

ĐẠI HỌC HUTECH
Khoa Công Nghệ Thông Tin



BÀI TẬP THỰC HÀNH
Môn : Lập Trình Window form

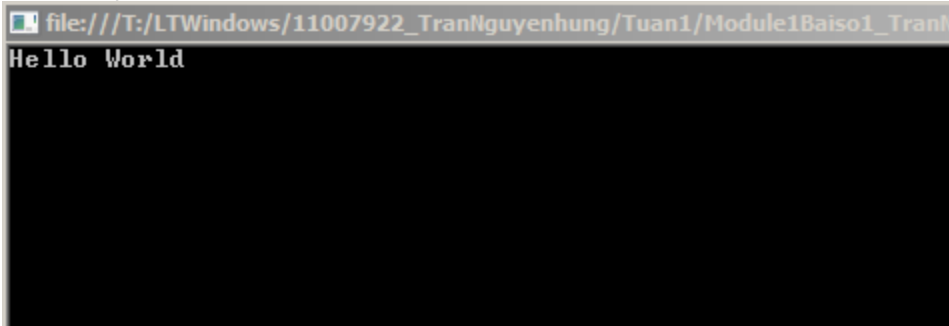
Giáo Viên : Ngô Thị Ngọc Thắm

Module 1 : Console

Bài 1

Viết ứng dụng console viết lời chào "Hello World"

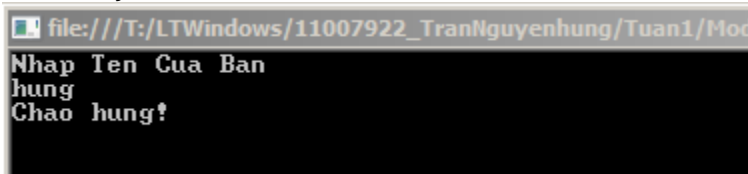
```
static void Main(string[] args)
{
    Console.Write("Hello World");
    Console.ReadKey();
}
```



Bài 2

Viết chương trình nhập vào tên mình và xuất ra màn hình "Hello + Tên"

```
static void Main(string[] args)
{
    Console.WriteLine("Nhap Ten Cua Ban");
    String ten = Console.ReadLine();
    Console.WriteLine("Chao {0}!", ten);
    Console.ReadKey();
}
```



Bài 3

Viết chương trình Tính tổng 2 số

```
static void Main(string[] args)
{
    int nSo1, nSo2, nTong = 0;
    Console.Clear();
    Console.Write("{0}", "Nhap So Thu Nhat: ");
    nSo1 = Convert.ToInt32(Console.ReadLine());
    Console.Write("Nhap So Thu 2: ");
    nSo2 = Convert.ToInt32(Console.ReadLine());
    nTong = nSo1 + nSo2;
    Console.WriteLine("Tong cua " + nSo1 + " Va " + nSo2 + " la : " + nTong);
    Console.ReadLine();
}
```

```
file:///T:/LTWindows/11007922_TranNguyenhung/Tuan1/Module1
Nhap So Thu Nhat: 3
Nhap So Thu 2: 7
Tong cua 3 Va 7 la : 10
```

Bài 4

Viết chương trình giải Phương trình bậc nhất

```
public static void GiaiPTBac1(int a, float b)
{
    if (a == 0 && b == 0)
    {
        Console.WriteLine("Phuong trinh vo so nghiem");
    }
    else if (a == 0 && b != 0)
    {
        Console.WriteLine("Phuong trinh vo nghiem");
    }
    else
    {
        Console.WriteLine("Phuong trinh co 1 nghiem {0}", -b / a);
    }
}
static void Main(string[] args)
{
    Console.WriteLine("***GIAI VA BIEN LUAN PHUONG TRINH BAC NHAT***");
    int a;
    float b;
    Console.Write("Nhap a: ");
    a = Convert.ToInt32(Console.ReadLine());
    Console.Write("Nhap b: ");
    b = float.Parse(Console.ReadLine());
    GiaiPTBac1(a,b);
    Console.ReadLine();
}
```

```
file:///T:/LTWindows/11007922_TranNguyenhung/Tuan1/Module1BaiSo4_
***GIAI VA BIEN LUAN PHUONG TRINH BAC NHAT***
Nhap a: 2
Nhap b: 9
Phuong trinh co 1 nghiem -4.5
```

Bài 5

Tạo chương trình giải phương trình bậc 2

```
public static void GiaiPTBac1(int a, int b)
{
    if (a == 0 && b == 0)
    {
        Console.WriteLine("Phuong trinh vo so nghiem");
    }
    else if (a == 0 && b != 0)
    {
        Console.WriteLine("Phuong trinh vo nghiem");
    }
    else
    {

```

```

        Console.WriteLine("Phuong trinh co 1 nghiem {0}", (double)-b / a);
    }
}
public static void GiaiPTBac2(int a,int b,int c)
{
    double delta = 0;
    double x, y;
    if (a == 0)
    {
        GiaiPTBac1(b, c);
    }
    else
    {
        delta = (double)Math.Pow(b, 2) - 4 * a * c;
        if (delta == 0)
        {
            Console.WriteLine("Phuong Trinh co 1 nghiem " + (double)-b / 2 * a);
        }
        else if (delta > 0)
        {
            x=(-b + Math.Sqrt(delta)) / (2 * a);
            y=(-b - Math.Sqrt(delta)) / (2 * a);
            Console.WriteLine("Phuong Trinh co 2 nghiem: ");
            Console.WriteLine("X = {0}, Y = {1}",Math.Round(x,2),Math.Round(y,2));
        }
        else
        {
            Console.WriteLine("Phuong trinh vo nghiem");
        }
    }
}
}
static void Main(string[] args)
{
    Console.WriteLine("***GIAI VA BIEN LUAN PHUONG TRINH BAC 2***");
    int a, b, c;
    Console.Write("Nhap a: ");
    a = Convert.ToInt32(Console.ReadLine());
    Console.Write("Nhap b: ");
    b = Convert.ToInt32(Console.ReadLine());
    Console.Write("Nhap c: ");
    c = Convert.ToInt32(Console.ReadLine());
    GiaiPTBac2(a, b, c);
    Console.ReadKey();
}

```

```

file:///T:/LTWindows/11007922_TranNguyenhung/Tuan1/Modu
***GIAI VA BIEN LUAN PHUONG TRINH BAC 2***
Nhap a: 1
Nhap b: 7
Nhap c: -8
Phuong Trinh co 2 nghiem:
X = 1, Y = -8

```

Bài 6

Tạo chương trình tính lượng giác từ 1 góc nhập vào

Sử dụng một số hàm lượng giác và nhập dữ liệu từ bàn phím Kết hợp định dạng

xuất dữ liệu.(tiếp tục làm quen với class Math)

Math.PI lấy số PI

Math.Sin(a) tính sin, với a là radian

Math.Cos(a) tính Cos, với a là radian

Math.Tan(a) tính Tan, với a là radian

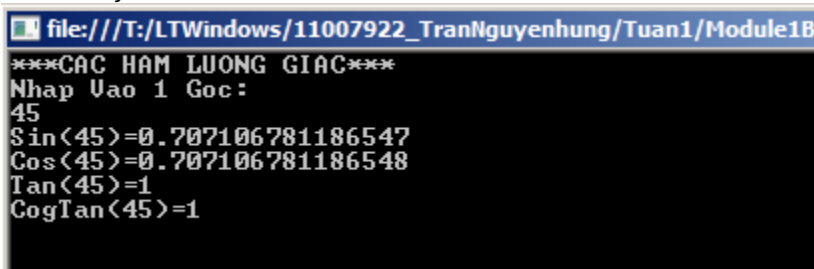
Yêu cầu:

+ Tạo Project mới với tên là CacHamLuongGiac:

+ Nhập 1 góc x. Tính và xuất sinx, cosx, tgx, cotgx.

+ Sinh viên chú ý rằng hàm lượng giác truyền vào là 1 số Radian, do đó các em phải đổi góc x ra radian, công thức đổi như sau $a = x * \pi / 180$.

```
static void Main(string[] args)
{
    double rad = 0;
    double goc;
    Console.WriteLine("***CAC HAM LUONG GIAC***");
    Console.WriteLine("Nhap Vao 1 Goc: ");
    goc = Convert.ToInt32(Console.ReadLine());
    rad = goc * Math.PI / 180;
    Console.WriteLine("Sin({0})={1}", goc, Math.Sin(rad));
    Console.WriteLine("Cos({0})={1}", goc, Math.Cos(rad));
    Console.WriteLine("Tan({0})={1}", goc, Math.Tan(rad));
    Console.WriteLine("CogTan({0})={1}", goc, 1/Math.Tan(rad));
    Console.ReadLine();
}
```



Bài 7

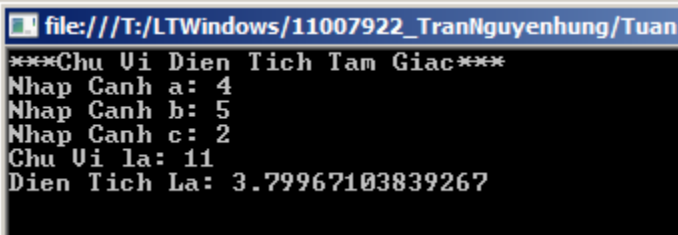
Tính chu vi diện tích tam giác

```
static void Main(string[] args)
{
    Console.WriteLine("***Chu Vi Dien Tich Tam Giac***");
    int a, b, c;
    double cv, dt, p;
    Console.Write("Nhap Canh a: ");
    a = Convert.ToInt32(Console.ReadLine());
    Console.Write("Nhap Canh b: ");
    b = Convert.ToInt32(Console.ReadLine());
    Console.Write("Nhap Canh c: ");
    c = Convert.ToInt32(Console.ReadLine());
    if (a + b > c || a + c > b || b + c > a)
    {
        cv = a + b + c;
        p = cv / 2;
        dt = Math.Sqrt(p * (p - a) * (p - b) * (p - c));
        Console.WriteLine("Chu Vi la: " + cv);
        Console.WriteLine("Dien Tich La: " + dt);
    }
}
```

```

    }
    else
        Console.WriteLine("Khong Tao Thanh Tam Giac");
    Console.ReadKey();
}

```



Bài 8

Viết chương trình tính chu vi diện tích hình tam giác, hình chữ nhật, hình vuông, hình tròn
viết chương trình tính chu vi và diện tích của hình tam giác, hình chữ nhật, hình vuông, hình tròn. Chương trình cho phép nhập thông số cho 1 trong 4 hình trên và xuất kết quả ra màn hình.

Yêu cầu viết hàm dạng

```
Static void main(string[] args)
```

```
{
```

Chọn chức năng

1 làm gì

2 làm gì

...

```
Switch(chon)
```

```
{
```

Case 1:

```
Goihamgiaitamgiac(??)
```

```
Break;
```

Case 2:

```
Goihamgiaihinhchunhat(??)
```

```
}
```

```
}
```

```

public static void GoiHamGiaiTamGiac()
{
    double cv, dt, p;
    int a, b, c;
    Console.WriteLine("***Ham Giai Hinh Tam Giac***");
    Console.WriteLine("Nhap Canh a: ");

```

```

a = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Nhap Canh b: ");
b = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Nhap Canh c: ");
c = Convert.ToInt32(Console.ReadLine());
if (a + b > c || a + c > b || b + c > a)
{
    cv = a + b + c;
    p = cv / 2;
    dt = Math.Sqrt(p * (p - a) * (p - b) * (p - c));
    Console.WriteLine("Chu Vi la: " + cv);
    Console.WriteLine("Dien Tich La: " + dt);
}
else
    Console.WriteLine("Khong Tao Thanh Tam Giac");
Console.ReadKey();
}
public static void GoiHamGiaiHinhChuNhat()
{
    double cv, dt;
    int a, b;
    Console.WriteLine("****Ham Giai Hinh Chu Nhat****");
    Console.WriteLine("Nhap Canh a: ");
    a = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Nhap Canh b: ");
    b = Convert.ToInt32(Console.ReadLine());
    cv = (a + b) * 2;
    dt = a * b;
    Console.WriteLine("Chu Vi la: " + cv);
    Console.WriteLine("Dien Tich La: " + dt);
    Console.ReadKey();
}
public static void GoiHamGiaiHinhVuong()
{
    double cv, dt;
    int a;
    Console.WriteLine("****Ham Giai Hinh Vuong****");
    Console.WriteLine("Nhap Canh Hinh Vuong: ");
    a = Convert.ToInt32(Console.ReadLine());
    cv = a*4;
    dt = Math.Pow(a,2);
    Console.WriteLine("Chu Vi la: " + cv);
    Console.WriteLine("Dien Tich La: " + dt);
    Console.ReadKey();
}
public static void GoiHamGiaiHinhTron()
{
    double cv, dt;
    int r;
    Console.WriteLine("****Ham Giai Hinh Tron****");
    Console.WriteLine("Nhap Ban Kinh: ");
    r = Convert.ToInt32(Console.ReadLine());
    cv = Math.PI * r;
    dt = Math.PI * Math.Pow(r, 2);
    Console.WriteLine("Chu Vi la: " + cv);
    Console.WriteLine("Dien Tich La: " + dt);
    Console.ReadKey();
}
static void Main(string[] args)
{
    int a;
    Console.WriteLine("Chon Chuc Nang:");
    Console.WriteLine("Nhan 1 de Giai Tam Giac");
}

```

```

Console.WriteLine("Nhan 2 de Giai Hinh Chu Nhat");
Console.WriteLine("Nhan 3 de Giai Hinh Vuong");
Console.WriteLine("Nhan 4 de Giai Hinh Tron");
a=Convert.ToInt32(Console.ReadLine());
switch (a)
{
    case 1:
        GoiHamGiaiTamGiac();
        break;
    case 2:
        GoiHamGiaiHinhChuNhat();
        break;
    case 3:
        GoiHamGiaiHinhVuong();
        break;
    case 4:
        GoiHamGiaiHinhTron();
        break;
    default:
        Console.WriteLine("Ban Nhap Sai Phim Chuc Nang");
        Console.ReadKey();
        break;
}
}

```

```

file:///T:/LTWindows/11007922_TranNguyenhung/Tuan
Chon Chuc Nang:
Nhan 1 de Giai Tam Giac
Nhan 2 de Giai Hinh Chu Nhat
Nhan 3 de Giai Hinh Vuong
Nhan 4 de Giai Hinh Tron
1
***Ham Giai Hinh Tam Giac***
Nhap Canh a:
5
Nhap Canh b:
5
Nhap Canh c:
2
Chu Vi la: 12
Dien Tich La: 4.89897948556636

```

Bài 9

Tính: $S(x, n) = x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$

```

public static double giaithua(int a)
{
    double kq=1;
    for (int i = 1; i <= a; i++)
        kq *= i;
    return kq;
}
public static void tongchuoi(int x,int n)
{
    double kq = 0;
    for (int i = 1; i <= n; i++)
    {
        kq += (double) Math.Pow(x, i) / giaithua(i);
    }
    Console.WriteLine("S({0},{1})={2}",x,n,Math.Round(kq,3));
    Console.ReadKey();
}
static void Main(string[] args)
{
    int x, n;

```



```

    Console.WriteLine("***TINH TONG CHUOI SO***");
    Console.WriteLine("Nhap x: ");
    x = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Nhap n: ");
    n = Convert.ToInt32(Console.ReadLine());
    tongchuoi(x, n);
}

```

```

file:///T:/LTWindows/11007922_TranNgu
***TINH TONG CHUOI SO***
Nhap x:
2
Nhap n:
2
S<2,2>=4

```

Bài 10

Giải bài toán:

Vừa gà vừa chó
 Bó lại cho tròn
 Ba mươi sáu con
 Một trăm chân chẵn

Tìm số gà, số chó?

```

static void Main(string[] args)
{
    //2x+4y=100
    //x+y=36
    int x=0, y=0;

    for (int i = 36; i > 0; i--)
    {
        x = i;
        y = 36 - x;
        if (2 * x + 4 * y == 100)
            break;
    }
    Console.WriteLine("So ga la: " + x);
    Console.WriteLine("So cho la: " + y);
    Console.ReadKey();
}

```

```

file:///T:/LTWindows/11007922_TranNgu
So ga la: 22
So cho la: 14

```

Module 2 : GUI

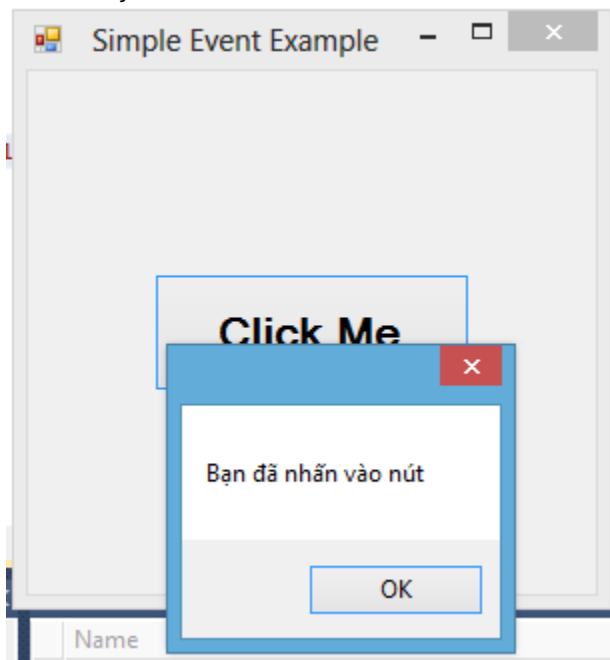
Bài 1

Thiết kế Form chứa nút Click Me, khi hiện lên sẽ hiện lời chào, khi click vào button hay form thì hiển thị message box tương ứng.

```
private void Form1_Load(object sender, EventArgs e)
{
    MessageBox.Show("Hi,Chào Mừng Bạn Đến Với Lập Trình C# 2010");
}

private void Form1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Bạn đã click vào Form");
}

private void btn1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Bạn đã nhấn vào nút");
}
```



Bài 2

thiết kế form sau:

Yêu cầu:

- chương trình cho phép nhập tên, năm sinh vào textbox yourname và year of birth tương ứng. nếu your name không nhập dữ liệu, year of birth không phải là số thì thông báo lỗi.
- Người dùng ấn nút show sẽ hiển thị message box bao gồm tên, tuổi(năm sinh hiện tại)

- Người dùng ấn nút clear sẽ xóa hết thông tin nhập trên các textbox bao gồm tên, tuổi(năm hiện tại- năm sinh)
- Nút exit xác nhận người dùng có thật sự muốn thoát khỏi chương trình không?
- Nhấn esc tương ứng với nút exit

```
public partial class FrmBaiTap1 : Form
{
    public FrmBaiTap1()
    {
        InitializeComponent();
    }
    int er = 1;

    private void btnExit_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void FrmBaiTap1_FormClosing(object sender, FormClosingEventArgs e)
    {
        DialogResult r;
        r= MessageBox.Show("Bạn chắc chắn muốn đóng?", "Thông báo",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Question,
            MessageBoxDefaultButton.Button1);
        if(r == DialogResult.No)
            e.Cancel=true;
    }

    private void txtUrName_Leave(object sender, EventArgs e)
    {
        Control ctr = (Control)sender;
        if (ctr.Text.Trim().Length == 0)
            this.errorProvider1.SetError(ctr, "Bạn phải điền tên vào");
        else
            this.errorProvider1.Clear();
    }

    private void txtNamSinh_TextChanged(object sender, EventArgs e)
    {
        Control ctr = (Control)sender;
        if (ctr.Text.Length > 0 && !Char.IsDigit(ctr.Text[ctr.Text.Length - 1]))
        {
            this.errorProvider1.SetError(ctr, "Năm sinh Không Đúng");
            er = 1;
        }
        else
        {
            this.errorProvider1.Clear();
            er = 0;
        }
    }

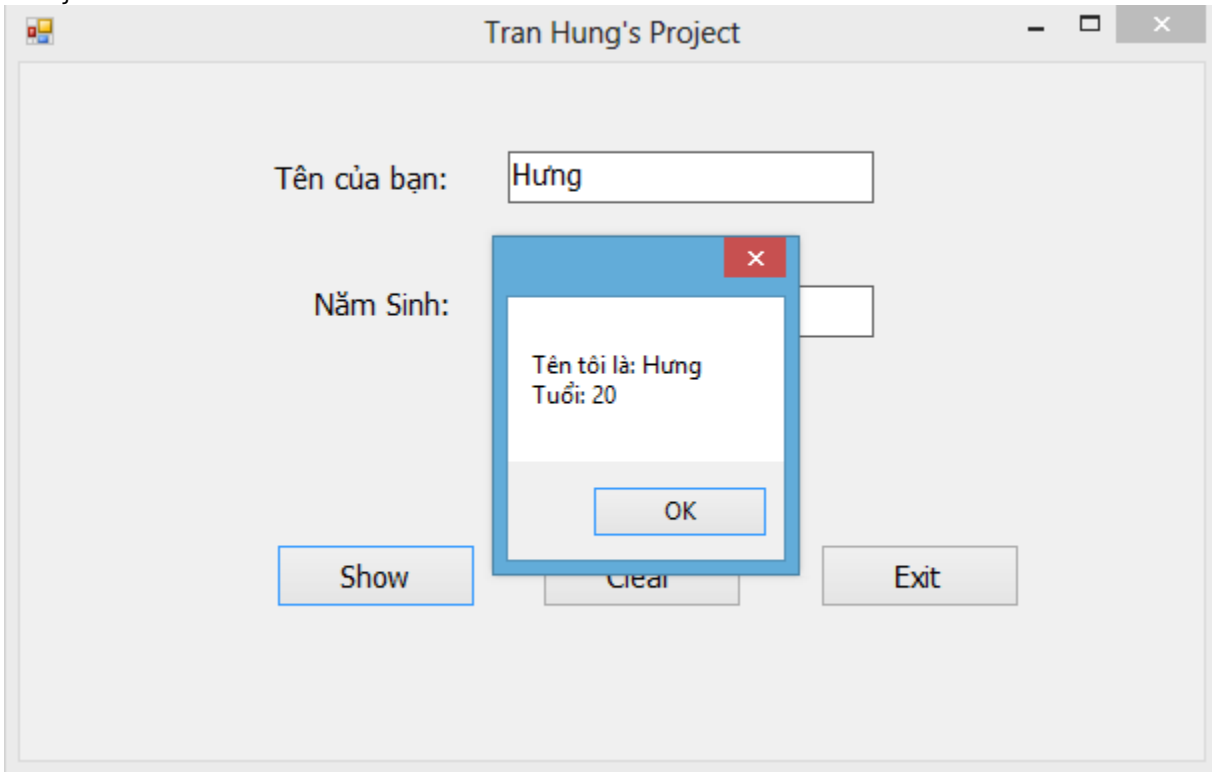
    private void btnShow_Click(object sender, EventArgs e)
    {
        if (er == 0)
        {
            int tuoi;
            String chuoi;
            chuoi = "Tên tôi là: " + txtUrName.Text + "\n";
            tuoi = DateTime.Now.Year - Convert.ToInt32(txtNamSinh.Text);
            chuoi += "Tuổi: " + tuoi.ToString();
        }
    }
}
```

```

        MessageBox.Show(chuoi);
    }
}

private void btnClear_Click(object sender, EventArgs e)
{
    txtUrName.Clear();
    txtNamSinh.Clear();
    txtUrName.Focus();
}
}

```



Bài 3

Thiết kế giao diện như hình sau:

Viết lệnh để chương trình hoạt động như sau:

- Nhấn vào Button btnHo thì gán nội dung của txtHo cho lblHoTen
- Nhấn vào Button btnTen thì gán nội dung của txtTen cho lblHoTen
- Nhấn vào Button btnHoTen thì gán nội dung của txtHo cộng với txtTen cho lblHo
- Nhấn Double click vào lblHoTen thì nội dung của lblHoTen bị xoá
- Nhấn vào btnKetThuc thì đóng chương trình

```

public partial class Frm01 : Form
{
    public Frm01()
    {
        InitializeComponent();
    }

    private void btnho_Click(object sender, EventArgs e)
    {
        lblHoTen.Text = txtHo.Text;
    }
}

```

```

private void btnnten_Click(object sender, EventArgs e)
{
    lblHoTen.Text = txtTen.Text;
}

private void btnHoTen_Click(object sender, EventArgs e)
{
    lblHoTen.Text = txtHo.Text + " " + txtTen.Text;
}

private void btnthoat_Click(object sender, EventArgs e)
{
    this.Close();
}

private void lblHoTen_DoubleClick(object sender, EventArgs e)
{
    lblHoTen.Text = "";
}
}

```

Bài 4

viết chương trình dùng để quy đổi tiền từ VND sang USD và ngược lại, từ VND sang EUR và ngược lại như hình:

Yêu cầu:

- Khi số tiền quy đổi không phải là số hay để trống thì thông báo lỗi vào erro provider
- Khi nút VND to USD: thực hiện quy đổi từ VND sang USD(với tỷ giá là: 1USD =17861 VND(và hiện thị vào lbl kết quả, làm tròn 2 chữ số thập phân
- Khi nút USD to VND: thực hiện quy đổi từ USD sang VND(với tỷ giá là: 1USD =17861 VND(và hiện thị vào lbl kết quả, làm tròn 2 chữ số thập phân
- Khi nút VND to EUR: thực hiện quy đổi từ VND sang EUR (với tỷ giá là: 1USD =27043 VND(và hiện thị vào lbl kết quả, làm tròn 2 chữ số thập phân
- Khi nút EUR to VND: thực hiện quy đổi từ EUR sangVND(với tỷ giá là: 1USD =27043 VND(và hiện thị vào lbl kết quả, làm tròn 2 chữ số thập phân.
- Sự kiện form closing: xác nhận người dùng có chắc chắn muốn thoát không.

```

public partial class Form1 : Form
{
    public Form1()

```

```

{
    InitializeComponent();
}
int er = 1;
decimal usd = 17861;
decimal eur = 27043;

private void btnVnd2Usd_Click(object sender, EventArgs e)
{
    if (er == 0)
    {
        decimal kq = (decimal)Convert.ToDecimal(txtTien.Text) / usd;
        txtKetQua.Text = Convert.ToString(Math.Round(kq, 2));
    }
}

private void Form1_Load(object sender, EventArgs e)
{
    txtTien.Focus();
}

private void btnVnd2Eur_Click(object sender, EventArgs e)
{
    if (er == 0)
    {
        decimal kq = (decimal)Convert.ToDecimal(txtTien.Text) / eur;
        txtKetQua.Text = Convert.ToString(Math.Round(kq, 2));
    }
}

private void btnUsd2Vnd_Click(object sender, EventArgs e)
{
    if (er == 0)
    {
        decimal kq = (decimal)Convert.ToDecimal(txtTien.Text) * usd;
        txtKetQua.Text = Convert.ToString(Math.Round(kq, 2));
    }
}

private void btnEur2Vnd_Click(object sender, EventArgs e)
{
    if (er == 0)
    {
        decimal kq = (decimal)Convert.ToDecimal(txtTien.Text) * eur;
        txtKetQua.Text = Convert.ToString(Math.Round(kq, 2));
    }
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult q;
    q = MessageBox.Show("Bạn thật sự muốn thoát?", "Xác nhận", MessageBoxButtons.YesNo,
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
    if (q == DialogResult.No)
        e.Cancel = true;
}

private void txtTien_Leave(object sender, EventArgs e)
{
    Control ctr = (Control)sender;
    if (ctr.Text.Length == 0 || !Char.IsDigit(ctr.Text[ctr.Text.Length - 1]))
    {
        this.errorProvider1.SetError(ctr, "Tiền Nhập Sai");
    }
}

```

```

        er = 1;
    }
    else
    {
        this.errorProvider1.Clear();
        er = 0;
    }
}

private void btndoiusd_Click(object sender, EventArgs e)
{
    if(err==0)
        usd = Convert.ToDecimal(txtTigia.Text);
}

private void btndoiEUR_Click(object sender, EventArgs e)
{
    if (err == 0)
        eur = Convert.ToDecimal(txtTigia.Text);
}
int err=0;
private void txtTigia_TextChanged(object sender, EventArgs e)
{
    Control ctr = (Control)sender;
    if (ctr.Text.Length == 0 || !Char.IsDigit(ctr.Text[ctr.Text.Length - 1]))
    {
        this.errorProvider1.SetError(ctr, "Tiền Nhập Sai");
        err = 1;
    }
    else
    {
        this.errorProvider1.Clear();
        err = 0;
    }
}
}
}

```

Bài 5

Thiết kế giao diện như hình bên dưới

Yêu cầu

1. khi Form load.
 - a. Radiobutton Red được chọn mặc định
 - b. Con trỏ xuất hiện ngay tại ô Nhập tên.
2. Khi gõ tên vào ô Nhập tên (gõ ký tự nào thì label lập trình chạy song song cùng giá trị trong ô nhập tên)

```
public partial class frm1 : Form
{
    public frm1()
    {
        InitializeComponent();
    }

    private void txtNhapTen_TextChanged(object sender, EventArgs e)
    {
        lblLapTrinh.Text = txtNhapTen.Text;
    }

    private void btnthoat_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void frm1_Load(object sender, EventArgs e)
    {
        txtNhapTen.Focus();
        radRed.Select();
    }

    private void radRed_CheckedChanged(object sender, EventArgs e)
    {
        //loc ra radiobutt?

        RadioButton r = (RadioButton)sender;
        changecolor(r);
    }
    public void changecolor(RadioButton r)
    {
        switch (r.Name)
        {
            case "radRed":
                txtNhapTen.ForeColor = Color.Red;
                lblLapTrinh.ForeColor = Color.Red;
                break;
            case "radGreen":
                txtNhapTen.ForeColor = Color.Green;
                lblLapTrinh.ForeColor = Color.Green;
                break;
            case "radBlue":
                txtNhapTen.ForeColor = Color.Blue;
                lblLapTrinh.ForeColor = Color.Blue;
                break;
            case "radBlack":
                txtNhapTen.ForeColor = Color.Black;
                lblLapTrinh.ForeColor = Color.Black;
                break;
            default:
                break;
        }
    }
}
```



```

    }

    FontStyle f = FontStyle.Regular;
    private void chkBold_CheckedChanged(object sender, EventArgs e)
    {
        CheckBox a = (CheckBox)sender;
        switch (a.Name)
        {
            case "chkBold":
                f = FontStyle.Bold;
                break;
            case "chkItalic":
                f = FontStyle.Italic;
                break;
            case "chkUnderline":
                f = FontStyle.Underline;
                break;
            default:
                break;
        }
        changefont();
    }

    public void changefont()
    {
        lblLapTrinh.Font = new Font(lblLapTrinh.Font.Name, lblLapTrinh.Font.Size,
        lblLapTrinh.Font.Style ^ f);
    }
}

```

Bài 6

Thiết kế giao diện như hình bên dưới

Viết lệnh để chương trình hoạt động như sau:

Khi form vừa load lên thì:

Con trỏ được đặt tại vị trí ô Name

Hình CDROM lớn hiện ngược lại hình nhỏ ẩn.

Hướng dẫn: hiệu chỉnh property Visible của picSmall thành false (trong lúc thiết kế)

Đưa chuột vào hình CDROM (lớn, nhỏ) hiện lên tooltip “Click Me”

Hiệu chỉnh TabIndex sao cho hợp lý (Tham khảo hình vào Menu View\Tab Order)

lblMessage

Nhấn vào Button Display hoặc phím Enter thì gán nội dung của ô Name và Message cho Label Message (lblMessage)

- Nhấn vào Button Clear thì xóa nội dung trong ô Name và Message
- Nhấn vào RadioButton (Red, Green ...) thì đổi màu chữ trong ô Message thành màu tương ứng.
- Ẩn hiện Message (lblMessage) phụ thuộc vào giá trị của Check box (chkVisible)
- Nhấn vào Button Exit hoặc phím Esc thì đóng chương trình
- Click chuột vào hình CDROM nào thì hình đó ẩn đi và hiện hình CDROM khác lên
- Khi chương trình chạy hoàn chỉnh hãy hiệu chỉnh hai hình CDROM chồng lên nhau sao cho đồng tâm.

```
public partial class frm1 : Form
{
    public frm1()
    {
        InitializeComponent();
    }

    private void picNho_Click(object sender, EventArgs e)
    {
        picNho.Visible = false;
        picLon.Visible = true;
    }

    private void picLon_Click(object sender, EventArgs e)
    {
        picNho.Visible = true;
        picLon.Visible = false;
    }

    private void btnDisplay_Click(object sender, EventArgs e)
    {
        lblMessage.Text = txtName.Text + " : " + txtMessage.Text;
    }

    private void btnExit_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void frm1_FormClosing(object sender, FormClosingEventArgs e)
    {
        DialogResult r;
        r = MessageBox.Show("Bạn thật sự muốn thoát?", "Xác Nhận", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (r == DialogResult.No)
```

```

        e.Cancel = true;
    }

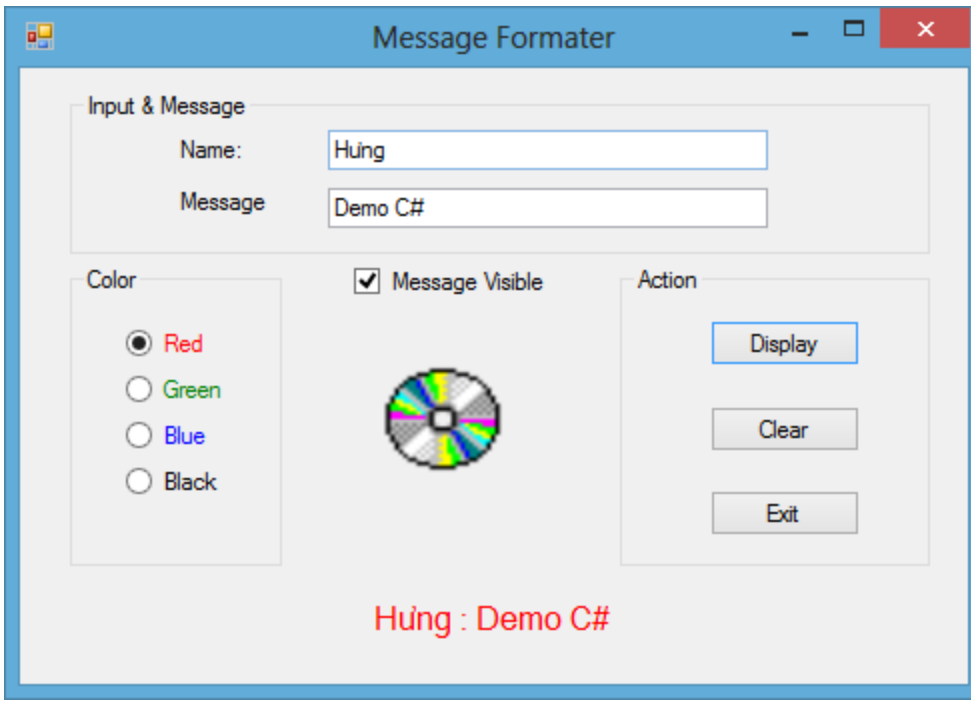
    private void btnClear_Click(object sender, EventArgs e)
    {
        txtMessage.Clear();
        txtName.Clear();
    }

    private void chkVisible_CheckedChanged(object sender, EventArgs e)
    {
        lblMessage.Visible = chkVisible.Checked;
    }

    private void radRed_CheckedChanged(object sender, EventArgs e)
    {
        RadioButton r = (RadioButton)sender;
        switch (r.Name)
        {
            case "radRed":
                lblMessage.ForeColor = Color.Red;
                break;
            case "radGreen":
                lblMessage.ForeColor = Color.Green;
                break;
            case "radBlue":
                lblMessage.ForeColor = Color.Blue;
                break;
            case "radBlack":
                lblMessage.ForeColor = Color.Black;
                break;
            default:
                break;
        }
    }

    private void frm1_Load(object sender, EventArgs e)
    {
        radBlack.Checked = true;
        chkVisible.Checked = true;
    }
}

```



Bài 7

Kết hợp sử dụng component ToolTip trên các control

Yêu cầu:

1. Khi chương trình vừa thực thi.
 - a. Form được hiển thị giữa màn hình
 - b. Các đối tượng được “neo” sao cho khi người dùng thay đổi kích thước form thì các đối tượng được resize kích thước sao cho “dễ nhìn”
 - c. picTurnOn được hiển thị, picTurnOff được ẩn ngay bên dưới picTurnOn.

Khi người dùng đưa chuột vào picTurnOn sẽ xuất hiện tip “Click me to Turn OFF the Light!”, và ngược lại trên picTurnOFF.

- d. Hiện tên mặc định vào ô Name
- e. Label ngay bên dưới pictureBox hiển thị thông báo

“Jack. Turn Off the Light ,please!”

```
public partial class FrmLight : Form
{
    public FrmLight()
    {
        InitializeComponent();
    }
    bool a = true;
    private void pictat_Click(object sender, EventArgs e)
    {
        pictat.Visible = false;
        picbat.Visible = true;
        a = false;
    }
}
```

```

        lblHienThi.Text = txtName.Text + ". Turn Off The Light, Pls!";
    }

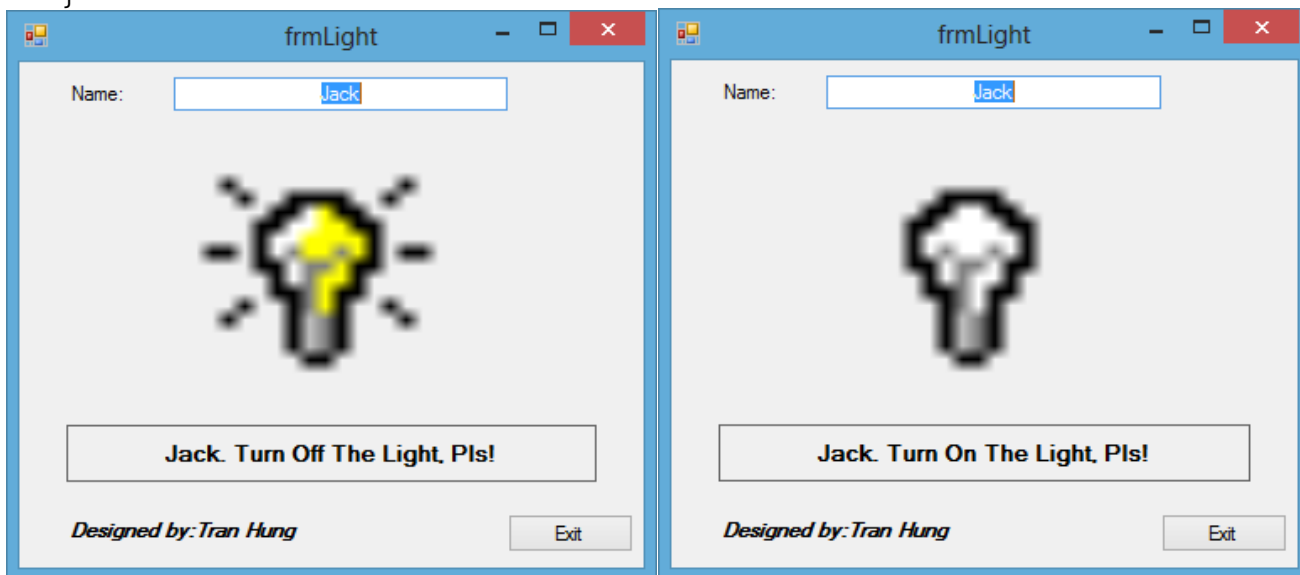
    private void picbat_Click(object sender, EventArgs e)
    {
        picbat.Visible = false;
        pictat.Visible = true;
        a = true;
        lblHienThi.Text = txtName.Text + ". Turn On The Light, Pls!";
    }

    private void FrmLight_Load(object sender, EventArgs e)
    {
        txtName.Text = "Jack";
        lblHienThi.Text = txtName.Text + ". Turn On The Light, Pls!";
    }

    private void btnexit_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void txtName_TextChanged(object sender, EventArgs e)
    {
        if(a==false)
            lblHienThi.Text = txtName.Text + ". Turn Off The Light, Pls!";
        else
            lblHienThi.Text = txtName.Text + ". Turn On The Light, Pls!";
    }
}

```



Bài 8

- Khi form hiện lên thì nút tính và nút xóa bị mờ. nếu dữ liệu nhập không hợp lệ thì báo lỗi. sau khi nhập dữ liệu đầy đủ và hợp lệ thì nút tính mới có tác dụng.
- Khi nhấn nút tính: tính nghiệm phương trình(xét tất cả trường hợp xảy ra:pt có 1 nghiệm, vô nghiệm, vô số nghiệm) và hiện kết quả vào nghiệm phương trình, khi đó nút xóa có tác dụng, nút tính bị mờ.
- Khi nhấn nút xóa: xóa các textbox và label, đặt con trỏ vào text box A, nút xóa bị mờ.
- Khi nhấn nút thoát: xác nhận người dùng có muốn thoát hay không,

```

public partial class frmptbn : Form
{
    public frmptbn()
    {

```

```

        InitializeComponent();
    }

    private void frmptbn_Load(object sender, EventArgs e)
    {
        txtA.Focus();
        btnTinh.Enabled = false;
        btnXoa.Enabled = false;
        txtKQ.Enabled = false;
    }

    public static String GiaiPTBac1(int a, float b)
    {
        if (a == 0 && b == 0)
        {
            return "Phuong trinh vo so nghiem";
        }
        else if (a == 0 && b != 0)
        {
            return "Phuong trinh vo nghiem";
        }
        else
        {
            return Convert.ToString(-b/a);
        }
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {
        if (txtA.Text.Length == 0 || txtB.Text.Length == 0 ||
            !Char.IsDigit(txtA.Text[txtA.Text.Length - 1]) || !Char.IsDigit(txtB.Text[txtB.Text.Length - 1]))
        {
            this.errorProvider1.SetError(txtA, "Giá Trị Nhập Chưa Đúng Hoặc Chưa đầy đủ");
            btnTinh.Enabled = false;
        }
        else
        {
            btnTinh.Enabled = true;
            this.errorProvider1.Clear();
        }
    }

    private void btnTinh_Click(object sender, EventArgs e)
    {
        txtKQ.Text = GiaiPTBac1(Convert.ToInt32(txtA.Text), Convert.ToInt32(txtB.Text));
        btnXoa.Enabled = true;
        btnTinh.Enabled = false;
    }

    private void btnXoa_Click(object sender, EventArgs e)
    {
        txtA.Clear();
        txtB.Clear();
        txtKQ.Clear();
        txtA.Focus();
        btnXoa.Enabled = false;
    }

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

```

```

private void frmptbn_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult r;
    r = MessageBox.Show("Bạn chắc chắn muốn thoát?", "Xác Nhận", MessageBoxButtons.YesNo,
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
    if (r == DialogResult.No)
        e.Cancel = true;
}
}

```

Bài 9

Sử dụng LinkLabel liên kết với một WebSite, mở ứng dụng

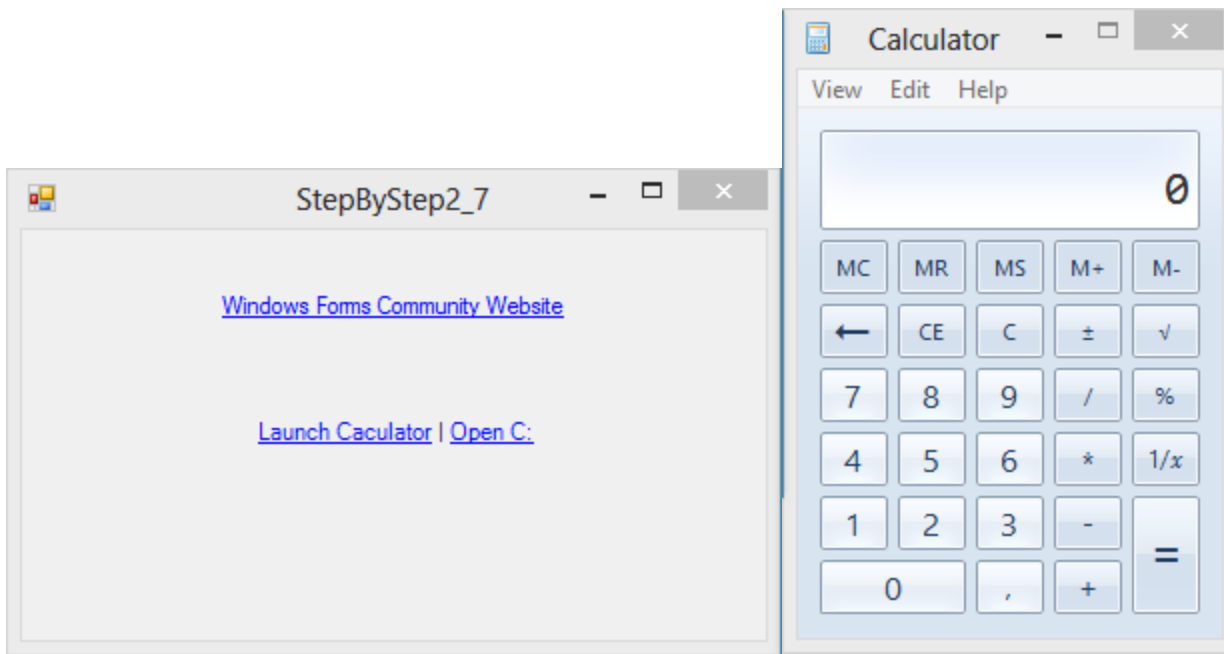
```

public partial class frm1 : Form
{
    public frm1()
    {
        InitializeComponent();
        lnkPrograms.Links.Add(0, "Launch Caculator".Length, "calc.exe");
        lnkPrograms.Links.Add(lnkPrograms.Text.IndexOf("Open C:"), "Open C:".Length, "C:\\");
    }

    private void lnkPrograms_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
    {
        System.Diagnostics.Process.Start(e.Link.LinkData.ToString());
    }

    private void lnk_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
    {
        lnkWinForms.LinkVisited = true;
        System.Diagnostics.Process.Start("FireFox", "http://www.google.com.vn");
    }
}

```

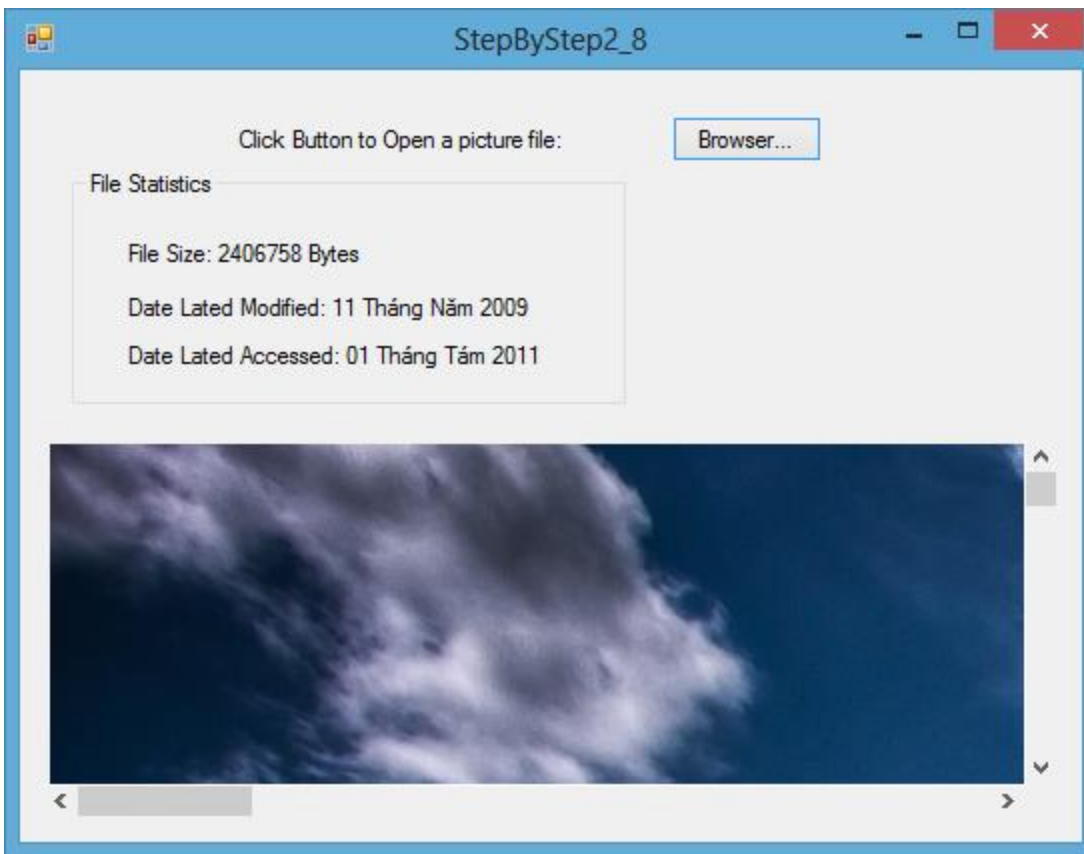


Bài 10

Sử dụng GroupBox và panel, PictureBox mở một File hình và xuất thông tin file đó

```
using System.IO;
public partial class frm1 : Form
{
    public frm1()
    {
        InitializeComponent();

        private void btnbrowser_Click(object sender, EventArgs e)
        {
            ofdPicture.Filter = "Image
Files(BMP,GIF,JPEG,etc.)|"+ "*.bmp;*.gif;*.jpg;*.jpeg;*.png;*.tif;*.tiff|" + "BMP Files(*.bmp)|*.bmp|" +
"PNG Files(*.png)|*.png|" +
"GIF Files (*.gif)|*.gif|" + "JPEG Files(*.jpg;*.jpeg)|*.jpg;*.jpeg|" + "TIF
Files(*.tif;*.tiff)|*.tif;*.tiff|" +
"All Files(*.*)|*.*";
            if (ofdPicture.ShowDialog() == DialogResult.OK)
            {
                FileInfo file = new FileInfo(ofdPicture.FileName);
                lblFileSize.Text=String.Format("File Size: {0} Bytes",file.Length.ToString());
                lblDateLM.Text = String.Format("Date Lated Modified: {0}",
file.LastWriteTime.ToLongDateString());
                lblDateLA.Text = String.Format("Date Lated Accessed: {0}",
file.LastAccessTime.ToLongDateString());
                picopen.Image = new Bitmap(ofdPicture.FileName);
            }
        }
    }
}
```

Bài 11

Viết chương trình định dạng Textbox theo các font chữ tương ứng radiobutton

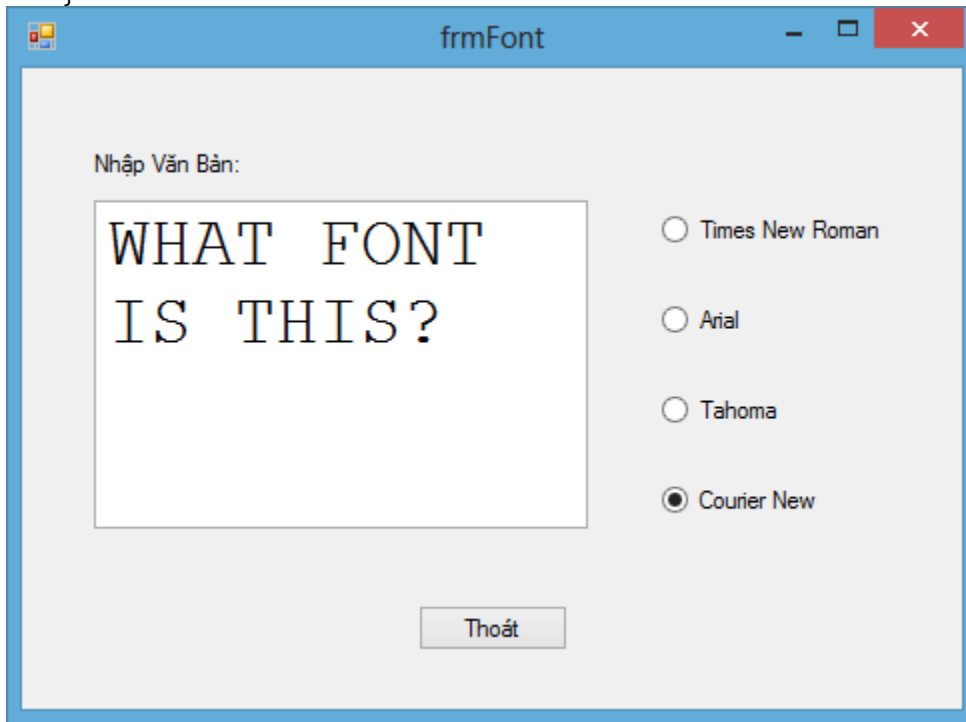
```
public partial class frmfont : Form
{
    public frmfont()
    {
        InitializeComponent();
    }

    private void frmfont_Load(object sender, EventArgs e)
    {
        radTNR.Checked = true;
    }

    private void radTNR_CheckedChanged(object sender, EventArgs e)
    {
        RadioButton r = (RadioButton)sender;
        switch (r.Name)
        {
            case "radTNR":
                txtchu.Font = new Font("Times New Roman", txtchu.Font.Size, txtchu.Font.Style);
                break;
            case "radArial":
                txtchu.Font = new Font("Arial", txtchu.Font.Size, txtchu.Font.Style);
                break;
            case "radTahoma":
                txtchu.Font = new Font("Tahoma", txtchu.Font.Size, txtchu.Font.Style);
                break;
            case "radCN":
                txtchu.Font = new Font("Courier New", txtchu.Font.Size, txtchu.Font.Style);
                break;
            default:
                break;
        }
    }
}
```

```
        break;
    }
}

private void btnthoat_Click(object sender, EventArgs e)
{
    this.Close();
}
}
```



Module 3 : GUI

Bài 1

Hãy tạo một Project dùng để quản lý giờ thuê máy của một phòng máy có giao diện và

yêu cầu như sau:

- Khi form vừa load lên màn hình thì
 - các CheckBox Bắt đầu có Enabled=True
 - các CheckBox Kết thúc có Enabled=False
 - Các Label có Caption= rỗng
- Khi người sử dụng Click vào CheckBox Bắt Đầu của một máy và nếu thuộc tính value của nó =1 thì:
 - Hiện Giờ hiện tại lên Label bắt đầu của máy
 - tương ứng
 - Cho thuộc tính Enabled của CheckBox Bắt đầu =False
 - Cho thuộc tính Enabled của CheckBox Kết thúc =True
- Khi người sử dụng Click vào CheckBox Kết thúc của một máy và nếu thuộc tính value của nó =1 thì:
 - Hiện Giờ hiện tại lên Label Kết thúc của máy tương ứng
 - Dựa vào Giờ và Phút (không cần quan tâm đến số giây) trên các label Bắt đầu và Kết thúc tính ra số tiền phải trả, rồi dùng MsgBox
 - để thông báo cho người sử dụng như hình bên (tiền thuê máy là 3000đ/giờ)
 - Xóa nội dung trong các Label Bắt đầu và Kết thúc
 - Cho thuộc tính Enabled của CheckBox Bắt đầu =True
 - Cho thuộc tính Enabled của CheckBox Kết thúc =False
 - Cho thuộc tính Value của CheckBox Bắt đầu =0
 - Cho thuộc tính Value của CheckBox Kết thúc =0
 - Khi người sử dụng Click vào nút Dừng chương trình thì kiểm tra nếu có máy đang thuê phải nhắc nhở người sử dụng bằng MsgBox, khi nào người sử dụng thật sự muốn kết thúc thì mới dừng chương trình.

```
public partial class frm1 : Form
{
    public frm1()
    {
        InitializeComponent();
    }

    private void frm1_Load(object sender, EventArgs e)
    {
        chkBatDau1.Enabled = true;
        chkBatDau2.Enabled = true;
        chkKetThuc1.Enabled = false;
        chkKetThuc2.Enabled = false;
        lblBatDau1.Text = "";
        lblKetThuc1.Text = "";
        lblBatDau2.Text = "";
        lblKetThuc2.Text = "";
    }

    private void chkBatDau1_CheckedChanged(object sender, EventArgs e)
    {
        if (chkBatDau1.Checked == true)
        {
            lblBatDau1.Text = DateTime.Now.ToLongTimeString();//Hien Thi gio.
            chkBatDau1.Enabled = false;
            chkKetThuc1.Enabled = true;
            chkKetThuc1.Checked = false;
        }
    }
}
```

```

    }
}

private void chkKetThuc1_CheckedChanged(object sender, EventArgs e)
{
    if (chkKetThuc1.Checked == true)
    {
        lblKetThuc1.Text = DateTime.Now.ToLongTimeString();

        DateTime batdau = Convert.ToDateTime(lblBatDau1.Text);
        DateTime kethuc = Convert.ToDateTime(lblKetThuc1.Text);
        TimeSpan sogiothue = kethuc - batdau;
        double dbgio = sogiothue.TotalHours;
        double dbtien = dbgio * 3000;
        MessageBox.Show("Số giờ thuê là: " + sogiothue + " h\n" + "Số Tiền phải trả: " +
Math.Round(dbtien,2) + " VND", "Thông báo máy 1!");

        chkBatDau1.Checked = false;
        chkBatDau1.Enabled = true;
        chkKetThuc1.Checked = false;
        chkKetThuc1.Enabled = false;

        lblBatDau1.Text = "";
        lblKetThuc1.Text = "";
    }
}

private void chkBatDau2_CheckedChanged(object sender, EventArgs e)
{
    if (chkBatDau2.Checked == true)
    {
        lblBatDau2.Text = DateTime.Now.ToLongTimeString();//Hien Thi gio.
        chkBatDau2.Enabled = false;
        chkKetThuc2.Enabled = true;
        chkKetThuc2.Checked = false;
    }
}

private void chkKetThuc2_CheckedChanged(object sender, EventArgs e)
{
    if (chkKetThuc2.Checked == true)
    {
        lblKetThuc2.Text = DateTime.Now.ToLongTimeString();

        DateTime batdau = Convert.ToDateTime(lblBatDau2.Text);
        DateTime kethuc = Convert.ToDateTime(lblKetThuc2.Text);
        TimeSpan sogiothue = kethuc - batdau;
        double dbgio = sogiothue.TotalHours;
        double dbtien = dbgio * 3000;
        MessageBox.Show("Số giờ thuê là: " + sogiothue + " h\n" + "Số Tiền phải trả: " +
Math.Round(dbtien, 2) + " VND", "Thông báo máy 2!");

        chkBatDau2.Checked = false;
        chkBatDau2.Enabled = true;
        chkKetThuc2.Checked = false;
        chkKetThuc2.Enabled = false;
    }
}

```

```

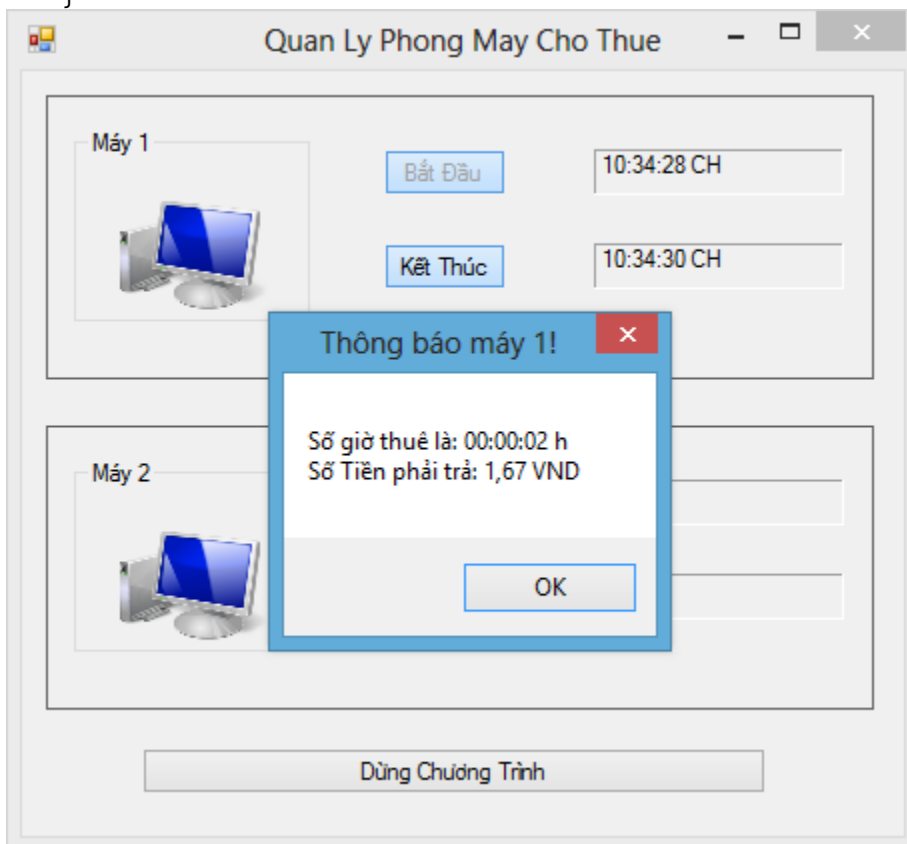
        lblBatDau2.Text = "";
        lblKetThuc2.Text = "";
    }

}

private void frm1_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult r;
    r = MessageBox.Show("Bạn chắc chắn muốn thoát?", "Xác Nhận!", MessageBoxButtons.YesNo,
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
    if (r == DialogResult.No)
        e.Cancel = true;
}

private void btnThoat_Click(object sender, EventArgs e)
{
    if (chkBatDau1.Checked == true || chkBatDau2.Checked == true)
        MessageBox.Show("Còn máy chưa tính tiền, hãy hoàn tất!", "Thông báo!",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
    else
        this.Close();
}
}

```



Bài 2

Viết chương trình hiển thị lá cờ tương ứng với nước được chọn trên radiobutton tương ứng

```

public partial class frmQG : Form
{
    public frmQG()
    {
        InitializeComponent();
    }
}

```

```

    }

    private void radVN_CheckedChanged(object sender, EventArgs e)
    {
        RadioButton r = (RadioButton)sender;
        if (radCach1.Checked == true)
            doihinhcach1(r);
        else if (radCach2.Checked == true)
            doihinhcach2(r);
    }

    void doihinhcach1(RadioButton r)
    {
        switch (r.Name)
        {
            case "radVN":
                picFlag.Image = Properties.Resources.VN;
                break;
            case "radUSA":
                picFlag.Image = Properties.Resources.USA;
                break;
            case "radItalian":
                picFlag.Image = Properties.Resources.Italian;
                break;
            case "radPhilippine":
                picFlag.Image = Properties.Resources.Philippine;
                break;
            default:
                break;
        }
    }

    void doihinhcach2(RadioButton r)
    {
        switch (r.Name)
        {
            case "radVN":
                picFlag.Image = this.imageList1.Images[0];
                break;
            case "radUSA":
                picFlag.Image = this.imageList1.Images[1];
                break;
            case "radItalian":
                picFlag.Image = this.imageList1.Images[2];
                break;
            case "radPhilippine":
                picFlag.Image = this.imageList1.Images[3];
                break;
            default:
                break;
        }
    }

    private void frmQG_Load(object sender, EventArgs e)
    {
        radCach1.Checked = true;
        radVN.Checked = true;
        picFlag.Image = Properties.Resources.VN;
    }
}

```



Bài 3

Viết chương trình cho phép nhập số vào ComboBox, khi chọn từ combobox thì xuất các ước số của số đó lên ListBox, Có các button tính tổng các ước số, số lượng ước số chẵn, số lượng ước số nguyên tố

Yêu cầu:

- Khi form vừa hiện lên, các textbox, combobox, listbox chưa có dữ liệu, con trỏ đặt tại textbox
- Nhấn nút “cập nhật” hoặc Enter: thêm số vừa nhập ở textbox vào combobox, đồng thời xóa nội dung textbox và đặt con trỏ tại textbox.
- Khi chọn 1 số trên combobox thì danh sách các ước số của số này sẽ hiện thị vào listbox bên phải tương ứng.

Khi nhấn các nút “tổng các ước số”, “số lượng các ước số chẵn”, “số lượng các ước số nguyên tố” sẽ hiện thị thông tin tương ứng vào messagebox dựa vào các ước số trên listbox.’

```
public partial class frmCBB : Form
{
    public frmCBB()
    {
        InitializeComponent();
    }

    private void frmCBB_Load(object sender, EventArgs e)
    {
        txtso.Clear();
        cboDanhsach.Text = "";
        lstUocSo.Text = "";
        txtso.Focus();
    }
    public bool kiemtraso(TextBox a)
    {
        for (int i = 0; i < a.Text.Length; i++)
            if (!Char.IsDigit(a.Text[i]))
                return true;
        return false;
    }
}
```

```

private void btncapnhat_Click(object sender, EventArgs e)
{
    if (txtso.Text.Length == 0 || kiemtraso(txtso))
        MessageBox.Show("Bạn nhập không đúng.", "Thông báo", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
    else
    {
        cboDanhsach.Items.Add(txtso.Text);
        txtso.Clear();
        txtso.Focus();
    }
}

private void cboDanhsach_SelectedIndexChanged(object sender, EventArgs e)
{
    int a = Convert.ToInt32(cboDanhsach.SelectedItem.ToString());
    lstUocSo.Items.Clear();
    for (int i = 1; i < a; i++)
        if (a % i == 0)
            lstUocSo.Items.Add(i);
}

private void btnTongUocSo_Click(object sender, EventArgs e)
{
    int tong = 0;
    for (int i = 0; i < lstUocSo.Items.Count; i++)
        tong += Convert.ToInt32(lstUocSo.Items[i].ToString());
    MessageBox.Show("Tổng các ước số là: " + tong, "Message", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}

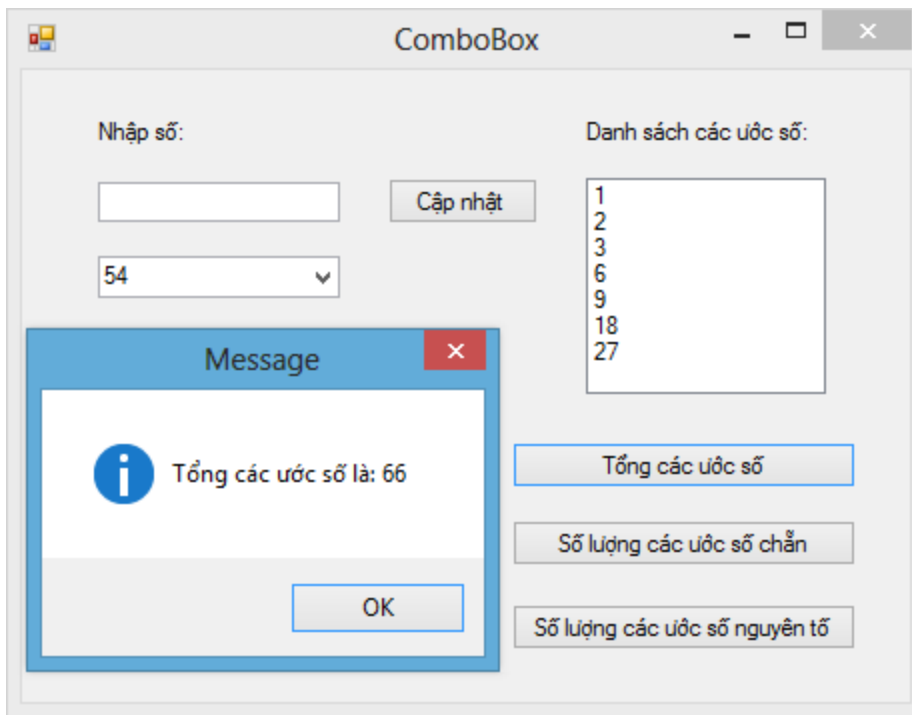
private void btnTongUocSoChan_Click(object sender, EventArgs e)
{
    int tong = 0;
    for (int i = 0; i < lstUocSo.Items.Count; i++)
        if (Convert.ToInt32(lstUocSo.Items[i].ToString())%2==0)
            tong += Convert.ToInt32(lstUocSo.Items[i].ToString());
    MessageBox.Show("Tổng các ước số chẵn là: " + tong, "Message", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}

public bool lasonguyento(int a)
{
    for (int i = 2; i < a; i++)
        if (a % i == 0)
            return false;
    return true;
}

private void btnTongUocSoNT_Click(object sender, EventArgs e)
{
    int tong = 0;
    for (int i = 1; i < lstUocSo.Items.Count; i++)
        if (lasonguyento(Convert.ToInt32(lstUocSo.Items[i].ToString())))
            tong++;
    MessageBox.Show("Số lượng ước số là số nguyên tố: " + tong, "Message",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}

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

```

Bài 4

Viết chương trình cho phép nhập phần tử cho ListBox, có các button thực hiện chức năng tính Tổng các phần tử trong ListBox, xóa phần tử đầu và cuối, xóa phần tử đang chọn, tăng mỗi phần tử lên 2, thay bằng bình phương, chọn số chẵn, chọn số lẻ

Mục đích:

Thao tác dữ liệu số trên Listbox

Yêu cầu

- Khi form vừa hiện lên, các textbox, listbox để trống, con trỏ đặt tại textbox
- Thiết kế form và thực hiện các chức năng sau:
- Khi người sử dụng nhập một số vào textbox rồi Enter hoặc nhấp vào nút cập nhật thì số này
- được thêm vào listbox, đồng thời nội dung trong textbox bị xóa và focus được chuyển về
- textbox.
- Người dùng nhấn vào nút nào thì thực hiện chức năng tương ứng của nút đó.
- Thiết lập thuộc tính Anchor hợp lý cho các control
- Thiết lập minimumSize cho form

```
public partial class frmListBox : Form
{
    public frmListBox()
    {
        InitializeComponent();
    }

    private void frmListBox_Load(object sender, EventArgs e)
    {
        txtNhap.Clear();
        lstDanhSach.Items.Clear();
        txtNhap.Focus();
    }
}
```

```

public bool kiemtraso(TextBox a)
{
    for (int i = 0; i < a.Text.Length; i++)
        if (!Char.IsDigit(a.Text[i]))
            return true;
    return false;
}

private void btnNhap_Click(object sender, EventArgs e)
{
    if (txtNhap.Text.Length == 0 || kiemtraso(txtNhap))
        MessageBox.Show("Bạn nhập không đúng.", "Thông báo", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
    else
    {
        lstDanhSach.Items.Add(txtNhap.Text);
        txtNhap.Clear();
        txtNhap.Focus();
    }
}

private void btnTongcacphantu_Click(object sender, EventArgs e)
{
    int tong = 0;
    for (int i = 0; i < lstDanhSach.Items.Count; i++)
        tong += Convert.ToInt32(lstDanhSach.Items[i].ToString());
    MessageBox.Show("Tổng các phần tử trong List là: " + tong, "Message", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}

private void btnXoadauvacuoi_Click(object sender, EventArgs e)
{
    int x = lstDanhSach.Items.Count;
    lstDanhSach.Items.RemoveAt(x-1);
    if (lstDanhSach.Items.Count!=0)
        lstDanhSach.Items.RemoveAt(0);
}

private void btnXoaPhanTuDangChon_Click(object sender, EventArgs e)
{
    int i = lstDanhSach.SelectedIndex;
    if(i== -1)
        MessageBox.Show("Bạn chưa chọn phần tử", "Thông báo!", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
    else
        lstDanhSach.Items.RemoveAt(i);
}

private void btnTanglen2_Click(object sender, EventArgs e)
{
    for (int i = 0; i < lstDanhSach.Items.Count; i++)
        lstDanhSach.Items[i] = Convert.ToInt32(lstDanhSach.Items[i].ToString()) + 2;
}

private void btnBinhPhuong_Click(object sender, EventArgs e)
{
    int i = lstDanhSach.SelectedIndex;
    if (i == -1)
        MessageBox.Show("Bạn chưa chọn phần tử", "Thông báo!", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
    else
        lstDanhSach.Items[i] = Math.Pow(Convert.ToInt32(lstDanhSach.Items[i].ToString()), 2);
}

```

```

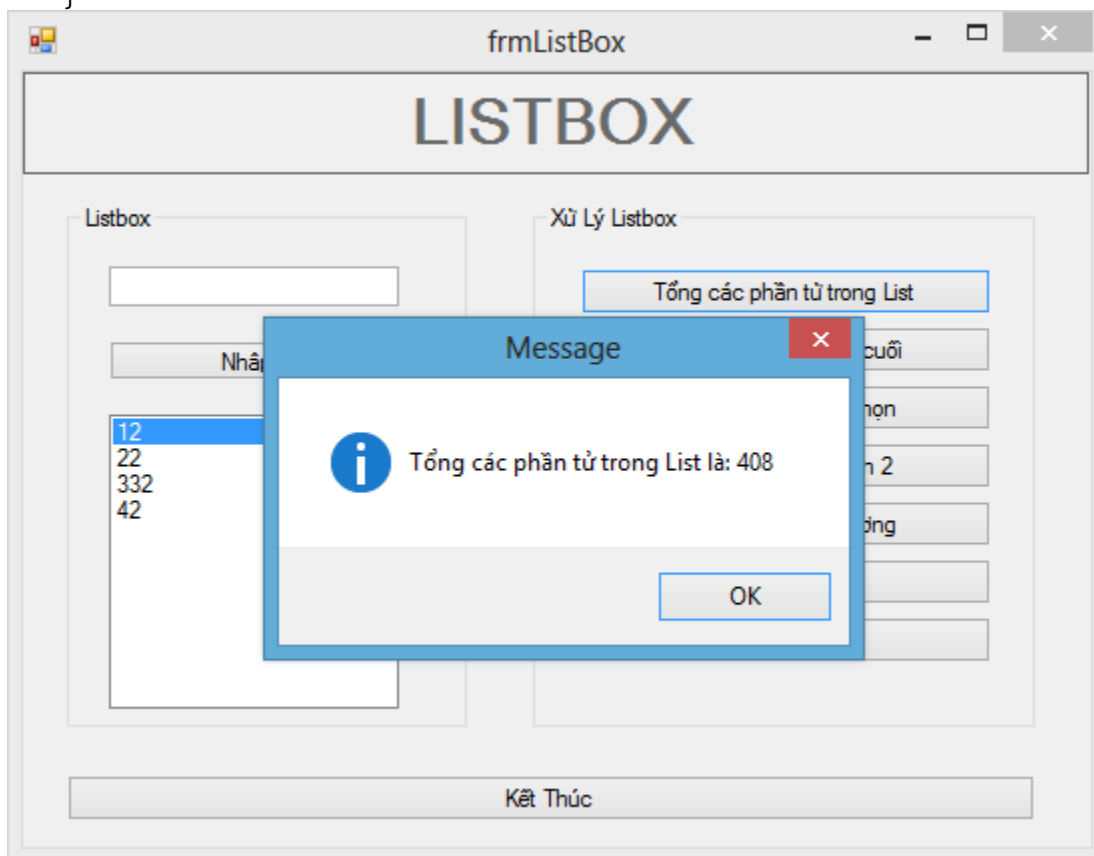
}

private void btnSoChan_Click(object sender, EventArgs e)
{
    for (int i = 0; i < lstDanhSach.Items.Count; i++)
        if (Convert.ToInt32(lstDanhSach.Items[i].ToString()) % 2 == 0)
        {
            lstDanhSach.SetSelected(i, true);
            break;
        }
}

private void btnSoLe_Click(object sender, EventArgs e)
{
    for (int i = 0; i < lstDanhSach.Items.Count; i++)
        if (Convert.ToInt32(lstDanhSach.Items[i].ToString()) % 2 != 0)
        {
            lstDanhSach.SetSelected(i, true);
            break;
        }
}

private void btnKetthuc_Click(object sender, EventArgs e)
{
    this.Close();
}
}

```



Bài 5

Tạo Form cho phép nhập SV cho 2 lớp là 2 ListBox, cho phép chuyển qua lại từng phần tử hoặc chuyển toàn bộ phần tử giữ 2 Lớp. cho phép xóa phần tử được chọn

Yêu cầu

Thiết kế form và thực hiện các chức năng sau

- Quy định form hiển thị giữa màn hình. Không cho người sử dụng thay đổi kích thước form.
- Quy định việc di chuyển Tab hợp lý.
- Các ListBox được phép chọn nhiều mục (kết hợp giữa phím Shift, Control và chuột)
- Khi người sử dụng Click nút “Cập nhật” hoặc nhấn phím Enter thì mất định nhập tên sinh viên từ textbox vào danh sách lớp A (không chấp nhận dữ liệu rỗng). Theo đó người sử dụng có thể chọn lớp để cập nhật từ combobox lớp.
- Các nút “>” và “<” khi được Click sẽ di chuyển tất cả các mục đang chọn sang ListBox bên kia tương ứng.
- Các nút “>>” và “<<” khi được Click sẽ di chuyển toàn các mục sang Listbox bên kia tương ứng.
- Nút lệnh “Xóa lớp A”, “Xóa lớp B” cho phép xóa các mục đang chọn trong list hiện hành.

```
public partial class frmupdate2lst : Form
{
    public frmupdate2lst()
    {
        InitializeComponent();
    }

    private void frmupdate2lst_Load(object sender, EventArgs e)
    {
        ngay();
        timer1.Start();
        cboLop.SelectedIndex = 0;
    }

    public void ngay()
    {
        lblngaythang.Text = DateTime.Now.ToLongDateString();
    }

    public void Gio()
    {
        lblgio.Text = DateTime.Now.ToLongTimeString();
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        Gio();
    }

    void capnhat()
    {
        if (txtTen.Text == "")
```

```

        {
            MessageBox.Show("Bạn chưa nhập tên", "Thông báo!", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
            txtTen.Focus();
        }
        else
        {
            switch (cboLop.SelectedIndex)
            {
                case 0:
                    lstA.Items.Add(txtTen.Text);
                    txtTen.Clear();
                    txtTen.Focus();
                    break;
                case 1:
                    lstB.Items.Add(txtTen.Text);
                    txtTen.Clear();
                    txtTen.Focus();
                    break;
                default:
                    break;
            }
        }
    }

    private void btncapnhat_Click(object sender, EventArgs e)
    {
        capnhat();
    }

    public void Chuyendulieu(ListBox a, ListBox b)
    {
        int i = 0;
        while (i < a.Items.Count)
            if (a.GetSelected(i))
            {
                b.Items.Add(a.Items[i].ToString());
                a.Items.RemoveAt(i);
            }
            else
                i++;
    }

    public void chuyentoanbo(ListBox a, ListBox b)
    {
        int i = 0;
        while (i < a.Items.Count)
        {
            b.Items.Add(a.Items[i].ToString());
            i++;
        }
        a.Items.Clear();
    }

    private void btnchuyen1AB_Click(object sender, EventArgs e)
    {
        DialogResult hoi;
        if (lstA.Items.Count > 0)
        {
            hoi = MessageBox.Show("Bạn chắc chắn muốn chuyển dữ liệu đang chọn?", "Xác nhận",
MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);

```

```

        if (hỏi == DialogResult.Yes)
            Chuyendulieu(lstA, lstB);
    }
    else
        MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
}

private void btnChuyenhetAB_Click(object sender, EventArgs e)
{
    DialogResult hỏi;
    if (lstA.Items.Count > 0)
    {
        hỏi = MessageBox.Show("Bạn chắc chắn muốn chuyển toàn bộ dữ liệu?", "Xác nhận",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (hỏi == DialogResult.Yes)
            chuyentoanbo(lstA, lstB);
    }
    else
        MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
}

private void btnchuyen1BA_Click(object sender, EventArgs e)
{
    DialogResult hỏi;
    if (lstB.Items.Count > 0)
    {
        hỏi = MessageBox.Show("Bạn chắc chắn muốn chuyển dữ liệu đang chọn?", "Xác nhận",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (hỏi == DialogResult.Yes)
            Chuyendulieu(lstB, lstA);
    }
    else
        MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
}

private void btnChuyenhetBA_Click(object sender, EventArgs e)
{
    DialogResult hỏi;
    if (lstB.Items.Count > 0)
    {
        hỏi = MessageBox.Show("Bạn chắc chắn muốn chuyển toàn bộ dữ liệu?", "Xác nhận",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (hỏi == DialogResult.Yes)
            chuyentoanbo(lstB, lstA);
    }
    else
        MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
}

public void xoadulieu(ListBox a)
{
    int i = 0;
    while (i < a.Items.Count)
        if (a.GetSelected(i))
            a.Items.RemoveAt(i);
        else
            i++;
}

private void btnXoaA_Click(object sender, EventArgs e)
{

```

```

        DialogResult hoi;
        if (lstA.Items.Count > 0)
        {
            hoi = MessageBox.Show("Bạn chắc chắn muốn xóa dữ liệu đang chọn?", "Xác nhận",
MessageBoxButtons.YesNo,
            MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
            if (hoi == DialogResult.Yes)
                xoadulieu(lstA);
        }
        else
            MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
    }

    private void btnXoaB_Click(object sender, EventArgs e)
    {
        DialogResult hoi;
        if (lstB.Items.Count > 0)
        {
            hoi = MessageBox.Show("Bạn chắc chắn muốn xóa dữ liệu đang chọn?", "Xác nhận",
MessageBoxButtons.YesNo,
            MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
            if (hoi == DialogResult.Yes)
                xoadulieu(lstB);
        }
        else
            MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
    }

    private void btnKetThuc_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void frmupdate2lst_FormClosing(object sender, FormClosingEventArgs e)
    {
        DialogResult r;
        r = MessageBox.Show("Bạn chắc chắn muốn thoát?", "Xác Nhận!", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (r == DialogResult.No)
            e.Cancel = true;
    }

    private void thôngTinToolStripMenuItem_Click(object sender, EventArgs e)
    {
        String thongtin = "Họ Tên: Trần Nguyên Hưng\nMASSV:11007922\n" + "Lớp:CDTH13K";
        MessageBox.Show(thongtin);
    }

    private void cậpNhapLớpBToolStripMenuItem_Click(object sender, EventArgs e)
    {
        cboLop.SelectedIndex = 1;
        capnhat();
    }

    private void cậpNhapLớpAToolStripMenuItem_Click(object sender, EventArgs e)
    {
        cboLop.SelectedIndex = 0;
        capnhat();
    }
}

```

Bài 6

Viết chương trình cho phép sinh viên đăng ký các môn học trong học kỳ

Yêu cầu:

Khi form hiện lên, các ô nhập đều để trống

Nút đăng kí : hiện thị các thông tin mà sinh viên đã đăng kí lên message box như hình

```
public partial class frmdkmonhoc : Form
{
    public frmdkmonhoc()
    {
        InitializeComponent();
    }

    private void frmdkmonhoc_Load(object sender, EventArgs e)
    {
        txtHovaTen.Clear();
        txtMssv.Clear();
        huycheckboxlistbox(chklistboxMonHoc);
        //Hủy chọn combobox
        cboLop.SelectedIndex = -1;
        cboNienKhoa.SelectedIndex = -1;
        radhk1.Checked = true;
        txtMssv.Focus();
    }

    public string CacMonhocdangchon()
    {
        int dem = 0;
        string chuoii="";
        //Duyệt từng item được chọn trong chklistboxMonHoc
        foreach (var item in chklistboxMonHoc.CheckedItems)
```



```

        {
            dem++;
            chuoi += dem.ToString() + ". " + item.ToString() + "\n";
        }
        return chuoi;
    }

    public void huycheckboxlistbox(CheckedListBox lst)
    {
        while (lst.CheckedIndices.Count > 0)
        {
            //lst.CheckedIndices[0] lấy vị trí được chọn đầu tiên
            lst.SetItemChecked(lst.CheckedIndices[0], false);
        }
    }
    int hk = 0;
    private void btnDK_Click(object sender, EventArgs e)
    {
        if (txtHovaTen.Text.Length == 0 || txtMssv.Text.Length == 0 || cboNienKhoa.SelectedIndex ==
        -1 || cboLop.SelectedIndex == -1 ||
            chkListBoxMonHoc.CheckedIndices.Count == 0)
            MessageBox.Show("Chưa đủ thông tin");
        else
        {
            string loinhon = "";
            loinhon += "Sinh Viên: " + txtHovaTen.Text.ToString() + " MSSV: " +
            txtMssv.Text.ToString() + "\n";
            loinhon += "Lớp: " + cboLop.SelectedItem.ToString() + "\n";
            loinhon += "Niên Khóa: " + cboNienKhoa.SelectedItem.ToString() + "\n";
            loinhon += "Đã đăng ký học Học kỳ " + hk + " Các môn như sau:\n";
            loinhon += CacMonhocdangchon();
            MessageBox.Show(loinhon);
        }
    }

    private void radhk1_CheckedChanged(object sender, EventArgs e)
    {
        RadioButton ra = (RadioButton)sender;
        switch (ra.Name)
        {
            case "radhk1":
                hk = 1;
                break;
            case "radhk2":
                hk = 2;
                break;
            case "radhk3":
                hk = 3;
                break;
            case "radhk4":
                hk = 4;
                break;
            default:
                break;
        }
    }

    private void btnHuy_Click(object sender, EventArgs e)
    {
        txtHovaTen.Clear();
        txtMssv.Clear();
        huycheckboxlistbox(chkListBoxMonHoc);
        radhk1.Checked = true;
        cboLop.SelectedIndex = -1;
    }

```

```

        cboNienKhoa.SelectedIndex = -1;
        txtMssv.Focus();
    }

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

```

Bài 7

Viết chương trình số 7 may mắn

Thiết kế giao diện như hình trên

- Groupbox chứa 3 labe.
- 2 Label dùng để hiển thị kết quả (canh lề phải)
- 3 button như hình vẽ.

Yêu cầu

Khi chương trình vừa bắt đầu

- 3 label sẽ hiển thị 3 số 7.
- Máy tính có 100 đồng
- Người chơi có 100 đồng

Khi Click nút quay số

Luật chơi:

Mỗi lần quay số, người chơi phải trả trước 30 đồng, tương ứng là máy sẽ được cộng thêm 30 đồng. Và nếu người chơi có số tiền ít hơn 30 đồng thì sẽ không được phép chơi tiếp

Nếu mỗi lần quay được số 7, người chơi sẽ được thưởng như sau

- Nếu ô đầu tiên là số 7 (random từ 0 đến 8), được thưởng 100 đồng + 50% số tiền của máy.

- Nếu ô số 2 là số 7(random từ 0 đến 9), người chơi được thưởng 30 đồng + 50% tiền của máy.

- Nếu ô số 3 là số 7(random từ 0 đến 10), người chơi được thưởng 10 đồng;

(số tiền người chơi sẽ được cộng dồn nếu cả 3 ô cùng là số 7, tiền máy sẽ giảm đi phần trăm tương ứng)

Nút game mới

Cho phép 1 lượt chơi mới hoàn toàn (set lại tên người chơi và tiền máy là 100)

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
    decimal tienmay = 100;
    decimal tiennguoi = 100;
    private void timer1_Tick(object sender, EventArgs e)
    {
        decimal tienthuong;
        int so1;
        int so2;
        int so3;
        timer1.Interval += 5;
        Random ran = new Random();
        so1 = ran.Next(0, 8);
        so2 = ran.Next(0, 9);
        so3 = ran.Next(0, 10);

        //xuất ra label
        lblSo1.Text = so1.ToString();
        lblSo2.Text = so2.ToString();
        lblSo3.Text = so3.ToString();

        if(timer1.Interval > 200)
        {
            timer1.Stop();
            if (so1 == 7)
            {
                tienthuong = 100 + (tienmay * 50) / 100;
                tiennguoi = tiennguoi + tienthuong;
                tienmay = tienmay - (tienmay * 50) / 100;
                MessageBox.Show("Bạn được thưởng " + tienthuong + " Đồng", "Lucky Seven!");
            }
            if (so2 == 7)
            {
                tienthuong = 30 + (tienmay * 50) / 100;
                tiennguoi += tienthuong;
                tienmay -= (tienmay * 50) / 100;
                MessageBox.Show("Bạn được thưởng " + tienthuong + " Đồng", "Lucky Seven!");
            }
        }
    }
}
```

```

        if (so3 == 7)
        {
            tienhuong = 100 + (tienmay * 50) / 100;
            tiennguoi += tienhuong;
            tienmay -= (tienmay * 50) / 100;
            MessageBox.Show("Bạn được thưởng " + tienhuong + " Đồng", "Lucky Seven!");
        }
        //Cập nhật số tiền
        lblTienMay.Text = tienmay.ToString();
        lblTienNguoi.Text = tiennguoi.ToString();
    }
}

private void btnQuaySo_Click(object sender, EventArgs e)
{
    if (tiennguoi > 30)
    {
        timer1.Enabled = true;
        timer1.Interval = 2;
        tienmay += 30;
        tiennguoi -= 30;
        lblTienNguoi.Text = tiennguoi.ToString();
        lblTienMay.Text = tienmay.ToString();

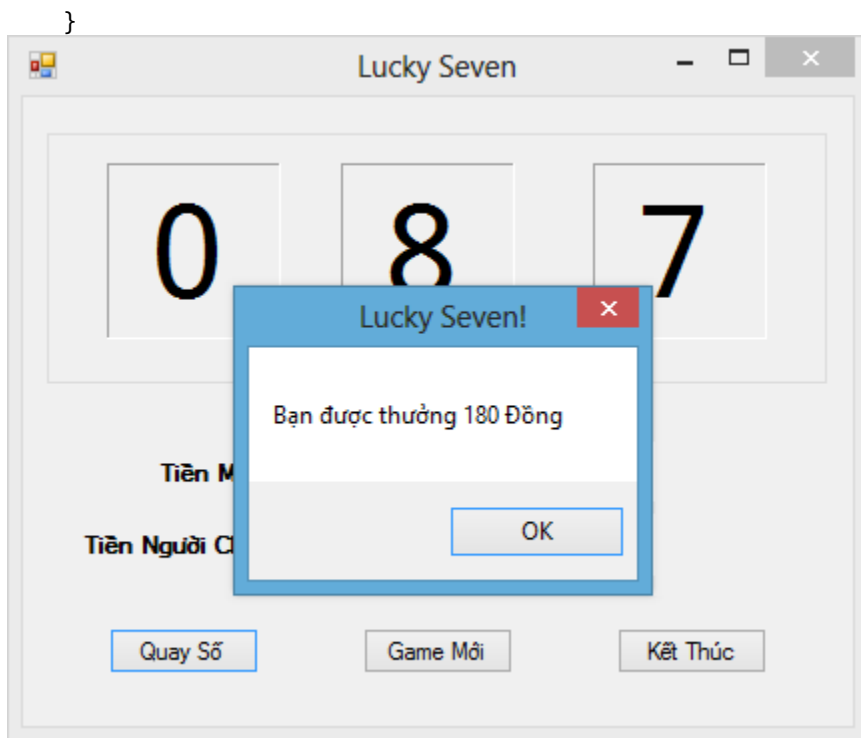
    }
    else
    {
        MessageBox.Show("Game Over.Bạn không đủ tiền để tiếp tục!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        timer1.Enabled = false;
        btnQuaySo.Enabled = false;
    }
}

private void btnGamemoi_Click(object sender, EventArgs e)
{
    lblSo1.Text = "7";
    lblSo2.Text = "7";
    lblSo3.Text = "7";
    lblTienMay.Text = "100";
    lblTienNguoi.Text = "100";
    btnQuaySo.Enabled = true;
    btnQuaySo.Focus();
    tienmay = 100;
    tiennguoi = 100;
}

private void btnKetThuc_Click(object sender, EventArgs e)
{
    this.Close();
}

private void Form1_Load(object sender, EventArgs e)
{
    lblSo1.Text = "7";
    lblSo2.Text = "7";
    lblSo3.Text = "7";
    lblTienMay.Text = "100";
    lblTienNguoi.Text = "100";
    btnQuaySo.Enabled = true;
    btnQuaySo.Focus();
    tienmay = 100;
    tiennguoi = 100;
}

```



Module 4 : GUI

Bài 1

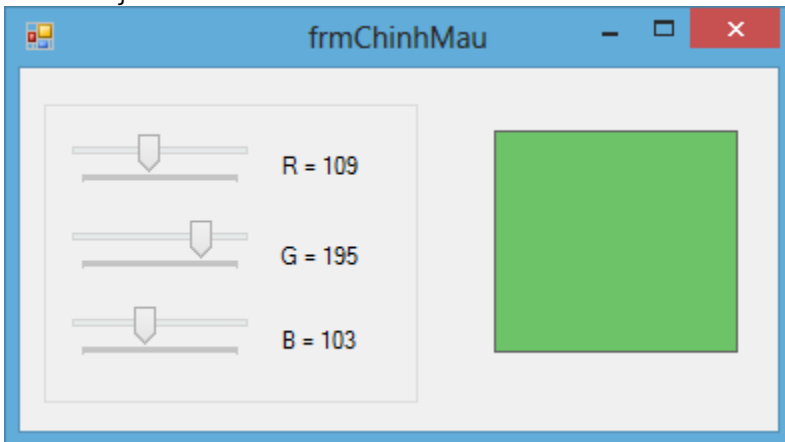
Thiết kế giao diện cho phép đổi màu panel như sau:

Để đổi màu panel, người dùng có thể kéo các Trackbar để thay đổi các giá trị màu red(R), green(G), và Blu(B). yêu cầu khi Trackbar nào kéo đến đâu thì giá trị của nó thay đổi đến đó tương ứng trên các label, các trackbar có giá trị từ 0 đến 255.

```
void ProgramInit()
{
    int nRed = trkRed.Value;
    int nGreen = trkGreen.Value;
    int nBlue = trkBlue.Value;
    lblRed.Text = "R = " + nRed;
    lblGreen.Text = "G = " + nGreen;
    lblBlue.Text = "B = " + nBlue;
    Color mau = Color.FromArgb(nRed, nGreen, nBlue);
    pnlColor.BackColor = mau;
}

private void trkRed_Scroll(object sender, EventArgs e)
{
    ProgramInit();
}

private void frm1_Load(object sender, EventArgs e)
{
    Color mau = Color.FromArgb(0,0,0);
    pnlColor.BackColor = mau;
}
```



Bài 2

Viết chương trình Lịch Tờ, Hiển thị ngày tháng được nhập theo kiểu chữ

Khi mới mở form sẽ hiện thị lịch theo ngày hiện hành. Khi thay đổi ngày, tháng, năm trên form(nhớ kiểm tra ngày hợp lệ) sau đó click nút xem lịch thì sẽ hiện thị đúng ngày tháng năm đó vào grupobox bên phải(bao gồm cả thứ trong tuần)

```
public partial class frmLich : Form
{
    public frmLich()
    {
        InitializeComponent();
    }
    //Mặc định PT DayOfWeek sẽ trả về tiếng anh nên chuyển về tiếng việt
    string DayOfWeekTextShow(object date)
    {

```

```

DateTime obj = (DateTime)date;
string ngaytrongtuan="";
switch (obj.DayOfWeek.ToString())
{
    case "Sunday":
        ngaytrongtuan = "Chủ Nhật";
        break;
    case "Monday":
        ngaytrongtuan = "Thứ Hai";
        break;
    case "Tuesday":
        ngaytrongtuan = "Thứ Ba";
        break;
    case "Wednesday":
        ngaytrongtuan = "Thứ Tư";
        break;
    case "Thursday":
        ngaytrongtuan = "Thứ Năm";
        break;
    case "Friday":
        ngaytrongtuan = "Thứ Sáu";
        break;
    case "Saturday":
        ngaytrongtuan = "Thứ Bảy";
        break;
    default:
        break;
}
return ngaytrongtuan;
}
// Month sẽ trả về tháng là số từ 1 đến 12 nên ta chuyển thành chữ
String MonthTextShow(object date)
{
    DateTime obj = (DateTime)date;
    string thangtrongnam = "";
    switch (obj.Month)
    {
        case 1:
            thangtrongnam = "Tháng Một";
            break;
        case 2:
            thangtrongnam = "Tháng Hai";
            break;
        case 3:
            thangtrongnam = "Tháng Ba";
            break;
        case 4:
            thangtrongnam = "Tháng Bốn";
            break;
        case 5:
            thangtrongnam = "Tháng Năm";
            break;
        case 6:
            thangtrongnam = "Tháng Sáu";
            break;
        case 7:
            thangtrongnam = "Tháng Bảy";
            break;
        case 8:
            thangtrongnam = "Tháng Tám";
            break;
        case 9:
            thangtrongnam = "Tháng Chín";
            break;
    }
}

```

```

        case 10:
            thangtrongnam = "Tháng Mười";
            break;
        case 11:
            thangtrongnam = "Tháng Mười Một";
            break;
        case 12:
            thangtrongnam = "Tháng Mười Hai";
            break;
        default:
            break;
    }
    return thangtrongnam;
}

void DateShow(object date) //Truyền vào kiểu đối tượng
{
    DateTime obj = (DateTime)date; //Ép kiểu về dạng ngày tháng
    grpNam.Text = "Lịch Năm " + nudNam.Value.ToString(); //Gán text của GrpNam bằng giá trị của
nudNam
    lblThang.Text = MonthTextShow(date);
    lblNgay.Text = nudNgay.Value.ToString();
    lblThu.Text = DayOfWeekTextShow(date);
}

private void frmLich_Load(object sender, EventArgs e)
{
    DateTime hientai = DateTime.Now;
    lblThu.Text = DayOfWeekTextShow(hientai);
    lblThang.Text = MonthTextShow(hientai);
    grpNam.Text = "Lịch Năm " + hientai.Year;
    lblNgay.Text = hientai.Day.ToString();
}

private void btnXemLich_Click(object sender, EventArgs e)
{
    int nam = (int)nudNam.Value;
    int thang = (int)nudThang.Value;
    int ngay = (int)nudNgay.Value;
    if (ngay > ngaytrongthang(thang, nam))
        MessageBox.Show("Ngày tháng không đúng", "Thông báo");
    else
    {
        DateTime date = new DateTime(nam, thang, ngay); //Tạo đối tượng kiểu DateTime
        DateShow(date); //Gọi hàm xuất thông tin ra giao diện
    }
}

//kiểm tra ngày hợp lệ, xét tháng nhập có bao nhiêu ngày
int ngaytrongthang(int thang, int nam)
{
    int songay = 0;
    switch (thang)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12: songay = 31; break;
        case 4:
        case 6:
        case 9:
        case 11: songay = 30; break;
    }
}

```



```

        case 2:
            if (nam % 400 == 0 || (nam % 4 == 0 && nam % 100 != 0)) // Năm Nhuận
            {
                songay = 29;
                break;
            }
            else
            {
                songay = 28;
                break;
            }
            default:
                break;
        }
        return songay;
    }
}

```

Bài 3

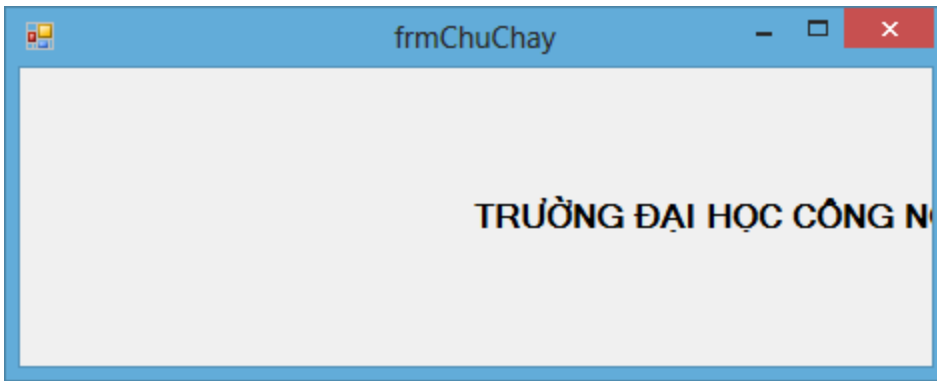
Thiết kế Form, khi form load lên thì dòng chữ “TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HCM” chạy ngang màn hình

```

private void Form1_Load(object sender, EventArgs e)
{
    timer1.Interval = 1;
    timer1.Enabled = true;
    timer1.Start();
}

private void timer1_Tick(object sender, EventArgs e)
{
    Point pos;
    pos = new Point(lblText.Location.X - 3, lblText.Location.Y);
    if (lblText.Location.X < -lblText.Size.Width)
    {
        pos = new Point(this.Size.Width, lblText.Location.Y);
    }
    lblText.Location = pos;
}

```

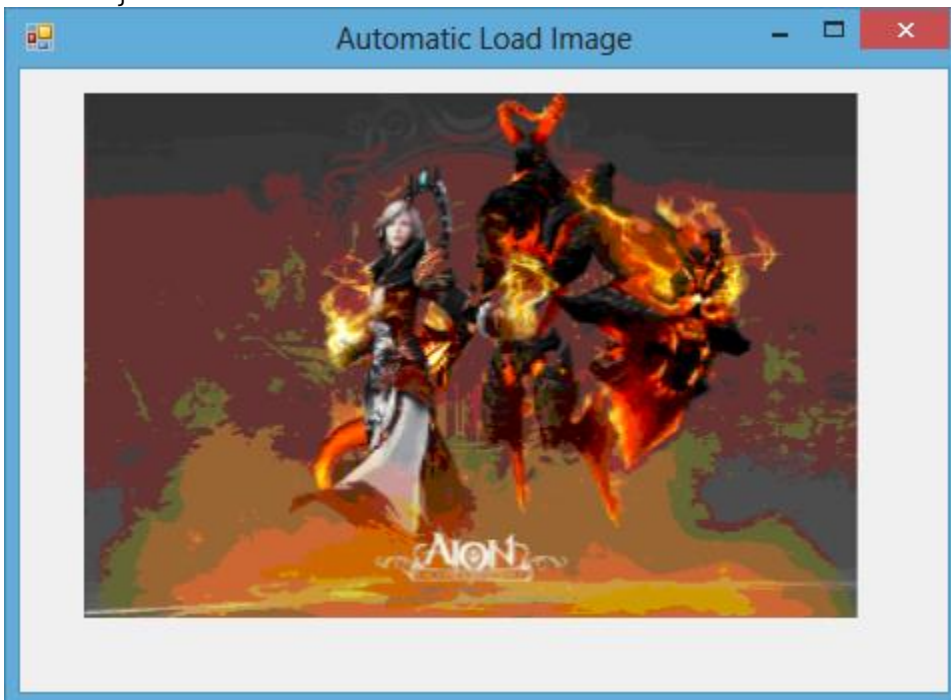


Bài 4

Viết chương trình tự thay đổi hình theo thời gian

```
private void timer1_Tick(object sender, EventArgs e)
{
    Random rnd = new Random();//Tạo biến Random
    int index = 0;
    index = rnd.Next(0, imageList1.Images.Count); //Lấy giá trị kiểu nguyên từ 0 đến số hình
    trong ImageList từ biến random
    picHinh.Image = imageList1.Images[index]; //Gán hình có thứ tự tương ứng vào PictureBox
}

private void frmAutoLoadImg_Load(object sender, EventArgs e)
{
    timer1.Enabled = true; //Timer có hiệu lực
    timer1.Start(); //Khởi chạy Timer
}
```



Bài 5

Thiết kế chương trình nhập thông tin nhân viên

Yêu cầu:

- Quy định của masktextbox phone là 000_000000
- Combobox country chỉ chứa 2 nước Việt Nam và Thailan(chứa 3 thành phố Pattaya, Chiangmai và Bangkok)

- Khi đang nhập 1 ô ,mà để trống và focus đến ô khác thì sẽ có thông báo lỗi và cho focus về ô cần nhập,
- Khi nhấn submit sẽ có 1 messagebox hiện thị đầy đủ thông tin vừa nhập

```

public partial class frmEmployeeDetails : Form
{
    public frmEmployeeDetails()
    {
        InitializeComponent();
    }

    bool FullDataInput()
    {
        int city=lstCity.SelectedIndex;
        int quali=lstQualification.SelectedIndex;
        int country=cboCountry.SelectedIndex;
        if (txtAddress.Text.Length == 0 ||
            txtEmail.Text.Length == 0 ||
            txtName.Text.Length == 0 ||
            mskBirth.Text.Length < 7 ||
            mskPhone.Text.Length < 7 ||
            city == -1 || quali == -1 || country == -1)
            return false;
        else
            return true;
    }

    string DataProcess()
    {
        string chuoi = "";
        if (!FullDataInput())
            return chuoi;
        chuoi += lblten.Text + txtName.Text + "\n";
        chuoi += lblNS.Text + mskBirth.Text + "\n" + lblDC.Text + txtAddress.Text + "\n" +
            lblTP.Text + lstCity.SelectedItem.ToString() + "\n" +
            lblQG.Text + cboCountry.SelectedItem.ToString() + "\n" +
            lblTrinhDo.Text + lstQualification.SelectedItem.ToString() + "\n" +
            lblSDT.Text + mskPhone.Text + "\n" +
            lblEmail.Text + txtEmail.Text + "\n" +
            lblThoiDiemVao.Text + dateJoining.Value.ToShortDateString();
        return chuoi;
    }

    private void btnSubmit_Click(object sender, EventArgs e)
    {
        if (DataProcess() == "")
            MessageBox.Show("Bạn chưa nhập đủ thông tin!", "Thông Báo");
        else
            MessageBox.Show(DataProcess(), "Thông Tin");
    }
    //Khi quốc gia thay đổi thì gọi hàm thay đổi thành phố
    private void cboCountry_SelectedIndexChanged(object sender, EventArgs e)
    {
        thanhpho();
    }
    //Hàm đưa thành phố ứng với quốc gia
    void thanhpho()
    {
        if (cboCountry.SelectedIndex == 0)
        {
            lstCity.Items.Clear();
            lstCity.Items.Add("Ho Chi Minh");
            lstCity.Items.Add("Nha Trang");
        }
    }
}

```

```

        lstCity.Items.Add("Ha Noi");
    }
    else if (cboCountry.SelectedIndex == 1)
    {
        lstCity.Items.Clear();
        lstCity.Items.Add("Pattaya");
        lstCity.Items.Add("ChiengMai");
        lstCity.Items.Add("Bangkok");
    }
}

private void btnExit_Click(object sender, EventArgs e)
{
    this.Close();
}
//Đối với Text Box
private void txtName_Leave(object sender, EventArgs e)
{
    TextBox t = (TextBox)sender;
    if (t.Text.Length == 0)
    {
        this.errorProvider1.SetError(t, "Bạn Chưa hoàn tất ô này");
        t.Focus();
    }
    else
        this.errorProvider1.Clear();
}

private void frmEmployeeDetails_Load(object sender, EventArgs e)
{
    cboCountry.SelectedIndex = 0;
}
//Đối với MaskedTextBox
private void mskBirth_Leave(object sender, EventArgs e)
{
    MaskedTextBox t = (MaskedTextBox)sender;
    if (t.Text.Length < 7)
    {
        this.errorProvider1.SetError(t, "Bạn Chưa hoàn tất ô này");
        t.Focus();
    }
    else
        this.errorProvider1.Clear();
}

private void lnkVNExpress_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    lnkVNExpress.LinkVisited = true;
    System.Diagnostics.Process.Start("Iexplore", "http://vnexpress.net/");
}
}

```

Bài 6

Viết chương trình sử dụng ProgressBar, sau 15s thì progressbar chạy xong thì form này tắt và hiện form bài 1

- Tạo giao diện form splash như hình, chứa progressbar và 1 nút Ok
- Khi khởi động chương trình thì form splash xuất hiện, form này dùng trong thời gian 15s.
- Thanh progressbar sẽ thể hiện tiến trình load form, sau 15s form load sẽ tự động tắt và khởi động form trong bài số 1
- Trong khi form splash đang hoạt động, nếu người dùng ấn vào nút OK thì Form này ngưng hoạt động, đồng thời form bài 1 được gọi hoạt động

```
private void frmAMM_Load(object sender, EventArgs e)
{
    timer1.Start(); //Khởi chạy Timer1
}
//Sau khoảng thời gian Interval=150ms thì cho progressBar1 nhảy 1 bước
private void timer1_Tick(object sender, EventArgs e)
{
    progressBar1.PerformStep();
}
```

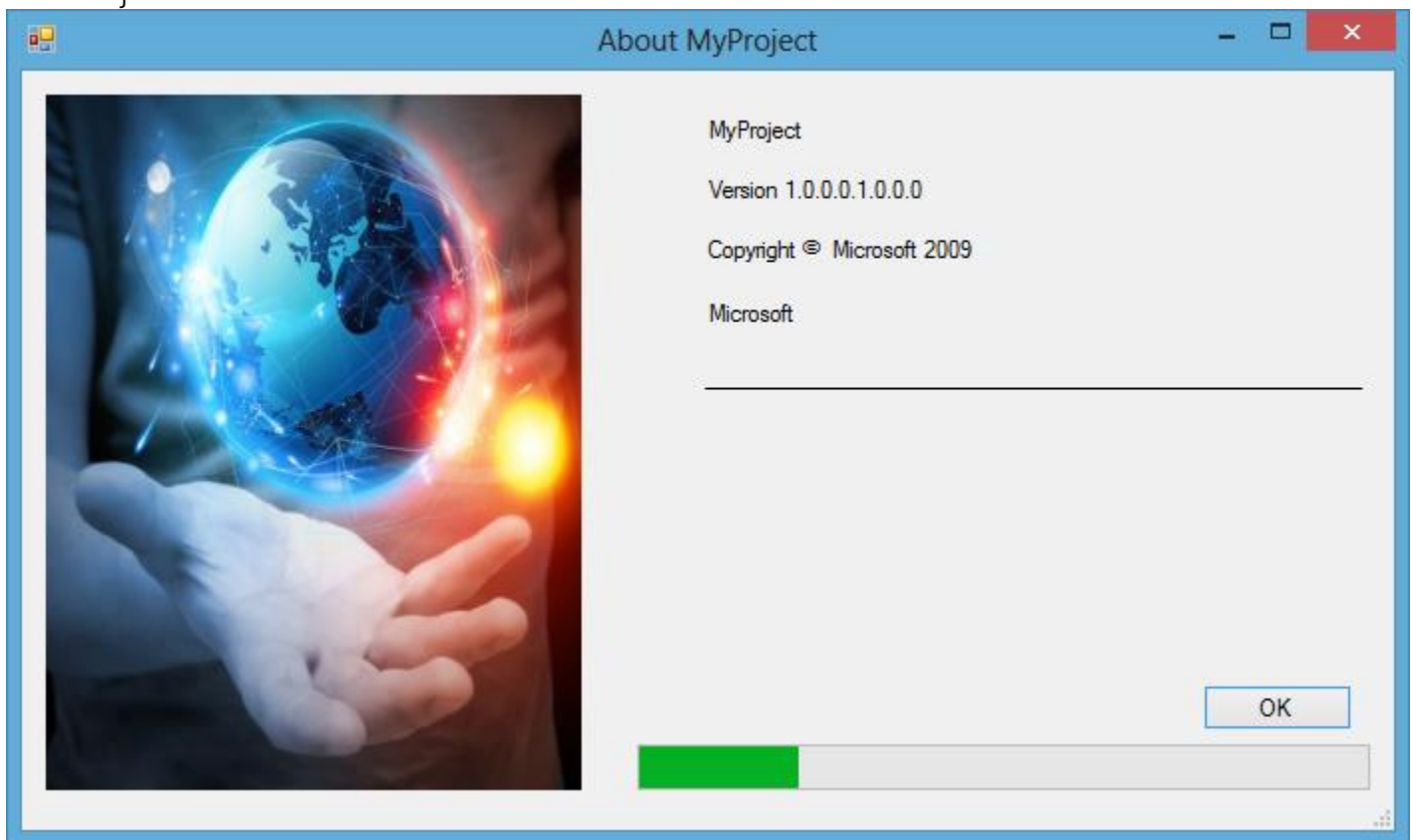
```

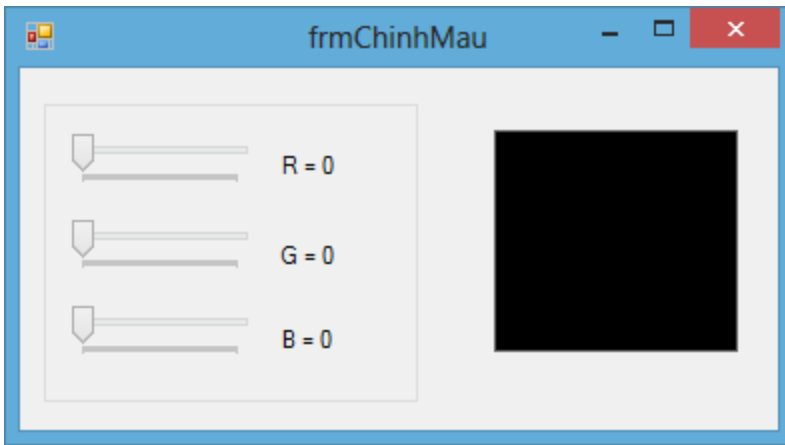
        if (progressBar1.Value >= progressBar1.Maximum) //Nếu progressBar1 chạy xong
        {
            timer1.Stop();
            this.DialogResult = DialogResult.OK;
        }
    }

    private void btnOK_Click(object sender, EventArgs e)
    {
        //timer1.Stop();

        this.DialogResult = DialogResult.OK;
    }
}
Program.cs
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    frmAMM frm = new frmAMM();
    if (frm.ShowDialog() == DialogResult.OK)
    {
        Application.Run(new Module4Baiso1_TranNguyenHung.frm1());
    }
}

```





Bài 7

Mục đích:

Sử dụng menu

Yêu cầu

Thiết kế bổ sung hệ thống menu cho Bài tập số 4 module 3

- Các menu item sẽ thực hiện chức năng tương tự như các button của bài 4
- Menu “Thông tin” khi được Click sẽ xuất messageBox, cho biết thông tin của sinh viên

thực hiện chương trình. `public partial class frmupdate2lst : Form`

```
{
    public frmupdate2lst()
    {
        InitializeComponent();
    }

    private void frmupdate2lst_Load(object sender, EventArgs e)
    {
        ngay();
        timer1.Start();
        cboLop.SelectedIndex = 0;
    }

    public void ngay()
    {
        lblngaythang.Text = DateTime.Now.ToLongDateString();
    }

    public void Gio()
    {
        lblgio.Text = DateTime.Now.ToLongTimeString();
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        Gio();
    }

    void capnhat()
    {
        if (txtTen.Text == "")
        {
```

```

        MessageBox.Show("Bạn chưa nhập tên", "Thông báo!", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
        txtTen.Focus();
    }
    else
    {
        switch (cboLop.SelectedIndex)
        {
            case 0:
                lstA.Items.Add(txtTen.Text);
                txtTen.Clear();
                txtTen.Focus();
                break;
            case 1:
                lstB.Items.Add(txtTen.Text);
                txtTen.Clear();
                txtTen.Focus();
                break;
            default:
                break;
        }
    }
}

private void btncapnhat_Click(object sender, EventArgs e)
{
    capnhat();
}

public void Chuyendulieu(ListBox a, ListBox b)
{
    int i = 0;
    while (i < a.Items.Count)
        if (a.GetSelected(i))
        {
            b.Items.Add(a.Items[i].ToString());
            a.Items.RemoveAt(i);
        }
        else
            i++;
}

public void chuyentoanbo(ListBox a, ListBox b)
{
    int i = 0;
    while (i < a.Items.Count)
    {
        b.Items.Add(a.Items[i].ToString());
        i++;
    }
    a.Items.Clear();
}

private void btnchuyen1AB_Click(object sender, EventArgs e)
{
    DialogResult hoi;
    if (lstA.Items.Count > 0)
    {
        hoi = MessageBox.Show("Bạn chắc chắn muốn chuyển dữ liệu đang chọn?", "Xác nhận",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (hoi == DialogResult.Yes)

```



```

        Chuyendulieu(lstA, lstB);
    }
    else
        MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
}

private void btnChuyenhetAB_Click(object sender, EventArgs e)
{
    DialogResult hoi;
    if (lstA.Items.Count > 0)
    {
        hoi = MessageBox.Show("Bạn chắc chắn muốn chuyển toàn bộ dữ liệu?", "Xác nhận",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (hoi == DialogResult.Yes)
            chuyentoanbo(lstA, lstB);
    }
    else
        MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
}

private void btnchuyen1BA_Click(object sender, EventArgs e)
{
    DialogResult hoi;
    if (lstB.Items.Count > 0)
    {
        hoi = MessageBox.Show("Bạn chắc chắn muốn chuyển dữ liệu đang chọn?", "Xác nhận",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (hoi == DialogResult.Yes)
            Chuyendulieu(lstB, lstA);
    }
    else
        MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
}

private void btnChuyenhetBA_Click(object sender, EventArgs e)
{
    DialogResult hoi;
    if (lstB.Items.Count > 0)
    {
        hoi = MessageBox.Show("Bạn chắc chắn muốn chuyển toàn bộ dữ liệu?", "Xác nhận",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (hoi == DialogResult.Yes)
            chuyentoanbo(lstB, lstA);
    }
    else
        MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
}

public void xoadulieu(ListBox a)
{
    int i = 0;
    while (i < a.Items.Count)
        if (a.GetSelected(i))
            a.Items.RemoveAt(i);
        else
            i++;
}

private void btnXoaA_Click(object sender, EventArgs e)
{
    DialogResult hoi;

```

```

        if (lstA.Items.Count > 0)
        {
            hoi = MessageBox.Show("Bạn chắc chắn muốn xóa dữ liệu đang chọn?", "Xác nhận",
MessageBoxButtons.YesNo,
            MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
            if (hoi == DialogResult.Yes)
                xoadulieu(lstA);
        }
        else
            MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
    }

    private void btnXoaB_Click(object sender, EventArgs e)
    {
        DialogResult hoi;
        if (lstB.Items.Count > 0)
        {
            hoi = MessageBox.Show("Bạn chắc chắn muốn xóa dữ liệu đang chọn?", "Xác nhận",
MessageBoxButtons.YesNo,
            MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
            if (hoi == DialogResult.Yes)
                xoadulieu(lstB);
        }
        else
            MessageBox.Show("Danh sách hiện đang rỗng!", "Chú ý");
    }

    private void btnKetThuc_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void frmupdate2lst_FormClosing(object sender, FormClosingEventArgs e)
    {
        DialogResult r;
        r = MessageBox.Show("Bạn chắc chắn muốn thoát?", "Xác Nhận!", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (r == DialogResult.No)
            e.Cancel = true;
    }

    private void thôngTinToolStripMenuItem_Click(object sender, EventArgs e)
    {
        String thongtin = "Họ Tên: Trần Nguyên Hưng\nMASSV:11007922\n" + "Lớp:CDTH13K";
        MessageBox.Show(thongtin);
    }

    private void cậpNhapLopBToolStripMenuItem_Click(object sender, EventArgs e)
    {
        cboLop.SelectedIndex = 1;
        capnhat();
    }

    private void cậpNhapLopAToolStripMenuItem_Click(object sender, EventArgs e)
    {
        cboLop.SelectedIndex = 0;
        capnhat();
    }
}
//các menu lắng nghe cùng sự kiện với button

```

Form Cập Nhật 2 Listbox

Cập Nhật | Hệ Thống

Cập Nhật Lớp A
 Cập Nhật Lớp B
 Chuyển Phần Tử Chọn Sang Lớp A
 Chuyển Phần Tử Chọn Sang Lớp B
 Chuyển Hết DS Sang Lớp A
 Chuyển Hết DS Sang Lớp B
 Xóa Danh Sách Lớp A
 Xóa Danh Sách Lớp B

Cập Nhật

<<

Xóa Lớp A | Kết Thúc | Xóa Lớp B

07 Tháng Bảy 2013 10:58:16 CH Designed by TranHung

Form Cập Nhật 2 Listbox

Cập Nhật | Hệ Thống

Thông Tin
 Kết Thúc

Tên SV:

Cập Nhật Vào: Lớp A

>
 >>
 <
 <<

Xóa Lớp A | Kết Thúc | Xóa Lớp B

07 Tháng Bảy 2013 10:59:24 CH Designed by TranHung

Module 5 : GUI

Bài 1

Thiết kế giao diện trình bày sơ đồ các chỗ ngồi của một rạp chiếu phim giúp người sử dụng

chọn vị trí muốn mua (xem hình bên dưới). Trên sơ đồ này cũng thể hiện những vị trí đã bán vé và những vị trí chưa bán vé bằng cách thể hiện màu khác nhau (ghế chưa bán vé màu trắng, ghế đã bán vé màu vàng, ghế đang chọn màu xanh).

Khi khởi động, ứng dụng phải nằm giữa màn hình. Không cho phép người dùng thay đổi kích thước form.

Khi người sử dụng nhấp chuột tại một vị trí trên sơ đồ thì:

- o Nếu đây là vị trí chưa bán vé thì đổi màu của vị trí này sang màu xanh để cho biết đây là vị trí đang chọn.
- o Nếu đây là vị trí đang chọn (có màu xanh) thì đổi màu của vị trí này trở về màu trắng
- o Nếu đây là một vị trí đã bán vé thì xuất hiện một thông báo cho người sử dụng biết thông qua messageBox.

Sau khi đã chọn các vị trí người sử dụng có thể nhấn vào nút CHỌN hoặc HỦY BỎ.

- ☐ Nếu click vào nút CHỌN (hoặc nhấn phím Enter) thì:

Đổi màu các vị trí đã chọn (màu xanh) trên sơ đồ sang màu vàng (cho biết vị trí đã bán vé) và xuất lên label thành tiền phải trả cho số vé đã mua (giả sử giá vé là 100đ/ghế)

- ☐ Nếu click vào nút HỦY BỎ (hoặc nhấn phím ESC) thì:

Đổi màu các vị trí đã chọn (màu xanh) trên sơ đồ sang màu trắng trở lại và xuất lên label thành tiền giá trị 0

```
void taodonglabel(int soghe)
{
    for (int i = 1 ; i <= soghe; i++)
    {
        Label lbl = new Label();
        lbl.Text = i.ToString();
        lbl.Font = new Font("Times new roman",16, FontStyle.Bold);
        lbl.TextAlign = ContentAlignment.MiddleCenter;
        lbl.BorderStyle=BorderStyle.Fixed3D;
        lbl.BackColor = Color.White;
        lbl.Width = 55;
        lbl.Height = 55;
        lbl.Name = "ghe" + i.ToString();
        lbl.Click += new EventHandler(EventDong_Click);
        fldmBanVe.Controls.Add(lbl);
    }
}

private void frmMuaVe_Load(object sender, EventArgs e)
{
    taodonglabel(15);
}
```

```

private void btnHuy_Click(object sender, EventArgs e)
{
    foreach (Label lbl in fldmBanVe.Controls)
        if (lbl.BackColor == Color.Blue)
            lbl.BackColor = Color.White;
    lblThanhTien.Text = "";
}
private void EventDong_Click(object sender, EventArgs e)
{
    Label lbl = (Label)sender;
    if (lbl.BackColor == Color.White)
        lbl.BackColor = Color.Blue;
    else if (lbl.BackColor == Color.Blue)
        lbl.BackColor = Color.White;
    else
        MessageBox.Show("Ghế "+lbl.Text+" đã được chọn.", "Thông báo");
}

private void btnChon_Click(object sender, EventArgs e)
{
    int nTien = 0;
    foreach (Label lbl in fldmBanVe.Controls)
    {
        if (lbl.BackColor == Color.Blue)
        {
            lbl.BackColor = Color.Yellow;
            nTien += 100;
        }
    }
    lblThanhTien.Text = string.Format("{0} VND", nTien);
}

private void btnKetThuc_Click(object sender, EventArgs e)
{
    this.Close();
}

private void frmMuaVe_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult hoi;
    hoi = MessageBox.Show("Bạn có chắc muốn thoát?", "Xác nhận", MessageBoxButtons.YesNo,
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
    if (hoi == DialogResult.No)
        e.Cancel=true;
}

```

TranNguyenHung_CDTH13K_11007922

Trần Nguyễn Hưng_11007922

Màn Ảnh

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

Thành Tiền: 200 VND

Bài 2

Viết chương trình nhập vào năm dương lịch và xuất năm âm lịch tương ứng.

Biết rằng : Năm Âm = Can + Chi;

Can = Năm dương % 10 :

0	1	2	3	4	5	6	7	8	9
Canh	Tân	Nhâm	Quý	Giáp	Ất	Bính	Đinh	Mậu	Kỷ

Chi = dương lịch % 12 :

0	1	2	3	4	5	6	7	8	9	10	11
Thân	Dậu	Tuất	Hợi	Tý	Sửu	Dần	Mẹo	Thìn	Tỵ	Ngọ	Mùi

```
private void button1_Click(object sender, EventArgs e)
{
    int nam = Convert.ToInt32(txtNhap.Text);
    lblKetQua.Text = can(nam) + " " + Chi(nam);
}

string can(int namduong)
{
    string strCan="";
    switch (namduong % 10)
    {
```

```

{
    case 0:
        strCan = "Canh";
        break;
    case 1:
        strCan = "Tân";
        break;
    case 2:
        strCan = "Nhâm";
        break;
    case 3:
        strCan = "Quý";
        break;
    case 4:
        strCan = "Giáp";
        break;
    case 5:
        strCan = "Ất";
        break;
    case 6:
        strCan = "Bính";
        break;
    case 7:
        strCan = "Đinh";
        break;
    case 8:
        strCan = "Mậu";
        break;
    case 9:
        strCan = "Kỷ";
        break;
    default:
        break;
}
return strCan;
}

```

```

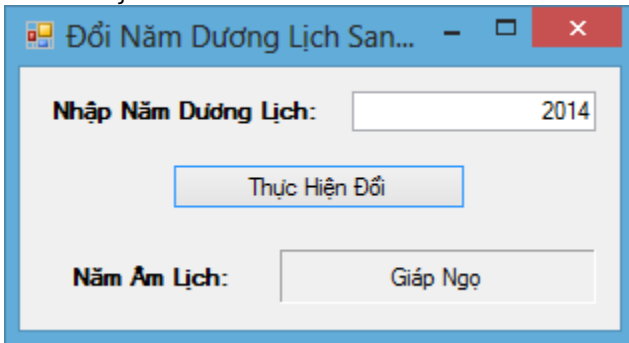
string Chi(int namduong)
{
    string strChi = "";
    switch (namduong % 12)
    {
        case 0:
            strChi = "Thân";
            break;
        case 1:
            strChi = "Dậu";
            break;
        case 2:
            strChi = "Tuất";
            break;
        case 3:
            strChi = "Hợi";
            break;
        case 4:
            strChi = "Tý";
            break;
        case 5:
            strChi = "Sửu";
            break;
        case 6:
            strChi = "Dần";
            break;
        case 7:

```

```

        strChi = "Mẹo";
        break;
    case 8:
        strChi = "Thìn";
        break;
    case 9:
        strChi = "Ty";
        break;
    case 10:
        strChi = "Ngọ";
        break;
    case 11:
        strChi = "Mùi";
        break;
    default:
        break;
    }
    return strChi;
}

```



Bài 3

Viết chương trình nhập dữ liệu vào ListView, bao gồm tên tuổi, số điện thoại

Yêu cầu:

- Người sử dụng nhập thông tin Last Name, First name, phone và sử dụng nút Add name để nhập vào ô listview:
- Các dòng trong list view có biểu tượng I con hiển thị như hình.
- Người sử dụng có thể thay đổi chế độ view của ListView bằng menu view
- Menu formatlist view hiển thị hộp thoại chọn màu dùng để thay đổi dàn grid của listview

```

private void btnAddName_Click(object sender, EventArgs e)
{
    string ten = txtLName.Text;
    string ho = txtFName.Text;
    string phone = txtPhone.Text;
    AddToListView(lstvInfo, ten, ho, phone);
}

void AddToListView(ListView lstvIn, String a, String b, String c)
{
    //Tạo phần tử listview
    ListViewItem lstvI = new ListViewItem(a, 0);
    lstvI.SubItems.Add(b);
    lstvI.SubItems.Add(c);
    //Thêm phần tử vào listview
    lstvIn.Items.Add(lstvI);
}

private void mnuLargeIcon_Click(object sender, EventArgs e)

```



```

{
    lstvInfo.View = View.LargeIcon;
}

private void mnuDetails_Click(object sender, EventArgs e)
{
    lstvInfo.View = View.Details;
}

private void mnuSmallIcon_Click(object sender, EventArgs e)
{
    lstvInfo.View = View.SmallIcon;
}

private void mnuList_Click(object sender, EventArgs e)
{
    lstvInfo.View = View.List;
}

private void mnuTitle_Click(object sender, EventArgs e)
{
    lstvInfo.View = View.Tile;
}

private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}

private void txtPhone_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar < 48 || e.KeyChar > 57)
        e.Handled = true;
}

private void showGridToolStripMenuItem_Click(object sender, EventArgs e)
{
    if(mnuShowGrid.Checked==true)
        lstvInfo.GridLines = true;
    else
        lstvInfo.GridLines = false;
}

```

The screenshot shows a Windows application window titled "ListView Demo". The window has a menu bar with three items: "File", "View", and "Format ListView". The main area is divided into two sections. On the left, there is a ListView control displaying a single item. The item has three columns: "Last Name", "First Name", and "Phone". The values in the item are "Tran", "Hung", and "01362343223" respectively. On the right, there are three text boxes labeled "Last Name", "First Name", and "Phone", each containing the same values as the item in the ListView. Below these text boxes is a button labeled "Add Name".

Last Name	First Name	Phone
Tran	Hung	01362343223

Last Name: Tran
 First Name: Hung
 Phone: 01362343223
 Add Name

Bài 4

Thiết kế chương trình quản lý sinh viên, cho phép nhập thông tin SV vào textbox. Sau đó thêm SV này vào ListView, có thể xóa SV từ ListView

Yêu cầu:

Thêm vào form 2 imagelist là ilsNho có kích thước mặc định 16x16, ilsLon có kích thước mặc định 48x48 phục vụ cho listview.

- Nhấn nút cập nhật Item thì đưa thông tin sinh viên vào listview theo các cột như hình
- Nhấn nút xóa Item là xóa item đang chọn trên listview, trước khi xóa cần xác nhận chọn Item nào chưa, xác nhận có chắc xóa không.
- Nhấn nút thêm mới thì xóa thông tin sinh viên đang đăng nhập và cho phép nhập thông tin sinh viên mới
- Nếu chọn một sinh viên nào trong listview thì hiện lại thông tin sinh viên đó lên các textbox tương ứng.
- Click phải vào các listview cho phép hiện menu ngữ cảnh để chọn chức năng view

```
public class SinhVien
{
    private string masv, hoten, diachi, ngaysinh, lop;

#region vùng dat va lay gia tri (Properties)
    public string Lop
    {
        get { return lop; }
        set { lop = value; }
    }

    public string Ngaysinh
    {
        get { return ngaysinh; }
        set { ngaysinh = value; }
    }

    public string Diachi
    {
        get { return diachi; }
        set { diachi = value; }
    }

    public string Hoten
    {
        get { return hoten; }
        set { hoten = value; }
    }

    public string Masv
    {
        get { return masv; }
        set { masv = value; }
    }
#endregion ket thuc vùng dat va lay gia tri

#region vùng cac methods
    public SinhVien()
    {
    }
    public SinhVien(string masv, string hoten, string diachi, string ngaysinh, string lop)
    {

```

```

        this.masv = masv;
        this.hoten = hoten;
        this.diachi = diachi;
        this.ngaysinh = ngaysinh;
        this.lop = lop;
    }
#endregion ket thuc khu vung cac methods

}

using System.IO;
public class DSSV
{
    ArrayList arr;
    public DSSV(){
        arr=new ArrayList();
    }
    public void themsv(SinhVien sv){
        arr.Add(sv);
    }
    public void xoasv(SinhVien sv){
        arr.Remove(sv);
    }
    public ArrayList LayDsSinhVien()
    {
        return arr;
    }

    public bool constaint(SinhVien sv)
    {
        if (arr.Contains(sv))
            return true;
        return false;
    }
}

public partial class frmBaiTapLstView : Form
{
    public frmBaiTapLstView()
    {
        InitializeComponent();
    }
    DSSV danhsach = new DSSV();
    private void btnCapNhat_Click(object sender, EventArgs e)
    {
        string masv = txtMaSV.Text;
        string hoten = txtHoTen.Text;
        string diachi = txtDiaChi.Text;
        string ngaysinh = dateNgaySinh.Value.ToShortDateString();
        string lop = cboLop.SelectedItem.ToString();
        SinhVien sv = new SinhVien(masv,hoten,diachi,ngaysinh,lop);
        danhsach.themsv(sv);
        UpdatelstView(danhsach);
    }
    void UpdatelstView(DSSV ds)
    {
        lstViewInfo.Items.Clear();
        foreach(SinhVien sv in ds.LayDsSinhVien())
        {
            //Tạo phần tử listview
            ListViewItem lstvI = new ListViewItem(sv.Masv, 0);
            lstvI.SubItems.Add(sv.Hoten);
            lstvI.SubItems.Add(sv.Diachi);
            lstvI.SubItems.Add(sv.Ngaysinh);
            lstvI.SubItems.Add(sv.Lop);
        }
    }
}

```

```

        lstvI.Tag = sv;
        //Thêm phần tử vào listview
        lstViewInfo.Items.Add(lstvI);
    }

}

private void btnXoa_Click(object sender, EventArgs e)
{
    if (lstViewInfo.SelectedItems.Count <= 0)
        MessageBox.Show("Bạn chưa chọn đối tượng để xóa!", "Thông báo");
    else
    {
        DialogResult hoi;
        hoi = MessageBox.Show("Bạn chắc chắn muốn xóa?", "Xác Nhận", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (hoi == DialogResult.Yes)
        {
            XoaItemListView();
            txtMaSV.Clear();
            txtHoTen.Clear();
            txtDiaChi.Clear();
            dateNgaySinh.Text = dateNgaySinh.MinDate.ToShortDateString();
            cboLop.SelectedIndex = 0;
        }
    }
}

void XoaItemListView()
{
    int i = 0;
    while (i < lstViewInfo.SelectedItems.Count)
    {
        danh sach.xoasv((SinhVien)lstViewInfo.SelectedItems[i].Tag);
        i++;
    }
    UpdateListView(danh sach);
}

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

private void mnuLargeIcon_Click(object sender, EventArgs e)
{
    ToolStripMenuItem a = (ToolStripMenuItem)sender;
    switch (a.Name)
    {
        case "mnuLargeIcon":
            lstViewInfo.View = View.LargeIcon;
            break;
        case "mnudetail":
            lstViewInfo.View = View.Details;
            break;
        case "mnuSmall":
            lstViewInfo.View = View.SmallIcon;
            break;
        case "mnuList":
            lstViewInfo.View = View.List;
            break;
        case "mnuTitle":
            lstViewInfo.View = View.Tile;
    }
}

```

```

        break;
    default:
        break;
    }
}

private void lstViewInfo_SelectedIndexChanged(object sender, EventArgs e)
{
    if(lstViewInfo.SelectedItems.Count>0)
    {
        ListViewItem item= lstViewInfo.SelectedItems[0];
        txtMaSV.Text = item.Text;
        txtHoTen.Text = item.SubItems[1].Text;
        txtDiaChi.Text = item.SubItems[2].Text;
        dateNgaySinh.Text = item.SubItems[3].Text;
        cboLop.Text = item.SubItems[4].Text;
    }
}

private void frmBaiTapLstView_Load(object sender, EventArgs e)
{
    cboLop.SelectedIndex = 0;
}

private void btnThemMoi_Click(object sender, EventArgs e)
{
    txtDiaChi.Clear();
    txtHoTen.Clear();
    txtMaSV.Clear();
    cboLop.SelectedIndex = -1;
    dateNgaySinh.Text = dateNgaySinh.MinDate.ToShortDateString();
}
}

```

The application window is titled "Bài Tập ListView". It contains a form titled "Thông tin sinh viên" with the following fields and controls:

- Mã SV: Textbox containing "11007922".
- Họ Tên: Textbox containing "Trần Hùng".
- Địa Chỉ: Textbox containing "Đồng Nai".
- Ngày Sinh: Textbox containing "01/01/1993" with a calendar icon.
- Lớp: Dropdown menu showing "Lop A".
- Buttons: "Thêm Mới", "Cập Nhật Item" (highlighted with a dashed border), "Xóa Item", and "Thoát".

Below the form is a ListView table with the following data:

Mã	Họ Tên	Địa Chỉ	Ngày Sinh	Lớp
1100...	Trần Hu...	Đồng Nai	01/01/1993	Lop A

Bài 5

Viết chương trình tính hóa đơn tiền điện

Yêu cầu:

thiết lập thuộc tính cho phép chọn nhiều dòng trên listview

Combobox có 3 khu vực: khu vực 1(định mức 50), khu vực 2(định mức 100). khu vực 3 định mức 150). Khi chọn khu vực nào thì hiện định mức khu vực ấy.

Nút tính tiền (hoặc enter trên các textbox) :kiểm tra dữ liệu nhập, nếu hợp lệ thì tính và xuất ra kết quả ra ô *tiêu thụ* và *thành tiền* , đồng thời thêm 1 dòng tương ứng ở listview và cập nhật ô tổng tiền.

Đơn giá điện, trong định mức là 500, ngoài định mức là 1000.

Nút Nhập mới: xóa các nội dụng trong textbox và label, đồng thời đặt con trỏ vào textbox đầu tiên.

Nút Xóa: cho phép xóa 1 dòng đang chọn trong listview , phải xác nhận lại trước khi xóa và cập nhật lại ô tổng tiền.

Nút Thoát(hoặc ấn ESC): thoát chương trình.

Quy định form hiện thị giữa màn hình.

Quy định việc di chuyển tab hợp lí,

Thiết lập thuộc tính Anchor hợp lí cho các control

Thiết lập minumunsize cho các control.

```

public class HoaDon
{
    private string hoten, khuvuc;
    private double dinhmuc, socu, somoi, tieuthu, thanhtien;

#region Get/Set

    public string Khuvuc
    {
        get { return khuvuc; }
        set { khuvuc = value; }
    }

    public string Hoten
    {
        get { return hoten; }
        set { hoten = value; }
    }

    public double Somoi
    {
        get { return somoi; }
        set { somoi = value; }
    }

    public double Socu
    {
        get { return socu; }
        set { socu = value; }
    }

    public double Thanhtien
    {
        get { return thanhtien; }
        set { thanhtien = value; }
    }

    public double Tieuthu
    {
        get { return tieuthu; }
        set { tieuthu = value; }
    }

    public double Dinhmuc
    {
        get { return dinhmuc; }
        set { dinhmuc = value; }
    }

#endregion end of set/get

#region methods
    public HoaDon()
    {

    }

    public HoaDon(string hoten, string khuvuc, double dinhmuc, double socu, double somoi)
    {
        this.hoten = hoten;
        this.khuvuc = khuvuc;
        this.dinhmuc = dinhmuc;
        this.socu = socu;
        this.somoi = somoi;
    }

```

```

        LayTieuThu();
        ThanhTien();
    }
    public void LayTieuThu(){
        tieuthu = somoi-socu;
    }
    public void ThanhTien()
    {
        if (tieuthu <= dinh muc)
        {
            thanhtien = 500 * tieuthu;
        }
        else
        {
            double trongdinh muc = dinh muc;
            double ngoaidinh muc = tieuthu - dinh muc;
            thanhtien = (500*trongdinh muc)+(ngoaidinh muc*1000);
        }
    }
}
#endregion end methods
}

using System.IO;
public class DSHoaDon
{
    ArrayList arr;
    public DSHoaDon()
    {
        arr = new ArrayList();
    }
    public void themhd(HoaDon hd){
        arr.Add(hd);
    }
    public void xoahd(HoaDon hd)
    {
        arr.Remove(hd);
    }
    public double laytongtien()
    {
        double tong = 0;
        foreach (HoaDon hd in arr)
            tong += hd.Thanhtien;
        return tong;
    }
    public ArrayList xuatds()
    {
        return arr;
    }
}
public partial class frmTinhTienDien : Form
{
    public frmTinhTienDien()
    {
        InitializeComponent();
    }
    DSHoaDon ds = new DSHoaDon();
    HoaDon hd;
    //Sét định mức theo khu vực được chọn
    private void cboKhuVuc_SelectedIndexChanged(object sender, EventArgs e)
    {
        switch (cboKhuVuc.SelectedIndex)
        {
            case 0:
                txtDinhMuc.Text = "50";

```



```

        break;
    case 1:
        txtDinhMuc.Text = "100";
        break;
    case 2:
        txtDinhMuc.Text = "150";
        break;
    default:
        break;
    }
}

private void btnTinhTien_Click(object sender, EventArgs e)
{
    if (kiemtradulieu()==0)
        MessageBox.Show("Bạn chưa nhập đủ dữ liệu!", "Thông báo", MessageBoxButtons.OK,
        MessageBoxIcon.Warning, MessageBoxDefaultButton.Button1);
    else if(kiemtradulieu()==1)
        MessageBox.Show("Số cũ phải nhỏ hơn số mới!", "Thông báo", MessageBoxButtons.OK,
        MessageBoxIcon.Warning, MessageBoxDefaultButton.Button1);
    else
    {
        tinh();
        txtTieuThu.Text = hd.Tieuthu+"";
        txtThanhTien.Text = hd.Thanhtien+"";
        UpdateListView(ds);
    }
}

//Hàm kiểm tra dữ liệu đầy đủ
int kiemtradulieu()
{
    if (txtDinhMuc.Text.Length == 0 || txtHoTen.Text.Length == 0 || cboKhuVuc.SelectedIndex ==
-1 ||
        txtSoCu.Text.Length == 0 || txtSoMoi.Text.Length == 0 )
        return 0;
    if (txtDinhMuc.Text.Length == 0 || txtHoTen.Text.Length == 0 || cboKhuVuc.SelectedIndex ==
-1 ||
        txtSoCu.Text.Length == 0 || txtSoMoi.Text.Length == 0 ||
        Convert.ToDouble(txtSoMoi.Text) < Convert.ToDouble(txtSoCu.Text))
        return 1;
    return 9999;
}

//Hàm Tính tiền
void tinh()
{
    try
    {
        hd = new HoaDon(txtHoTen.Text, cboKhuVuc.SelectedItem.ToString(),
        Convert.ToDouble(txtDinhMuc.Text), Convert.ToDouble(txtSoCu.Text),
        Convert.ToDouble(txtSoMoi.Text));
        ds.themhd(hd);
    }
    catch (Exception e)
    {
        MessageBox.Show("Lỗi: "+e.Message, "Thông báo");
    }
}

void UpdateListView(DSHoaDon DS)
{

```

```

lstvThongTin.Items.Clear();
foreach (HoaDon hd in DS.xuatds())
{
    //Tạo phần tử listview
    ListViewItem lstvI = new ListViewItem(hd.Hoten);
    lstvI.SubItems.Add(hd.Khuvuc);
    lstvI.SubItems.Add(hd.Dinhmuc.ToString());
    lstvI.SubItems.Add(hd.Tieuthu.ToString());
    lstvI.SubItems.Add(hd.Thanhtien.ToString());
    lstvI.Tag = hd;
    //Thêm phần tử vào listview
    lstvThongTin.Items.Add(lstvI);
    lblTongTien.Text = DS.laytongtien().ToString();
}
}

private void btnNhapMoi_Click(object sender, EventArgs e)
{
    txtHoTen.Clear();
    txtDinhMuc.Clear();
    txtSoCu.Clear();
    txtSoMoi.Clear();
    cboKhuVuc.SelectedIndex = -1;
    txtThanhTien.Clear();
    txtTieuThu.Clear();
}

//Chỉ cho nhập số
private void txtSoCu_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar < 48 || e.KeyChar > 57)
        e.Handled = true;
}

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

private void btnXoa_Click(object sender, EventArgs e)
{
    if (lstvThongTin.SelectedItems.Count <= 0)
        MessageBox.Show("Bạn chưa chọn đối tượng để xóa!", "Thông báo");
    else
    {
        DialogResult hoi;
        hoi = MessageBox.Show("Bạn chắc chắn muốn xóa?", "Xác Nhận", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (hoi == DialogResult.Yes)
        {
            XoaItemListView(ds);
        }
    }
}

//Hàm xóa Item Trong ListView
void XoaItemListView(DSHoaDon ds)
{
    int i = 0;
    while (i < lstvThongTin.SelectedItems.Count)
    {
        ds.xoahd((HoaDon)lstvThongTin.SelectedItems[i].Tag);
        i++;
    }
}

```

```

        lblTongTien.Text = ds.laytongtien().ToString();
        UpdateListView(ds);
    }
}

```

Module 6 : GUI

Bài 1

Viết chương trình nhập danh bạ, sử dụng treeView là bảng chữ cái từ A-Z, khi người dùng thêm tên thì tự đưa vào Node tương ứng

Yêu cầu:

- Khi chương trình vừa hiển thị, treeview thị tất cả các chữ cái từ a-> z.
- Nhằm mục đích tiện lợi cho người sử dụng khi tìm tên, khi người sử dụng nhập tên của 1 người nào đó, chương trình sẽ đưa tên người đó vào treeview ở vị trí node tương ứng với chữ cái đầu tiên của tên

```

private void frmTreeView_Load(object sender, EventArgs e)
{
    ProgramInit();
}

private void ProgramInit()
{
    for (char szChar = 'A'; szChar <= 'Z'; szChar++)
    {
        trwMain.Nodes.Add(szChar.ToString());
    }
}

public void AddName(Object First, Object Last, Object sender)
{
    TextBox fstName = (TextBox)First;
    TextBox lstName = (TextBox>Last;
}

```

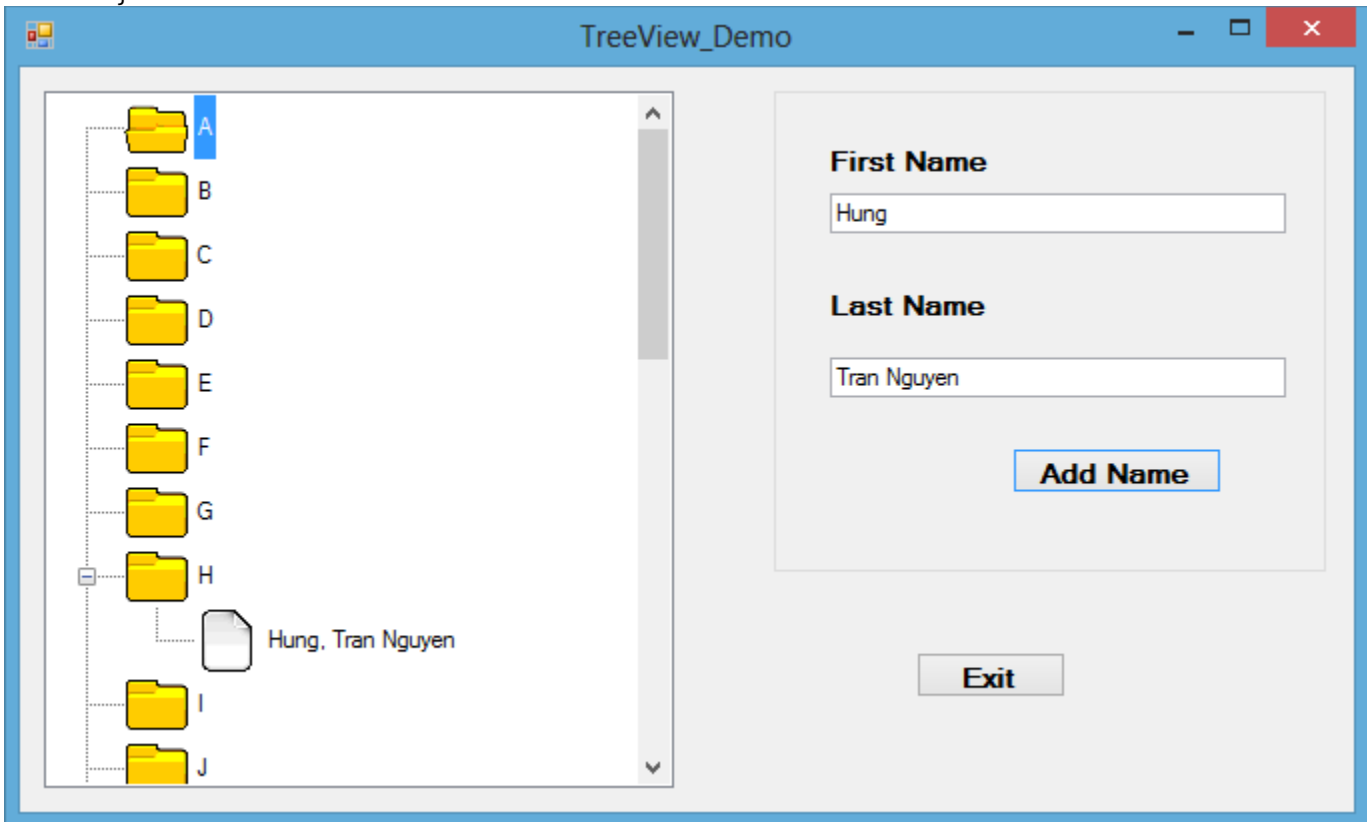
```

TreeView obj = (TreeView)sender;
for (int i = 0; i < obj.Nodes.Count; i++ )
{
    TreeNode root = obj.Nodes[i];
    if(Char.ToUpper(fstName.Text[0]).Equals(Char.ToUpper(obj.Nodes[i].Text[0])))
    {
        TreeNode nod = new TreeNode(fstName.Text + ", " + lstName.Text);
        nod.ImageIndex = 2;
        nod.SelectedImageIndex = 2;
        root.Nodes.Add(nod);
        return;
    }
}

private void btnAddName_Click(object sender, EventArgs e)
{
    AddName(txtFN, txtLN, trwMain);
}

private void btnExit_Click(object sender, EventArgs e)
{
    this.Close();
}

```



Bài 2

Viết chương trình xem danh sách sinh viên khoa tin học, sử dụng TreeView và ListView, cho phép tìm SV dựa vào tên SV

Yêu cầu:

- Khi form hiện lên. Tree view hiển thị danh sách các lớp- sinh viên, chưa có nút nào được chọn, con trỏ đặt tại ô đặt tên
- Khi dùng chọn nút cấp khoa, chương trình hiện toàn bộ danh sách Sv thuộc khoa vào listview

- Khi người dùng chọn một khoa bất kì thì chương trình hiện toàn bộ danh sách SV thuộc lớp đang chọn vào listview
- Khi chọn 1 sv bất kì thì chỉ hiện sinh viên đó vào Listview
- Nút tìm: cho phép tìm sv có họ tên chứa chuỗi nhập trong textbox, hiện kết quả ra listview

```
public class SinhVien
{
    string tenSV, Lop;

    public SinhVien()
    {
    }

    public SinhVien(string ten, string lop)
    {
        this.tenSV = ten;
        this.Lop = lop;
    }

    public string Lop1
    {
        get { return Lop; }
        set { Lop = value; }
    }

    public string TenSV
    {
        get { return tenSV; }
        set { tenSV = value; }
    }

    public override string ToString()
    {
        return tenSV;
    }
}
```

```
using System.IO;
public class Lop
{
    string tenlop;

    public string Tenlop
    {
        get { return tenlop; }
        set { tenlop = value; }
    }
    ArrayList dssv;

    public ArrayList Dssv
    {
        get { return dssv; }
        set { dssv = value; }
    }
    public Lop(string ten)
    {
        this.tenlop = ten;
        dssv = new ArrayList();
    }
    public void themsv(SinhVien sv)
    {

```

```

        dssv.Add(sv);
    }
    public ArrayList laydssv()
    {
        return Dssv;
    }
    public override string ToString()
    {
        return Tenlop;
    }
}
public class Khoa
{
    string tenkhoa;

    public string Tenkhoa
    {
        get { return tenkhoa; }
        set { tenkhoa = value; }
    }
    ArrayList dsLop;

    public ArrayList DsLop
    {
        get { return dsLop; }
        set { dsLop = value; }
    }
    public Khoa(string ten)
    {
        this.tenkhoa = ten;
        dsLop = new ArrayList();
    }
    public void ThemLop(Lop lop)
    {
        dsLop.Add(lop);
    }
    public ArrayList laydslop()
    {
        return DsLop;
    }
}
public partial class frmKhoaTinHoc : Form
{
    public frmKhoaTinHoc()
    {
        InitializeComponent();
    }

    private void KhoaTinHoc_Load(object sender, EventArgs e)
    {
        khoitaotree();
        treDanhSach.ExpandAll();
    }

    public void khoitaotree()
    {
        Khoa khoa = new Khoa("Khoa Tin Hoc");

        Lop lop1 = new Lop("THTH5A");
        lop1.themsv(new SinhVien("Nguyen Van Tuan", "THTH5A"));
        lop1.themsv(new SinhVien("Nguyen Thi Lan", "THTH5A"));
        lop1.themsv(new SinhVien("Nguyen Van Luong", "THTH5A"));
        khoa.ThemLop(lop1);
    }
}

```

```

Lop lop2 = new Lop("THTH5B");
lop2.themsv(new SinhVien("Le Nghiep", "THTH5B"));
lop2.themsv(new SinhVien("Tran Long", "THTH5B"));
lop2.themsv(new SinhVien("Ly Hai", "THTH5B"));
khoa.ThemLop(lop2);

Lop lop3 = new Lop("THTH5C");
lop3.themsv(new SinhVien("Le Trung", "THTH5C"));
lop3.themsv(new SinhVien("Ton Thi Mai", "THTH5C"));
lop3.themsv(new SinhVien("Tran Minh", "THTH5C"));
khoa.ThemLop(lop3);

TreeNode root = new TreeNode(khoa.ToString());
root.Tag = khoa;

int i = 1; //Icon Tung Lop
foreach (Lop lop in khoa.laydslop())
{
    TreeNode node1 = new TreeNode(lop.ToString());
    node1.Tag = lop;
    node1.ImageIndex = i;
    node1.SelectedImageIndex = i;
    root.Nodes.Add(node1);
    i++;
    foreach (SinhVien sv in ((Lop)node1.Tag).laydssv())
    {
        TreeNode node2 = new TreeNode(sv.ToString());
        node2.Tag = sv;
        node2.ImageIndex = 4;
        node2.SelectedImageIndex = 4;
        node1.Nodes.Add(node2);
    }
}

treDanhSach.Nodes.Add(root);
}

private void treDanhSach_AfterSelect(object sender, TreeViewEventArgs e)
{
    lstvHienThi.Items.Clear();
    TreeNode slnode = treDanhSach.SelectedNode;
    switch (slnode.Level)
    {
        case 0:
            hiencapkhoa(slnode);
            break;
        case 1:
            hiencaploi(slnode);
            break;
        case 2:
            hiencapsv(slnode);
            break;
        default:
            break;
    }
}

public void hiencapkhoa(TreeNode nod)
{
    foreach (TreeNode n in nod.Nodes)
    {
        if (n.Level == 1)
        {

```

```

        foreach (SinhVien sv in ((Lop)n.Tag).laydssv())
        {
            ListViewItem item = new ListViewItem(sv.TenSV);
            item.SubItems.Add(sv.Lop1);
            lstvHienThi.Items.Add(item);
        }
    }
}

public void hiencaPLOP(TreeNode slnode)
{
    foreach (SinhVien sv in ((Lop)slnode.Tag).laydssv())
    {
        ListViewItem item = new ListViewItem(sv.TenSV);
        item.SubItems.Add(sv.Lop1);
        lstvHienThi.Items.Add(item);
    }
}

public void hiencaPSV(TreeNode slnode)
{
    SinhVien sv=(SinhVien)slnode.Tag;
    ListViewItem item = new ListViewItem(sv.TenSV);
    item.SubItems.Add(sv.Lop1);
    lstvHienThi.Items.Add(item);
}

public void timSV(string hoten,TreeNode nod){
    bool flag = false;
    switch (nod.Level)
    {
        case 0:
            foreach (TreeNode n in nod.Nodes)
            {
                if (n.Level == 1)
                {
                    foreach (TreeNode node in n.Nodes )
                    {
                        SinhVien sv = (SinhVien)node.Tag;
                        if (hoten.ToLower().Equals(sv.TenSV.ToLower()))
                        {
                            treDanhSach.SelectedNode = node;
                            flag = true;
                        }
                    }
                }
            }
            break;
        case 1:
            foreach (TreeNode node in nod.Nodes)
            {
                SinhVien sv = (SinhVien)node.Tag;
                if (hoten.ToLower().Equals(sv.TenSV.ToLower()))
                {
                    treDanhSach.SelectedNode = node;
                    flag = true;
                }
            }
            break;
        default:
            break;
    }
    if (flag == false)

```



```

        MessageBox.Show("Không tìm thấy " + hoten + " trong cấp này");
    }

    private void btnTim_Click(object sender, EventArgs e)
    {
        string hotencantim = txtTim.Text;
        TreeNode slNode = treDanhSach.SelectedNode;
        if (slNode == null)
        {
            MessageBox.Show("Bạn chưa chọn cấp để tìm");
        }
        else
        {
            timSV(hotencantim, slNode);
        }
    }
}

```

Ten SV	Lop
Nguyen Van Tuan	THTH5A
Nguyen Thi Lan	THTH5A
Nguyen Van Luong	THTH5A

Bài 3

Viết chương trình quản lý sinh viên cho các lớp, chức năng gồm thêm và xóa

Thực hiện các yêu cầu sau:

- ☐ Thiết lập HideSelection = False.
- ☐ Nút Cập Nhật: Thêm 1 SV vào lớp đang chọn trên TreeView với nội dung các nút như hình.
- Trước khi thêm phải kiểm tra thông tin nhập gồm: các ô nhập không được để trống, không được trùng mã SV. Ngoài ra còn phải kiểm tra nút chọn trên TreeView có phải là nút lớp
- không (chỉ được thêm vào nút lớp).
- ☐ Nút Xóa: cho phép xóa nút đang chọn trong TreeView, phải xác nhận lại trước khi xóa và chỉ được xóa khi chọn nút chứa mã sv.
- ☐ Khi click chọn nút mã sv hoặc địa chỉ thì hiện thông tin sv đó qua các textbox.
- ☐ Quy định Form hiển thị giữa màn hình.
- ☐ Quy định việc di chuyển tab hợp lý.
- ☐ Thiết lập thuộc tính Dock hợp lý cho TreeView.

□ Thiết lập MinimumSize cho form

```
public class SinhVien
{
    string mssv, hoten, diachi;

    public string Diachi
    {
        get { return diachi; }
        set {
            try
            {
                if (value.Length == 0)
                    throw new Exception("Nhập thiếu Địa Chỉ");
                diachi = value;
            }
            catch (Exception err)
            {
                throw err;
            }
        }
    }

    public string Hoten
    {
        get { return hoten; }
        set {
            try
            {
                if (value.Length == 0)
                    throw new Exception("Nhập thiếu Họ Tên");
                hoten = value;
            }
            catch (Exception err)
            {
                throw err;
            }
        }
    }

    public string Mssv
    {
        get { return mssv; }
        set {
            try
            {
                if (value.Length == 0)
                    throw new Exception("Nhập thiếu MSSV");
                mssv = value;
            }
            catch (Exception err)
            {
                throw err;
            }
        }
    }

    public SinhVien(string mssv, string hoten, string diachi)
    {
        this.Mssv=mssv;
        this.Hoten=hoten;
    }
}
```

```

        this.Diachi = diachi;
    }
    public override string ToString()
    {
        return Mssv+" - "+Hoten;
    }
    public override bool Equals(object obj)
    {
        return this.Mssv.Equals(((SinhVien)obj).Mssv);
    }
    public override int GetHashCode()
    {
        return base.GetHashCode();
    }
}

using System.IO;
public class DSSV
{
    ArrayList ds;
    public DSSV()
    {
        ds = new ArrayList();
    }
    public void themsv(SinhVien sv)
    {
        if (ds.Contains(sv))
            throw new Exception("Trùng MSSV");
        else
            ds.Add(sv);
    }
    public void xoaSV(SinhVien sv)
    {
        ds.Remove(sv);
    }
}

public partial class frmQLSV : Form
{
    public frmQLSV()
    {
        InitializeComponent();
    }

    private void frmQLSV_Load(object sender, EventArgs e)
    {
        treDanhSach.ExpandAll();
    }
    DSSV ds = new DSSV();
    private void btnCapNhat_Click(object sender, EventArgs e)
    {
        ThemSV();
    }

    public void ThemSV()
    {
        try
        {
            string mssv = txtMSSV.Text;
            string hoten = txtHoTen.Text;
            string diachi = txtDiaChi.Text;
            SinhVien sv = new SinhVien(mssv, hoten, diachi);

```

```

        TreeNode slnode = treDanhSach.SelectedNode;
        if (slnode == null)
            MessageBox.Show("Bạn chưa chọn lớp để thêm");
        else
        {
            if (slnode.Level != 1)
                throw new Exception("Chỉ được Thêm SV Vào Lớp");
            else
            {
                ds.themsv(sv);
                TreeNode newnode = new TreeNode(sv.ToString());
                newnode.Tag = sv;
                TreeNode nodedc = new TreeNode(sv.Diachi);
                nodedc.Tag = sv;
                newnode.Nodes.Add(nodedc);
                slnode.Nodes.Add(newnode);
            }
        }
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message, "Lỗi!", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1);
    }
}

private void btnXoa_Click(object sender, EventArgs e)
{
    XoaSV();
}

public void XoaSV()
{
    try
    {
        TreeNode slnode = treDanhSach.SelectedNode;
        if (slnode.Level != 2)
            throw new Exception("Chỉ được xóa nút chứa mssv");
        else
        {
            DialogResult hoi;
            hoi = MessageBox.Show("Bạn chắc chắn muốn xóa?", "Xác
nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
            if (hoi == DialogResult.Yes)
            {
                SinhVien sv=(SinhVien)slnode.Tag;
                treDanhSach.Nodes.Remove(slnode);
                ds.xoaSV(sv);
            }
        }
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message);
    }
}

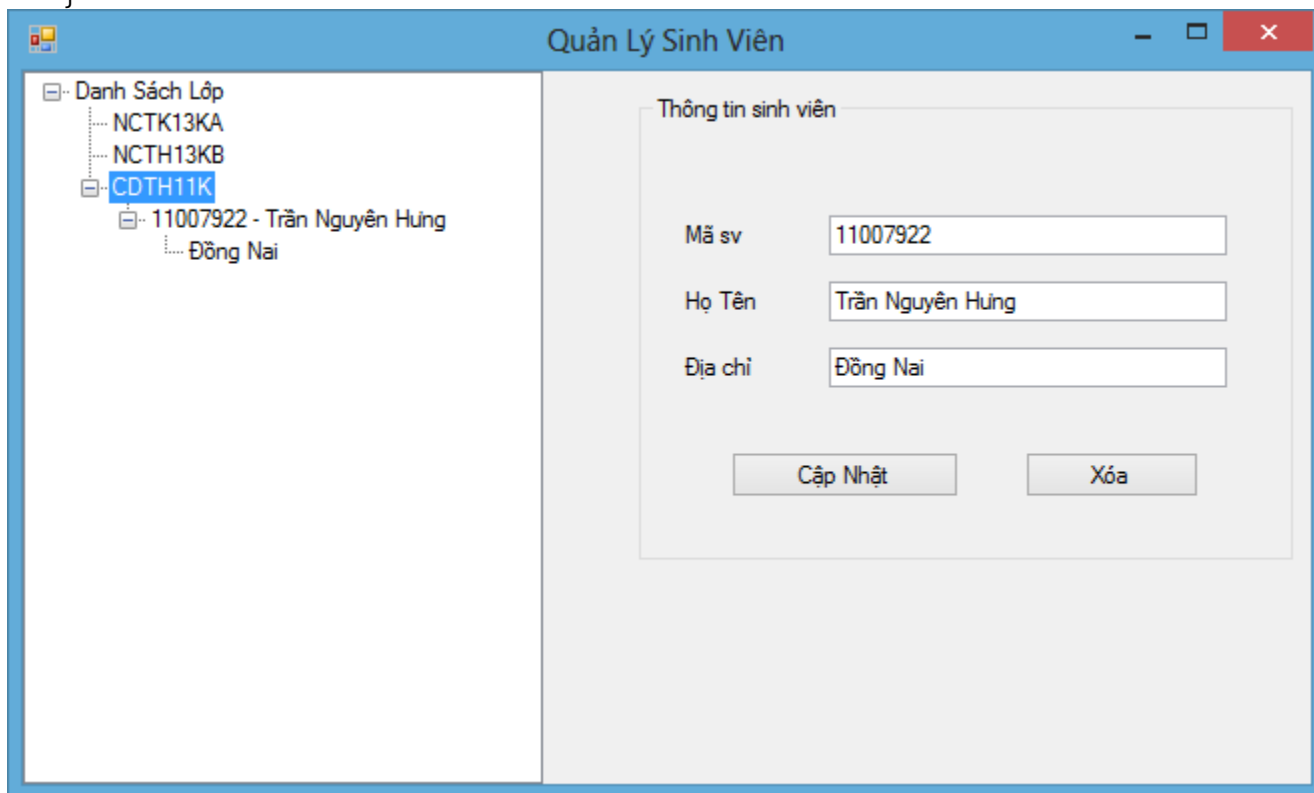
private void treDanhSach_AfterSelect(object sender, TreeViewEventArgs e)
{
    hienthongtin();
}
public void hienthongtin()

```

```

{
    TreeNode slnode=treDanhSach.SelectedNode;
    if (slnode.Level >= 2)
    {
        SinhVien sv = (SinhVien)slnode.Tag;
        txtMSSV.Text = sv.Mssv;
        txtHoTen.Text = sv.Hoten;
        txtDiaChi.Text = sv.Diachi;
    }
}
}

```



Module 7 : OOP

Bài 1

Viết đối tượng Account, Form để quản lý tài khoản ngân hàng, gồm các việc tạo tài khoản, gửi tiền, rút tiền, in báo cáo

```

public class Account
{
    private string ho, ten;
    private int sotk;
    private decimal sotien;
    #region Set/Get
    public decimal Sotien
    {
        get { return sotien; }
        set { sotien = value; }
    }
    public string Ten
    {
        get { return ten; }
        set { ten = value; }
    }

    public string Ho

```

```

    {
        get { return ho; }
        set { ho = value; }
    }

    public int Sotk
    {
        get { return sotk; }
        set { sotk = value; }
    }
}
#endregion End of Set/Get

#region cac methods
public Account()
{
}
public Account(int sotk, string ho, string ten, decimal sotien)
{
    this.sotk = sotk;
    this.ho = ho;
    this.ten = ten;
    this.sotien = sotien;
}
public void withdraw(decimal amount)
{
    if (0 < amount && amount <= sotien)
    {
        sotien -= amount;
    }
    else
    {
        throw new Exception("Khong du tien de rut");
    }
}
public bool deposit(decimal amount)
{
    if (amount > 0)
    {
        sotien += amount;
        return true;
    }
    else
    {
        throw new Exception("So tien nhap khong dung.");
    }
}

}
public string toString()
{
    return "Account Num: "+sotk+"\nName: "+ten+"\nHo: "+ho+"\nBalance: "+sotien;
}
}
#endregion End of cac methods

public partial class frmBankAcc : Form
{
    public frmBankAcc()
    {
        InitializeComponent();
    }
    Account acc;
    private void btnCreateNewAcc_Click(object sender, EventArgs e)
    {
        int sotk = Convert.ToInt32(txtAccNumb.Text);
        string ten = txtFrstName.Text;

```

```

        string ho = txtLstName.Text;
        decimal sotien = Convert.ToDecimal(txtBalance.Text);
        acc = new Account(sotk, ho, ten, sotien);
    }

    private void btnPrint_Click(object sender, EventArgs e)
    {
        if (acc == null)
            MessageBox.Show("Ban Chua Tao Tai Khoan");
        else
            MessageBox.Show(acc.toString(), "Thong tin");
    }

    private void btnWithdraw_Click(object sender, EventArgs e)
    {
        try
        {
            acc.withdraw(Convert.ToDecimal(txtWithdraw.Text));
        }
        catch (Exception err)
        {
            MessageBox.Show(err.Message, "Loi");
        }
    }

    private void btnDeposit_Click(object sender, EventArgs e)
    {
        try
        {
            acc.deposit(Convert.ToDecimal(txtDeposit.Text));
        }
        catch (Exception err)
        {
            MessageBox.Show(err.Message, "Loi");
        }
    }

    private void btnExit_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void txtAccNumb_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (!Char.IsDigit(e.KeyChar) && !Char.IsControl(e.KeyChar))
            e.Handled = true;
    }
}

```

Bài 2

Viết lớp Hóa Đơn gồm: Tên hàng, số lượng, đơn giá. Thiết kế form có chức năng tính tiền, thống kê

Yêu cầu:

Viết lớp Hóa Đơn đáp ứng yêu cầu xử lý của giao diện như sau:

- Khi chương trình hiện lên thì nút Tiếp mờ đi.
- Nút "Tính" (hoặc Enter) cho phép tính thành tiền sau khi nhập tên hàng, số lượng, đơn giá hợp lệ. Sau đó làm mờ nút Tính, hiện nút Tiếp.
- Nút "Tiếp" cho phép xóa các nội dung trong các ô nhập liệu và ô thành tiền, đồng thời làm mờ nút Tiếp và hiện nút Tính.
- Nút "Thống kê" cho phép thống kê các thông tin đã nhập.
- Nút "Kết thúc" (hoặc Esc) để đóng chương trình.

```
public class HoaDon
{
    string tenhang;
    int soluong;
    decimal dongia, thanhtien;
#region Get/Set
    public string Tenhang
    {
        get { return tenhang; }
        set { tenhang = value; }
    }
    public int Soluong
    {
        get { return soluong; }
        set { soluong = value; }
    }
    public decimal ThanhTien
    {
```



```

        get { return thanhtien; }
        set { thanhtien = value; }
    }
    public decimal Dongia
    {
        get { return dongia; }
        set { dongia = value; }
    }
#endregion end of Get/Set

#region methods
    public HoaDon()
    {
    }
    public HoaDon(string tenhang,int soluong,decimal dongia)
    {
        this.tenhang = tenhang;
        this.soluong = soluong;
        this.dongia = dongia;
        taothanhtien();
    }
    private void taothanhtien()
    {
        thanhtien = Convert.ToDecimal(soluong* dongia);
    }
#endregion end of methods
}

using System.IO;
public class DSHoaDon
{
    ArrayList ds;
    public DSHoaDon()
    {
        ds = new ArrayList();
    }
    public void themHD(HoaDon hd)
    {
        ds.Add(hd);
    }
    public void xoaHD(HoaDon hd)
    {
        ds.Remove(hd);
    }
    public ArrayList XuatDSHD()
    {
        return ds;
    }
    public int LaySoluongHD()
    {
        return ds.Count;
    }
    public decimal laydoanhthu(){
        decimal doanhthu=0;
        foreach (HoaDon hd in ds)
            doanhthu += hd.Thanhtien;
        return doanhthu;
    }
}

public partial class frmQLBH : Form
{
    public frmQLBH()
    {
        InitializeComponent();
    }
}

```

```

    }
    DSHoaDon ds = new DSHoaDon();
    HoaDon hd;
    private void btnTinh_Click(object sender, EventArgs e)
    {
        string tenhang = txtTenHang.Text;
        int soluong = Convert.ToInt32(txtSoLuong.Text);
        decimal dongia = Convert.ToDecimal(txtDonGia.Text);
        hd = new HoaDon(tenhang, soluong, dongia);
        txtThanhTien.Text = hd.Thanhtien.ToString();
        ds.themHD(hd);
        btnTinh.Enabled = false;
        btnTiep.Enabled = true;
    }

    private void frmQLBH_Load(object sender, EventArgs e)
    {
        btnTiep.Enabled = false;
    }

    private void btnTiep_Click(object sender, EventArgs e)
    {
        txtThanhTien.Clear();
        txtDonGia.Clear();
        txtSoLuong.Clear();
        txtTenHang.Clear();
        btnTinh.Enabled = true;
        btnTiep.Enabled = false;
    }

    private void btnThongKe_Click(object sender, EventArgs e)
    {
        txtSoDonDaBan.Text = ds.LaySoluongHD().ToString();
        txtDoanhThu.Text = ds.laydoanhthu().ToString();
    }

    private void btnKetThuc_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

```

Bài 3

Ứng dụng lớp Hóa Đơn của bài 2, viết chương trình có giao diện như sau:

- Nút "Thêm" cho phép xuất thành tiền và cập nhật thông tin thống kê, đồng thời thêm mặt hàng đang nhập vào ListBox (chỉ hiển thị tên hàng). Kiểm tra dữ liệu nhập.
- Nút "Xóa" cho phép xóa dòng đang chọn trong ListBox, đồng thời cập nhật thông tin thống kê, trước khi xóa phải xác nhận lại.
- Nút "Tìm" cho phép tìm hàng đã bán khi nhập vào tên hàng, nếu tìm thấy thì cho dòng đó sáng lên trong ListBox.
- Khi chọn một dòng trên ListBox thì hiện thông tin tương ứng lên các textbox và label.

```
public class HoaDon
{
    string tenhang;
    int soluong;
    decimal dongia, thanhTien;
    #region Get/Set
    public string Tenhang
    {
        get { return tenhang; }
        set { tenhang = value; }
    }
    public int Soluong
    {
        get { return soluong; }
        set { soluong = value; }
    }
    public decimal ThanhTien
    {
        get { return thanhTien; }
        set { thanhTien = value; }
    }
    public decimal Dongia
    {
        get { return dongia; }
        set { dongia = value; }
    }
    #endregion End of get/set
}
```

```

#region Methods
public HoaDon()
{
}
public HoaDon(string tenhang, int soluong, decimal dongia)
{
    this.tenhang = tenhang;
    this.soluong = soluong;
    this.dongia = dongia;
    tinhhtien();
}
public void tinhhtien()
{
    thanhtien = soluong * dongia;
}

public override string ToString()
{
    return this.tenhang;
}

//public override bool Equals(object obj)
//{
//    return this.Tenhang.Equals(((HoaDon)obj).Tenhang);
//}

#endregion End of methods
}

using System.IO;
public class DSHoaDon
{
    ArrayList ds;
    public DSHoaDon()
    {
        ds = new ArrayList();
    }
    public void themHD(HoaDon hd)
    {
        ds.Add(hd);
    }
    public void xoaHD(HoaDon hd)
    {
        ds.Remove(hd);
    }
    public ArrayList XuatDSHD()
    {
        return ds;
    }
    public int LaySoluongHD()
    {
        return ds.Count;
    }
    public decimal laydoanhthu()
    {
        decimal doanhthu = 0;
        foreach (HoaDon hd in ds)
            doanhthu += hd.Thanhtien;
        return doanhthu;
    }

    public bool kiemtra(HoaDon hd)

```

```

        {
            return this.ds.Contains(hd);
        }
    }
}
public partial class frmThongTinBanHang : Form
{
    public frmThongTinBanHang()
    {
        InitializeComponent();
    }
    DSHoaDon ds = new DSHoaDon();
    HoaDon hd;
    private void btnThem_Click(object sender, EventArgs e)
    {
        try
        {
            string tenhang = txtTenHang.Text;
            int soluong = Convert.ToInt32(txtSoLuong.Text);
            decimal dongia = Convert.ToDecimal(txtDonGia.Text);
            hd = new HoaDon(tenhang, soluong, dongia);
            if (tenhang.Length == 0 || soluong <= 0 || dongia <= 0)
                MessageBox.Show("Du Lieu Nhap Sai", "Thong Bao");
            //else if (ds.kiemtra(hd)==true)
            //{
            //    MessageBox.Show("Hoa Don nay da co", "Thong Bao");
            //}
            else
            {
                txtThanhTien.Text = hd.Thanhtien.ToString();
                ds.themHD(hd);
                CapNhatListBox();
            }
        }
        catch (Exception)
        {
            MessageBox.Show("Du Lieu Nhap Sai", "Thong Bao");
        }
    }
    public void CapNhatListBox()
    {
        lstThongTin.Items.Clear();
        foreach (HoaDon hd in ds.XuatDSHD())
        {
            lstThongTin.Items.Add(hd);
        }
        txtTongHoaDon.Text = ds.LaySoluongHD().ToString();
        txtDoanhThu.Text = ds.laydoanhthu().ToString();
    }
    private void btnXoa_Click(object sender, EventArgs e)
    {
        if (lstThongTin.SelectedIndex == -1)
        {
            MessageBox.Show("Ban chua chon gia tri", "Thong bao");
        }
        else
        {
            DialogResult hoi;
            hoi = MessageBox.Show("Ban Co Chac Muon Xoa", "Xac Nhan", MessageBoxButtons.YesNo,
                MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
            if (hoi == DialogResult.Yes)
            {
                ds.xoaHD((HoaDon)lstThongTin.SelectedItem);
            }
        }
    }
}

```

```

        CapNhatListBox();
    }
}

private void btnTim_Click(object sender, EventArgs e)
{
    string giatri = "";
    frmNhapGiaTriTim frm = new frmNhapGiaTriTim();
    frm.Text = "Tim Kiem";
    if (frm.ShowDialog() == DialogResult.OK)
    {
        giatri = frm.txtGiaTri.Text;
        Boolean flag = false;
        foreach (HoaDon hd in ds.XuatDSHD())
        {
            if (hd.Tenhang.Equals(giatri.Trim()))
            {
                lstThongTin.SelectedItem = hd;
                flag = true;
                break;
            }
        }
        if (flag == false)
            MessageBox.Show("Khong tin thay " + giatri, "Thong Bao");
    }
}

private void lstThongTin_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        HoaDon hd1 = (HoaDon)lstThongTin.SelectedItem;
        txtTenHang.Text = hd1.Tenhang;
        txtSoLuong.Text = hd1.Soluong.ToString();
        txtDonGia.Text = hd1.Dongia.ToString();
        txtThanhTien.Text = hd1.ThanhTien.ToString();
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message);
    }
}
}

```

The screenshot shows a Windows application window titled "Thông Tin Ban Hàng". It features a form with the following elements:

- Tên Hàng:** A text box containing "Linh Tinh".
- Số Lượng:** A text box containing "12".
- Đơn giá:** A text box containing "1222".
- Thành Tiền:** A text box containing "14664".
- Buttons:** Three buttons are located below the form: "+ Thêm" (Add), "Xóa" (Delete), and "Tìm" (Find).
- Thống Kê (Summary):** A section containing two text boxes: "Tổng Số Hóa Đơn" (Total Number of Invoices) with value "1", and "Doanh Thu" (Revenue) with value "14664".
- Output Area:** A large text area on the right side of the window displaying "Linh Tinh".

Bài 4

Thực hiện lại bài 3 thay ListBox thành ListView

```
public partial class frmM7BT4 : Form
{
    public frmM7BT4()
    {
        InitializeComponent();
    }
    DSHoaDon ds = new DSHoaDon();
    HoaDon hd;
    private void btnThem_Click(object sender, EventArgs e)
    {
        try
        {
            string tenhang = txtTenHang.Text;
            int soluong = Convert.ToInt32(txtSoLuong.Text);
            decimal dongia = Convert.ToDecimal(txtDonGia.Text);
            hd = new HoaDon(tenhang, soluong, dongia);
            if (tenhang.Length == 0 || soluong <= 0 || dongia <= 0)
                MessageBox.Show("Du Lieu Nhap Sai", "Thong Bao");
            //else if (ds.kiemtra(hd)==true)
            //{
            //    MessageBox.Show("Hoa Don nay da co", "Thong Bao");
            //}
            else
            {
                txtThanhTien.Text = hd.Thanhtien.ToString();
                ds.themHD(hd);
                CapNhatListView();
            }
        }
        catch (Exception)
        {
            MessageBox.Show("Du Lieu Nhap Sai", "Thong Bao");
        }
    }
    public void CapNhatListView()
    {
        lstvThongKe.Items.Clear();
    }
}
```

```

foreach (HoaDon hd in ds.XuatDSHD())
{
    ListViewItem item = new ListViewItem(hd.Tenhang);
    item.SubItems.Add(hd.Soluong.ToString());
    item.SubItems.Add(hd.Dongia.ToString());
    item.SubItems.Add(hd.Thanhtien.ToString());
    item.Tag = hd;
    lstvThongKe.Items.Add(item);
}
txtTongHoaDon.Text = ds.LaySoluongHD().ToString();
txtDoanhThu.Text = ds.laydoanhthu().ToString();
}

private void btnXoa_Click(object sender, EventArgs e)
{
    if (lstvThongKe.SelectedItems.Count == 0)
    {
        MessageBox.Show("Ban chua chon gia tri", "Thong bao");
    }
    else
    {
        DialogResult hoi;
        hoi = MessageBox.Show("Ban Co Chac Muon Xoa", "Xac Nhan", MessageBoxButtons.YesNo,
            MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
        if (hoi == DialogResult.Yes)
        {
            foreach (ListViewItem item in lstvThongKe.SelectedItems)
            {
                HoaDon hd = (HoaDon)item.Tag;
                ds.xoaHD(hd);
            }
            CapNhatListView();
        }
    }
}

private void btnTim_Click(object sender, EventArgs e)
{
    string giatri = "";
    frmNhapGiaTriTim frm = new frmNhapGiaTriTim();
    frm.Text = "Tim Kiem";
    if (frm.ShowDialog() == DialogResult.OK)
    {
        giatri = frm.txtGiaTri.Text;
        Boolean flag = false;
        foreach (ListViewItem item in lstvThongKe.Items)
        {
            HoaDon hd = (HoaDon)item.Tag;
            if (hd.Tenhang.Equals(giatri.Trim()))
            {
                lstvThongKe.Focus();
                lstvThongKe.Items[item.Index].Selected = true;
                flag = true;
                break;
            }
        }
        if (flag == false)
            MessageBox.Show("Khong tin thay " + giatri, "Thong Bao");
    }
}

private void lstvThongKe_SelectedIndexChanged(object sender, EventArgs e)
{
    try

```



```

{
    if (lstvThongKe.SelectedItems.Count > 0)
    {
        HoaDon hd1 = (HoaDon)lstvThongKe.SelectedItems[0].Tag;
        txtTenHang.Text = hd1.Tenhang;
        txtSoLuong.Text = hd1.Soluong.ToString();
        txtDonGia.Text = hd1.Dongia.ToString();
        txtThanhTien.Text = hd1.Thanhtien.ToString();
    }
}
catch (Exception err)
{
    MessageBox.Show(err.Message);
}
}
}

```

Tên Hàng	Số Lượng	Đơn Giá	Thành Tiền
Linh Tinh	22	1222	26884

Module 8 : Array and Collection

Bài 1

Thiết kế form gồm các Button thực hiện các chức năng sau: cho phép nhập vào số phần tử tối đa cho mảng 1 chiều, tính tổng mảng, đếm và tính tổng số phần tử lẻ, tìm phần tử nhỏ nhất, tăng mỗi phần tử lên 2, sắp mảng tăng, sắp mảng giảm.

```

public partial class frmArrayDemo : Form
{
    public frmArrayDemo()
    {
        InitializeComponent();
    }
    int soPT;
    private void btnSoPhanTu_Click(object sender, EventArgs e)
    {

```

```

frmNhapSoPT frm = new frmNhapSoPT();
try
{
    if (frm.ShowDialog() == DialogResult.OK)
        soPT = Convert.ToInt32(frm.txtSoPT.Text);
    PhatSinhMang();
    XuatMang();
}
catch (Exception)
{
    MessageBox.Show("Số phần tử không đúng");
}
}
int[] Mang;
private void PhatSinhMang()
{
    Mang = new int[soPT];
    Random rd = new Random();
    for (int i = 0; i < Mang.Length; i++)
    {
        Mang[i] = rd.Next(1, 101);
    }
}

private void XuatMang()
{
    string tt="";
    for (int i = 0; i < Mang.Length; i++)
    {
        tt += " " + Mang[i];
    }
    txtMangGoc.Text = tt;
}

private int TongMang()
{
    int tong = 0;
    for (int i = 0; i < Mang.Length; i++)
    {
        tong += Mang[i];
    }
    return tong;
}

private int soPTLe()
{
    int dem = 0;
    for (int i = 0; i < Mang.Length; i++)
    {
        if (Mang[i] % 2 != 0)
            dem++;
    }
    return dem;
}

private int TongPTLe(){
    int tong = 0;
    for (int i = 0; i < Mang.Length; i++)
    {
        if (Mang[i] % 2 != 0)
            tong += Mang[i];
    }
    return tong;
}

```

```

private int PTNhoNhat()
{
    int ptnhonhat = Mang[0];
    for (int i = 1; i < Mang.Length; i++)
    {
        if (ptnhonhat > Mang[i])
            ptnhonhat = Mang[i];
    }
    return ptnhonhat;
}

private void TangMoiPTLen2()
{
    for (int i = 0; i < Mang.Length; i++)
    {
        Mang[i] += 2;
    }
    XuatMang();
}

private void SapMangTang()
{
    for (int i = 0; i < Mang.Length-1; i++)
    {
        for (int j = i+1; j < Mang.Length; j++)
        {
            if (Mang[j] < Mang[i])
            {
                int temp = Mang[i];
                Mang[i] = Mang[j];
                Mang[j] = temp;
            }
        }
    }
    XuatMang();
}

private void SapMangGiam()
{
    for (int i = 0; i < Mang.Length - 1; i++)
    {
        for (int j = i + 1; j < Mang.Length; j++)
        {
            if (Mang[j] > Mang[i])
            {
                int temp = Mang[i];
                Mang[i] = Mang[j];
                Mang[j] = temp;
            }
        }
    }
    XuatMang();
}

private void btnTongMang_Click(object sender, EventArgs e)
{
    if (Mang != null && Mang.Length > 0)
        lblKetQua.Text = "Tổng Mảng = " + TongMang();
    else
        MessageBox.Show("Bạn chưa nhập mảng");
}

private void btnDemSoLe_Click(object sender, EventArgs e)
{

```

```

        if (Mang != null && Mang.Length > 0)
            lblKetQua.Text = "Số Phần tử lẻ = " + soPTLe();
        else
            MessageBox.Show("Bạn chưa nhập mảng");
    }

    private void btnTongSoLe_Click(object sender, EventArgs e)
    {
        if (Mang != null && Mang.Length > 0)
            lblKetQua.Text = "Tổng số Phần tử lẻ = " + TongPTLe();
        else
            MessageBox.Show("Bạn chưa nhập mảng");
    }

    private void btnSoNhoNhat_Click(object sender, EventArgs e)
    {
        if (Mang != null && Mang.Length > 0)
            lblKetQua.Text = "Phần tử nhỏ nhất = " + PTNhoNhat();
        else
            MessageBox.Show("Bạn chưa nhập mảng");
    }

    private void btnTangLen2_Click(object sender, EventArgs e)
    {
        if (Mang != null && Mang.Length > 0)
            TangMoiPTLen2();
        else
            MessageBox.Show("Bạn chưa nhập mảng");
    }

    private void btnSapTangDan_Click(object sender, EventArgs e)
    {
        if (Mang != null && Mang.Length > 0)
            SapMangTang();
        else
            MessageBox.Show("Bạn chưa nhập mảng");
    }

    private void btnSapGiamDan_Click(object sender, EventArgs e)
    {
        if (Mang != null && Mang.Length > 0)
            SapMangGiam();
        else
            MessageBox.Show("Bạn chưa nhập mảng");
    }
}

```

Bài 2

Làm lại bài 1 thay Mảng 1 chiều bằng ArrayList

```
public partial class frmDemoArrayList : Form
{
    public frmDemoArrayList()
    {
        InitializeComponent();
    }
    int soPT;
    private void btnSoPhanTu_Click(object sender, EventArgs e)
    {
        frmNhapSoPT frm = new frmNhapSoPT();
        try
        {
            if (frm.ShowDialog() == DialogResult.OK)
            {
                soPT = Convert.ToInt32(frm.txtSoPT.Text);
                PhatSinhMang();
                XuatMang();
            }
        }
        catch (Exception)
        {
            MessageBox.Show("Số phần tử không đúng");
        }
    }
    ArrayList Mang=new ArrayList();
    private void PhatSinhMang()
    {
        Mang.Clear();
        Random rd = new Random();
        for (int i = 0; i < soPT; i++)
        {
            Mang.Add(rd.Next(1, 101));
        }
    }
    private void XuatMang()
    {

```

```

        string tt = "";
        for (int i = 0; i < Mang.Count; i++)
        {
            tt += " " + Mang[i];
        }
        txtMangGoc.Text = tt;
    }

    private int TongMang()
    {
        int tong = 0;
        foreach (int i in Mang)
        {
            tong += i;
        }
        return tong;
    }

    private int soPTLe()
    {
        int dem = 0;
        foreach (int i in Mang)
        {
            if (i % 2 != 0)
                dem++;
        }
        return dem;
    }

    private int TongPTLe()
    {
        int tong = 0;
        foreach (int i in Mang)
        {
            if (i % 2 != 0)
                tong += i;
        }
        return tong;
    }

    private int PTNhoNhat()
    {
        int ptnhonhat = (int)Mang[0];
        foreach (int i in Mang)
        {
            if (ptnhonhat > i)
                ptnhonhat = i;
        }
        return ptnhonhat;
    }

    private void TangMoiPTLen2()
    {
        int i=0;
        while(i<Mang.Count){
            Mang[i] = (int)Mang[i]+ 2;
            i++;
        }
        XuatMang();
    }

    private void SapMangTang()
    {
        Mang.Sort();
        XuatMang();
    }
}

```

```

private void SapMangGiam()
{
    Mang.Reverse();
    XuatMang();
}

private void btnTongMang_Click(object sender, EventArgs e)
{
    if(Mang.Count>0)
        lblKetQua.Text = "Tổng Mảng = " + TongMang();
}

private void btnDemSoLe_Click(object sender, EventArgs e)
{
    if (Mang.Count > 0)
        lblKetQua.Text = "Số Phần tử lẻ = " + soPTLe();
}

private void btnTongSoLe_Click(object sender, EventArgs e)
{
    if (Mang.Count > 0)
        lblKetQua.Text = "Tổng số Phần tử lẻ = " + TongPTLe();
}

private void btnSoNhoNhat_Click(object sender, EventArgs e)
{
    if (Mang.Count > 0)
        lblKetQua.Text = "Phần tử nhỏ nhất = " + PTNhoNhat();
}

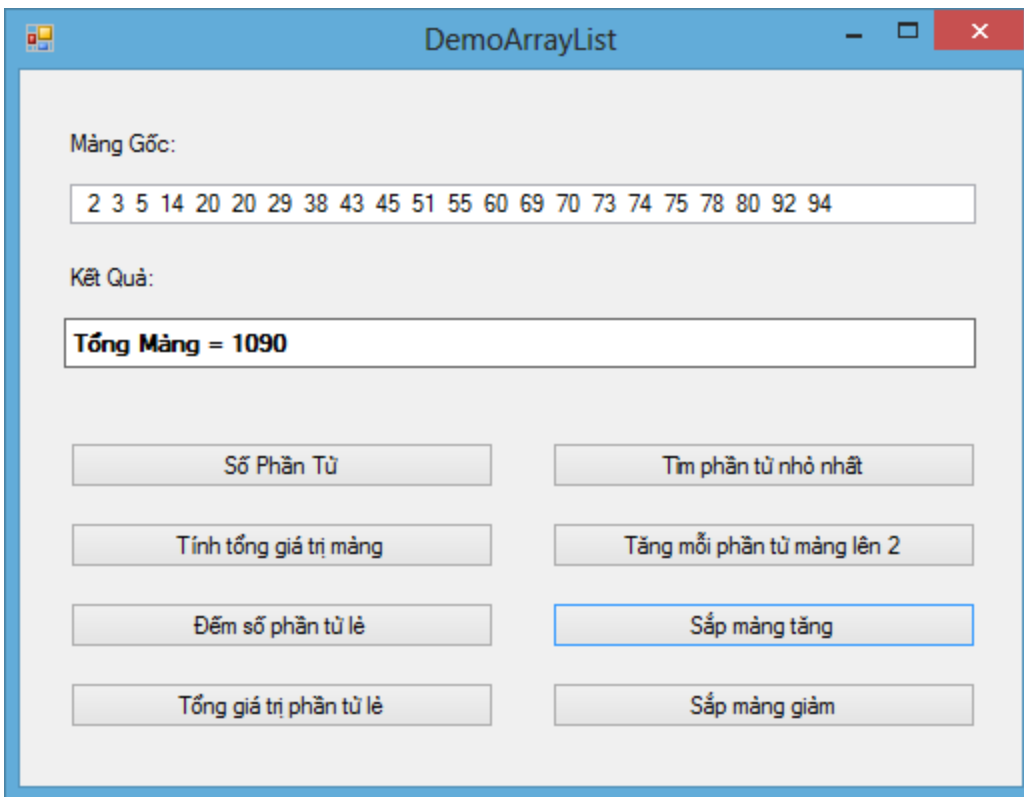
private void btnTangLen2_Click(object sender, EventArgs e)
{
    if (Mang.Count > 0)
        TangMoiPTLen2();
}

private void btnSapTangDan_Click(object sender, EventArgs e)
{
    if (Mang.Count > 0)
        SapMangTang();
}

private void btnSapGiamDan_Click(object sender, EventArgs e)
{
    if (Mang.Count > 0)
        SapMangGiam();
}

}

```



Bài 3

Viết chương trình quản lý thông tin sinh viên gồm các chức năng thêm, xóa, sửa và tìm kiếm

```
public class SinhVien
{
    String masv, hoten, diachi;

    public SinhVien(String masv, String hoten, String diachi)
    {
        this.Masv = masv;
        this.Hoten = hoten;
        this.Diachi = diachi;
    }

    public String Diachi
    {
        get { return diachi; }
        set { diachi = value; }
    }

    public String Hoten
    {
        get { return hoten; }
        set { hoten = value; }
    }

    public String Masv
    {
        get { return masv; }
        set { masv = value; }
    }

    public override bool Equals(object obj)
    {
        return this.Masv.Equals(((SinhVien)obj).Masv);
    }
}
```



```

    }

using System.IO;
public class DSSV
{
    ArrayList list;
    public DSSV()
    {
        list = new ArrayList();
    }

    public void ThemSV(SinhVien sv)
    {
        if (list.Contains(sv))
            throw new Exception("Trùng MSSV");
        else
            list.Add(sv);
    }

    public void XoaSV(SinhVien sv)
    {
        if (list.Contains(sv))
            list.Remove(sv);
        else
            throw new Exception("Không có sinh viên này trong danh sách");
    }

    public void SuaSV(SinhVien sv)
    {
        if (list.Contains(sv))
            list[list.IndexOf(sv)] = sv;
        else
            throw new Exception("Không có sinh viên này trong danh sách");
    }

    public SinhVien TimSinhVien(SinhVien sv)
    {
        if (list.Contains(sv))
            return (SinhVien)list[list.IndexOf(sv)];
        else
            return null;
    }

    public ArrayList DSSV1()
    {
        return list;
    }
}

public partial class frmQLTTSV : Form
{
    public frmQLTTSV()
    {
        InitializeComponent();
    }

    DSSV ds = new DSSV();
    private void btnThoat_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    public void CapNhatListView()

```

```

{
    lstvThongTin.Items.Clear();
    foreach (SinhVien sv in ds.DSSV1())
    {
        ListViewItem item = new ListViewItem(sv.Masv);
        item.SubItems.Add(sv.Hoten);
        item.SubItems.Add(sv.Diachi);
        item.Tag = sv;
        lstvThongTin.Items.Add(item);
    }
}

private void btnThem_Click(object sender, EventArgs e)
{
    try
    {
        string mssv = txtMSSV.Text;
        string hoten = txtHoTen.Text;
        string diachi = txtDiaChi.Text;
        SinhVien sv = new SinhVien(mssv, hoten, diachi);
        ds.ThemSV(sv);
        CapNhatListView();
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message);
    }
}

private void btnXoa_Click(object sender, EventArgs e)
{
    try
    {
        string mssv = txtMSSV.Text;
        string hoten = txtHoTen.Text;
        string diachi = txtDiaChi.Text;
        SinhVien sv = new SinhVien(mssv, hoten, diachi);
        ds.XoaSV(sv);
        CapNhatListView();
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message);
    }
}

private void btnSua_Click(object sender, EventArgs e)
{
    try
    {
        string mssv = txtMSSV.Text;
        string hoten = txtHoTen.Text;
        string diachi = txtDiaChi.Text;
        SinhVien sv = new SinhVien(mssv, hoten, diachi);
        ds.SuaSV(sv);
        CapNhatListView();
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message);
    }
}

private void btnTim_Click(object sender, EventArgs e)

```

```

{
    String tim = "";
    frmTimKiem frm=new frmTimKiem();
    if (frm.ShowDialog() == DialogResult.OK)
    {
        tim = frm.txtGiaTri.Text;
        TimKiem(tim);
    }
}

public void TimKiem(string tim)
{
    SinhVien sv = new SinhVien(tim, "", "");
    if (ds.TimSinhVien(sv) == null)
        MessageBox.Show("Không tìm thấy sinh viên này");
    else
    {
        SinhVien svtimdc = ds.TimSinhVien(sv);
        foreach (ListViewItem item in lstvThongTin.Items)
        {
            if (item.Tag == svtimdc)
            {
                lstvThongTin.Focus();
                item.Selected = true;
            }
        }
    }
}

private void lstvThongTin_SelectedIndexChanged(object sender, EventArgs e)
{
    if (lstvThongTin.SelectedItems.Count > 0)
    {
        SinhVien sv = (SinhVien)lstvThongTin.SelectedItems[0].Tag;
        txtMSSV.Text = sv.Masv;
        txtHoTen.Text = sv.Hoten;
        txtDiaChi.Text = sv.Diachi;
    }
}
}

```

Quản lý thông tin sinh viên

THÔNG TIN SINH VIÊN

Mã số SV:

Họ và Tên:

Địa chỉ:

Mã SV	Họ Tên	Địa Chỉ
11007922	Trần Nguyễn Hưng	Đồng Nai

Module 9 : File And Stream

Bài 2

Viết lớp Sinh viên gồm: mã sv, học tên sv, địa chỉ, email. Trong đó phân biệt bởi MSSV.

Xây dựng lớp Lớp học gồm: Mã lớp, tên lớp, GVCN. Các lớp phân biệt bằng MSLớp. Một lớp học có thể có nhiều sinh viên. Lớp này thực hiện các chức năng thêm, xóa, sửa sinh viên, trích 1 sinh viên khi biết mssv hoặc vị trí, lấy số lượng sinh viên trong ds.

Xử Dụng Tree :

```
[Serializable]
public class HocSinh
{
    string mahs, hoten, diachi, Email, Lop;

    public HocSinh(string mshs, string hoten, string diachi, string email, string lop)
    {
        this.Mahs = mshs;
        this.Hoten = hoten;
        this.Diachi = diachi;
        this.Email1 = email;
        this.Lop1 = lop;
    }

    public string Lop1
    {
        get { return Lop; }
        set {
            try
            {
                if (value.Length == 0)
                    throw new Exception("Lớp học không được để trống");
                else
                    Lop = value;
            }
        }
    }
}
```

```

        catch (Exception err)
        {

            throw err;

        }
    }

    public string Email1
    {
        get { return Email; }
        set { Email = value; }
    }

    public string Diachi
    {
        get { return diachi; }
        set { diachi = value; }
    }

    public string Hoten
    {
        get { return hoten; }
        set {
            try
            {
                if (value.Length == 0)
                    throw new Exception("Họ Tên không được để trống");
                else
                    hoten = value;
            }
            catch (Exception err)
            {

                throw err;

            }
        }
    }

}

public string Mahs
{
    get { return mahs; }
    set {
        try
        {
            if (value.Length == 0)
                throw new Exception("MSHS không được để trống");
            else
                mahs = value;
        }
        catch (Exception err)
        {

            throw err;

        }
    }
}

public override bool Equals(object obj)
{
    return this.Mahs.Equals(((HocSinh)obj).Mahs);
}

```

```

    }
    using System.Collections;

    [Serializable]
    public class LopHoc
    {
        string maLop, tenlop, gvcn;
        ArrayList dSHS;

        public ArrayList DSHS
        {
            get { return dSHS; }
            set { dSHS = value; }
        }

        public LopHoc(string mslop, string tenlop, string gvcn)
        {
            this.MaLop = mslop;
            this.Tenlop = tenlop;
            this.Gvcn = gvcn;
            dSHS = new ArrayList();
        }

        public string MaLop
        {
            get { return maLop; }
            set {
                if (value.Length == 0)
                    throw new Exception("Mã Lớp không được để trống");
                else
                    maLop = value; }
        }

        public string Tenlop
        {
            get { return tenlop; }
            set {
                if (value.Length == 0)
                    throw new Exception("Tên Lớp không được để trống");
                else
                    tenlop = value; }
        }

        public string Gvcn
        {
            get { return gvcn; }
            set {
                if (value.Length == 0)
                    throw new Exception("GVCN không được để trống");
                else
                    gvcn = value; }
        }

        public void ThemHS(HocSinh hs)
        {
            if (!this.dSHS.Contains(hs))
                dSHS.Add(hs);
            else
                throw new Exception("Trùng MS Học Sinh");
        }

        public void SuaHS(HocSinh hs)
        {

```

```

        if (this.dSHS.Contains(hs))
        {
            dSHS[dSHS.IndexOf(hs)]=hs;
        }
        else
            throw new Exception("Bạn không được sửa mã HS");
    }

    public void XoaHS(HocSinh hs)
    {
        if (dSHS.Contains(hs))
        {
            dSHS.Remove(hs);
        }
        else
            throw new Exception("Không tồn tại Học Sinh này");
    }

    public ArrayList LayDSHS()
    {
        return dSHS;
    }

    public override bool Equals(object obj)
    {
        return this.MaLop.Equals(((LopHoc)obj).MaLop);
    }

    public bool kths(HocSinh hs )
    {
        return dSHS.Contains(hs);
    }
}

using System.Collections;

[Serializable]
public class DSLopHoc
{
    ArrayList DSLH;
    public DSLopHoc()
    {
        DSLH = new ArrayList();
    }

    public void themLophoc(LopHoc lh)
    {
        if (this.DSLH.Contains(lh))
            throw new Exception("Trùng mã lớp");
        else
            DSLH.Add(lh);
    }

    public void SuaLopHoc(LopHoc lh )
    {
        if (DSLH.Contains(lh))
        {
            DSLH[DSLH.IndexOf(lh)] = lh;
        }
        else
            throw new Exception("Lớp học này không tồn tại");
    }

    public void XoaLophoc(LopHoc lh)

```

```

    {
        if (DSLH.Contains(lh))
            DSLH.Remove(lh);
        else
            throw new Exception("Không có lớp học này");
    }
    public ArrayList LayDSLH()
    {
        return DSLH;
    }

    public bool kttrungma(HocSinh hs){
        foreach (LopHoc lh in DSLH)
        {
            if (lh.kths(hs))
                return true;
        }
        return false;
    }
}

using System.IO;
using System.Runtime.Serialization.Formatters.Binary;

public class DocGhiFile
{
    public void GhiFile(string PathFile, object obj)
    {
        FileStream fs = new FileStream(PathFile, FileMode.OpenOrCreate, FileAccess.Write,
        FileShare.None);
        BinaryFormatter bf = new BinaryFormatter();
        try
        {
            bf.Serialize(fs, obj);
        }
        catch (Exception)
        {
            throw new Exception("Không thể ghi file");
        }
        finally
        {
            fs.Close();
        }
    }

    public object DocFile(string PathFile)
    {
        FileStream fs = new FileStream(PathFile, FileMode.Open, FileAccess.Read, FileShare.Read);
        BinaryFormatter bf = new BinaryFormatter();
        object obj;
        try
        {
            obj = bf.Deserialize(fs);
        }
        catch (Exception)
        {
            throw new Exception("Không thể đọc file");
        }
        fs.Close();
        return obj;
    }
}

```



```

using System.IO;

public partial class frmQLLopHoc : Form
{
    public frmQLLopHoc()
    {
        InitializeComponent();
    }
    public frmQLLopHoc(DSLopHoc ds)
    {
        InitializeComponent();
        this.ds = ds;
    }
    public DSLopHoc ds;
    private void btnThem_Click(object sender, EventArgs e)
    {
        try
        {
            string mslop = txtMaLop.Text;
            string tenlop = txtTenLop.Text;
            string gvcn = txtGVCN.Text;
            LopHoc lh = new LopHoc(mslop, tenlop, gvcn);
            ds.themLophoc(lh);
            capnhatListView();
        }
        catch (Exception err)
        {
            MessageBox.Show(err.Message);
        }
    }

    public void capnhatListView()
    {
        listView1.Items.Clear();
        foreach (LopHoc lh in ds.LayDSLH())
        {
            ListViewItem item = new ListViewItem(lh.MaLop);
            item.SubItems.Add(lh.Tenlop);
            item.SubItems.Add(lh.Gvcn);
            item.Tag = lh;
            listView1.Items.Add(item);
        }
    }

    private void btnThoat_Click(object sender, EventArgs e)
    {
        this.DialogResult = DialogResult.Yes;
    }

    private void btnXoa_Click(object sender, EventArgs e)
    {
        if(listView1.SelectedItems.Count>0)
            foreach (ListViewItem item in listView1.SelectedItems)
            {
                LopHoc lh = (LopHoc)item.Tag;
                ds.Xoalophoc(lh);
            }
        capnhatListView();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        listView1.Focus();
    }
}

```

```

        listView1.TopItem.Selected = true;
    }

    private void frmQLLopHoc_Load(object sender, EventArgs e)
    {
        if (ds.LayDSLH().Count > 0)
        {
            capnhatListView();
            listView1.Items[0].Selected = true;
        }
    }

    private void btnSua_Click(object sender, EventArgs e)
    {
        if (listView1.SelectedItems.Count > 0)
        {
            try
            {
                LopHoc lhd = (LopHoc)listView1.SelectedItems[0].Tag;
                string mslop = txtMaLop.Text;
                string tenlop = txtTenLop.Text;
                string gvcn = txtGVCN.Text;
                LopHoc temp = lhd;
                temp.Tenlop = tenlop;
                temp.Gvcn = gvcn;
                ds.SualopHoc(temp);
                capnhatListView();
            }
            catch (Exception err)
            {
                MessageBox.Show(err.Message);
            }
        }
        else
        {
            MessageBox.Show("Bạn chưa chọn lớp để sửa");
        }
    }

    private void listView1_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (listView1.SelectedItems.Count > 0)
        {
            LopHoc lhd = (LopHoc)listView1.SelectedItems[0].Tag;
            txtMaLop.Text=lhd.MaLop;
            txtTenLop.Text=lhd.Tenlop;
            txtGVCN.Text=lhd.Gvcn;
            lopdangchon();
        }
    }

    private void button5_Click(object sender, EventArgs e)
    {
        listView1.Focus();
        listView1.Items[listView1.Items.Count-1].Selected = true;
    }

    private void button4_Click(object sender, EventArgs e)
    {
        if (listView1.SelectedItems.Count > 0)
        {
            listView1.Focus();
            int i = listView1.SelectedItems[0].Index;
            i++;
            if (i < listView1.Items.Count)
            {

```

```

        listView1.Items[i].Selected = true;
    }
}

private void button2_Click(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count > 0)
    {
        int i = listView1.SelectedItems[0].Index;
        i--;
        if (i >= 0)
        {
            listView1.Focus();
            listView1.Items[i].Selected = true;
        }
    }
}

public void lopdangchon()
{
    string lop = (listView1.SelectedItems[0].Index+1).ToString();
    string tongsolop = listView1.Items.Count.ToString();
    lblhienthi.Text = lop + "/" + tongsolop;
}

private void btnThem_Click_1(object sender, EventArgs e)
{
    txtGVCN.Clear();
    txtMaLop.Clear();
    txtTenLop.Clear();
}

private void frmQLLopHoc_FormClosing(object sender, FormClosingEventArgs e)
{
    this.DialogResult = DialogResult.Yes;
}
}

using System.IO;

public partial class frmQLSV : Form
{
    public frmQLSV()
    {
        InitializeComponent();
    }
    public DSLopHoc DSLH = new DSLopHoc();
    DocGhiFile file = new DocGhiFile();
    private void btnLopHoc_Click(object sender, EventArgs e)
    {
        frmQLLopHoc frm = new frmQLLopHoc(DSLH);
        if (frm.ShowDialog() == DialogResult.Yes)
            XuatDSLop();
    }

    private void frmQLSV_Load(object sender, EventArgs e)
    {
        if (File.Exists(Application.StartupPath + "\\data.dat"))
        {
            DSLH = (DSLopHoc)file.DocFile(Application.StartupPath + "\\data.dat");
            XuatDSLop();
        }
    }
}

```

```

}

public void XuatDSLop(){
    treLopHoc.Nodes.Clear();
    foreach (LopHoc lh in DSLH.LayDSLH())
    {
        TreeNode node = new TreeNode(lh.Tenlop);
        node.Tag = lh;
        treLopHoc.Nodes.Add(node);
    }
}

private void treLopHoc_AfterSelect(object sender, TreeViewEventArgs e)
{
    XuatListView();
}

public void XuatListView()
{
    LopHoc lh = (LopHoc)treLopHoc.SelectedNode.Tag;
    txtLop.Text = lh.MaLop;
    lstvDSHS.Items.Clear();
    foreach (HocSinh hs in lh.LayDSHS())
    {
        ListViewItem item = new ListViewItem(hs.MaHS);
        item.SubItems.Add(hs.Hoten);
        item.SubItems.Add(hs.Diachi);
        item.SubItems.Add(hs.Email1);
        item.Tag = hs;
        lstvDSHS.Items.Add(item);
    }
}

private void lstvDSHS_SelectedIndexChanged(object sender, EventArgs e)
{
    if (lstvDSHS.SelectedItems.Count > 0)
    {
        HocSinh hs = (HocSinh)lstvDSHS.SelectedItems[0].Tag;
        txtMaHS.Text=hs.MaHS;
        txtHoTen.Text = hs.Hoten;
        txtEmail.Text = hs.Email1;
        txtDiaChi.Text = hs.Diachi;
        txtLop.Text = hs.Lop1;
    }
}

private void btnThem_Click(object sender, EventArgs e)
{
    TreeNode node12 = treLopHoc.SelectedNode;
    if (node12 == null)
        MessageBox.Show("Bạn chưa chọn lớp để thêm");
    else
    {
        try
        {
            LopHoc lh = (LopHoc)treLopHoc.SelectedNode.Tag;
            string mshs = txtMaHS.Text;
            string hoten = txtHoTen.Text;
            string email = txtEmail.Text;
            string diachi = txtDiaChi.Text;
            string lophoc = txtLop.Text;
            HocSinh hs = new HocSinh(mshs, hoten, diachi, email, lophoc);
            if (DSLH.kttrungma(hs))
                MessageBox.Show("Trùng ms học sinh");
            else

```

```

        lh.ThemHS(hs);
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message);
    }
    XuatListView();
}

private void frmQLSV_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult hoi;
    hoi = MessageBox.Show("Bạn chắc chắn muốn thoát?", "Xác
nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
    if (hoi == DialogResult.Yes)
        file.GhiFile(Application.StartupPath + "\\data.dat", DSLH);
    else
        e.Cancel = true;
}

private void btnSua_Click(object sender, EventArgs e)
{
    try
    {
        string mshs = txtMaHS.Text;
        string hoten = txtHoTen.Text;
        string email = txtEmail.Text;
        string diachi = txtDiaChi.Text;
        string lophoc = txtLop.Text;
        HocSinh hs = new HocSinh(mshs, hoten, diachi, email, lophoc);
        LopHoc lh = (LopHoc)treLopHoc.SelectedNode.Tag;
        lh.SuaHS(hs);
        XuatListView();
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message);
    }
}

private void btnXoa_Click(object sender, EventArgs e)
{
    if (lstvDSHS.SelectedItems.Count > 0)
    {
        try
        {
            HocSinh hs = (HocSinh)lstvDSHS.SelectedItems[0].Tag;
            LopHoc lh = (LopHoc)treLopHoc.SelectedNode.Tag;
            lh.XoaHS(hs);
            XuatListView();
        }
        catch (Exception err)
        {
            MessageBox.Show(err.Message);
        }
    }
}

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

```

```
private void btnThem_Click_1(object sender, EventArgs e)
{
    txtDiaChi.Clear();
    txtEmail.Clear();
    txtHoTen.Clear();
    txtMaHS.Clear();
}
}
```

frmQLLopHoc

LỚP HỌC

Thông tin lớp học:

Mã Lớp: LH002

Tên Lớp: Lập Trình Java NC

Giáo Viên CN: Võ Văn Hải

Thêm

Lưu Thêm

Sửa

Xóa

Thoát

< << 1/1 >> >|

Mã Lớp	Tên Lớp	Giáo Viên CN
LH001	Lập Trình CSharp	Thắng Nguyễn
LH002	Lập Trình Java NC	Võ Văn Hải

Form1

QUẢN LÝ LỚP HỌC

Lập Trình CSharp

Lập Trình Java NC

Lớp Học

Mã HS	Họ Tên	Địa Chỉ	Email
HS003	Trần Nguyễn Hưng	Đồng Nai	chuabiet@dahu.com

Thông tin chi tiết của Học Sinh

Mã HS: HS003

Họ tên HS: Trần Nguyễn Hưng

Địa Chỉ: Đồng Nai

Email: chuabiet@dahu.com

Lớp: LH001

Thêm

Lưu Thêm

Sửa

Xóa

Thoát

Xử Dụng ListBox thay TreeView :

```
public partial class frmLopHoc : Form
{
    public frmLopHoc()
    {
        InitializeComponent();
    }
    public DSLopHoc ds= new DSLopHoc();
    public frmLopHoc(DSLopHoc ds)
    {
        InitializeComponent();
        this.ds = ds;
    }

    private void btnThem_Click(object sender, EventArgs e)
    {
        txtGVCN.Clear();
        txtMaLop.Clear();
        txtTenLop.Clear();
    }

    public void capnhatListView()
    {
        listView1.Items.Clear();
        foreach (LopHoc lh in ds.LayDSLH())
        {
            ListViewItem item = new ListViewItem(lh.MaLop);
            item.SubItems.Add(lh.Tenlop);
            item.SubItems.Add(lh.Gvcn);
            item.Tag = lh;
            listView1.Items.Add(item);
        }
    }

    private void btnLuuThem_Click(object sender, EventArgs e)
    {
        try
        {
            string mslop = txtMaLop.Text;
            string tenlop = txtTenLop.Text;
            string gvcn = txtGVCN.Text;
            LopHoc lh = new LopHoc(mslop, tenlop, gvcn);
            ds.themLophoc(lh);
            capnhatListView();
            string tongsolop = listView1.Items.Count.ToString();
            lblhienthi.Text = "?/" + tongsolop;
        }
        catch (Exception err)
        {
            MessageBox.Show(err.Message);
        }
    }

    private void btnSua_Click(object sender, EventArgs e)
    {
        if (listView1.SelectedItems.Count > 0)
        {
            try
            {
                LopHoc lhd = (LopHoc)listView1.SelectedItems[0].Tag;
                string mslop = txtMaLop.Text;
                string tenlop = txtTenLop.Text;
                string gvcn = txtGVCN.Text;
                LopHoc temp = lhd;
```



```

        temp.Tenlop = tenlop;
        temp.Gvcn = gvcn;
        ds.SuaLopHoc(temp);
        capnhatListView();
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message);
    }
}

}

private void btnXoa_Click(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count > 0)
        foreach (ListViewItem item in listView1.SelectedItems)
        {
            LopHoc lh = (LopHoc)item.Tag;
            ds.Xoalophoc(lh);
        }
    capnhatListView();
}

private void btnThoat_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.OK;
}

private void frmLopHoc_FormClosing(object sender, FormClosingEventArgs e)
{
    this.DialogResult = DialogResult.OK;
}

private void btnTop_Click(object sender, EventArgs e)
{
    listView1.Focus();
    listView1.TopItem.Selected = true;
}

private void btnFow_Click(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count > 0)
    {
        int i = listView1.SelectedItems[0].Index;
        i--;
        if (i >= 0)
        {
            listView1.Focus();
            listView1.Items[i].Selected = true;
        }
    }
}

private void btnNext_Click(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count > 0)
    {
        listView1.Focus();
        int i = listView1.SelectedItems[0].Index;
        i++;
        if (i < listView1.Items.Count)
        {

```

```

        listView1.Items[i].Selected = true;
    }
}

private void btnEnd_Click(object sender, EventArgs e)
{
    listView1.Focus();
    listView1.Items[listView1.Items.Count - 1].Selected = true;
}

private void frmLopHoc_Load(object sender, EventArgs e)
{
    if (ds.LayDSLH().Count > 0)
    {
        capnhatListView();
        listView1.Items[0].Selected = true;
    }
}

public void lopdangchon()
{
    string lop = (listView1.SelectedItems[0].Index + 1).ToString();
    string tongsolop = listView1.Items.Count.ToString();
    lblhienthi.Text = lop + "/" + tongsolop;
}

private void listView1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count > 0)
    {
        LopHoc lhd = (LopHoc)listView1.SelectedItems[0].Tag;
        txtMaLop.Text = lhd.MaLop;
        txtTenLop.Text = lhd.Tenlop;
        txtGVCN.Text = lhd.Gvcn;
        lopdangchon();
    }
}
}

public partial class frmM9Bai2LstBox : Form
{
    public frmM9Bai2LstBox()
    {
        InitializeComponent();
    }
    private DSLopHoc DSLH = new DSLopHoc();
    DocGhiFile file = new DocGhiFile();

    public void XuatDSLop()
    {
        lstLopHoc.Items.Clear();
        foreach (LopHoc lh in DSLH.LayDSLH())
        {
            lstLopHoc.Items.Add(lh);
        }
    }

    private void btnLopHoc_Click(object sender, EventArgs e)
    {
        frmLopHoc frm = new frmLopHoc(DSLH);
        if (frm.ShowDialog() == DialogResult.OK)
        {
            this.DSLH = frm.ds;
            XuatDSLop();
        }
    }
}

```

```

}

private void btnThem_Click(object sender, EventArgs e)
{
    txtDiaChi.Clear();
    txtEmail.Clear();
    txtHoTen.Clear();
    txtMaHS.Clear();
}

public void XuatListView()
{
    LopHoc lh = (LopHoc)lstLopHoc.SelectedItem;
    txtLop.Text = lh.MaLop;
    lstvDSHS.Items.Clear();
    foreach (HocSinh hs in lh.LayDSHS())
    {
        ListViewItem item = new ListViewItem(hs.MaHS);
        item.SubItems.Add(hs.Hoten);
        item.SubItems.Add(hs.DiaChi);
        item.SubItems.Add(hs.Email);
        item.Tag = hs;
        lstvDSHS.Items.Add(item);
    }
}

private void btnLuuThem_Click(object sender, EventArgs e)
{
    if (lstLopHoc.SelectedIndex == -1)
        MessageBox.Show("Bạn chưa chọn lớp để thêm");
    else
    {
        try
        {
            LopHoc lh = (LopHoc)lstLopHoc.SelectedItem;
            string mshs = txtMaHS.Text;
            string hoten = txtHoTen.Text;
            string email = txtEmail.Text;
            string diachi = txtDiaChi.Text;
            string lophoc = txtLop.Text;
            HocSinh hs = new HocSinh(mshs, hoten, diachi, email, lophoc);
            if (DSLH.kttrungma(hs))
                MessageBox.Show("Trùng ms học sinh");
            else
                lh.ThemHS(hs);
        }
        catch (Exception err)
        {
            MessageBox.Show(err.Message);
        }
        XuatListView();
    }
}

private void lstLopHoc_SelectedIndexChanged(object sender, EventArgs e)
{
    XuatListView();
}

private void btnSua_Click(object sender, EventArgs e)
{
    if (lstvDSHS.SelectedItems.Count > 0)
    {
        try
    
```

```

    {
        string mshs = txtMaHS.Text;
        string hoten = txtHoTen.Text;
        string email = txtEmail.Text;
        string diachi = txtDiaChi.Text;
        string lophoc = txtLop.Text;
        HocSinh hs = new HocSinh(mshs, hoten, diachi, email, lophoc);
        LopHoc lh = (LopHoc)lstLopHoc.SelectedItem;
        lh.SuaHS(hs);
        XuatListView();
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message);
    }
}
else
    MessageBox.Show("Bạn chưa chọn SV để Sửa");
}

private void btnXoa_Click(object sender, EventArgs e)
{
    if (lstvDSHS.SelectedItems.Count > 0)
    {
        if (lstLopHoc.SelectedIndex != -1)
        {
            try
            {
                DialogResult hoi;
                hoi = MessageBox.Show("Bạn chắc chắn muốn xóa SV này?", "Xác Nhận",
                MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
                if (hoi == DialogResult.Yes)
                {
                    HocSinh hs = (HocSinh)lstvDSHS.SelectedItems[0].Tag;
                    LopHoc lh = (LopHoc)lstLopHoc.SelectedItem;
                    lh.XoaHS(hs);
                    XuatListView();
                }
            }
            catch (Exception err)
            {
                MessageBox.Show(err.Message);
            }
        }
    }
    else
        MessageBox.Show("Bạn chưa chọn SV để xóa");
}

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

private void frmM9Bai2LstBox_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult hoi;
    hoi = MessageBox.Show("Bạn chắc chắn muốn thoát?", "Xác nhận", MessageBoxButtons.YesNo,
    MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);
    if (hoi == DialogResult.Yes)
        file.GhiFile(Application.StartupPath + "\\data.dat", DSLH);
    else
        e.Cancel = true;
}

```

```

    }

    private void lstvDSHS_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (lstvDSHS.SelectedItems.Count > 0)
        {
            HocSinh hs = (HocSinh)lstvDSHS.SelectedItems[0].Tag;
            txtMaHS.Text = hs.Mahs;
            txtHoTen.Text = hs.Hoten;
            txtEmail.Text = hs.Email1;
            txtDiaChi.Text = hs.Diachhi;
            txtLop.Text = hs.Lop1;
        }
    }

    private void frmM9Bai2LstBox_Load(object sender, EventArgs e)
    {
        if (File.Exists(Application.StartupPath + "\\data.dat"))
        {
            DSLH = (DSLopHoc)file.DocFile(Application.StartupPath + "\\data.dat");
            XuatDSLop();
        }
    }
}

```

The screenshot shows a Windows application window titled "frmQLLH". The main content area has a light gray background with the title "QUẢN LÝ LỚP HỌC" in large, bold, orange letters. On the left side, there is a list box containing three items: "Lập Trình CSharp", "Lập Trình Java", and "Lập Trình c++". Below the list box is a button labeled "Lớp Học". In the center, there is a table with four columns: "Mã HS", "Họ Tên", "Địa Chỉ", and "Email". The table contains one row of data: "HS003", "Trần Nguyễn Hưng", "Biển Hòa", and "chuabiet@dahu.com". Below the table is a horizontal scrollbar. On the right side, there is a form titled "Thông tin chi tiết của Học Sinh" (Detailed information of the student). The form has five input fields: "Mã HS:" (containing "HS003"), "Họ tên HS:" (containing "Trần Nguyễn Hưng"), "Địa Chỉ:" (containing "Đồng Nai"), "Email:" (containing "chuabiet@dahu.com"), and "Lớp:" (containing "LH001"). To the right of these fields are five buttons: "Thêm" (Add), "Lưu Thêm" (Save Add), "Sửa" (Edit), "Xóa" (Delete), and "Thoát" (Exit).

MỤC LỤC

Module 1 : Console.....	1
Bài 1.....	1
Bài 2.....	1
Bài 3.....	1
Bài 4.....	2
Bài 5.....	2
Bài 6.....	3
Bài 7.....	4
Bài 8.....	5
Bài 9.....	7
Bài 10.....	8
Module 2 : GUI	9
Bài 1.....	9
Bài 2.....	9
Bài 3.....	11
Bài 4.....	12
Bài 5.....	14
Bài 6.....	16
Bài 7.....	19
Bài 8.....	20
Bài 9.....	22
Bài 10.....	23
Bài 11.....	24
Module 3 : GUI	26
Bài 1.....	26
Bài 2.....	28
Bài 3.....	30
Bài 4.....	32
Bài 5.....	34
Bài 6.....	39
Bài 7.....	41
Module 4 : GUI	45
Bài 1.....	45

Bài 2.....	45
Bài 3.....	48
Bài 4.....	49
Bài 5.....	49
Bài 6.....	52
Bài 7.....	54
Module 5 : GUI	59
Bài 1.....	59
Bài 2.....	61
Bài 3.....	63
Bài 4.....	65
Bài 5.....	69
Module 6 : GUI	74
Bài 1.....	74
Bài 2.....	75
Bài 3.....	80
Module 7 : OOP	84
Bài 1.....	84
Bài 2.....	87
Bài 3.....	90
Bài 4.....	94
Module 8 : Array and Collection	96
Bài 1.....	96
Bài 2.....	100
Bài 3.....	103
Module 9 : File And Stream	107
Bài 2.....	107