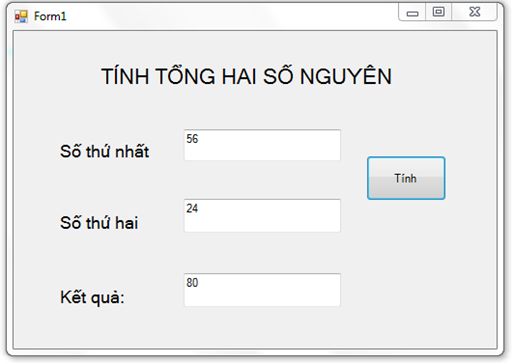
# Bài 1. Viết chương trình nhập 2 số nguyên và tính tổng 2 số đó (Sử dụng Windows Form Application)

* **Điều kiện:** Nếu dữ liệu nhập vào Số thứ nhất và Số thứ hai không phải là số nguyên thì hiển thị hộp thoại báo: “Lỗi! Cần nhập lại”

****

private void btnTinh\_Click(object sender, EventArgs e)

{

if(int.TryParse(txtA.Text,out int a) && int.TryParse(txtB.Text, out int b)){

int tong;

tong = a + b;

lblKetqua.Text=tong.ToString();

}

else

{

MessageBox.Show("Lỗi! Cần nhập lại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

# Bài 2. Viết chương trình nhập vào 2 số A và B, so sánh 2 số A, B và hiển thị với giao diện như sau:

* **Điều kiện:** Nếu dữ liệu nhập vào số A và B không phải là số thì hiển thị hộp thoại báo: “Lỗi! Cần nhập lại”

**A screenshot of a computer

Description automatically generated**

private void button1\_Click(object sender, EventArgs e)

{

if(int.TryParse(txtSoA.Text, out int a)&& int.TryParse(txtSoB.Text, out int b))

{

if (a > b)

{

lblsolon.Text = a.ToString();

lblsobe.Text = b.ToString();

}

else

{

lblsolon.Text = b.ToString();

lblsobe.Text = a.ToString();

}

}

else

{

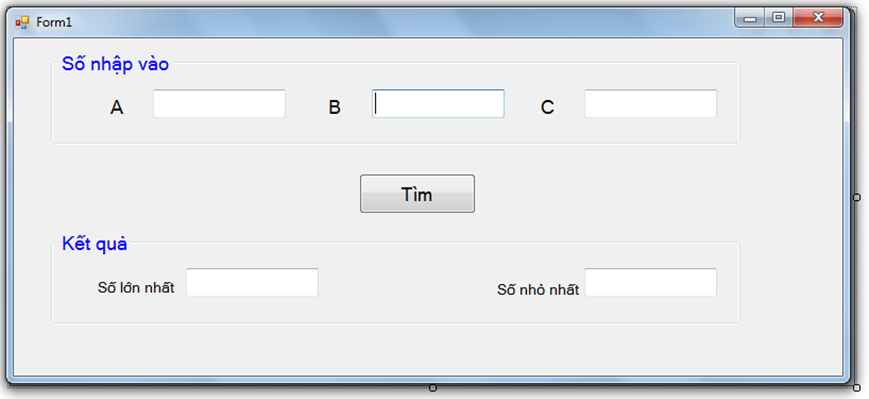
MessageBox.Show("Lỗi! Cần nhập lại","Thông báo",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

# Bài 3. Viết chương trình nhập vào 3 số A , B và C và cho biết số lớn nhất và số nhỏ nhất trong 3 số A, B, C với giao diện như sau:

* **Điều kiện:** Nếu dữ liệu nhập vào số A, B và C không phải là số thì hiển thị hộp thoại báo: “Lỗi! Cần nhập lại”

****

private void btnTim\_Click(object sender, EventArgs e)

{

if (int.TryParse(txtA.Text, out int a) && int.TryParse(txtB.Text, out int b) && int.TryParse(txtC.Text, out int c))

{

if (a > b && a > c)

{

if (b > c)

{

lblSolon.Text = a.ToString();

lblSobe.Text = c.ToString();

}

else

{

lblSolon.Text = a.ToString();

lblSobe.Text = b.ToString();

}

}

else if (b > a && b > c)

{

if (a > c)

{

lblSolon.Text = b.ToString();

lblSobe.Text = c.ToString();

}

else

{

lblSolon.Text = b.ToString();

lblSobe.Text = a.ToString();

}

}

else

{

if (a > b)

{

lblSolon.Text = c.ToString();

lblSobe.Text = b.ToString();

}

else

{

lblSolon.Text = c.ToString();

lblSobe.Text = a.ToString();

}

}

}

else

{

MessageBox.Show("Lỗi", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

# Bài 4: Nhập vào một số nguyên từ 0 đến 9, hiển thị bằng “chữ” với số vừa nhập. Với giao diện như sau:

* **Ví dụ:**

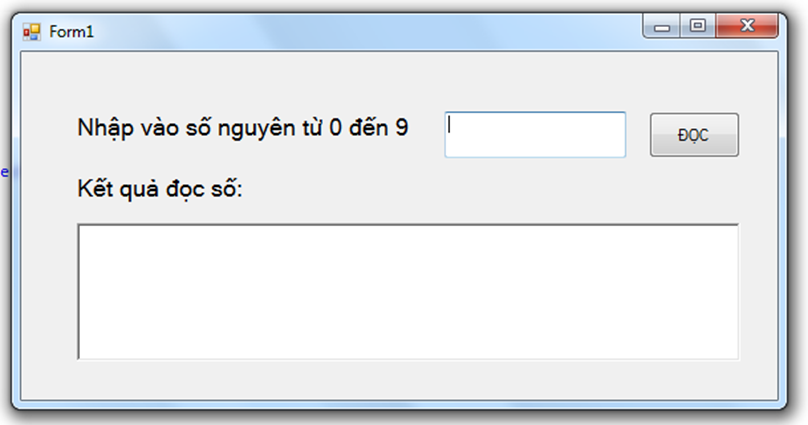
Nhập 1: “Một”

Nhập 2: “Hai”

…

Nhập 9: “Chín”

* **Điều kiện:** Nếu dữ liệu nhập vào không phải là số nguyên từ 0 đến 9 thì hiển thị hộp thoại báo: “Lỗi! Cần nhập lại”



private void button1\_Click(object sender, EventArgs e)

{

if (int.TryParse(textInput.Text, out int soNhap) && soNhap >= 0 && soNhap <= 9)

{

if (int.Parse(textInput.Text) == 1)

{

lblKetqua.Text = "Một";

}

if (int.Parse(textInput.Text) == 2)

{

lblKetqua.Text = "Hai";

}

if (int.Parse(textInput.Text) == 3)

{

lblKetqua.Text = "Ba";

}

if (int.Parse(textInput.Text) == 4)

{

lblKetqua.Text = "Bốn";

}

if (int.Parse(textInput.Text) == 5)

{

lblKetqua.Text = "Năm";

}

if (int.Parse(textInput.Text) == 6)

{

lblKetqua.Text = "Sáu";

}

if (int.Parse(textInput.Text) == 7)

{

lblKetqua.Text = "Bảy";

}

if (int.Parse(textInput.Text) == 8)

{

lblKetqua.Text = "Tám";

}

if (int.Parse(textInput.Text) == 9)

{

lblKetqua.Text = "Chín";

}

}

else {

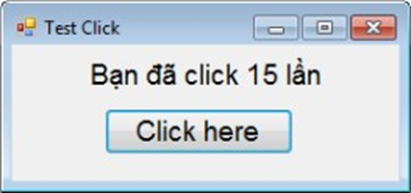
MessageBox.Show("Lỗi!Cần nhập lại","Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

# Bài 5: Viết chương trình gồm 1 form trong đó có một button và 1 label hiển thị số lần khi nhấn button đó.



private void btnClick\_Click(object sender, EventArgs e)

{

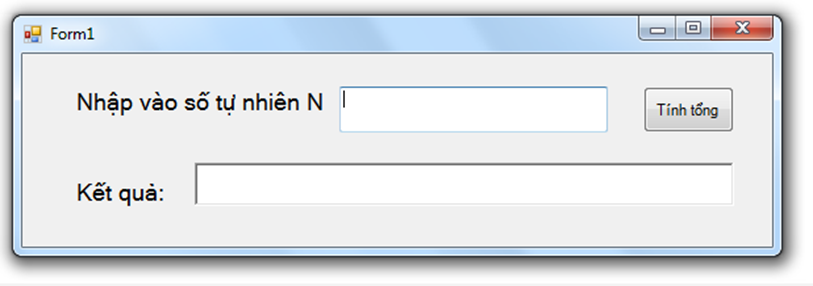
dem++;

lblKetqua.Text = $"Bạn đã click {dem} lần";

}

**Bài 6:** Viết chương trình tính tổng S = 1 + 2 + 3 + 4 + ……. + N với N là một số tự nhiên được nhập từ bàn phím có giao diện như sau:

* **Điều kiện:** Nếu dữ liệu nhập vào không phải là số tự nhiên thì hiển thị hộp thoạibáo: “Lỗi! Cần nhập lại”



private void btnTinh\_Click(object sender, EventArgs e)

{

if(int.TryParse(txtNhap.Text,out int a))

{

int sum = 0;

for(int i = 1; i <= a; i++)

{

sum += i;

}

lblKetqua.Text = sum.ToString();

}

else

{

MessageBox.Show("Lỗi","Thông báo",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

# Bài 7: Viết chương trình tính tổng: S(X,N) = X1 + X2 + X3 + X4 + …… + XN với X, N là số tự nhiên nhập từ bàn phím có giao diện như sau:

* **Điều kiện:** Nếu dữ liệu nhập vào hai số X và N không phải là số tự nhiên thì hiển thị hộp thoại báo: “Lỗi! Cần nhập lại”

A screenshot of a computer

Description automatically generated

private void btnTinh\_Click(object sender, EventArgs e)

{

if(double.TryParse(txtN.Text,out double a )&& double.TryParse(txtX.Text, out double b))

{

double tong = 0;

for(int i = 1; i <= a; i++)

{

tong += Math.Pow(b, i);

}

lblKetqua.Text = tong.ToString();

}

else

{

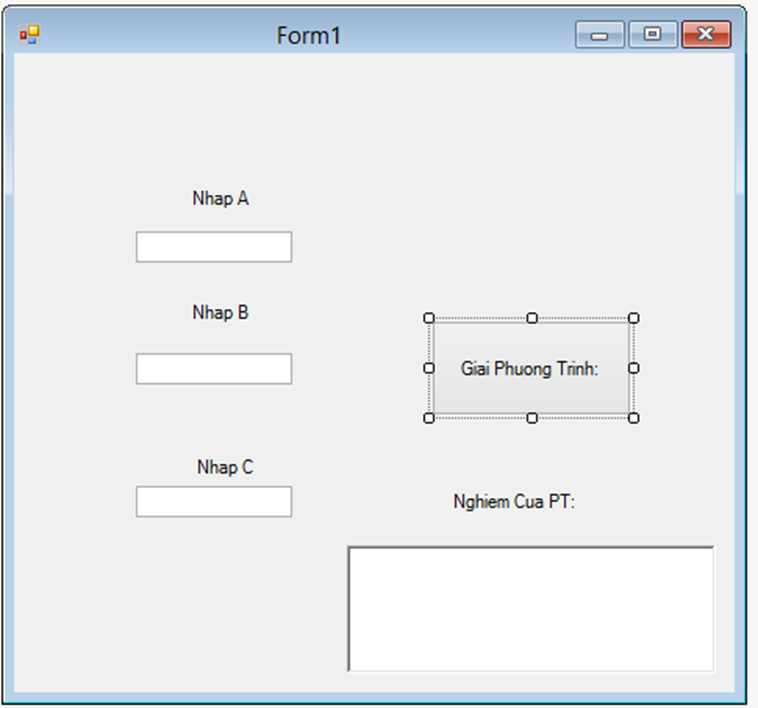
MessageBox.Show("lỗi","Thông báo",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

# *Bài 8*: Viết chương trình giải hệ phương trình bậc 2 F(x) = Ax2 + Bx + C với giao diện như sau:

* **Điều kiện:** A ≠ 0



private void btnTinh\_Click(object sender, EventArgs e)

{

if(double.TryParse(txtA.Text,out double a)

&& double.TryParse(txtB.Text, out double b) && double.TryParse(txtC.Text, out double c)

&& double.Parse(txtA.Text) != 0)

{

double denta = Math.Pow(b, 2) - 4 \* a \* c;

if (denta<0)

{

lblKetqua.Text = "PT Vo nghiem";

}else if (denta == 0)

{

float x = (float)(b / (2 \* a));

lblKetqua.Text = $"PT có nghiệm kép là: {x}";

}

else

{

float x1 = (float)((-b + Math.Sqrt(denta)) / (2 \* a));

float x2 = (float)((-b - Math.Sqrt(denta)) / (2 \* a));

lblKetqua.Text = $"Nghiệm thứ nhất là: {x1} Nghiệm thứ hai là: {x2}";

}

}

else

{

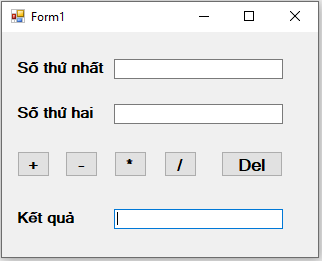
MessageBox.Show("Lỗi","Thông báo",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

# Bài 9: Tạo ứng dụng máy tính bỏ túi với giao diện như sau:

* **Điều kiện:** Nếu dữ liệu nhập vào Số thứ nhất và Số thứ hai không phải là số thì hiển thị hộp thoại báo: “Lỗi! Cần nhập lại”



private void btnDelete\_Click(object sender, EventArgs e)

{

txtA.Text = "";

txtB.Text = "";

lblKetqua.Text = "";

}

private void btnCong\_Click(object sender, EventArgs e)

{

if(double.TryParse(txtA.Text,out double a)&&double.TryParse(txtB.Text, out double b))

{

double tong;

tong = a + b;

lblKetqua.Text = tong.ToString();

}

else

{

MessageBox.Show("lỗi", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnTru\_Click(object sender, EventArgs e)

{

if (double.TryParse(txtA.Text, out double a) && double.TryParse(txtB.Text, out double b))

{

double hieu;

hieu = a - b;

lblKetqua.Text = hieu.ToString();

}

else

{

MessageBox.Show("lỗi", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnNhan\_Click(object sender, EventArgs e)

{

if (double.TryParse(txtA.Text, out double a) && double.TryParse(txtB.Text, out double b))

{

double tich;

tich = a \* b;

lblKetqua.Text = tich.ToString();

}

else

{

MessageBox.Show("lỗi", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnChia\_Click(object sender, EventArgs e)

{

if (double.TryParse(txtA.Text, out double a) && double.TryParse(txtB.Text, out double b))

{

double chia;

chia = (float)(a/ b);

lblKetqua.Text = chia.ToString();

}

else

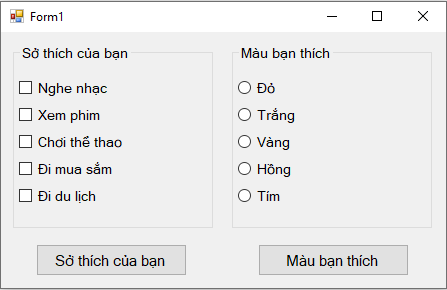
{

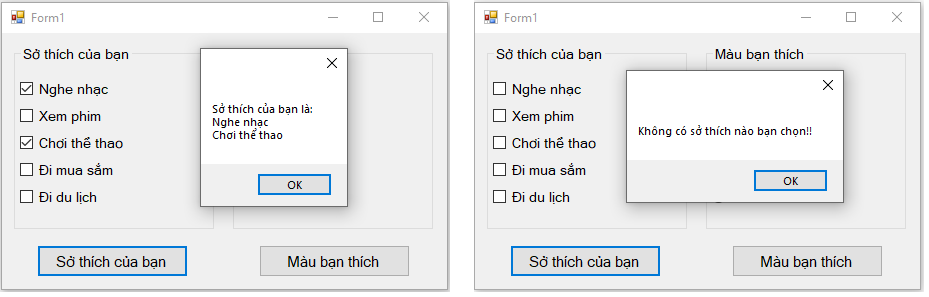
MessageBox.Show("lỗi", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

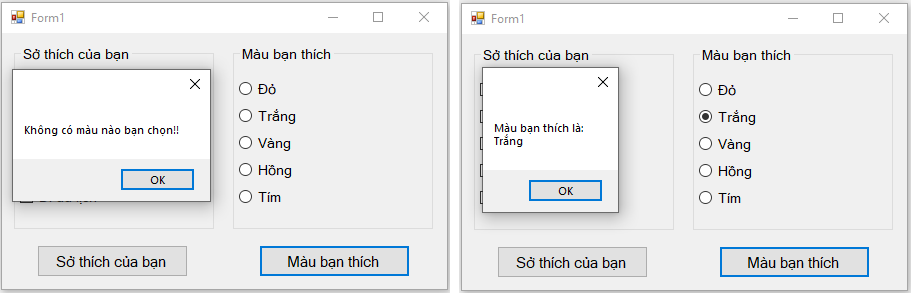
}

}

# Bài 10. Lập trình ra ứng dụng sau:







public partial class Form1 : Form

{

private string sothich;

private string mau;

public Form1()

{

InitializeComponent();

sothich = "";

mau = "";

}

private void btnSothch\_Click(object sender, EventArgs e)

{

CheckedListBox.CheckedItemCollection items = chkListBox.CheckedItems;

string strItem = "";

foreach (string s in items)

{

strItem += s +"\n";

}

MessageBox.Show(strItem);

}

private void btnmau\_Click(object sender, EventArgs e)

{

if (radDo.Checked)

{

mau = "Đỏ";

}

else if (radTrang.Checked)

{

mau = "Trắng";

}

else if (radHong.Checked)

{

mau = "Hồng";

}

else if (radTim.Checked)

{

mau = "Tím";

}

else if (radVang.Checked)

{

mau = "Vàng";

}

MessageBox.Show ("Màu sắc yêu thích: " + mau,"Thông báo");

}

private void checkedListBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

# Bài 11. Lập trình ra ứng dụng khai báo điện tử sau:

* **Yêu cầu:**

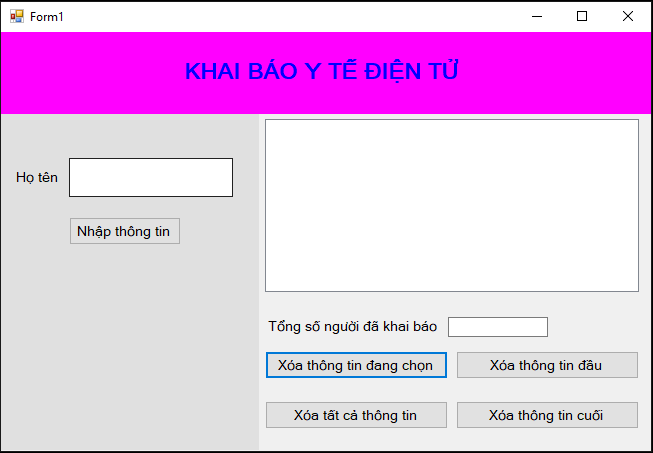
+ Button "**Nhập thông tin**": Khi người dùng nhập xong Họ tên, sau đó nhấn nút Button " **Nhập thông tin** " thì trong ListView sẽ chèn một dòng vào cuối với Họ tên vừa nhập.

+ Button "**Xóa thông tin đang chọn**": Khi người dùng chọn vào Họ tên cần xóa và nhấn nút "**Xóa thông tin đang chọn**" thì sẽ xóa Họ tên vừa chọn trong ListView.

+ Button "**Xóa thông tin đầu**": Khi người dùng nhấn nút "**Xóa thông tin đang chọn**" thì sẽ xóa Họ tên đầu tiên trong ListView.

+ Button "**Xóa thông tin cuối**": Khi người dùng nhấn nút "**Xóa thông tin cuối**" thì sẽ xóa Họ tên cuối cùng trong ListView.

+ Button "**Xóa tất cả thông tin** ": Khi người dùng nhấn nút "**Xóa tất cả thông tin**" thì sẽ xóa tất cả Họ tên trong ListView.



A screenshot of a computer

Description automatically generated

private void btnNhapThongTin\_Click(object sender, EventArgs e)

{

if (!String.IsNullOrEmpty(txtHoTen.Text))

{

listView1.Items.Add(txtHoTen.Text);

txtHoTen.Clear();

UpdateTongSoNguoi();

}

}

private void btnXoaChon\_Click(object sender, EventArgs e)

{

if (listView1.SelectedItems.Count > 0)

{

listView1.Items.Remove(listView1.SelectedItems[0]);

UpdateTongSoNguoi();

}

}

private void btnXoaDau\_Click(object sender, EventArgs e)

{

if (listView1.Items.Count > 0)

{

listView1.Items.RemoveAt(0);

UpdateTongSoNguoi();

}

}

private void btnXoaCuoi\_Click(object sender, EventArgs e)

{

if (listView1.Items.Count > 0)

{ listView1.Items.RemoveAt(listView1.Items.Count - 1);

UpdateTongSoNguoi();

}

}

private void btnXoaTatCa\_Click(object sender, EventArgs e)

{

listView1.Items.Clear();

UpdateTongSoNguoi();

}

private void UpdateTongSoNguoi()

{

txtTongSoNguoi.Text = listView1.Items.Count.ToString();

}

# Bài 12. Lập trình ra ứng dụng đăng nhập với giao diện sau:

**A screenshot of a computer screen

Description automatically generated**

* **Yêu cầu:**

+ Thiết kế giao diện cho Form giống như Form mẫu ở trên.

+ Khi rê chuột vào ô TextBox "**Tên đăng nhập**" thì hiển thị dòng ghi chú "*Chỉ được nhập ký tự a-z và 0-9*".

+ Khi rê chuột vào ô TextBox "**Mật khẩu**" thì hiển thị dòng ghi chú "*Chỉ được nhập ký tự từ 0-9*".

+ Khi load Form thì nội dung trong ô TextBox "**Mật khẩu**" sẽ được mã hóa thành dấu \*, khi Tick vào ô "**Hiển thị mật khẩu**" thì mật khẩu sẽ được hiển thị.

+ Xử lí nút Button "**Đăng nhập**", nếu hai ô TextBox ở trên không để trống thì thông báo đăng nhập thành công, ngược lại nếu một trong hai để trống thì thông báo điền đầy đủ thông tin.

+ Xử lý nút Button "**Thoát**" để thoát khỏi chương trình.

private void btnDangnhap\_Click(object sender, EventArgs e)

{

string ten = txtDangnhap.Text;

string mk = txtMatkhau.Text;

if(string.IsNullOrEmpty(ten)&& string.IsNullOrEmpty(mk))

{

MessageBox.Show("Không được để trống tên và mật khẩu","Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

MessageBox.Show("Đăng nhập thành công","Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void chkMatkhau\_CheckedChanged(object sender, EventArgs e)

{

txtMatkhau.PasswordChar = chkMatkhau.Checked ? '\0' : '\*';

}

# Bài 13. Lập trình ra ứng dụng quản lý phòng ban với giao diện như sau:

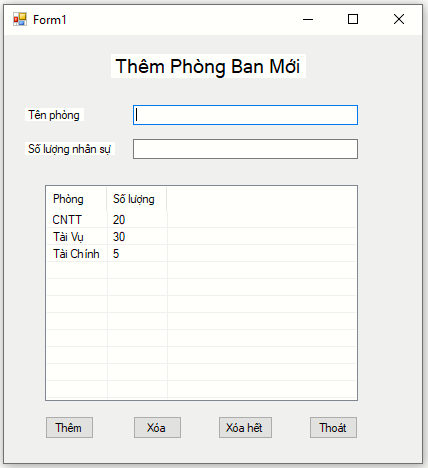
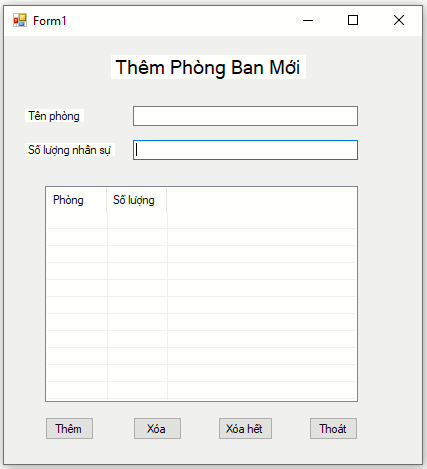
* **Yêu cầu:**

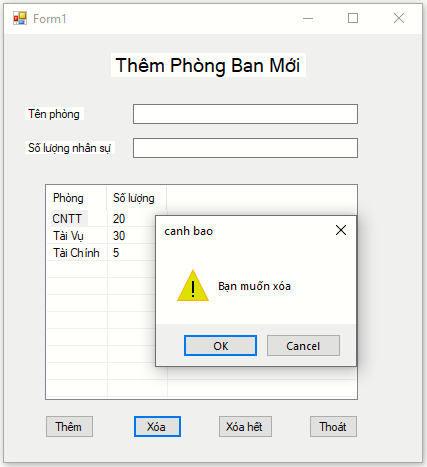
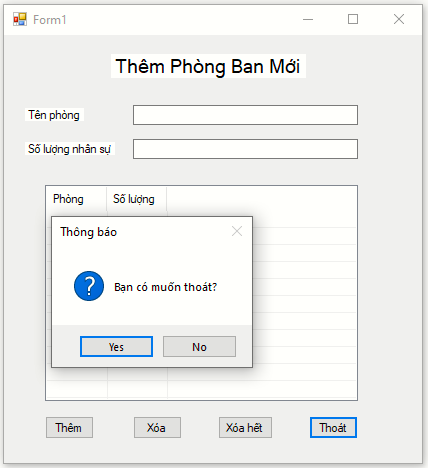
+ Button "**Thêm**": Khi người dùng nhập xong tên phòng và số lượng nhân sự, sau đó nhấn nút Button "**Thêm**" thì trong ListView sẽ chèn một dòng vào cuối với tên phòng và số lượng vừa nhập.

+ Button "**Xóa**": Khi người dùng chọn vào tên phòng ban cần xóa và nhấn nút xóa thì sẽ xóa phòng vừa chọn trong ListView. Nếu chọn OK thì thực thi lệnh xóa, nếu chọn Cacel thì hủy bỏ lệnh xóa.

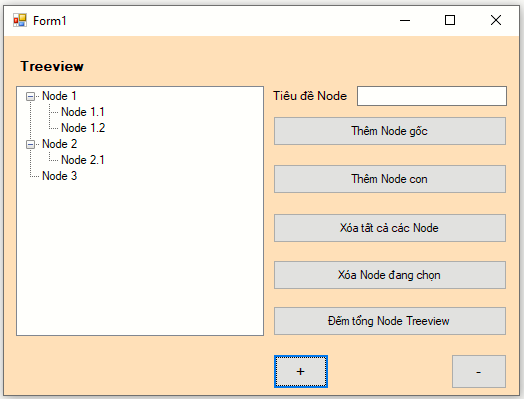
+ Button "**Xóa hết**": Khi người dùng chọn vào Button sẽ xóa tất cả các Item trong ListView. Nếu chọn **OK** thì thực thi lệnh xóa hết, nếu chọn **Cancel** thì hủy bỏ lệnh xóa hết.

+ Button "**Thoát**": Thông báo "**Bạn có muốn thoát?**" xác nhận người dùng muốn đóng Form hiện hành, nếu chọn **Yes** thì kết thúc chương trình thực thi, nếu chọn **No** thì hủy bỏ lệnh kết thúc.





# Bài 14. Lập trình ứng dụng quản lý thư mục như sau:



**A screenshot of a computer

Description automatically generated**

Tạo hệ thống cây TreeView gồm:

+ Button "**Thêm Node gốc**": Thêm một Node gốc.

+ Button "**Thêm Node con**": Thêm một Node con.

+ Button "**Xóa tất cả các Node**": Hủy tất cả các Node trong TreeView.

+ Butotn "**Xóa Node được chọn**": Xóa Node được chọn khỏi TreeView.

+ Button "**-**": Thu hẹp lại vị trí chọn - thu hẹp lại các Node là con của Node.

+ Button "**+**": Mở rộng tại vị trí chọn - mở rộng các Node là con của Node.

+ Button "**Đếm tổng Node TreeView**": Đếm tổng Node có trên TreeView. Khi người dùng ấn vào nút này sẽ hiển thị một cửa sổ nhỏ với thông tin tổng số Node có trên Treeview.

namespace Bài\_14

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

// Thêm một Node gốc

TreeNode rootNode = new TreeNode(textBox1.Text);

treeView1.Nodes.Add(rootNode);

}

private void button2\_Click(object sender, EventArgs e)

{

// Them một node con vào node được chọn

TreeNode selectedNode = treeView1.SelectedNode;

if (selectedNode != null)

{

TreeNode childNode = new TreeNode(textBox1.Text);

selectedNode.Nodes.Add(childNode);

selectedNode.Expand(); // Mở rộng node cha để hiển thị node con

}

else

{

MessageBox.Show("Vui lòng chọn một Node để thêm Node con.",

"Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button3\_Click(object sender, EventArgs e)

{

//Xóa tất cả các Node

treeView1.Nodes.Clear();

}

private void button4\_Click(object sender, EventArgs e)

{

TreeNode selectedNode = treeView1.SelectedNode;

if (selectedNode != null)

{

selectedNode.Remove();

}

else

{

MessageBox.Show("Vui lòng chọn một Node để xóa.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button5\_Click(object sender, EventArgs e)

{

//Đếm tổng số Ndoe trên treeview

int totalNodes = CountNodes(treeView1.Nodes);

MessageBox.Show($"Tổng số Node trên Treeview: {totalNodes}",

"Thông tin", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private int CountNodes(TreeNodeCollection nodes)

{

int cound = nodes.Count;

foreach (TreeNode node in nodes)

{

cound += CountNodes(node.Nodes);

}

return cound;

}

private void button6\_Click(object sender, EventArgs e)

{

// Mở rộng vị trí

TreeNode selectedNode = treeView1.SelectedNode;

if (selectedNode != null)

{

selectedNode.Expand();

}

}

private void button7\_Click(object sender, EventArgs e)

{

// Thu hepj vị trí

TreeNode selectedNode = treeView1.SelectedNode;

if (selectedNode != null)

{

selectedNode.Collapse();

}

}

}

}

# Bài 15. Lập trình ra ứng dụng đăng ký khóa học trực tuyến sau:

A screenshot of a computer

Description automatically generated

Chương trình có các chức năng như sau:

+ TextBox "**Họ Tên**": Chỉ cho phép nhập ký tự hoa hoặc thường, cho phép nhập khoảng trắng và cho phép xóa lùi.

+ TextBox "**Số Điện Thoại**": Chỉ cho phép nhập số và được xóa lùi.

+ DateTimePicker "**Thời Gian**": Chỉ cho phép chọn thời gian đăng ký học lớn hơn thời gian hiện tại.

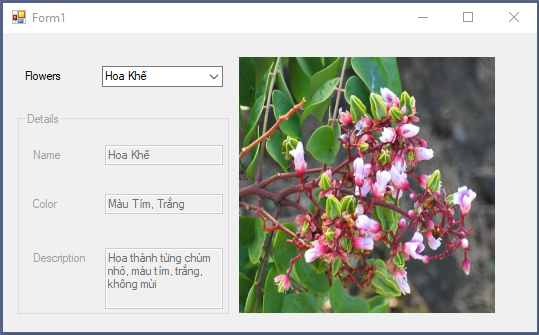
+ Button "**Đăng ký**": Hiển thị hộp thoại bao gồm các nội dung trong ô TextBox và thời gian được chọn trong DateTimePicker.

+ Button "**Thoát**": Đóng chương trình.

+ Nếu nhập sai quy định của các trường thông tin trên thì hiển thị hộp thoại thông báo: “Yêu cầu nhập lại”

# Bài 16: Sử dụng comboBox để lưu trữ 1 danh sách các loài hoa. Khi chọn tên 1 loài hoa bất kì sẽ hiển thị thông tin về hoa như: tên, màu sắc, và mô tả về hoa, cộng bức ảnh của loài hoa đó.

**Yêu cầu:** Sử dụng pictureBox hiển thị ảnh, ImageListBox lưu trữ ảnh, GroupBox quản lý thông tin về hoa. Khi chọn hoa thì các thuộc tính mô tả về hoa được ẩn đi.



# Bài 17. Sử dụng TreeView hiển thị danh sách các phần mềm Antivirut theo dạng "Cây" thư mục. Listview hiển thị thông tin chi tiết về sản phẩm khi được chọn ở bên TreeView.

* **Yêu cầu:**

+ 1 NumbericUpDown Control dùng để chọn số lượng sản phẩm muốn mua (**Quantity**)

+ 1 DomainUpDown Control để lựa chọn thời gian sử dụng sản phẩm (**Package**).

+ Khi muốn mua sản phẩm, khách hàng cần chọn sản phẩm muốn mua bên **ListApplications** sau đó, khách hàng mới được tiếp tục chọn số lượng sản phẩm và thời gian sử dụng, khi đó **Total Amount** sẽ hiển thị giá tiền các sản phẩm đã chọn.

+ Khi nhấn button "**Submit**" sẽ hiển thị một hộp thoại thông tin chi tiết về sản phẩm mà khách hàng chọn mua.

A screenshot of a software program

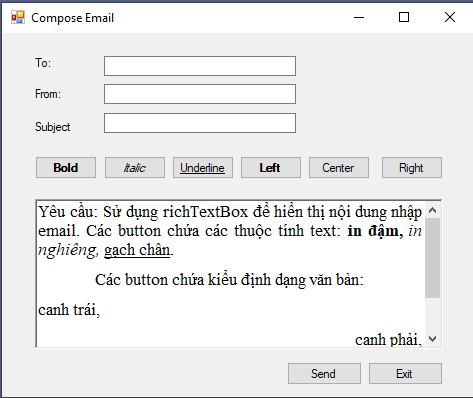
Description automatically generated

A screen shot of a computer

Description automatically generated

# Bài 18. Giả lập 1 chuơng trình gửi email đơn giản.

* **Yêu cầu:** Sử dụng RichTextBox để hiển thị nội dung nhập email. Các button chứa các thuộc tính text: in đậm, in nghiêng, gạch chân. Các button chứa kiểu định dạng văn bản: căn trái, căn phải, căn giữa.



# Bài 19. Giả lập 1 ứng dụng bán hàng đơn giản trong đó bạn là nhân viên bán hàng với giao diện như sau:

* **Yêu cầu:**

+ Sử dụng các textBox để nhập thông tin: 1 textBox mã đơn hàng (**Order No**), 1 textBox tên khách hàng (**Customer Name**).

+ Tạo 1 DateTimePicker để hiển thị ngày làm hóa đơn.

+ 1 groupBox chứa các thông tin về tên sản phẩm (**Name**), đơn giá (**Rate**), và số lượng sản phẩm (**Quanity**) mà hàng khách mua.

+ Tạo 1 button tính tổng tiền (**Calculate**) khi khách hàng nhấn nút **Calculate**.

+ Tạo 1 label để hiển thị tổng số tiền của các sản phẩm đã chọn (**Total Amount**)

+ Tạo 1 button **Report** sẽ báo cáo dữ liệu vừa nhập sang 1 form mới.Form mới này cài đặt 1 MonthCalendar để hiển thị ngày tháng. Khi chọn tháng (hoặc ngày) khác thì sẽ hiển thị hàng đã bán trong tháng (ngày đó).

+ Tạo 1 button **Exit** để thoát khỏi ứng dụng.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Bài 20. Giả lập 1 ứng dụng thanh toán trong đó bạn là nhân viên lập hóa đơn thanh toán với giao diện như sau:

* **Yêu cầu:**

+ Sử dụng các textBox : 1 textBox để nhập tên khách hàng, 1 textbox để hiện thị tổng tiền dịch vụ đã sử dụng.

+ 1 groupBox chứa các thông tin về tên về các dịch vụ đã sử dụng bao gồm các checkbox: Cạo vôi, Tẩy trắng, Chụp hình răng và Trám răng tương ứng bên cạnh là giá của mỗi dịch vụ: 100.000, 1200000, 200000, 800000 VNĐ

+ Tạo 1 button tính tổng tiền (Tính tiền) khi khách hàng nhấn nút tính tiền. Hệ thống sẽ tự động tính tiền theo giá dịch vụ đã cho ở trên và tính tổng tiền dịch vụ (Lưu ý: button chỉ hoạt động khi đã nhập tên Khách hàng).

+ Tạo một NumericUpDown để hiện thị số lượng răng trám (số lượng phải không được nhỏ hơn không và lớn hơn 10)

