

Lab 04:

LẬP TRÌNH WINDOWS FORM VỚI CÁC CONTROL NÂNG CAO

(Các control trình bày dữ liệu dưới dạng danh sách)

A. MỤC TIÊU:

- ✓ Sử dụng Visual Studio .NET 2010/2012/2013 tạo ứng dụng dạng Windows Forms với các control nâng cao dạng danh sách.
- ✓ Làm quen với việc sử dụng các control trên form như:
 - NumericUpDown: Hộp chọn tăng giảm giá trị số.
 - Combobox: Hộp chọn 1 giá trị trong danh sách giá trị
 - ListBox: Danh sách các mục chọn, cho phép chọn 1 hoặc nhiều mục
 - GroupBox: Nhóm các đối tượng về cùng nhóm

B. NỘI DUNG:

Bài tập 1:

- ✓ Tạo một ứng dụng Windows Form cơ bản tính tiền công dịch vụ tại phòng nha.
- ✓ Với mỗi khách hàng, các dịch vụ cung cấp gồm: tẩy răng, cạo vôi, chụp hình răng và trám răng. Mỗi loại sẽ có chi phí riêng. Cuối cùng tính tổng các chi phí mà người khách phải trả. Lưu ý: chỉ tính tiền khi phần thông tin tên khách hàng đã được nhập (nếu thông tin này chưa có thì chương trình phát sinh MessageBox cảnh báo).
- ✓ Ứng dụng có giao diện đơn giản như hình 1 bên dưới.

Hình 1: Màn hình chính của ứng dụng

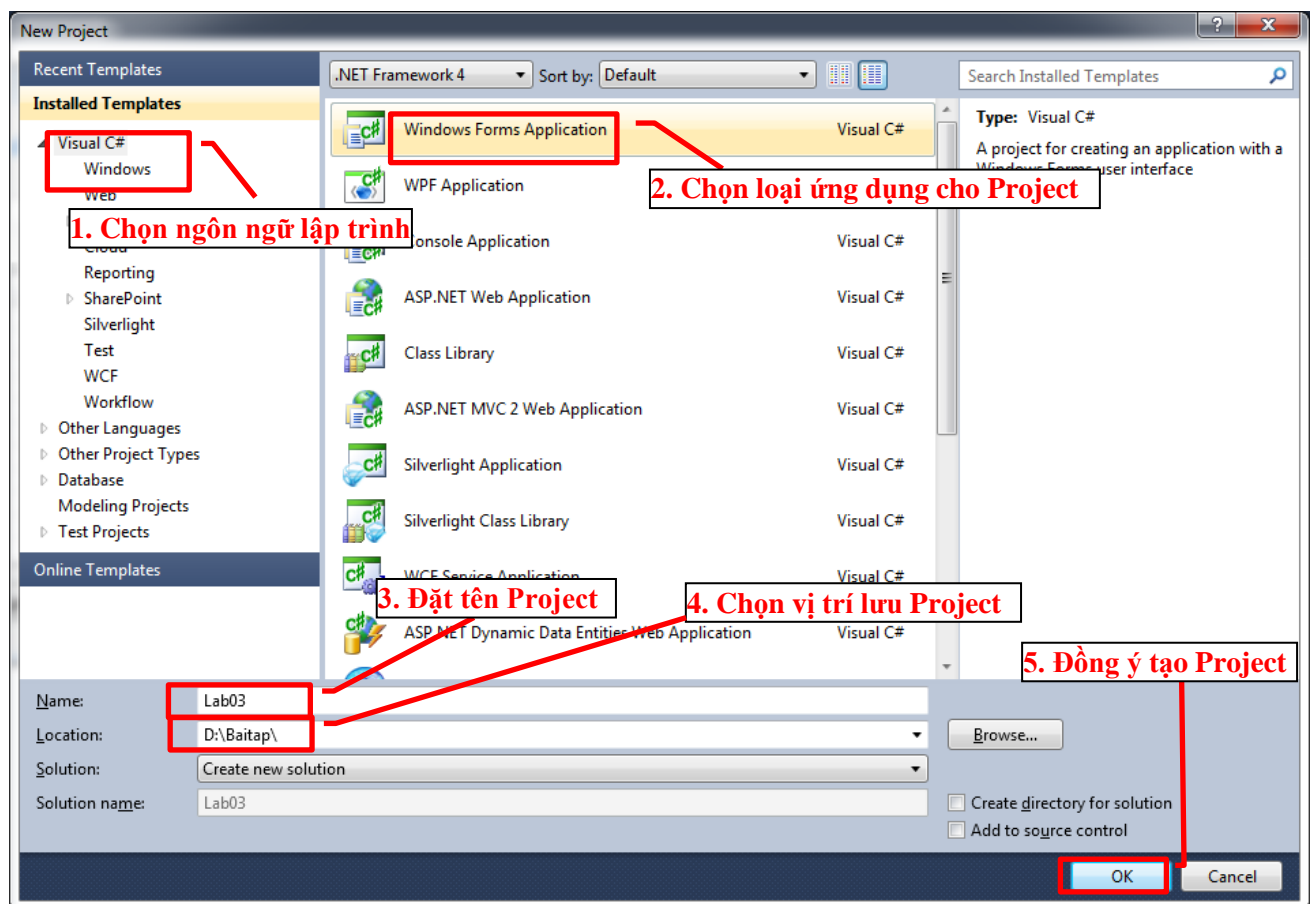
Hướng dẫn:

Bước 1. Tạo ứng dụng Windows Form có tên **DentalPaymentApp** theo các bước sau

- ✓ Tạo project mới (Ctrl+Shift +N),

- ✓ Trong cửa sổ new project chọn Visual C# - Windows
- ✓ Phần template chọn Windows Forms Application
- ✓ Đặt tên project trong phần Name

Xem hình mô tả các bước tạo ứng dụng Windows Form

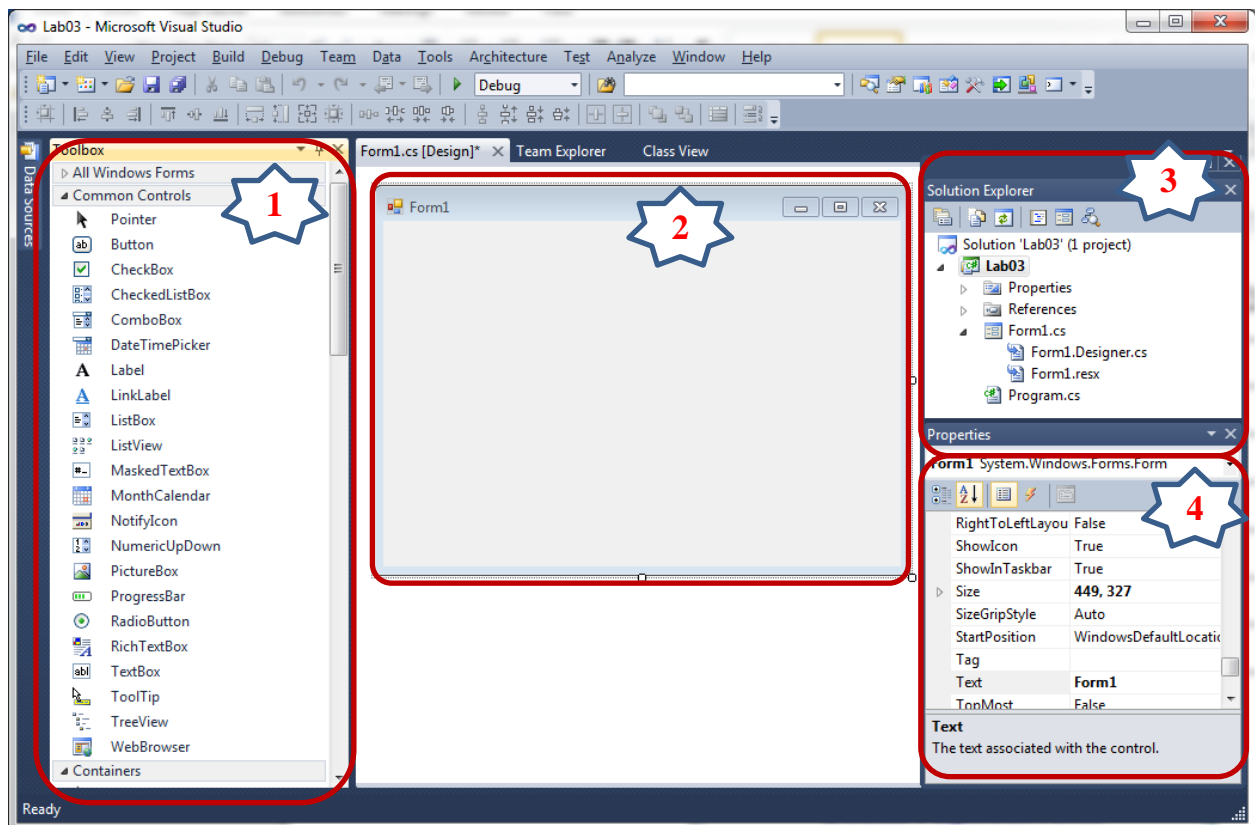


Hình 2: Tạo ứng dụng Windows Form

Bước 2. Sau khi hoàn tất các bước trên VS.NET sẽ phát sinh ra một project Windows Form mẫu, cho phép người lập trình bắt đầu xây dựng các ứng dụng. Giao diện của VS.NET cho ứng dụng vừa tạo có dạng như hình 3 bên dưới.

Màn hình VS.NET cho ứng dụng Windows Form bao gồm các phần cơ bản

- (1): Toolbox: Chứa các control cho phép kéo thả vào Form
- (2): Màn hình thiết kế Form, có thể chuyển sang phần code editor...
- (3): Cửa sổ Solution Explorer: Cho phép người lập trình có thể quản lý các thành phần trong project, hỗ trợ định vị nhanh chóng đến các file mã nguồn.
- (4): Cửa sổ property: cho phép user có thể custom lại các thành phần control trên form như: thiết lập các thuộc tính cho control, form, component, cho phép khai báo trình xử lý sự kiện của các control trên form...



Hình 3: Màn hình VS. NET phục vụ cho việc tạo project Windows Form

Bước 3. Thiết kế form theo mô tả như sau

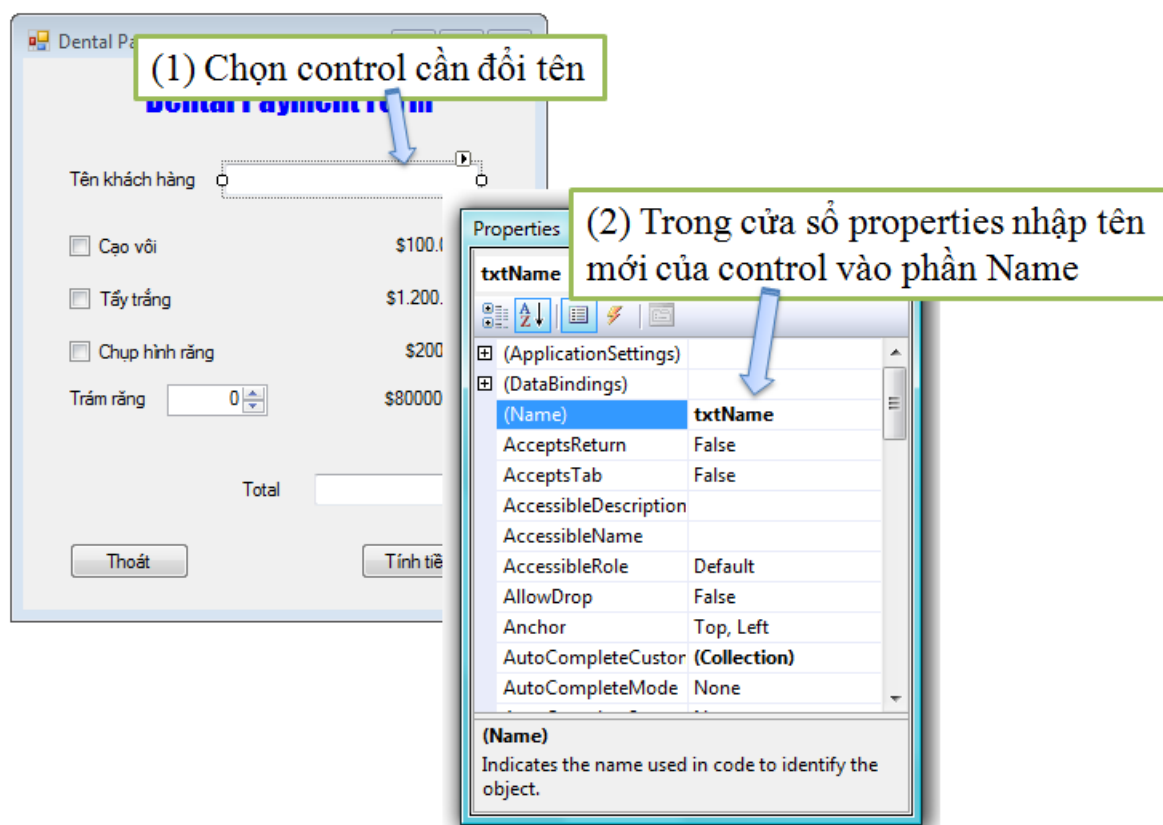
Bảng 1: Mô tả các control trên form

STT	Name	Control	Thiết lập các thuộc tính cho control
1	lblTitle	Label	Text = “Dental Payment Form”, Font = “Impact, Size = 17”, ForeColor = Blue
2	lblName	Label	Text = “Tên khách hàng”
3	txtName	TextBox	
4	chkClean	CheckBox	Text = “Cạo vôi”
5	lblCleanCost	Label	Text=”\$100000”
6	chkWhitening	CheckBox	Text=”Tẩy trắng”
7	lblWhiteningCost	Label	Text=”\$1200000”
8	chkXRay	CheckBox	Text=”Chụp hình răng”
9	lblXRayCost	Label	Text=”\$200000”
10	lblFilling	Label	Text=”Trám răng”
11	numFilling	NumericUpDown	
12	lblFillCost	Label	Text=”\$80000”
13	lblTotal	Label	Text=”Total”
14	txtTotal	TextBox	Enable = False
15	btnExit	Button	Text=”Thoát”
16	btnCalc	Button	Text=”Tính tiền”

Hình 4: Giao diện của Form chương trình

Cách thực hiện:

- Chọn trong ToolBox control tương ứng rồi kéo thả vào vị trí xác định trên form.
- Sử dụng cửa sổ properties thiết lập các thông tin:
 - o Đổi tên của control theo mô tả ở bảng trên
 - o Thiết lập các giá trị cho control theo mô tả ở bảng 1

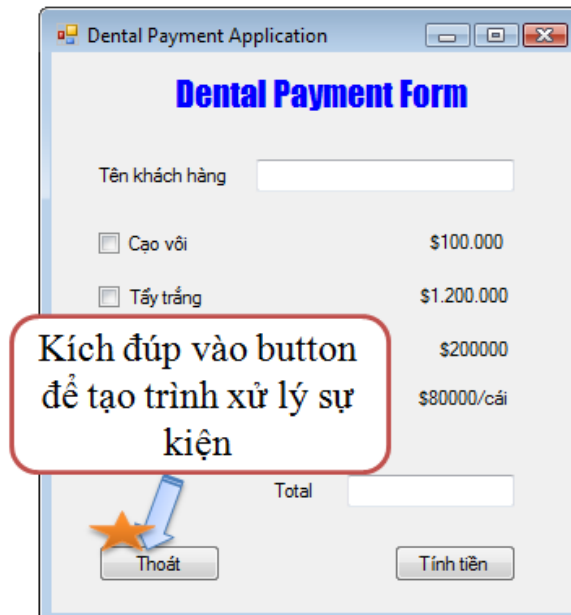


Hình 5: Minh họa việc đổi tên của TextBox trên Form

Bước 4. Tạo trình xử lý sự kiện click cho button “Thoát”: chức năng này khi thực hiện sẽ kết thúc ứng dụng (đóng form lại)

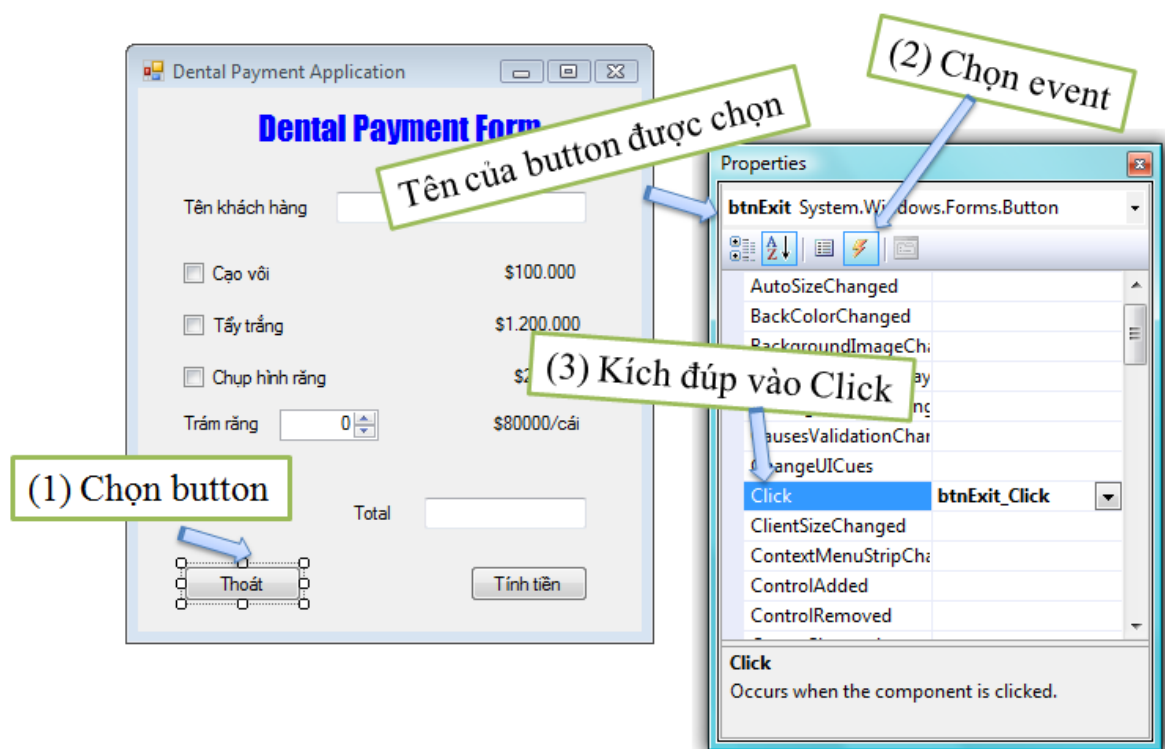
Cách tạo trình xử lý sự kiện: có thể làm theo một trong hai cách

- **Cách 1:** Double Click vào button cần tạo trình xử lý sự kiện trong màn hình Form design view: khi đó VS sẽ tạo trình xử lý sự kiện gắn với sự kiện Click của button “Thoát”



Hình 6: Minh họa cách kích đúp vào button để tạo event handler

- **Cách 2:** chọn button cần tạo trình xử lý, sau đó kích tab event trong cửa sổ Properties, kích đúp vào mục Click trong cửa sổ event.



Hình 7: Minh họa các bước khai báo event handler từ cửa sổ properties của button

Nội dung của trình xử lý sự kiện Click của button btnExit như sau

```
private void btnExit_Click(object sender, EventArgs e)
{
    // đóng form lại ==> kết thúc ứng dụng
    this.Close();
}
```

Bước 5. Tạo chức năng tính tiền, chức năng này được kích hoạt khi button “Tính tiền” được chọn.

Mô tả chức năng **GetPay()** như sau (GetPay()) là phương thức thành viên của lớp Form chính:

- + Kiểm tra xem tên khách hàng có được nhập hay không?
- + Nếu chưa: ➔ xuất thông báo, yêu cầu nhập tên khách.
- + Đã nhập: thực hiện các bước sau

Total = 0

If (cạo vôi) Total += 100.000

If (tẩy trắng) Total += 1.200.000

If (chụp hình răng) Total +=200000

Total += (số răng trám)*80000

Xuất số tiền ra TextBox txtTotal

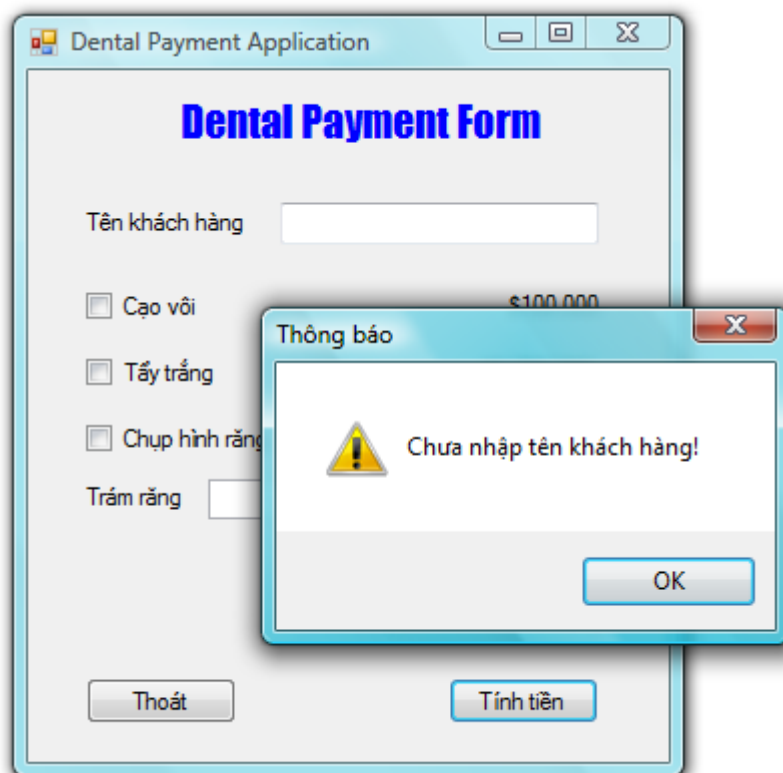
Sinh viên tự viết phương thức GetPay theo mô tả bên trên!

Tạo trình xử lý sự kiện cho button btnCalc rồi trong trình xử lý sự kiện này gọi chức năng GetPay.

```
private void btnCalc_Click(object sender, EventArgs e)
{
    // gọi chức năng tính tiền
    GetPay();
}
```

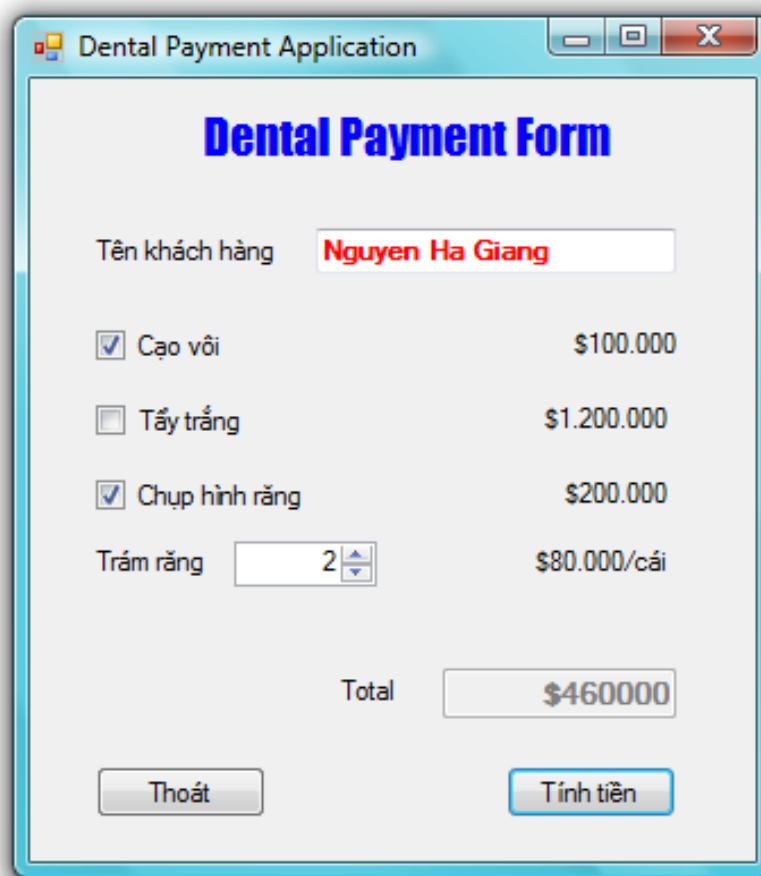
Kết quả chương trình

- ✓ Trường hợp không nhập tên khách hàng: phát sinh message box cảnh báo



Hình 8: Không tính tiền khi chưa nhập tên khách hàng

- ✓ Màn hình sau khi tính tiền cho khách



Hình 9: Màn hình tính tiền cho khách

Bài tập 2: Từ bài tập 1 hãy thực hiện bổ sung

- ✓ Bổ sung một ListBox vào form tính tiền, ListBox này dùng để lưu trữ các thông tin tính tiền của khách hàng. Mỗi thông tin tính tiền sẽ được lưu trên một dòng trong ListBox (một item của listbox). Một item gồm hai thông tin: <tên khách hàng> - <tổng số tiền thanh toán>
- ✓ Bổ sung chức năng Lưu: cho phép lưu trữ các thông tin tính tiền của khách hàng trong một file text. File text này có định dạng mỗi dòng là một thông tin tính tiền: gồm tên khách hàng + tổng số tiền.
- ✓ Bổ sung chức năng Đọc file: Cho phép load thông tin tính tiền khách hàng từ một file lưu trữ
- ✓ Tạo chức năng tùy chọn: Cho phép người tính tiền phòng nha có thể chỉnh lại đơn giá cho từng dịch vụ. Hiện tại ứng dụng trên các đơn giá là fix, ta sẽ cải tiến lại chức năng này. Sinh viên nên tạo file chứa đơn giá từng dịch vụ, tạo form cho phép người quản lý phòng nha có thể hiệu chỉnh lại giá tiền này, lưu lại file đó, và mỗi lần ứng dụng chạy thì đọc file đó để lấy đơn giá.

Bài tập 3: Dành cho sinh viên khá, giỏi.

Các chức năng trong bài tập 2 là dùng file dạng text. Phần nâng cao này sinh viên tìm hiểu file XML và xây dựng lại các file theo định dạng XML phục vụ các yêu cầu đã mô tả (*Sinh viên tìm hiểu phân đọc và lưu trữ file dạng XML trong C#*)

Bài tập 4: Tạo Form frmThaotacso với Listbox như sau :

Yêu cầu:

- Chỉ cho nhập số vào textbox **Nhập số**
- Khi nhập dữ liệu vào textbox **Nhập số** và nhấn Enter(hoặc click vào **Cập Nhật**) thì số mới nhập này được thêm vào Listbox đồng thời dữ liệu trong textbox bị xóa và focus được chuyển về textbox

- Khi nhấn vào các phím chức năng, yêu cầu sẽ được thực hiện trên listBox

Hướng dẫn :

- ✓ **Bước 1: Thiết kế Form với các con tról được chỉ định như sau:**

STT	Tên Control	Loại	Tên hàm xử lý
1	txtSonguyen	TextBox	
2	lstKetqua	ListBox	
4	btCapnhat	Button	btCapnhat_Click
5	btTang2	Button	btTang2_Click
6	btChonchandau	Button	btChonchandau_Click
7	btLecuoil	Button	btLecuoil_Click
8	btXoaphantudangchon	Button	btXoaphantudangchon_Click
9	btXoaphantudau	Button	btXoaphantudau_Click
10	btXoaphantucuoil	Button	btXoaphantudau_Click
11	btKetthuc	Button	btKettuc_Click

Bước 2 : Viết Code cho sự kiện như sau

- ✓ Chỉ cho nhập số vào textbox **Nhập số**

```
private void txtSonguyen_TextChanged(object sender, EventArgs e)
{
    Control ctrSonguyen = (Control)sender;
    if (ctrSonguyen.Text.Length > 0)
        if (!char.IsDigit(ctrSonguyen.Text[ctrSonguyen.Text.Length - 1]))
            this.errSonguyen.SetError(ctrSonguyen, "Vui lòng nhập số nguyên");
        else
            this.errSonguyen.Clear();
}
```

- ✓ Đóng Form có thông báo hỏi đáp.

```
private void frmThaotactrenListbox_FormClosing(object sender, FormClosingEventArgs e)
{
    if (MessageBox.Show("Bạn muốn thoát?", "Thông báo", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.No)
        e.Cancel = true;
}
```

- ✓ Sự liên Click cho các button khác

```
private void btCapnhat_Click(object sender, EventArgs e)
{
    lstKetqua.Items.Add(txtSonguyen.Text);
    txtSonguyen.Text = "";
    txtSonguyen.Focus();
}

private void btTang2_Click(object sender, EventArgs e)
{
    int n = lstKetqua.Items.Count; //Số mục trong lst
    for (int i = 0; i < n; i++)
    {
        int tam = int.Parse(lstKetqua.Items[i].ToString()) + 2;
        lstKetqua.Items.RemoveAt(i);
        lstKetqua.Items.Insert(i, tam.ToString());
    }
}
```

```

    }
}

private void btChonchandau_Click(object sender, EventArgs e)
{
    int n = lstKetqua.Items.Count; //Số mục trong lst
    for (int i = 0; i < n-1; i++)
    {
        if(int.Parse(lstKetqua.Items[i].ToString())%2==0)
        {
            string s = lstKetqua.Items[i].ToString();//Lay gia tri muc gia tri chan
            lstKetqua.SelectedItem = s;//Chon muc co gia tri chan
            break; //Thuc hien duoc so chn dau thi ket thuc vong lap
        }
    }
}

private void btLecuoi_Click(object sender, EventArgs e)
{
    int n = lstKetqua.Items.Count; //Số mục trong lst
    for (int i = n-1; i >=0; i--)
    {
        if (int.Parse(lstKetqua.Items[i].ToString()) % 2 == 1)
        {
            string s = lstKetqua.Items[i].ToString();//Lay gia tri muc gia tri le
            lstKetqua.SelectedItem = s;//Chon muc co gia tri le
            break; //Thuc hien duoc so le dau (tu duoi len) thi ket thuc vong lap
        }
    }
}

private void btXoaphantudangchon_Click(object sender, EventArgs e)
{
    int n = lstKetqua.SelectedItems.Count; //Số mục đang được chọn trong lstKetqua
    for (int i = n-1; i >= 0; i--)
        lstKetqua.Items.Remove(lstKetqua.SelectedItems[i].ToString());
}

private void btXoaphantudau_Click(object sender, EventArgs e)
{
    lstKetqua.Items.Remove(lstKetqua.Items[0].ToString());
}

private void btXoaphantucuoiclick_Click(object sender, EventArgs e)
{
    int n = lstKetqua.Items.Count;
    lstKetqua.Items.Remove(lstKetqua.Items[n-1].ToString());
}

```

Bài tập 5 : Thêm vào project Form frmSinhvien thực hiện thao tác nhập danh sách sinh viên:

Yêu cầu:

- Nút > dùng để di chuyển các mục(còn gọi là item) được chọn từ listbox trái qua listbox phải và ngược lại cho <
- Nút >> dùng để di chuyển tất cả các mục(kể cả không được chọn) từ listbox trái qua listbox phải và ngược lại cho <<
- Nút Xóa dùng để xóa nhưng item được chọn trên textbox trái

Hướng dẫn :

Bước 1: Thiết kế Form với các con tról được chỉ định như sau:

STT	Tên Control	Loại	Tên hàm xử lý
1	txtHoten	TextBox	
2	lstBandau	ListBox	
3	lstKetqua	ListBox	
4	btCapnhat	Button	btCapnhat_Click
5	btTraisangphai	Button	btTraisangphai_Click
6	btTatcatraisangphai	Button	btTatcatraisangphai_Click
7	btPhaisangtrai	Button	btPhaisangtrai_Click
8	btTatcaphaisangtrai	Button	btTatcaphaisangtrai_Click
9	btXoa	Button	btXoa_Click
10	btKetthuc	Button	btKettuc_Click

Bước 2 : Viết Code cho sự kiện các button như sau

```
private void btCapnhat_Click(object sender, EventArgs e)
{
    lstBandau.Items.Add(txtHoten.Text);
    txtHoten.Text = "";
    txtHoten.Focus();
}
private void Traisangphai_Click(object sender, EventArgs e)
{
    int n = lstBandau.SelectedItems.Count; //Tong so muc duoc chon
    for(int i=0;i<=n-1; i++)
        lstKetqua.Items.Add(lstBandau.SelectedItems[i].ToString());
    for (int j = n - 1; j >= 0; j--)
        lstBandau.Items.RemoveAt(j);
}
```

```
private void TatcaTraisangphai_Click(object sender, EventArgs e)
{
    int n = lstBandau.Items.Count;
    for (int i = 0; i <= n - 1; i++)
        lstKetqua.Items.Add(lstBandau.Items[i].ToString());
    for (int j = n - 1; j >= 0; j--)
        lstBandau.Items.RemoveAll();
}
```

Tương tự cho các Button còn lại.

-----**Hết Lab 04** -----