

## Chương 2: XÂY DỰNG WINDOWS FORMS APPLICATION

### Bài 3: XÂY DỰNG WINDOWS FORMS APPLICATION

#### I. Sử dụng Microsoft Visual Studio

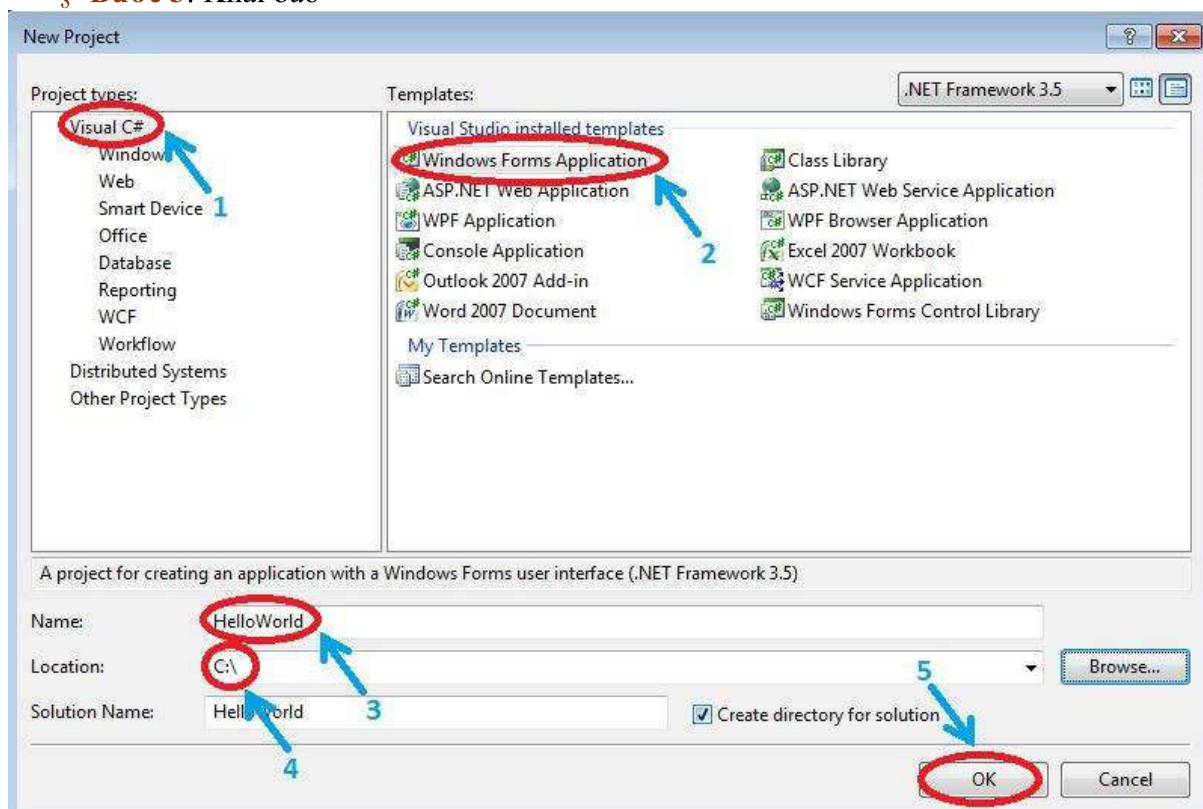
##### 1. Khởi động

§ **Bước 1:** Khởi động Visual Studio

Start | All Programs | Microsoft Visual Studio | Microsoft Visual Studio

§ **Bước 2:** Vào menu File | New | Project


§ **Bước 3:** Khai báo



- Mở hộp ToolBox: Menu View | ToolBox à chứa các control
- Mở cửa sổ Properties: Menu View | Properties à chứa thuộc tính
- Mở cửa sổ Solution Explorer: Menu View | Solution Explorer à cửa sổ Project xuất hiện.

§ **Bước 4:** Thiết kế Form – Viết code

- Thiết kế form: Nhấp vào  View Designer (trong cửa sổ Solution Explorer)
- Viết code: Nhấp vào  View Code (trong cửa sổ Solution Explorer)

§ **Bước 5:** Để chạy chương trình, nhấn F5 hoặc nhấp vào nút 

Để dừng chương trình, nhấn Shift + F5 hoặc nhấp vào nút 

#### \* Các thao tác với Project / Solution

##### a. Tạo Project

- C1. Vào menu File | New | Project
- C2. Ctrl + Shift + N
- C3. Chọn công cụ **New Project** trên thanh Standart

**b. Mở Project / Solution:**

- C1. Vào menu File | Open | Project / Solution
- C2. Ctrl + Shift + O

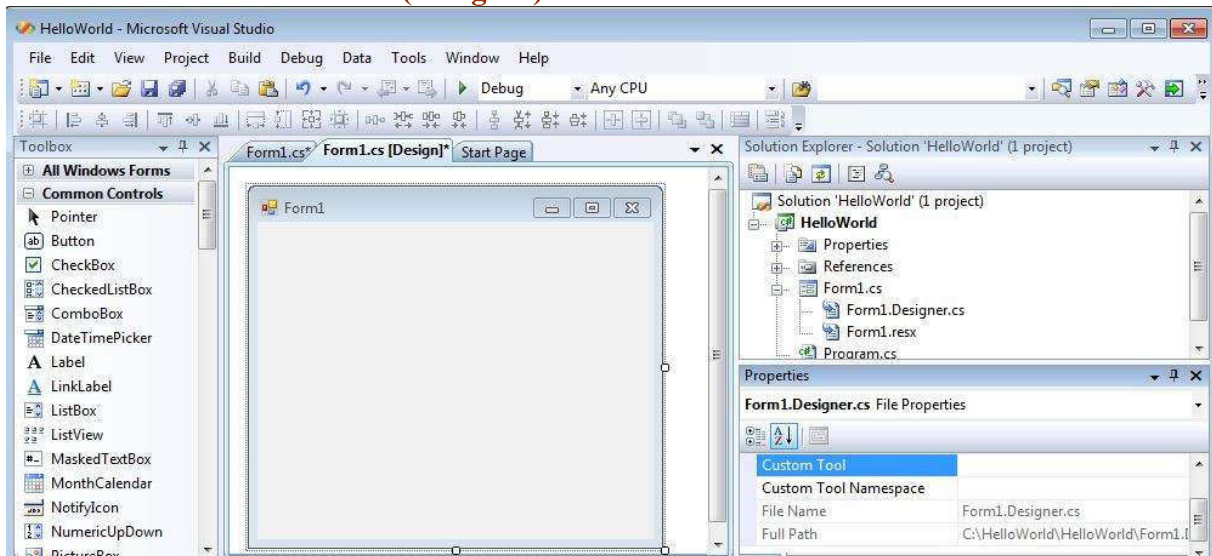
**c. Lưu Project / Solution**

- C1. Vào menu File | Save All
- C2. Chọn công cụ **Save All** trên thanh Standart

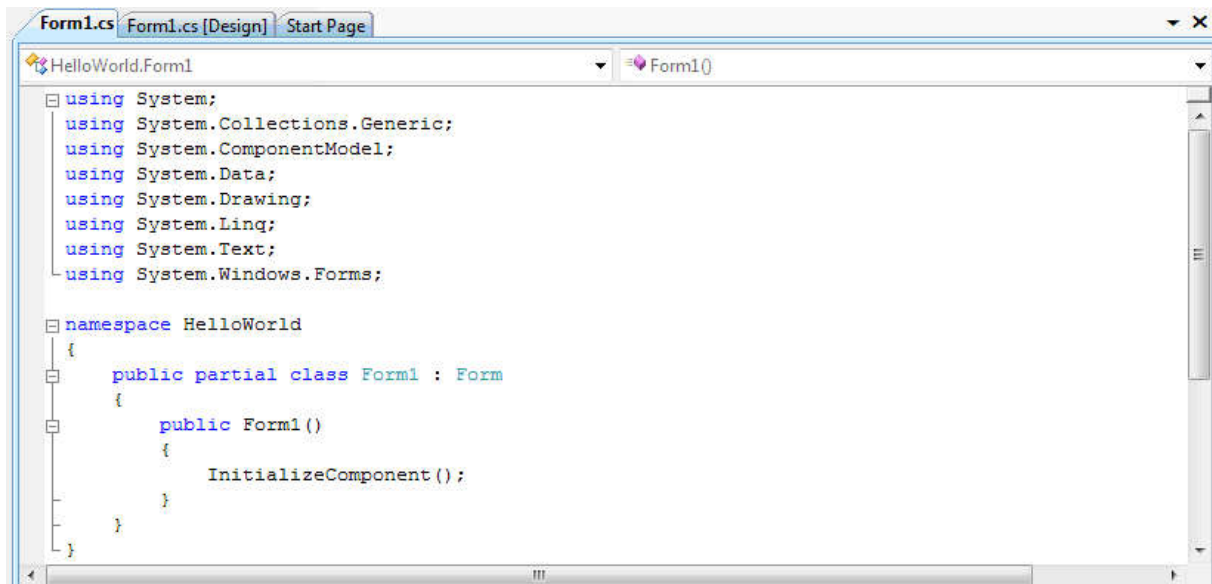
**d. Đóng Solution:** Vào menu File | Close Solution

## 2. Màn hình giao diện của Windows Forms

**a. Cửa sổ thiết kế Form (Designer):**



**b. Cửa sổ thiết viết code:**



**\* Các thao tác với Form**

**a. Thêm một Form mới vào Project:**

- b1. C1. Vào menu Project | Add New Item ...
  - C2. Chọn công cụ **Add New Item** trên thanh Standart
- b2. Khai báo
  - + Categories: chọn Windows Forms
  - + Templates: chọn Windows Form
  - + Name: đặt tên Form
- b3. Nhấp Add

**b. Thêm một Form có sẵn vào Project:**

- b1. Vào menu Project | Add Existing Item ...
- b2. Chọn Form
- b3. Nhấp Add

**c. Xóa bỏ một Form đang có trong Project:**

- b1. Chọn Form cần gỡ bỏ (ở cửa sổ Solution Explorer)
- b2. Vào menu Edit | Delete

**d. Lưu Form**

- Vào menu File | Save Form.cs
- Ctrl + S

**\* Ghi chú**

- Ta đem “bỏ vào” form các đối tượng như: Label, TextBox, Button, ...
  - + Label, TextBox, Button, ... được gọi là control hay còn gọi là component.
  - + Form được gọi là control “chứa”.
- Khi thay đổi nội dung của Label, TextBox, Button, ... ta thay đổi vào Text. Text được gọi là Property của control.

**3. Control là gì?**

- Control là lớp (class) các thành phần được thêm vào Windows Forms để tương tác giữa người sử dụng với Windows.
- Có rất nhiều loại control trong Windows Forms như: Label, TextBox, ListBox, ComboBox, Button, ...
- Các control sử dụng trên Windows Forms dùng namespace **System.Windows.Forms**.

**4. Properties (thuộc tính) của control**

- Properties là những thông tin mà ta có thể thay đổi nội dung, cách trình bày ... của người thiết kế để ứng dụng vào control.
- Mỗi lớp (class) có nhiều property khác nhau. Tuy nhiên, vẫn có một số property giống nhau được xây dựng từ lớp ban đầu.

**\* Bảng trình bày các thuộc tính (Properties) giống nhau**

Thuộc tính	Mô tả
Anchor	Có 4 hướng được định nghĩa là: top, bottom, left, right để cố định (neo). Khi control chứa nó thay đổi kích thước thì nó sẽ bị thay đổi kích thước nếu nếu các hướng left / right / top / bottom bị cố định (neo).
BackColor	Màu nền của control.
Bottom	Là khoảng cách theo chiều dọc từ cạnh đáy của control đến cạnh trên của control chứa nó.
Dock	Giống như Anchor nhưng việc cố định (neo) này theo một cạnh nào đó của control (hoặc cả 4 cạnh) với control chứa nó.
Enabled	Control được phép tương tác (True) hay không được phép tương tác (False)) với người dùng.

ForeColor	Màu chữ của control.
Height	Là chiều cao của control tính từ cạnh trên của control đến cạnh dưới của control.
Left	Là khoảng cách theo chiều ngang từ cạnh trái của control đến cạnh trái của control chứa nó.
Name	Tên của control.
Parent	Chỉ đến control chứa control hiện hành.
Right	Là khoảng cách theo chiều ngang từ cạnh phải của control đến cạnh trái của control chứa nó.
TabIndex	Thứ tự focus khi nhấn phím Tab (trên bàn phím) của control so với các control khác cùng nằm trong control chứa nó.
TabStop	Chỉ định control có được phép “bắt” (True) / không được phép “bắt” (False) phím Tab. Nếu không được phép thì TabIndex cũng không dùng được.
Tag	Là nhãn phân biệt giữa các control giống nhau trong cùng form.
Text	Nội dung hiện trong control.
Top	Là khoảng cách theo chiều dọc từ cạnh trên của control đến cạnh trên của control chứa nó.
Visible	Cho phép control hiện (True) / không hiện (False) khi chạy ứng dụng.
Width	Là chiều rộng của control tính từ cạnh trái của control đến cạnh phải của control.

**\* Bảng trình bày các phương thức (Method) xử lý trên chuỗi**

Phương thức	Mô tả
Clear()	Xóa nội dung
ResetText()	Xóa nội dung Text
Trim()	Cắt bỏ khoảng trắng thừa hai bên chuỗi


## II. Các control cơ bản

### 1. Label ( Label )

#### a. Công dụng:

- Hiện thị chuỗi ký tự không thay đổi trên form (nhãn).

#### b. Tạo Label:

- Chọn công cụ 

- rê chuột và vẽ Label trên form.

#### c. Thuộc tính:


Thuộc tính	Mô tả
AutoSize	Điều chỉnh kích thước đối tượng cho vừa với chiều dài chuỗi ký tự
Font	
Name	Quy định font chữ cho văn bản
Bold	True: đậm / False: bỏ đậm
Italic	True: nghiêng / False: bỏ nghiêng
Size	Quy định cỡ chữ cho văn bản
Underline	True: gạch dưới / False: bỏ gạch dưới
TextAlign	Canh lề (Left / Center / Right)

### 2. TextBox ( TextBox )

### a. Công dụng:

- Dùng trình bày văn bản và cho phép người dùng được thay đổi nội dung văn bản.
- Công dụng chính là cho người dùng nhập văn bản.

### b. Tạo TextBox:

- Chọn công cụ  TextBox
- rê chuột và vẽ TextBox trên form.

### c. Thuộc tính:


Thuộc tính	Mô tả
PasswordChar	Quy định ký tự hiển thị cho ô mật khẩu.
Multiline	True: hiện thanh cuộn / False: không hiện thanh cuộn
ScrollBars	Thanh cuộn (None / Horizontal / Vertical / Both)

## 3. Button ( Button )

### a. Công dụng:

- Dùng để thực thi lệnh.
- Khi nhấp chuột lên button, chương trình nhận được tín hiệu Click và lệnh được thi hành.

### b. Tạo Button:

- Chọn công cụ  Button
- rê chuột và vẽ Button trên form.

### c. Thuộc tính:


Thuộc tính	Mô tả
Text	Nhập nội dung vào Button

## 4. CheckBox ( CheckBox )

### a. Công dụng:

- Cho phép người dùng chọn hoặc không chọn.

### b. Tạo CheckBox:

- Chọn công cụ  CheckBox
- rê chuột và vẽ CheckBox trên form.

### c. Thuộc tính:

Thuộc tính	Mô tả
Checked	Không có dấu check (False) / Có dấu check (True)

### Ví dụ 3.1:

\* Khởi tạo một ứng dụng Windows Forms Application, lưu với tên là Vi Du 3.1 như sau:



\* Yêu cầu:

- Nhập:
  - + Username vào TextBox **Tên đăng nhập** (Name: *txtUser*)
  - + Password vào TextBox **Mật khẩu** (Name: *txtPass*)

- Chọn hoặc không chọn ô CheckBox **Ghi nhớ** (Name: *chkNho*)
- Nhấp button **Đăng nhập** thì hiện ra hộp thông báo chứa **Tên đăng nhập**, **Mật khẩu**; Và **“Bạn có ghi nhớ”** (nếu *chkNho* có đánh dấu chọn).
- Nhấp button **Xóa** thì xóa trống TextBox **Tên đăng nhập** và TextBox **Mật khẩu**, đồng thời di chuyển con trỏ vào *txtUser*.

- Nhấp button **Dừng** thì dừng chương trình.

\* Hướng dẫn:

- Thiết kế Form như yêu cầu, trong đó form có các thuộc tính sau:

- + AutoSize: True
- + Font: Times New Roman
- + Size: 12
- + Text: Form Đăng Nhập
- + Icon: logo.ico

- Nhấp đúp vào button **Đăng nhập**, thêm đoạn code sau:

```
string thongbao;
thongbao = "Tên đăng nhập là: " ;
thongbao += this.txtUser.Text ;
thongbao += "\n\rMật khẩu là: " ;
thongbao += this.txtPass.Text;
if (this.chkNho.Checked==true) {
    thongbao += "\n\rBạn có ghi nhớ.";
}
MessageBox.Show(thongbao, "Thông báo");
```

- Nhấp đúp vào button **Xóa**, thêm đoạn code sau:

```
this.txtUser.Clear();
this.txtPass.Clear();
this.txtUser.Focus();
```

- Nhấp đúp vào button **Dừng**, thêm đoạn code sau:

```
Application.Exit();
```

- + Có thể thay button **Xóa** bằng button **Reset** với đoạn code như sau:

```
this.txtUser.ResetText();
this.txtPass.ResetText();
this.txtUser.Focus();
```

### \* **Phần Bổ sung:**

1. **MessageBox.Show:** hiện hộp thông báo `MessageBox.Show("Thông báo", "Tiêu đề");` `MessageBox.Show("Thông báo", "Tiêu đề", Buttons, Icon);`

+ Buttons (nút lệnh):

```
MessageBoxButtons.OK
MessageBoxButtons.OKCancel
MessageBoxButtons.YesNo
MessageBoxButtons.YesNoCancel
```

+ Icon (biểu tượng):

```
MessageBoxIcon.Information
MessageBoxIcon.Question
MessageBoxIcon.Warning
```

+ DialogResult (kết quả trả về):

```
DialogResult.Ok
```



```
DialogResult.Yes
```

## 2. Dừng chương trình:

```
this.Close();  
Application.Exit();
```

## 3. Event (sự kiện) là gì?

- Sự kiện là những phản ứng của đối tượng. Nói cách khác, sự kiện là những tín hiệu phát ra khi người dùng thao tác trên đối tượng.

- Nhờ có event, người lập trình sẽ nhận được những tín hiệu và xử lý những tín hiệu đó để phản hồi lại cho người dùng, tạo nên sự nhịp nhàng cho chương trình.

## 4. Bảng trình bày các sự kiện (Events) của control

Sự kiện	Mô tả
Click	Gọi đến khi control bị Click. Trong một vài control, event này cũng xảy ra khi người dùng nhấn phím Enter.
DoubleClick	Gọi đến khi control bị Double-Click. Trong một vài control, event này không bao giờ được gọi. Ví dụ: control Button.
DragDrop	Gọi đến khi việc “Drag and Drop” được hoàn tất.
DragEnter	Gọi đến khi đối tượng vừa được “Drag” đến biên của control.
DragLeave	Gọi đến khi đối tượng vừa được “Drag” ra ngoài biên của control.
DragOver	Gọi đến khi đối tượng được “Drag” bên trong control.
KeyDown	Gọi đến khi vừa bấm một phím bất kỳ từ 1 control đang focus. Sự kiện này luôn được gọi trước sự kiện KeyUp.
KeyPress	Gọi đến khi vừa bấm một phím bất kỳ từ 1 control được focus. Sự kiện này được gọi sau sự kiện KeyUp.
KeyUp	Gọi đến khi vừa bấm một phím bất kỳ rồi thả ra từ 1 control đang focus. Sự kiện này luôn được gọi sau sự kiện KeyDown.
GotFocus	Gọi đến khi control được focus.
LostFocus	Gọi đến khi control bị mất focus.
MouseDown	Gọi đến khi con trỏ chuột nằm trên 1 control và nút chuột được nhấp nhưng chưa thả ra.
MouseMove	Gọi đến khi con trỏ chuột đi qua 1 control.
MouseUp	Gọi đến khi con trỏ chuột nằm trên 1 control và nút chuột vừa được thả.
Paint	Gọi đến khi control được vẽ.
Validated	Gọi đến khi control focus, property CausesValidation được đặt là true và sau khi gọi việc kiểm tra bằng Validating.
Validating	Gọi đến khi control mất focus, property CausesValidation được đặt là true.

### Ví dụ 3.2:

\* Cải tiến Ví Du 3.1 cho button **Dừng** như sau:

- Khi nhấp vào button Dừng thì xuất hiện hộp thoại hỏi đáp có 2 button Ok, Cancel.

- Chương trình chỉ dừng khi nhấp tiếp vào nút Ok.

\* Hướng dẫn: Sửa lại button **Dừng** như sau

```
DialogResult traloi;  
traloi = MessageBox.Show("Chắc không?", "Trả lời",  
    MessageBoxButtons.OKCancel, MessageBoxIcon.Question);  
if ( traloi == DialogResult.OK) Application.Exit();
```

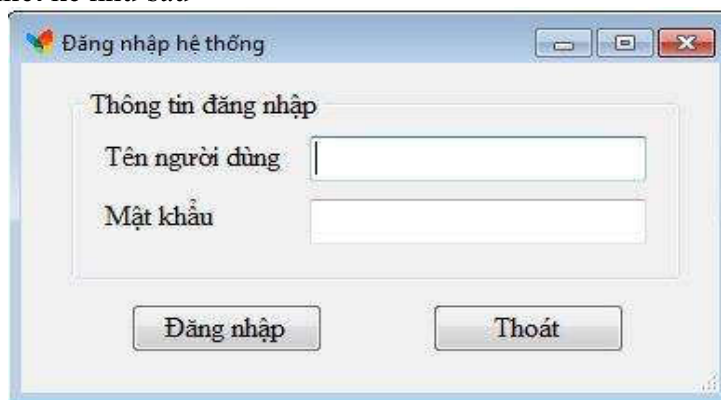
### Ví dụ 3.3:

\* Khởi tạo một ứng dụng Windows Forms Application, lưu với tên là Ví Du 3.3 gồm:

- Form1: được thiết kế như sau



- Form2: được thiết kế như sau



(txtUser, txtPass, btnDangnhap, btnThoat)

\* Yêu cầu:

- Khi chạy chương trình thì Form2 (Đăng nhập hệ thống) được mở lên trước.

- Nhập:

- + Tên người dùng vào TextBox **txtUser**

- + Mật khẩu vào TextBox **txtPass**

- Nhấp button **Đăng nhập** thì thực hiện kiểm tra

- + Nếu **txtUser** = "**teonv**" và **txtPass** = "**123**" thì chuyển sang Form1.

- + Ngược lại thì thông báo "**Không đúng tên người dùng / mật khẩu !!!**"

- Nhấp button **Thoát** thì hiển thị thông báo "Chắc không? "

- + Nếu chọn **Yes** thì kết thúc chương trình.

- + Ngược lại thì trở lại màn hình Đăng nhập hệ thống.

\* Hướng dẫn:

- **Form1**

- . Thiết kế Form như yêu cầu, trong đó form có các thuộc tính sau:

- + AutoSize: True

- + Font: Times New Roman

- + Size: 12

- + Text: Màn hình chính

- + Icon: star.ico

. Form load:

```
Form frm = new Form2();
```

```
frm.ShowDialog();
```

- **Form2**

- . Thiết kế Form như yêu cầu, trong đó form có các thuộc tính sau:

- + AutoSize: True

- + Font: Times New Roman



- + Size: 12
- + Text: Đăng nhập hệ thống
- + Icon: logo.ico

. **Đăng nhập:** Nhấp đúp vào button btnDangnhap, gõ vào đoạn code sau

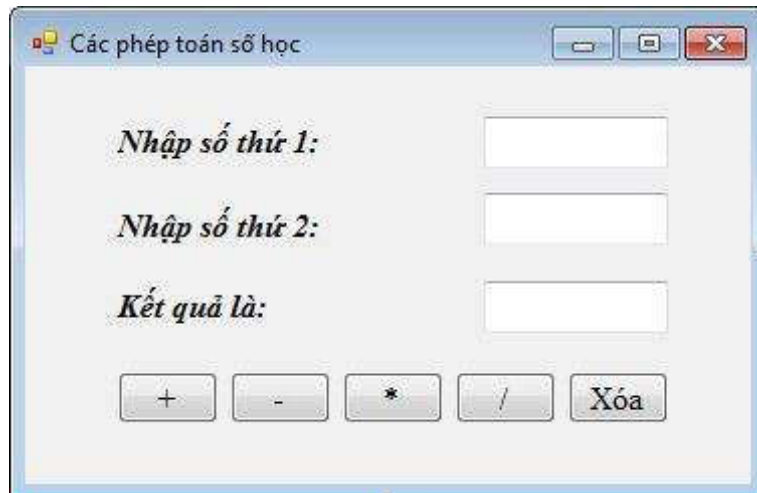
```
if ( (this.txtUser.Text=="teonv") && (this.txtPass.Text=="123") )
    this.Close();
else
{
    MessageBox.Show("Không đúng tên người dùng / mật khẩu !!!", "Thông báo");
    this.txtUser.Focus();
}
```

. **Thoát:** Nhấp đúp vào button btnThoat, gõ vào đoạn code sau

```
DialogResult traloi;
traloi = MessageBox.Show("Chắc không?", "Trả lời",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Question);
if ( traloi == DialogResult.OK)
    Application.Exit();
```

## Bài tập

1. - Thiết kế form như mẫu.



(txtsSo1, txtsSo2, txtKQ, btnCong, btnTru, btnNhan, btnChia, btnXoa)

- Viết chương trình làm các phép toán: cộng, trừ, nhân, chia cho các button btnCong, btnTru, btnNhan, btnChia.
  - **Xóa:** Xóa trắng các TextBox.
  - Thêm vào button **Dừng** (btnDung), khi nhấp vào btnDung thì dừng chương trình.
  - Cải tiến:  
Khi nhấp button btnChia, nếu txtSo2 là 0 thì xuất hiện hộp thông báo lỗi rồi xóa trống txtSo2 và di chuyển con trỏ đến TextBox này.
2. Viết chương trình giải phương trình bậc nhất:  $bx + c = 0$



(txtB, txtC, btnGiai, txtKQ)

- Thêm vào button **Dừng**, khi nhấp vào button này thì dừng chương trình.
- 3. Viết chương trình giải phương trình bậc hai:  $ax^2 + bx + c = 0$
- 4. Viết chương trình nhập vào: họ tên (txtHoTen), nữ (chkNu), điểm văn (txtVan), điểm toán (txtToan), điểm ngoại ngữ (txtNN).
  - Nhấp vào nút **Tính** (btnTinh) thì in ra điểm thấp nhất (txtDTN), điểm kết quả (txtDKQ), xếp loại (txtXL). Biết rằng:
    - + *Điểm thấp nhất*: txtDTN là điểm thấp nhất trong 3 điểm: văn, toán, ngoại ngữ.
    - + *Điểm thêm*: DThem = 0.5 nếu là nữ; DThem = 0 nếu là nam.
    - + *Điểm kết quả*: txtKQ = txtVan \* 2 + txtToan \* 2 + txtNN + DThem
    - + *Xếp loại* theo tiêu chuẩn:
      - . *Giỏi*: nếu txtKQ >= 40 và txtDTN >= 7
      - . *Khá*: nếu txtKQ >= 35 và txtDTN >= 6
      - . *Trung bình*: nếu txtKQ >= 25 và txtDTN >= 5
      - . *Yếu*: các trường hợp còn lại
  - Nhấp vào nút **Xóa** (btnXoa) thì xóa hết các nội dung trong các TextBox.
  - Thêm vào button **Dừng**, khi nhấp vào button này thì dừng chương trình.
- 5. Viết chương trình tạo một ứng dụng gồm:
  - Form1: Màn hình chính có 5 button: **Bài tập 1** (btnBT1), **Bài tập 2** (btnBT2), **Bài tập 3** (btnBT3), **Bài tập 4** (btnBT4), **Thoát** (btnThoat).
  - + Khi nhấp vào Button **Bài Tập 1**: mở Form2
  - + Khi nhấp vào Button **Bài Tập 2**: mở Form3
  - + Khi nhấp vào Button **Bài Tập 3**: mở Form4
  - + Khi nhấp vào Button **Bài Tập 4**: mở Form5
  - Form2: thực hiện bài tập 1. Bổ sung Button **Trở về** để đóng Form2.
  - Form3: thực hiện bài tập 2. Bổ sung Button **Trở về** để đóng Form3.
  - Form4: thực hiện bài tập 3. Bổ sung Button **Trở về** để đóng Form4.
  - Form5: thực hiện bài tập 4. Bổ sung Button **Trở về** để đóng Form5.

--- oOo ---