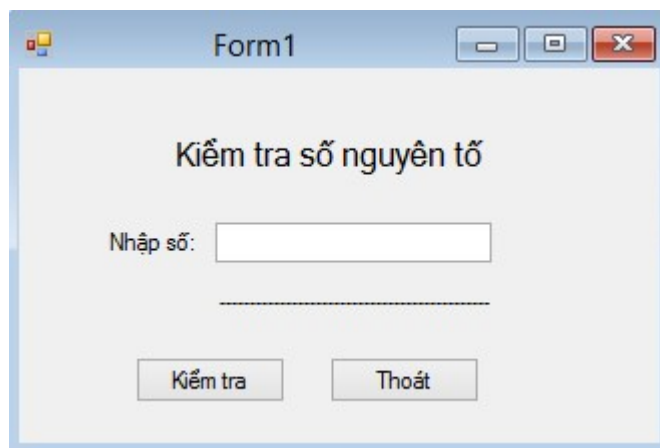
```
UserControl1.Controls["lblNhan"].Text = "Xin chào các bạn";
```

Ví dụ 9.3: Xây dựng kiểm tra số nguyên tố, giao diện xây dựng từ User Control gồm các điều khiển Label, TextBox để nhập số nguyên như hình 9.16.



Hình 9.16: Giao diện User Control kiểm tra số nguyên tố

Sau khi xây dựng xong User Control, tiến hành kéo User Control vào form. Trên form thêm hai Button “Kiểm tra” để kiểm tra số nhập vào có phải số nguyên tố hay không và một Button để thoát chương trình. Giao diện form như hình 9.17.



Hình 9.17: Giao diện form kiểm tra số nguyên tố

Yêu cầu: Khi nhấn Button “Kiểm tra”, nếu giá trị trong textBox1 là số nguyên tố thì hiển thị trên label1 chuỗi “Là số nguyên tố”, nếu không phải số nguyên tố thì hiển thị trên label1 chuỗi “Không phải số nguyên tố”.

Hướng dẫn:

Bước 1: Xây dựng giao diện ban đầu:

- Thêm mới User Control: chọn Project > Add User Control. User Control có tên UserControl1 được thêm vào. Trên UserControl1 kéo các điều khiển Label và TextBox từ cửa sổ Toolbox vào như hình 9.18.

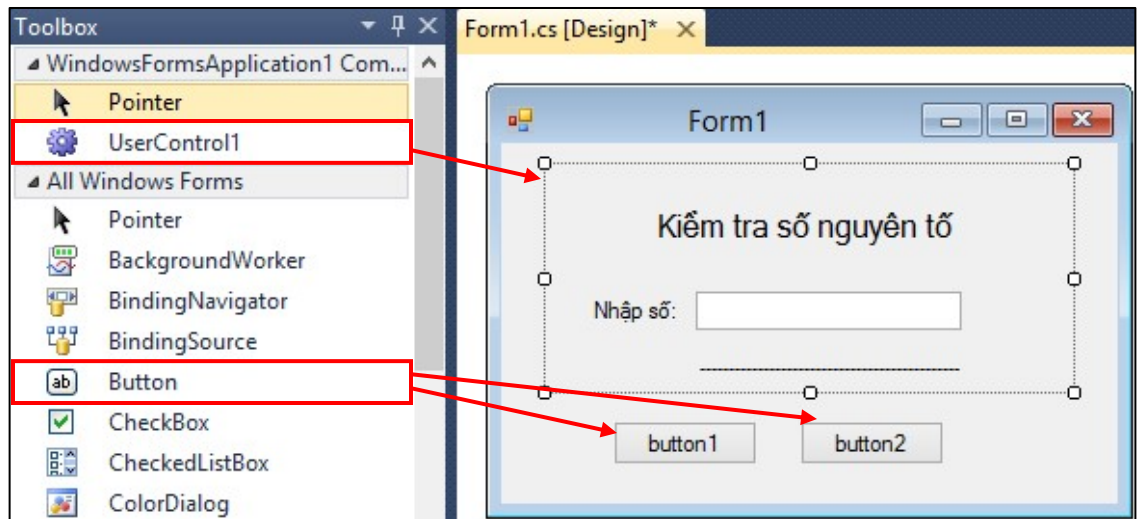


Hình 9.18: Giao diện ban đầu UserControl1

- Thiết lập thuộc tính cho label3:
Thuộc tính *Text*: “Kiểm tra số nguyên tố”
Thuộc tính *Font Size*: 14
- Thiết lập thuộc tính cho label2:
Thuộc tính *Text*: “Nhập số:”
- Thiết lập thuộc tính cho label1:
Thuộc tính *Text*: “-----”
Thuộc tính *Name*: lblKetQua
- Thiết lập thuộc tính textBox1:
Thuộc tính *Name*: txtNhapSo

Khi thiết lập xong thuộc tính sẽ được giao diện UserControl1 như hình 9.16

- Trên form, kéo User Control vừa tạo vào form và thêm hai Button như hình 9.19.



Hình 9.19: Giao diện form khi thêm UserControl1 và Button

- Thiết lập thuộc tính cho button1:
Thuộc tính Text: “Kiểm tra”
Thuộc tính Name: btnKiemTra
- Thiết lập thuộc tính cho button2:
Thuộc tính Text: “Thoát”
Thuộc tính Name: btnThoat

Khi thiết lập xong thuộc tính sẽ được giao diện form như hình 9.17.

Bước 2: Viết mã lệnh cho điều khiển

- Viết hàm kiểm tra số nguyên tố:

```
bool kiemTraSoNguyenTo(int x)
{
    bool kt = true;
    for (int i = 2; i <= x / 2; i++)
        if (x % i == 0)
            kt = false;
    return kt;
}
```

- Sự kiện Click nút btnThoat:

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

- Sự kiện Click nút btnKiemTra:

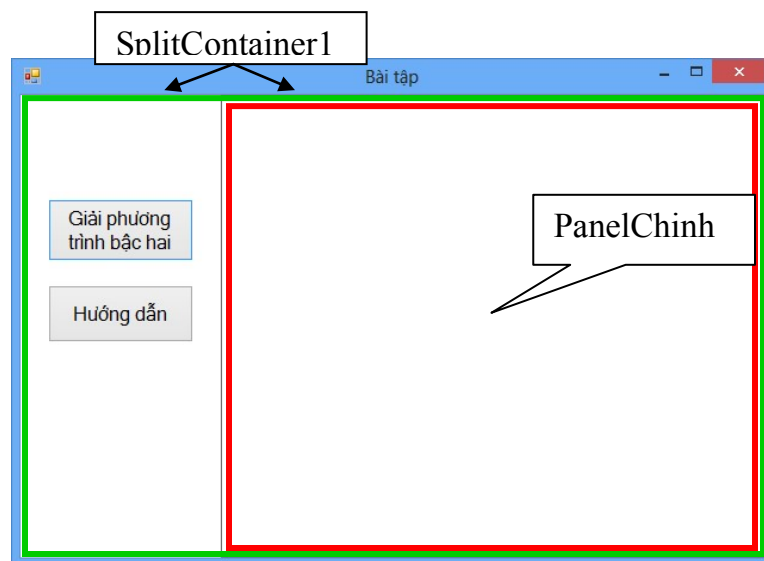
```
private void btnKiemtra_Click(object sender, EventArgs e)
{
    int gt =
        int.Parse(userControl11.Controls["txtNhapSo"].Text);
    if (kiemTraSoNguyenTo(gt) == true)
        userControl11.Controls["lblKetQua"].Text = " là số
                                                    nguyên tố";
    else
        userControl11.Controls["lblKetQua"].Text =
            gt.ToString() + " không là số nguyên tố";
    userControl11.Controls["txtNhapSo"].Text = "";
}
```

9.4. Cách sử dụng những thư viện tạo sẵn để thiết kế giao diện

Trong thực tế, có những thư viện tạo sẵn hỗ trợ lập trình viên xây dựng giao diện với mục đích tiết kiệm thời gian như: DotNetBar, SandDock, Phần lớn các thư viện này đều xây dựng ở dạng tập tin .DLL, do đó lập trình viên có thể dễ dàng sử dụng bằng cách thêm điều khiển vào cửa sổ Toolbox, sau đó kéo thả điều khiển đó vào form. Xem cách sử dụng tại mục 9.2.3.

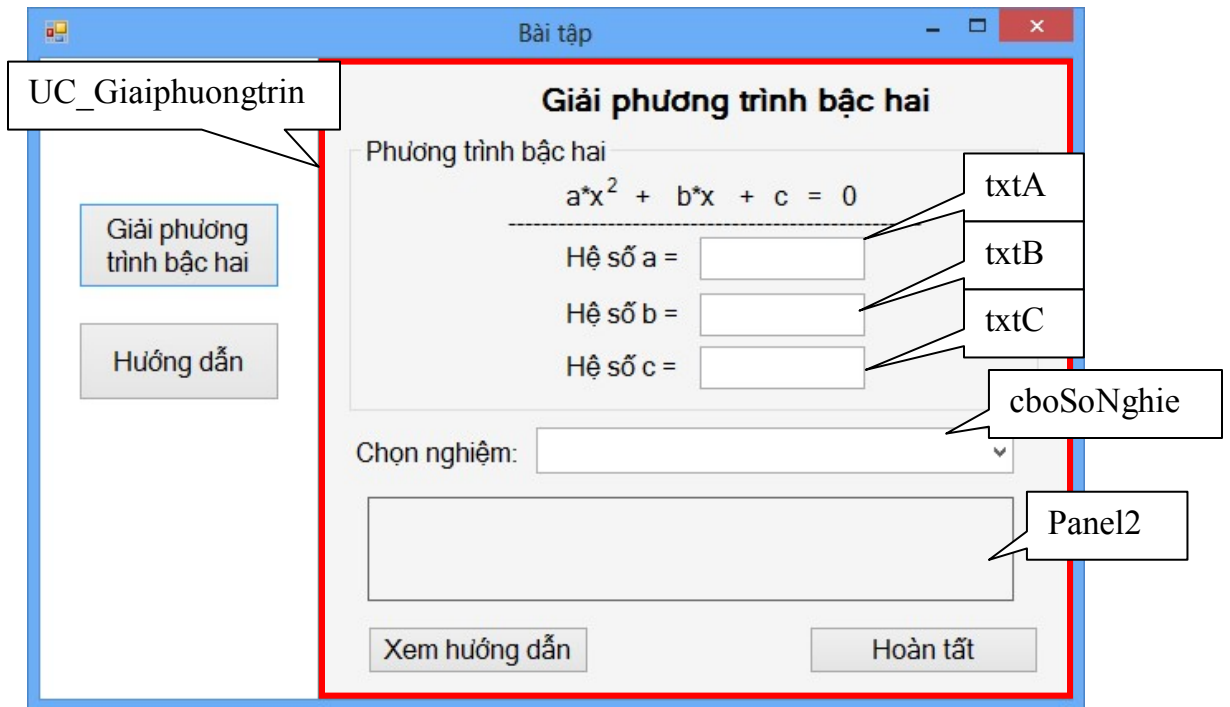
9.5. Bài tập cuối chương

Viết chương trình hỗ trợ người dùng học giải phương trình bậc hai có giao diện như hình 9.20.



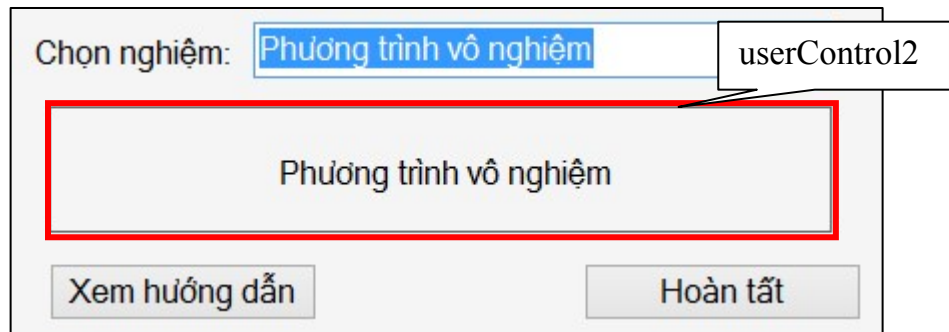
Hình 9.20: Giao diện chính hỗ trợ người dùng học giải phương trình bậc hai

- Khi người dùng nhấn chuột chọn Button “Giải phương trình bậc hai” trên giao diện chính:
 - Chương trình sẽ hiển thị giao diện như hình 9.21. Giao diện “Giải phương trình bậc hai” hiển thị trong PanelChinh là UserControl “UC_Giaiphuongtrin”.



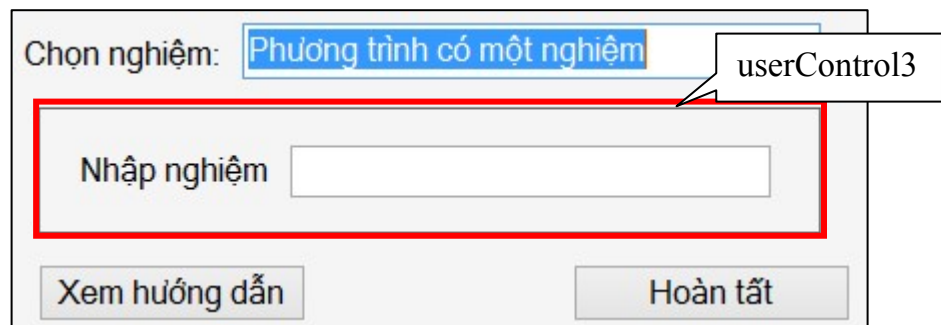
Hình 9.21: Giao diện hiển thị phương trình bậc hai

- Các TextBox: txtA, txtB, txtC chứa các hệ số a, b, c của phương trình và được phát sinh ngẫu nhiên.
- ComboBox “cboSoNghiem: Chứa 3 lựa chọn cho người dùng lựa chọn bao gồm 3 lựa chọn: “Phương trình vô nghiệm”, “Phương trình có một nghiệm”, “Phương trình có hai nghiệm”.
- Với mỗi sự lựa chọn ở cboSoNghiem thì sẽ hiển thị các UserControl: “UC_VoNghiem”, “UC_MotNghiem”, “UC_HaiNghiem” tương ứng như sau:
 - Nếu người dùng chọn mục “Phương trình vô nghiệm” thì Panel2 sẽ hiển thị UserControl “UC_VoNghiem bên trong như hình 9.22.



Hình 9.22: Giao diện hiển thị UC_VoNghiem khi phương trình vô nghiệm

- Nếu người dùng chọn “Phương trình có một nghiệm” thì Panel2 sẽ hiển thị UserControl “UC_MotNghiem” bên trong như hình 9.23. Trong UserControl “UC_MotNghiem” có một điều khiển TextBox cho phép người dùng nhập nghiệm đúng của phương trình. Sau khi nhập xong người dùng nhấn Button “Hoàn tất” để kiểm tra kết quả. Nếu kết quả đúng sẽ hiển thị MessageBox với nội dung “Bạn đã giải đúng”, nếu nhập sai sẽ hiển thị MessageBox với nội dung “Bạn đã giải sai”.



Hình 9.23: Giao diện hiển thị userControl3 khi phương trình có một nghiệm

- Nếu người dùng chọn “Phương trình có hai nghiệm” thì Panel2 sẽ hiển thị UserControl “UC_HaiNghiem” bên trong như hình 9.24. Trong UserControl “UC_HaiNghiem” có hai điều khiển TextBox cho phép người dùng nhập nghiệm đúng của phương trình. Sau khi nhập xong người dùng nhấn Button “Hoàn tất” để kiểm tra kết quả. Nếu kết quả đúng sẽ hiển thị MessageBox với nội dung “Bạn đã giải đúng”, nếu nhập sai sẽ hiển thị MessageBox với nội dung “Bạn đã giải sai”.

Chọn nghiệm: Phương trình có hai nghiệm

Nhập nghiệm 1

Nhập nghiệm 2

Xem hướng dẫn Hoàn tất

userControl4

Hình 9.24: Giao diện hiển thị userControl3 khi phương trình có hai nghiệm

- Nhấn Button “Xem hướng dẫn” thì sẽ hiển thị UserControl “UC_HuongDan” trên PanelChinh mô tả cách giải phương trình như hình 9.25.

Bài tập

Hướng dẫn giải phương trình bậc 2

Bước 1: Tính Delta = $b*b - 4*a*c$
 Bước 2: Xét giá trị Delta

- Nếu Delta < 0: Kết luận phương trình vô nghiệm
- Nếu Delta = 0: Phương trình có nghiệm kép:

$$x1 = x2 = \frac{-b}{2*a}$$
- Nếu Delta > 0: Phương trình có hai nghiệm phân biệt:

$$x1 = \frac{-b - \text{Math.Sqrt}(\text{Delta})}{2*a}$$

$$x2 = \frac{-b + \text{Math.Sqrt}(\text{Delta})}{2*a}$$

Trở lại

Giải phương trình bậc hai

Hướng dẫn

Hình 9.25: Giao diện UC_Huongdan trên PanelChinh

Nhấn Button “Trở lại” sẽ quay về màn hình “Giải phương trình bậc hai” như hình 9.21.

- Khi người dùng nhấn Button “Hướng dẫn” trên giao diện chính: thì sẽ hiển thị UserControl “UC_Huongdan” trên PanelChinh như hình 9.25.