Bài Lab dành cho các cháu lớp 1, đầu cấp làm quen với lập trình

**Tên: Trần Phạm Thiên Ân**

**11CNPM2**

LAB 2:

1. **Mục tiêu**
   * Hướng dẫn sinh viên xây dựng chương trình Windows Form đơn giản.
   * Hướng dẫn làm quen môi trường lập trình C# với phần mềm Microsoft

Visual Studio 2022.

* + Hướng dẫn sinh viên tập làm quen mới soạn thảo, chỉnh sửa mã nguồn, biên dịch, gỡ lỗi (debug), thực thi chương trình.
  + Làm quen với các control cơ bản trên Winform như:

+ Label: hiển thị thông tin.

+ TextBox: nhập thông tin dữ liệu.

+ Button: cho phép người dùng click để thực hiện chức năng nào đó.

+ CheckBox: cho phép người dùng chọn một hoặc nhiều tùy chọn. +

RadioButton: cho phép người dùng chọn một tùy chọn.

+ MessageBox: hiển thị thông tin đến người dùng theo dạng hộp thoại. + 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.

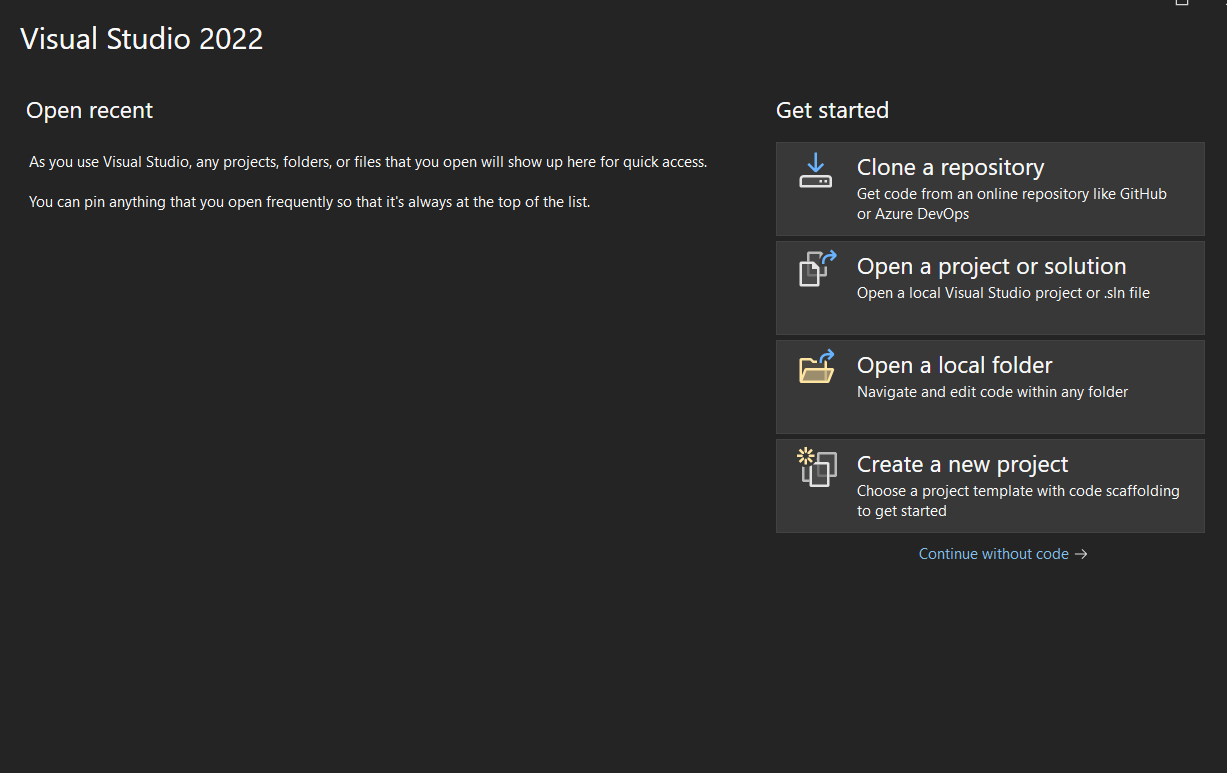
* + Làm quen khai báo và xử lý các sự kiện với các control khác nhau.
  + Làm quen với cách đặt tên biến lập trình của các control.

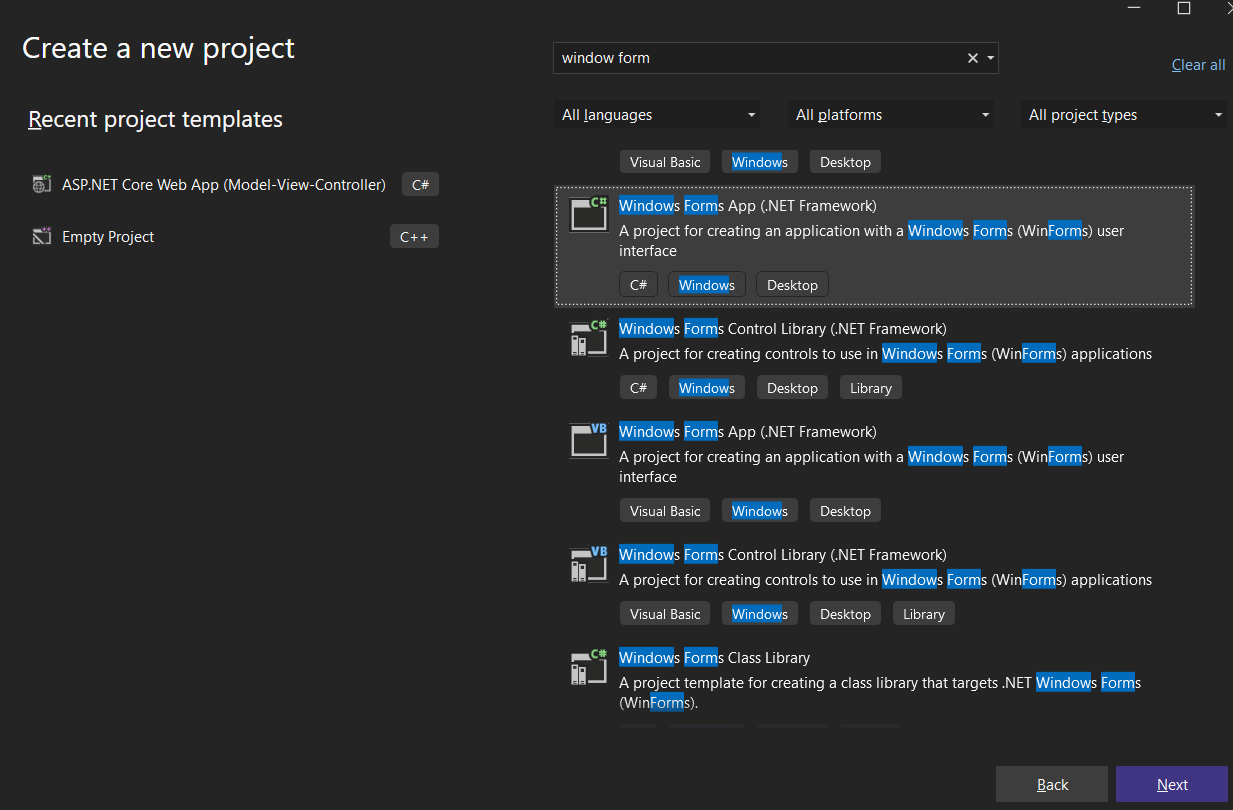
1. **Nội dung**

# 1. Tạo Project Windows Forms Application trên Microsoft visualstudio 2022

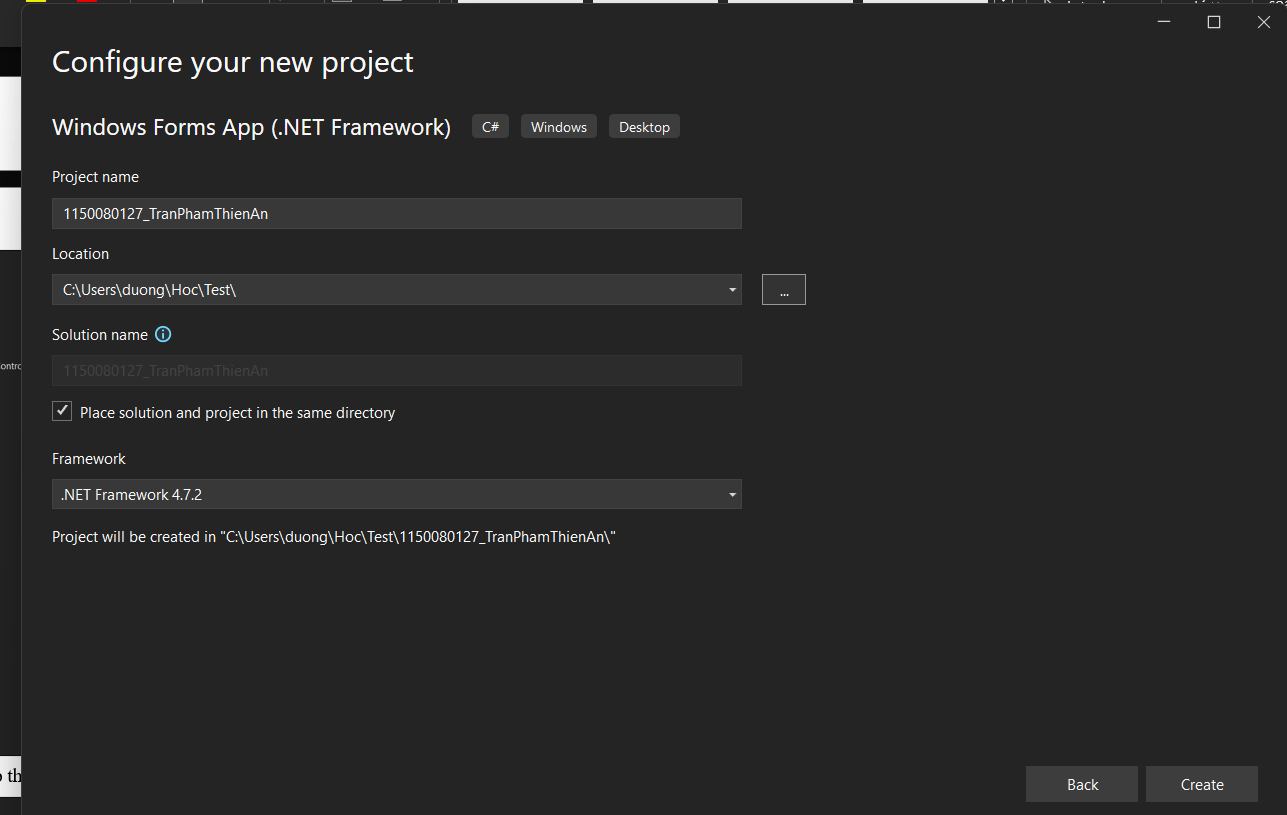
**Bước 1:** Khởi động chương trình visual studio 2022. Hình ảnh dưới đây là màn hình khởi động của ứng dụng:

Bài Lab dành cho các cháu lớp 1, đầu cấp làm quen với lập trình

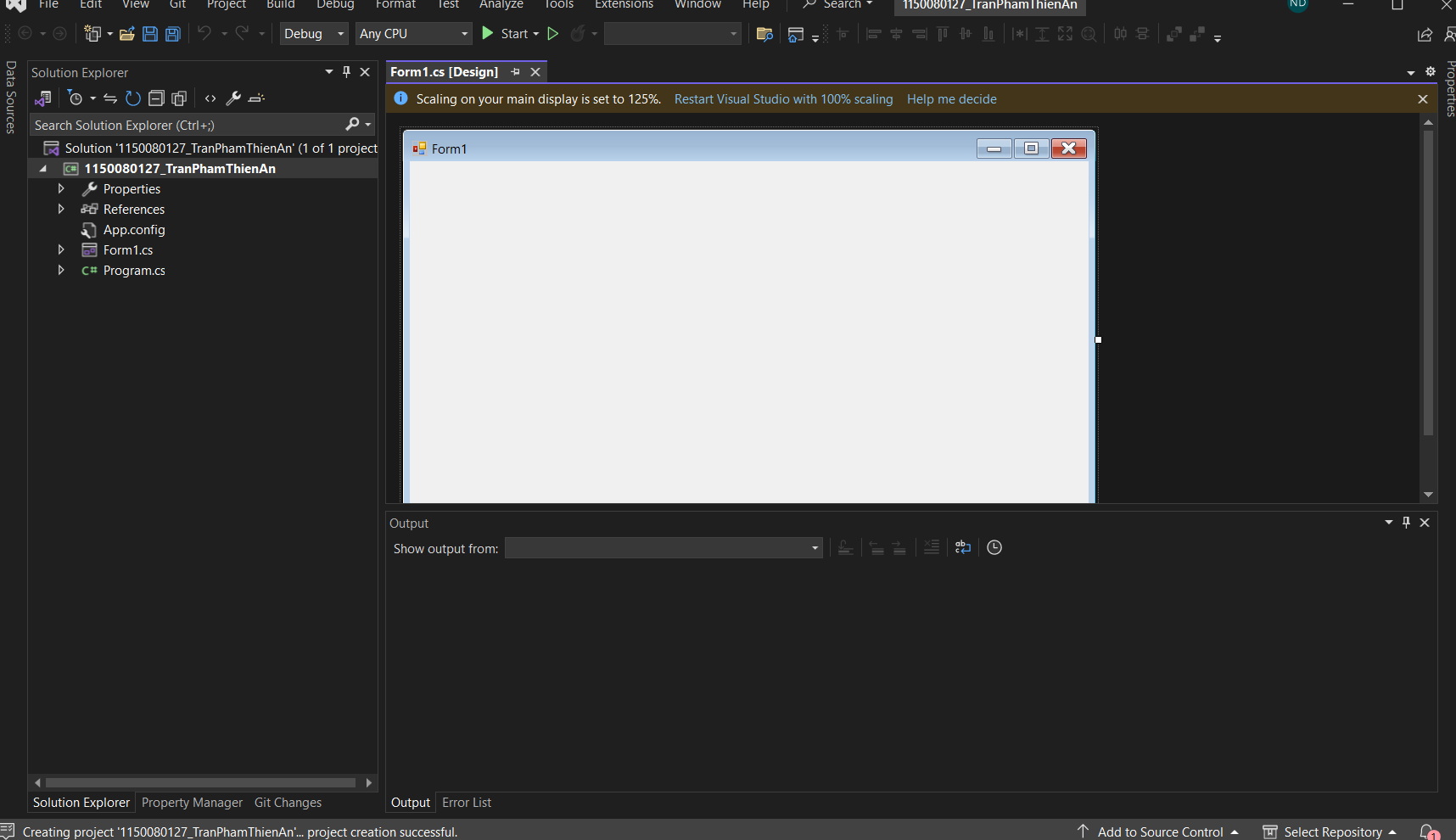
 **Bước 2:** Chọn Create a new project trên màn hình khởi động. Trong màn hình tiếp theo, ở mục All languages → chọn C# trong danh sách ngôn ngữ lập trình. Mục All platforms → chọn Windows. Ở mục All project types → chọn Desktop.



**Bước 3:** Trong cửa sổ tiếp theo, bạn điền đầy đủ các thông tin về Project.



Đây là kết quả!



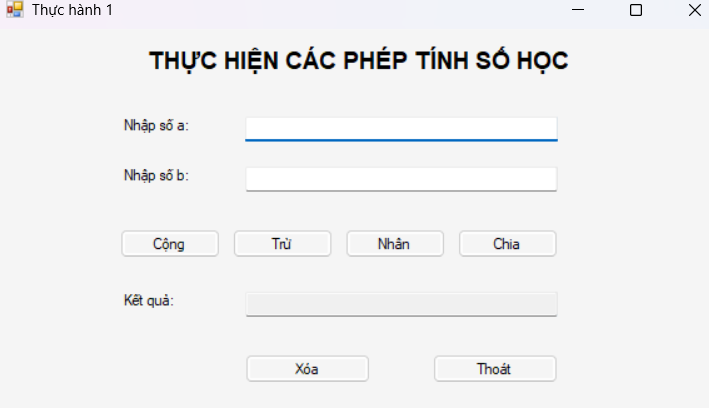
Màn hình làm việc cho ứng dụng Winform bao gồm các phần cơ bản sau: Số 1: Toolbox → chứa các control cho phép kéo thả vào form.

Số 2: màn hình thiết kế → nơi thiết kế trực quan, có thể chuyển đổi qua lại giữa code editor và giao diện.

Số 3: Solution Exlorer → lập trình viên quản lý các thành phần trong dự án. Số 4: Property → cho phép tùy chỉnh lại các thành phần của control và form.

**2. Thực hành**

**Thực hành 1:** Thiết kế form như sau:



Form này sẽ cho phép người dùng nhập các giá trị là số thực vào 2 biến a và b. Tùy vào các chức năng của các phép toán (cộng, trừ, nhân, chia) mà hiển thị ở phần kết quả.

Code:

using System;

using System.Windows.Forms;

using System.Drawing;

namespace \_1150080127\_TranPhamThienAn

{

public class Form1 : Form

{

TextBox txtA, txtB, txtKetQua;

Button btnCong, btnTru, btnNhan, btnChia, btnXoa, btnThoat;

public Form1()

{

BuildUI();

}

private void BuildUI()

{

this.Text = "Thực hành 1";

this.Width = 600;

this.Height = 350;

this.StartPosition = FormStartPosition.CenterScreen;

this.BackColor = Color.WhiteSmoke;

Label lblTitle = new Label()

{

Text = "THỰC HIỆN CÁC PHÉP TÍNH SỐ HỌC",

Font = new Font("Arial", 14, FontStyle.Bold),

AutoSize = false,

TextAlign = ContentAlignment.MiddleCenter,

Dock = DockStyle.Top,

Height = 50

};

this.Controls.Add(lblTitle);

Label lblA = new Label() { Text = "Nhập số a:", Left = 100, Top = 70, Width = 100 };

txtA = new TextBox() { Left = 200, Top = 70, Width = 250 };

Label lblB = new Label() { Text = "Nhập số b:", Left = 100, Top = 110, Width = 100 };

txtB = new TextBox() { Left = 200, Top = 110, Width = 250 };

btnCong = new Button() { Text = "Cộng", Left = 100, Top = 160, Width = 80 };

btnTru = new Button() { Text = "Trừ", Left = 190, Top = 160, Width = 80 };

btnNhan = new Button() { Text = "Nhân", Left = 280, Top = 160, Width = 80 };

btnChia = new Button() { Text = "Chia", Left = 370, Top = 160, Width = 80 };

Label lblKQ = new Label() { Text = "Kết quả:", Left = 100, Top = 210, Width = 100 };

txtKetQua = new TextBox() { Left = 200, Top = 210, Width = 250, ReadOnly = true };

btnXoa = new Button() { Text = "Xóa", Left = 200, Top = 260, Width = 100 };

btnThoat = new Button() { Text = "Thoát", Left = 350, Top = 260, Width = 100 };

btnCong.Click += btnCong\_Click;

btnTru.Click += btnTru\_Click;

btnNhan.Click += btnNhan\_Click;

btnChia.Click += btnChia\_Click;

btnXoa.Click += btnXoa\_Click;

btnThoat.Click += btnThoat\_Click;

this.Controls.AddRange(new Control[]

{ lblA, txtA, lblB, txtB, btnCong, btnTru, btnNhan, btnChia, lblKQ, txtKetQua, btnXoa, btnThoat });

}

private void btnCong\_Click(object sender, EventArgs e) => XuLy((a, b) => a + b);

private void btnTru\_Click(object sender, EventArgs e) => XuLy((a, b) => a - b);

private void btnNhan\_Click(object sender, EventArgs e) => XuLy((a, b) => a \* b);

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

XuLy((a, b) => b == 0 ? double.NaN : a / b);

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

{

txtA.Clear();

txtB.Clear();

txtKetQua.Clear();

txtA.Focus();

}

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

{

if (MessageBox.Show("Bạn có muốn thoát?", "Xác nhận",

MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

this.Close();

}

}

private void XuLy(Func<double, double, double> phepTinh)

{

try

{

double a = Convert.ToDouble(txtA.Text.Trim());

double b = Convert.ToDouble(txtB.Text.Trim());

double kq = phepTinh(a, b);

if (double.IsNaN(kq))

MessageBox.Show("Không chia cho 0 được!");

else

txtKetQua.Text = kq.ToString();

}

catch

{

MessageBox.Show("Dữ liệu nhập không hợp lệ!");

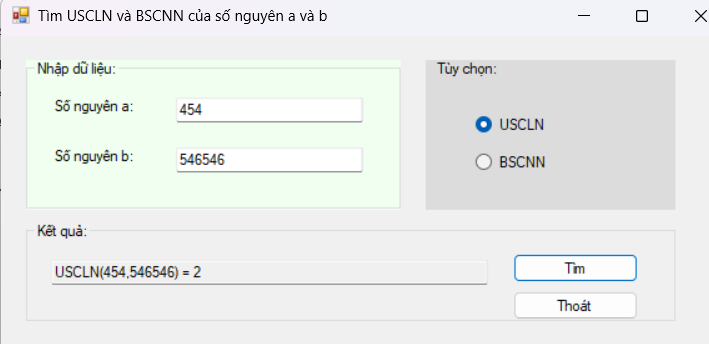
}

}

}

}

**Áp dụng 1:** Thêm vào một project có tên là “ApDung1”, thực hiện tìm ước số chung lớn nhất và bội số chung nhỏ nhất của 2 số nguyên a và b:



# Yêu cầu:

* Viết 2 hàm để tính ước số chung lớn nhất và bội số chung nhỏ nhất của 2 số a và b. Khi nhấn vào nút Tìm, nếu người dùng chọn tùy chọn tìm

USCLN thì hiển thị ước số chung lớn nhất, ngược lại hiển thị bội số chung nhỏ nhất.

* Nút thoát để thoát khỏi chương trình.

using System;

using System.Windows.Forms;

using System.Drawing;

namespace ApDung1

{

public class Form1 : Form

{

TextBox txtA, txtB, txtKetQua;

RadioButton rdoUSCLN, rdoBSCNN;

Button btnTim, btnThoat;

public Form1()

{

BuildUI();

}

private void BuildUI()

{

this.Text = "Tìm USCLN và BSCNN của số nguyên a và b";

this.Width = 600;

this.Height = 300;

this.StartPosition = FormStartPosition.CenterScreen;

// Group nhập dữ liệu

GroupBox grpNhap = new GroupBox()

{

Text = "Nhập dữ liệu:",

Left = 20,

Top = 20,

Width = 300,

Height = 120,

BackColor = Color.Honeydew

};

Label lblA = new Label() { Text = "Số nguyên a:", Left = 20, Top = 30, Width = 100 };

txtA = new TextBox() { Left = 120, Top = 30, Width = 150 };

Label lblB = new Label() { Text = "Số nguyên b:", Left = 20, Top = 70, Width = 100 };

txtB = new TextBox() { Left = 120, Top = 70, Width = 150 };

grpNhap.Controls.AddRange(new Control[] { lblA, txtA, lblB, txtB });

// Group tùy chọn

GroupBox grpChon = new GroupBox()

{

Text = "Tùy chọn:",

Left = 340,

Top = 20,

Width = 200,

Height = 120,

BackColor = Color.Gainsboro

};

rdoUSCLN = new RadioButton() { Text = "USCLN", Left = 40, Top = 40, Checked = true };

rdoBSCNN = new RadioButton() { Text = "BSCNN", Left = 40, Top = 70 };

grpChon.Controls.AddRange(new Control[] { rdoUSCLN, rdoBSCNN });

// Group kết quả

GroupBox grpKQ = new GroupBox()

{

Text = "Kết quả:",

Left = 20,

Top = 150,

Width = 520,

Height = 80

};

txtKetQua = new TextBox() { Left = 20, Top = 30, Width = 350, ReadOnly = true };

btnTim = new Button() { Text = "Tìm", Left = 390, Top = 25, Width = 100 };

btnThoat = new Button() { Text = "Thoát", Left = 390, Top = 55, Width = 100 };

btnTim.Click += btnTim\_Click;

btnThoat.Click += btnThoat\_Click;

grpKQ.Controls.AddRange(new Control[] { txtKetQua, btnTim, btnThoat });

// Add vào form

this.Controls.AddRange(new Control[] { grpNhap, grpChon, grpKQ });

}

private void InitializeComponent()

{

this.SuspendLayout();

//

// Form1

//

this.ClientSize = new System.Drawing.Size(282, 253);

this.Name = "Form1";

this.Load += new System.EventHandler(this.Form1\_Load);

this.ResumeLayout(false);

}

private void Form1\_Load(object sender, EventArgs e)

{

}

// Hàm tính USCLN

private int USCLN(int a, int b)

{

while (b != 0)

{

int t = b;

b = a % b;

a = t;

}

return a;

}

// Hàm tính BSCNN

private int BSCNN(int a, int b)

{

return (a \* b) / USCLN(a, b);

}

// Sự kiện nút Tìm

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

{

try

{

int a = int.Parse(txtA.Text);

int b = int.Parse(txtB.Text);

if (rdoUSCLN.Checked)

txtKetQua.Text = $"USCLN({a},{b}) = {USCLN(a, b)}";

else

txtKetQua.Text = $"BSCNN({a},{b}) = {BSCNN(a, b)}";

}

catch

{

MessageBox.Show("Vui lòng nhập số nguyên hợp lệ!");

}

}

// Sự kiện nút Thoát

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

{

if (MessageBox.Show("Bạn có muốn thoát?", "Xác nhận",

MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

this.Close();

}

}

}

}

**Áp dụng 2:** Xây dựng ứng dụng mô phỏng thiết bị Security Panel theo yêu cầu:

Một phòng Lab muốn lắp thiết bị bảo vệ đặt được ở bên ngoài cửa ra vào có tên là Security. Chỉ cho phép những cá nhân nào nhập đúng password đã được quy định mới được vào phòng. Bảng dưới là password được cấp cho những nhóm thành viên:

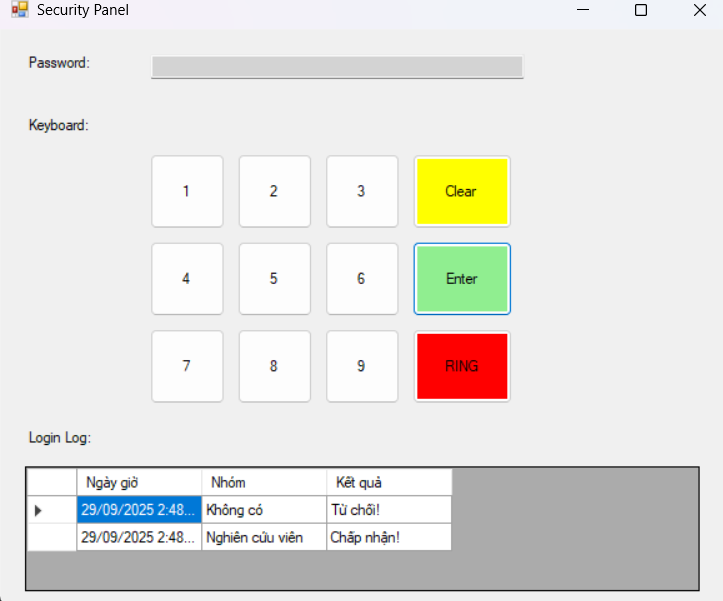
# Nhóm thành viên Password

Phát triển công nghệ 1496 hoặc 2673

|  |  |
| --- | --- |
| Nghiên cứu viên | 7462 |
| Thiết kế mô hình | 8884 hoặc 3842 hoặc 3383 |

Mỗi một lần nhập password sẽ có 2 trạng thái: chấp nhận hoặc từ chối. Tất cả các thông tin truy cập sẽ được hiển thị trong khung thông tin phía dưới bàn phím số. Nếu là chấp nhận thì ghi lại và hiển thị thông về về ngày, thời gian, tên nhóm và chấp nhận. Nếu là từ chối thì ghi lại và hiển thị thông về về ngày, thời gian, từ chối.

Giao diện ứng dụng:

****

using System;

using System.Windows.Forms;

using System.Drawing;

namespace ApDung2

{

public class Form1 : Form

{

TextBox txtPassword;

DataGridView dgvLog;

public Form1()

{

BuildUI();

}

private void BuildUI()

{

this.Text = "Security Panel";

this.Width = 600;

this.Height = 500;

this.StartPosition = FormStartPosition.CenterScreen;

// Label + TextBox Password

Label lblPass = new Label() { Text = "Password:", Left = 20, Top = 20, Width = 100 };

txtPassword = new TextBox() { Left = 120, Top = 20, Width = 300, ReadOnly = true, BackColor = Color.LightGray };

this.Controls.Add(lblPass);

this.Controls.Add(txtPassword);

// Label Keyboard

Label lblKeyboard = new Label() { Text = "Keyboard:", Left = 20, Top = 70, Width = 100 };

this.Controls.Add(lblKeyboard);

// Tạo bàn phím số

int x = 120, y = 100, size = 60;

int num = 1;

for (int row = 0; row < 3; row++)

{

for (int col = 0; col < 3; col++)

{

Button btnNum = new Button()

{

Text = num.ToString(),

Left = x + col \* (size + 10),

Top = y + row \* (size + 10),

Width = size,

Height = size

};

btnNum.Click += (s, e) => txtPassword.Text += ((Button)s).Text;

this.Controls.Add(btnNum);

num++;

}

}

// Nút Clear

Button btnClear = new Button()

{

Text = "Clear",

Left = x + 3 \* (size + 10),

Top = y,

Width = size + 20,

Height = size,

BackColor = Color.Yellow

};

btnClear.Click += (s, e) => txtPassword.Clear();

this.Controls.Add(btnClear);

// Nút Enter

Button btnEnter = new Button()

{

Text = "Enter",

Left = x + 3 \* (size + 10),

Top = y + (size + 10),

Width = size + 20,

Height = size,

BackColor = Color.LightGreen

};

btnEnter.Click += BtnEnter\_Click;

this.Controls.Add(btnEnter);

// Nút Ring

Button btnRing = new Button()

{

Text = "RING",

Left = x + 3 \* (size + 10),

Top = y + 2 \* (size + 10),

Width = size + 20,

Height = size,

BackColor = Color.Red

};

btnRing.Click += (s, e) => MessageBox.Show("BÁO ĐỘNG!!!", "ALERT", MessageBoxButtons.OK, MessageBoxIcon.Warning);

this.Controls.Add(btnRing);

// Label Log

Label lblLog = new Label() { Text = "Login Log:", Left = 20, Top = 320, Width = 100 };

this.Controls.Add(lblLog);

// DataGridView Log

dgvLog = new DataGridView()

{

Left = 20,

Top = 350,

Width = 540,

Height = 100,

AllowUserToAddRows = false,

ReadOnly = true,

ColumnCount = 3

};

dgvLog.Columns[0].Name = "Ngày giờ";

dgvLog.Columns[1].Name = "Nhóm";

dgvLog.Columns[2].Name = "Kết quả";

this.Controls.Add(dgvLog);

}

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

{

string pass = txtPassword.Text.Trim();

string group = "Không có";

string result = "Từ chối!";

// Kiểm tra password

if (pass == "1496" || pass == "2673") { group = "Phát triển công nghệ"; result = "Chấp nhận!"; }

else if (pass == "7462") { group = "Nghiên cứu viên"; result = "Chấp nhận!"; }

else if (pass == "8884" || pass == "3842" || pass == "3383") { group = "Thiết kế mô hình"; result = "Chấp nhận!"; }

// Thêm log

dgvLog.Rows.Add(DateTime.Now.ToString(), group, result);

// Reset pass

txtPassword.Clear();

}

}

}

**Thực hành 2:** Tính tiền công dịch vụ tại phòng khám nha khoa.

Giả sử, tại một phòng khám nha khoa cần tính tiền các loại dịch vụ khác nhau. Một biểu mẫu thanh toán gồm:

* Tên khách hàng.

– Các loại dịch vụ như: lấy cao răng (50.000đ), tẩy trắng răng (100.000đ), hàn răng (100.000đ), bẻ răng (10.000đ), bọc răng (1.000.000đ).

* Hàn răng, bẻ răng và bọc răng được tính theo chiếc răng. Giá tiền trên được tính theo 1 chiếc răng.
* Biểu mẫu tính tiền không được để trống tên khách hàng. Có cảnh báo nếu trống.

Tham khảo giao diện ứng dụng sau:



using System;

using System.Windows.Forms;

using System.Drawing;

namespace ThucHanh2

{

public class Form1 : Form

{

TextBox txtTenKH, txtThanhTien;

CheckBox chkLayCaoRang, chkTayTrangRang, chkHanRang, chkBeRang, chkBocRang;

NumericUpDown numHanRang, numBeRang, numBocRang;

Button btnTinhTien, btnThoat;

public Form1()

{

BuildUI();

}

private void BuildUI()

{

this.Text = "Phòng khám nha khoa";

this.Width = 600;

this.Height = 400;

this.StartPosition = FormStartPosition.CenterScreen;

// Tiêu đề

Label lblTitle = new Label()

{

Text = "PHÒNG KHÁM NHA KHOA HẢI ÂU",

Font = new Font("Arial", 14, FontStyle.Bold),

BackColor = Color.Green,

ForeColor = Color.White,

Dock = DockStyle.Top,

Height = 50,

TextAlign = ContentAlignment.MiddleCenter

};

this.Controls.Add(lblTitle);

// Tên khách hàng

Label lblTenKH = new Label() { Text = "Tên khách hàng:", Left = 20, Top = 70, Width = 120 };

txtTenKH = new TextBox() { Left = 150, Top = 70, Width = 300 };

this.Controls.Add(lblTenKH);

this.Controls.Add(txtTenKH);

// Dịch vụ

GroupBox grpDV = new GroupBox() { Text = "Dịch vụ tại phòng khám:", Left = 20, Top = 110, Width = 540, Height = 180 };

chkLayCaoRang = new CheckBox() { Text = "Lấy cao răng (50.000đ / 2 hàm)", Left = 20, Top = 30, Width = 250 };

chkTayTrangRang = new CheckBox() { Text = "Tẩy trắng răng (100.000đ / 2 hàm)", Left = 20, Top = 60, Width = 280 };

chkHanRang = new CheckBox() { Text = "Hàn răng (100.000đ / răng)", Left = 20, Top = 90, Width = 250 };

numHanRang = new NumericUpDown() { Left = 300, Top = 90, Width = 50, Minimum = 1, Maximum = 20, Value = 1 };

chkBeRang = new CheckBox() { Text = "Bẻ răng (10.000đ / răng)", Left = 20, Top = 120, Width = 250 };

numBeRang = new NumericUpDown() { Left = 300, Top = 120, Width = 50, Minimum = 1, Maximum = 20, Value = 1 };

chkBocRang = new CheckBox() { Text = "Bọc răng (1.000.000đ / răng)", Left = 20, Top = 150, Width = 250 };

numBocRang = new NumericUpDown() { Left = 300, Top = 150, Width = 50, Minimum = 1, Maximum = 20, Value = 1 };

grpDV.Controls.AddRange(new Control[] { chkLayCaoRang, chkTayTrangRang, chkHanRang, numHanRang, chkBeRang, numBeRang, chkBocRang, numBocRang });

this.Controls.Add(grpDV);

// Kết quả

Label lblKQ = new Label() { Text = "Thành tiền:", Left = 20, Top = 310, Width = 100 };

txtThanhTien = new TextBox() { Left = 120, Top = 310, Width = 200, ReadOnly = true };

this.Controls.Add(lblKQ);

this.Controls.Add(txtThanhTien);

// Nút chức năng

btnTinhTien = new Button() { Text = "Tính tiền", Left = 350, Top = 310, Width = 80 };

btnThoat = new Button() { Text = "Thoát", Left = 450, Top = 310, Width = 80 };

btnTinhTien.Click += btnTinhTien\_Click;

btnThoat.Click += (s, e) => this.Close();

this.Controls.Add(btnTinhTien);

this.Controls.Add(btnThoat);

}

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

{

if (string.IsNullOrWhiteSpace(txtTenKH.Text))

{

MessageBox.Show("Tên khách hàng không được để trống!");

txtTenKH.Focus();

return;

}

double thanhTien = 0;

if (chkLayCaoRang.Checked) thanhTien += 50000;

if (chkTayTrangRang.Checked) thanhTien += 100000;

if (chkHanRang.Checked) thanhTien += (int)numHanRang.Value \* 100000;

if (chkBeRang.Checked) thanhTien += (int)numBeRang.Value \* 10000;

if (chkBocRang.Checked) thanhTien += (int)numBocRang.Value \* 1000000;

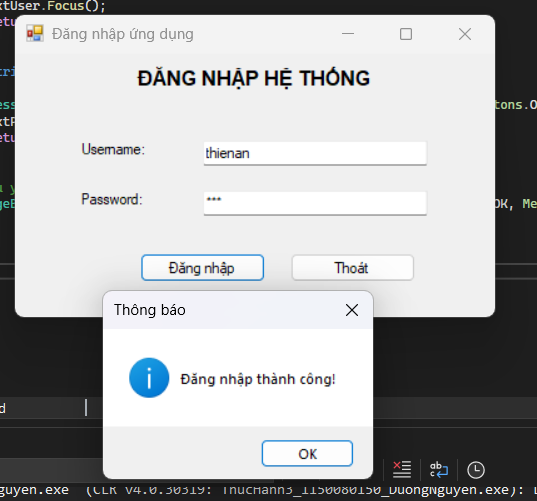
txtThanhTien.Text = thanhTien.ToString("N0") + " VND";

}

}

}

**Áp dụng 3:** Thiết kế và lập trình form Đăng nhập của một ứng dụng. Thông tin cần đăng nhập gồm username và password. Yêu cầu người dùng không được để trống trường thông tin nào. Nếu để trống trường thông tin nào thì cảnh báo người dùng để bổ sung.



using System;

using System.Windows.Forms;

using System.Drawing;

namespace ApDung3

{

public class Form1 : Form

{

TextBox txtUser, txtPass;

Button btnLogin, btnThoat;

public Form1()

{

BuildUI();

}

private void BuildUI()

{

this.Text = "Đăng nhập ứng dụng";

this.Width = 400;

this.Height = 250;

this.StartPosition = FormStartPosition.CenterScreen;

// Tiêu đề

Label lblTitle = new Label()

{

Text = "ĐĂNG NHẬP HỆ THỐNG",

Font = new Font("Arial", 12, FontStyle.Bold),

Dock = DockStyle.Top,

Height = 40,

TextAlign = ContentAlignment.MiddleCenter

};

this.Controls.Add(lblTitle);

// Username

Label lblUser = new Label() { Text = "Username:", Left = 50, Top = 70, Width = 80 };

txtUser = new TextBox() { Left = 150, Top = 70, Width = 180 };

// Password

Label lblPass = new Label() { Text = "Password:", Left = 50, Top = 110, Width = 80 };

txtPass = new TextBox() { Left = 150, Top = 110, Width = 180, PasswordChar = '\*' };

// Nút chức năng

btnLogin = new Button() { Text = "Đăng nhập", Left = 100, Top = 160, Width = 100 };

btnThoat = new Button() { Text = "Thoát", Left = 220, Top = 160, Width = 100 };

btnLogin.Click += BtnLogin\_Click;

btnThoat.Click += (s, e) => this.Close();

this.Controls.AddRange(new Control[] { lblUser, txtUser, lblPass, txtPass, btnLogin, btnThoat });

}

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

{

if (string.IsNullOrWhiteSpace(txtUser.Text))

{

MessageBox.Show("Vui lòng nhập Username!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtUser.Focus();

return;

}

if (string.IsNullOrWhiteSpace(txtPass.Text))

{

MessageBox.Show("Vui lòng nhập Password!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtPass.Focus();

return;

}

// Nếu yêu cầu chỉ check trống thì dừng ở đây

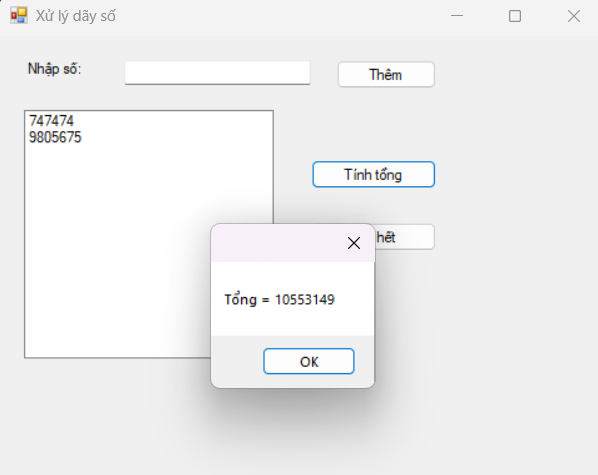
MessageBox.Show("Đăng nhập thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

**Thực hành 3:** Thiết kế form frmXuLySo với control ListBox như



# Yêu cầu:

* Chỉ cho nhập số vào textbox Nhập số nguyên
* Khi nhập dữ liệu vào textbox Nhập số nguyên và nhấn Enter (hoặc click vào nút Nhập số) 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.

Code tham khảo:

using System;

using System.Windows.Forms;

using System.Drawing;

namespace ThucHanh3

{

public class Form1 : Form

{

TextBox txtNhap;

ListBox lstSo;

Button btnThem, btnTong, btnXoa;

private void InitializeComponent()

{

this.SuspendLayout();

//

// Form1

//

this.ClientSize = new System.Drawing.Size(282, 253);

this.Name = "Form1";

this.Load += new System.EventHandler(this.Form1\_Load);

this.ResumeLayout(false);

}

private void Form1\_Load(object sender, EventArgs e)

{

}

public Form1()

{

BuildUI();

}

private void BuildUI()

{

this.Text = "Xử lý dãy số";

this.Width = 500;

this.Height = 400;

Label lblNhap = new Label() { Text = "Nhập số:", Left = 20, Top = 20, Width = 80 };

txtNhap = new TextBox() { Left = 100, Top = 20, Width = 150 };

btnThem = new Button() { Text = "Thêm", Left = 270, Top = 20, Width = 80 };

btnThem.Click += (s, e) =>

{

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

{

lstSo.Items.Add(so);

txtNhap.Clear();

}

else

{

MessageBox.Show("Nhập số hợp lệ!");

}

};

lstSo = new ListBox() { Left = 20, Top = 60, Width = 200, Height = 200 };

btnTong = new Button() { Text = "Tính tổng", Left = 250, Top = 100, Width = 100 };

btnTong.Click += (s, e) =>

{

int tong = 0;

foreach (var item in lstSo.Items)

tong += Convert.ToInt32(item);

MessageBox.Show("Tổng = " + tong);

};

btnXoa = new Button() { Text = "Xóa hết", Left = 250, Top = 150, Width = 100 };

btnXoa.Click += (s, e) => lstSo.Items.Clear();

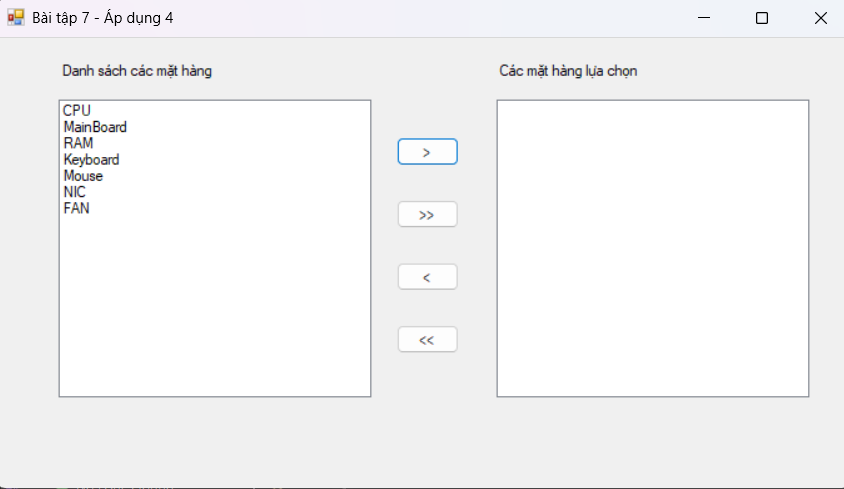
this.Controls.AddRange(new Control[] { lblNhap, txtNhap, btnThem, lstSo, btnTong, btnXoa });

}

}

}

**Áp dụng 4:** làm bài tập 7 tại Link.



using System;

using System.Windows.Forms;

using System.Drawing;

namespace ApDung4

{

public class Form1 : Form

{

ListBox lstMatHang, lstDaChon;

Button btnChon1, btnChonAll, btnTra1, btnTraAll;

public Form1()

{

BuildUI();

}

private void BuildUI()

{

this.Text = "Bài tập 7 - Áp dụng 4";

this.Width = 700;

this.Height = 400;

this.StartPosition = FormStartPosition.CenterScreen;

// Label

Label lblDanhSach = new Label() { Text = "Danh sách các mặt hàng", Left = 50, Top = 20, Width = 200 };

Label lblDaChon = new Label() { Text = "Các mặt hàng lựa chọn", Left = 400, Top = 20, Width = 200 };

// ListBox trái

lstMatHang = new ListBox() { Left = 50, Top = 50, Width = 250, Height = 250, SelectionMode = SelectionMode.MultiExtended };

lstMatHang.Items.AddRange(new string[] { "CPU", "MainBoard", "RAM", "Keyboard", "Mouse", "NIC", "FAN" });

// ListBox phải

lstDaChon = new ListBox() { Left = 400, Top = 50, Width = 250, Height = 250, SelectionMode = SelectionMode.MultiExtended };

// Nút chức năng

btnChon1 = new Button() { Text = ">", Left = 320, Top = 80, Width = 50 };

btnChon1.Click += BtnChon1\_Click;

btnChonAll = new Button() { Text = ">>", Left = 320, Top = 130, Width = 50 };

btnChonAll.Click += BtnChonAll\_Click;

btnTra1 = new Button() { Text = "<", Left = 320, Top = 180, Width = 50 };

btnTra1.Click += BtnTra1\_Click;

btnTraAll = new Button() { Text = "<<", Left = 320, Top = 230, Width = 50 };

btnTraAll.Click += BtnTraAll\_Click;

this.Controls.AddRange(new Control[] { lblDanhSach, lblDaChon, lstMatHang, lstDaChon, btnChon1, btnChonAll, btnTra1, btnTraAll });

}

// Chuyển phần tử chọn từ trái sang phải

private void BtnChon1\_Click(object sender, EventArgs e)

{

var items = lstMatHang.SelectedItems;

for (int i = items.Count - 1; i >= 0; i--)

{

lstDaChon.Items.Add(items[i]);

lstMatHang.Items.Remove(items[i]);

}

}

// Chuyển tất cả sang phải

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

{

foreach (var item in lstMatHang.Items)

lstDaChon.Items.Add(item);

lstMatHang.Items.Clear();

}

// Chuyển phần tử chọn từ phải sang trái

private void BtnTra1\_Click(object sender, EventArgs e)

{

var items = lstDaChon.SelectedItems;

for (int i = items.Count - 1; i >= 0; i--)

{

lstMatHang.Items.Add(items[i]);

lstDaChon.Items.Remove(items[i]);

}

}

// Chuyển tất cả sang trái

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

{

foreach (var item in lstDaChon.Items)

lstMatHang.Items.Add(item);

lstDaChon.Items.Clear();

}

}