

1 Nội dung

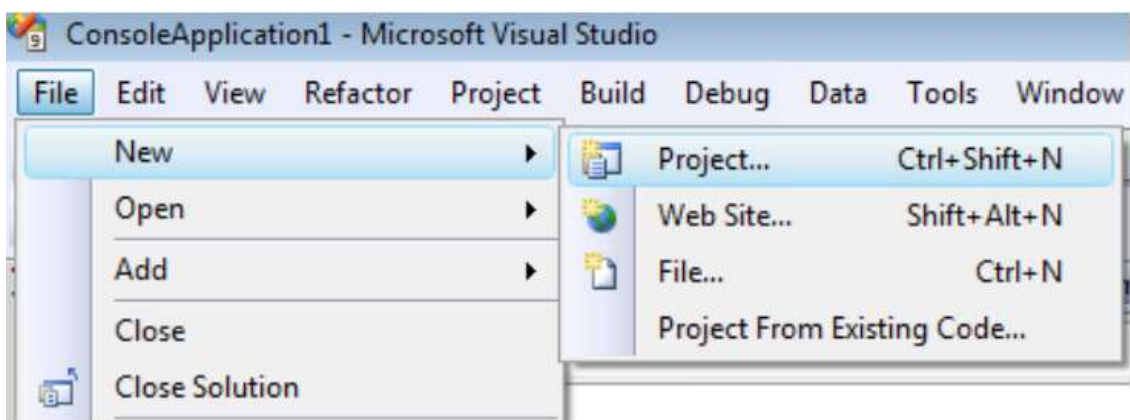
- Làm quen với các control Windows Form
- Lập trình Hướng đối tượng
- Làm việc với File
- Vẽ hình với GDI+
- Lập trình kết nối CSDL Sql Server

2 Làm quen với ứng dụng Windows

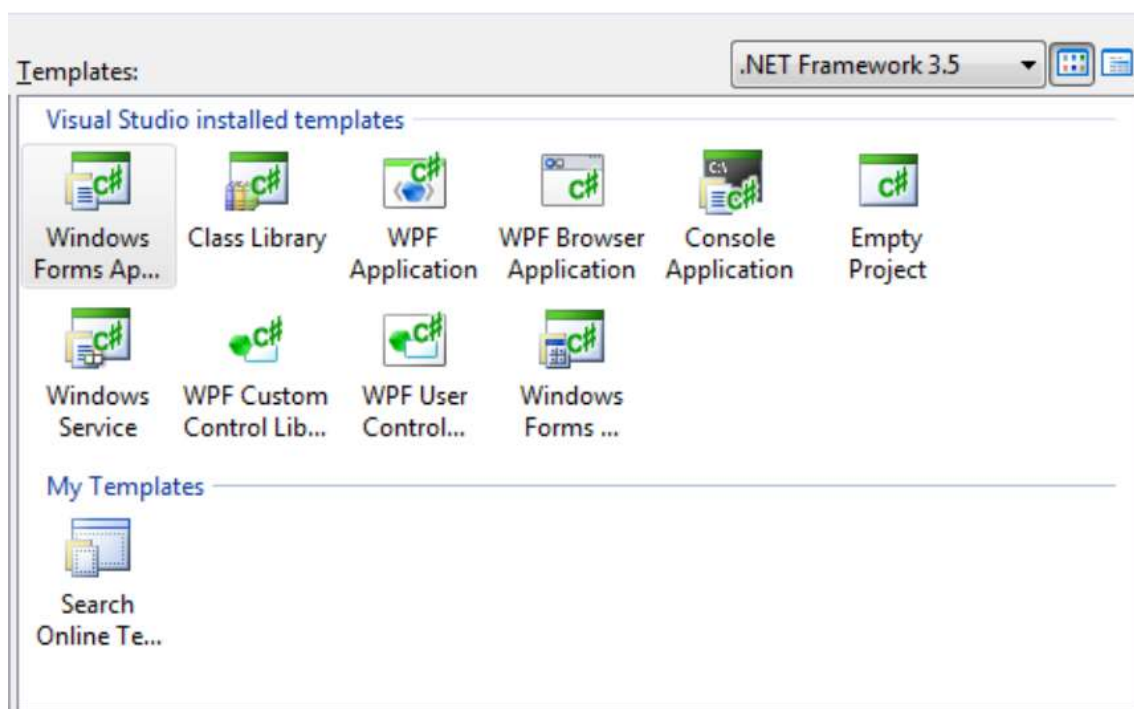
2.1 Bài tập mở đầu

2.1.1 Khởi tạo Project

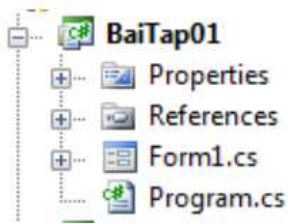
Từ Visual Studio 2008/2010 ta chọn **File → New → Project**.



Một hộp thoại xuất hiện, ta chọn **Template** là **Windows Form Application**.

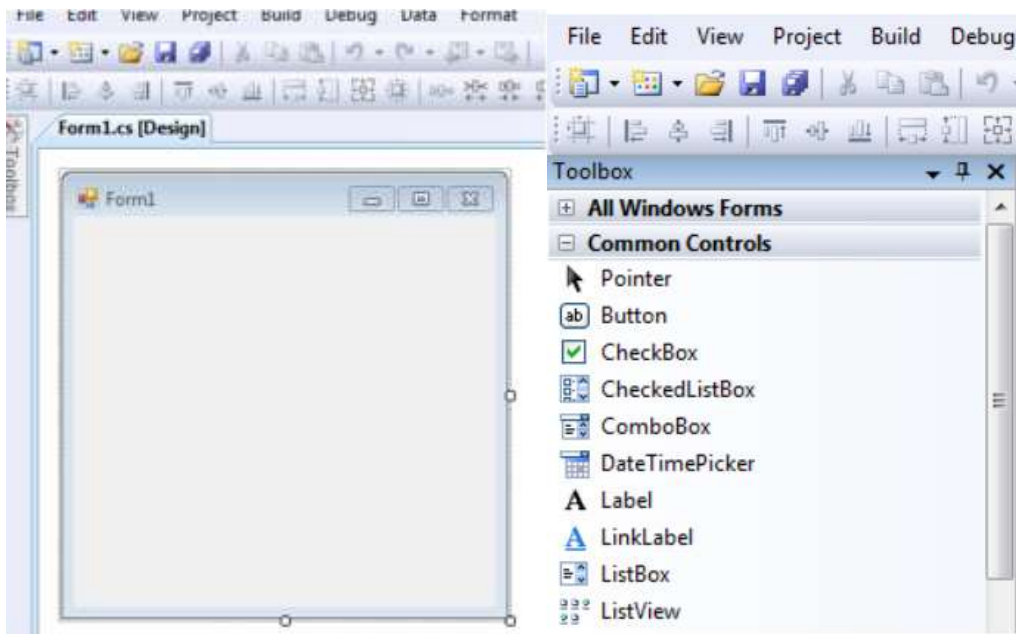


Ứng dụng được tạo ra như sau:



2.1.2 Ứng dụng mở đầu:

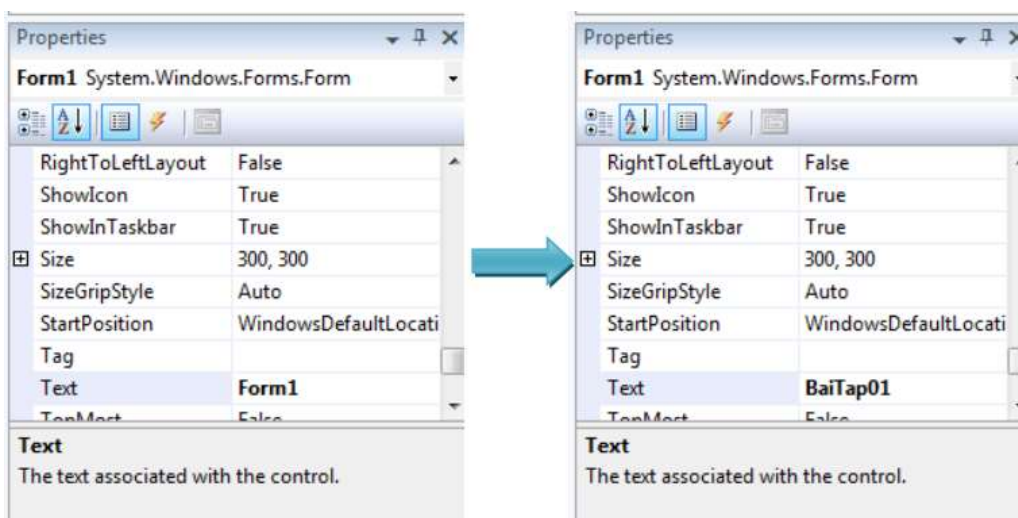
Project BaiTap01 gồm có thành phần Program.cs giống trong ứng dụng Console Application và còn có một thành phần mới là **Form1.cs** đang ở chế độ **Design** với giao diện như sau:



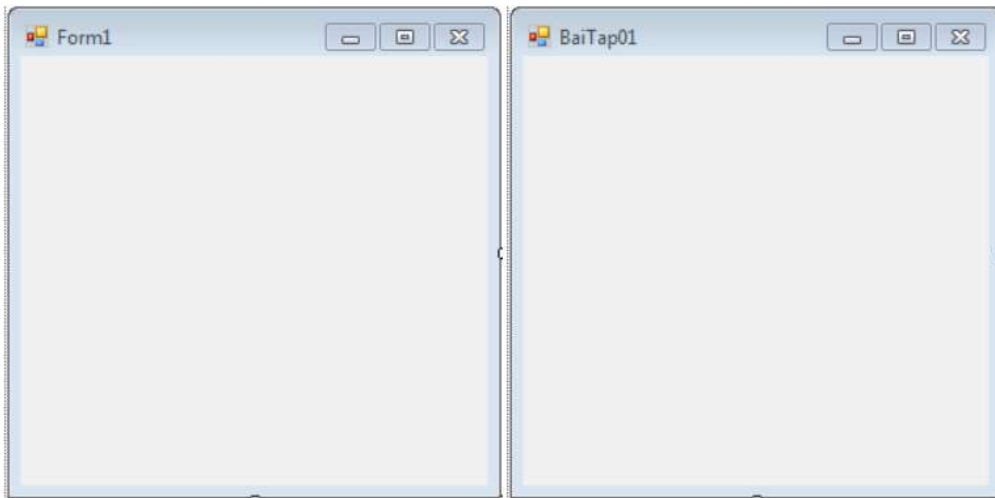
Nếu không thấy cửa sổ **Toolbox** này, ta chọn menu **View → Toolbox**.

2.1.3 Đổi tên form:

Click lên Form1 ở cửa sổ Design, quan sát trên cửa sổ Properties, ta thấy có thuộc tính Text, giá trị mặc định là Form1, ta đổi thành BaiTap01.

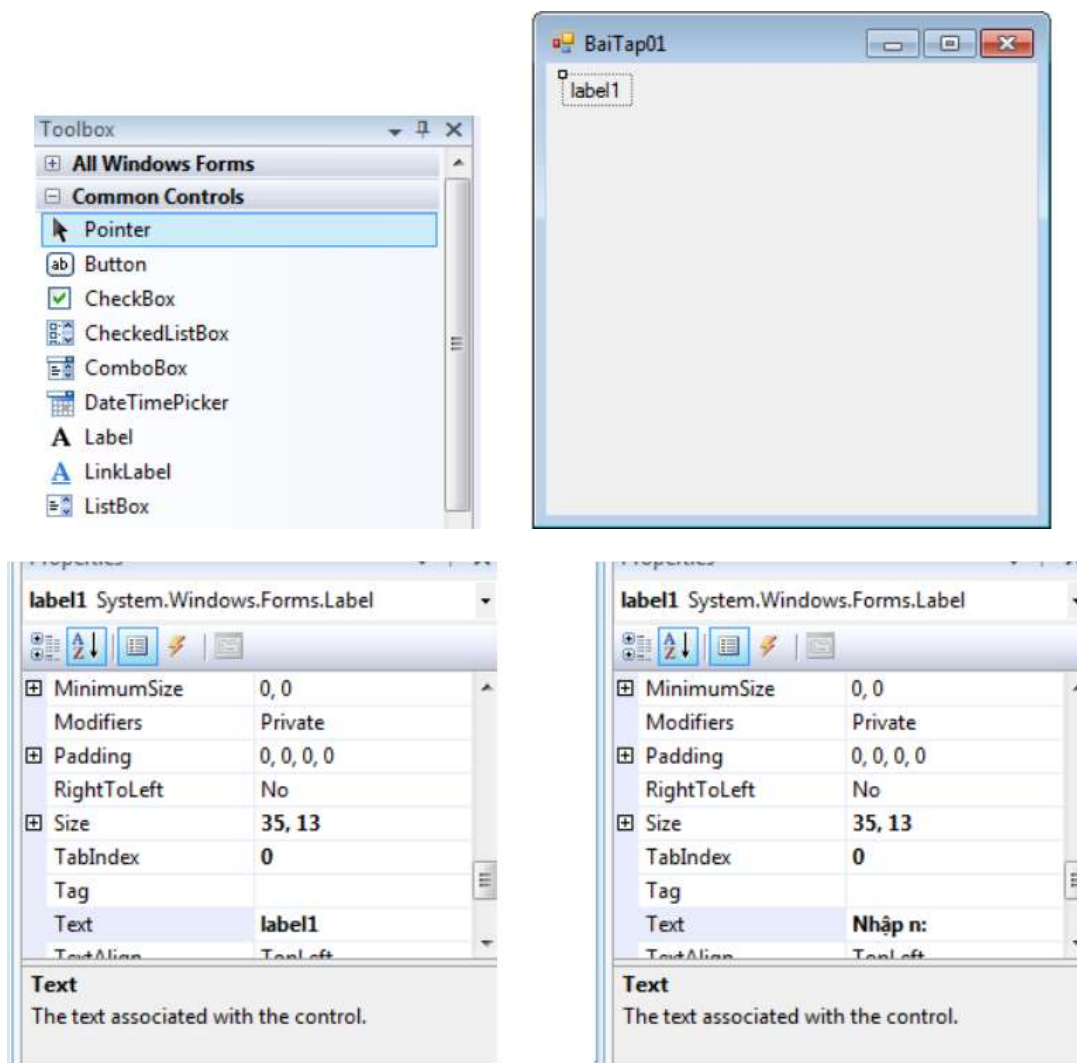


Ta thấy thanh tiêu đề Form1 đổi sang BaiTap01.

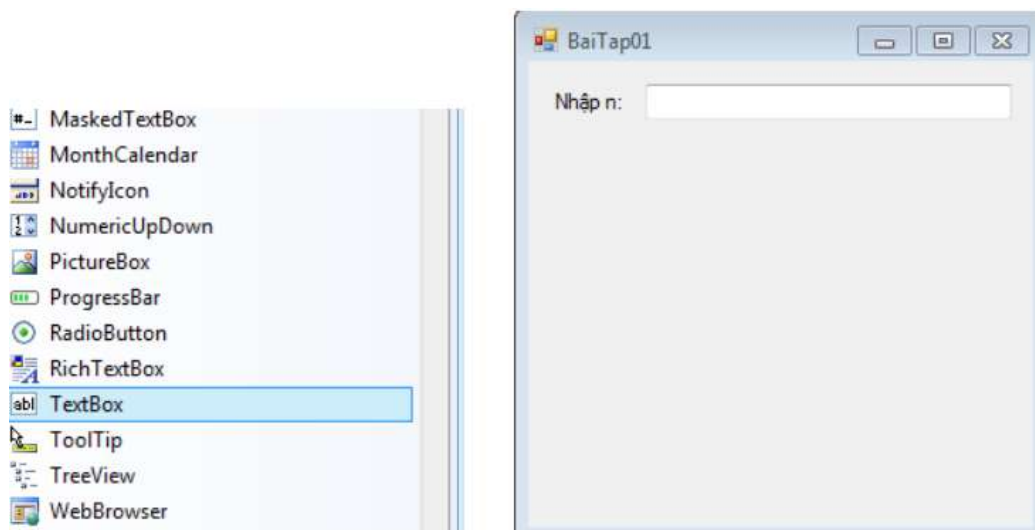


2.1.4 Kéo thả các control từ Toolbox vào Form

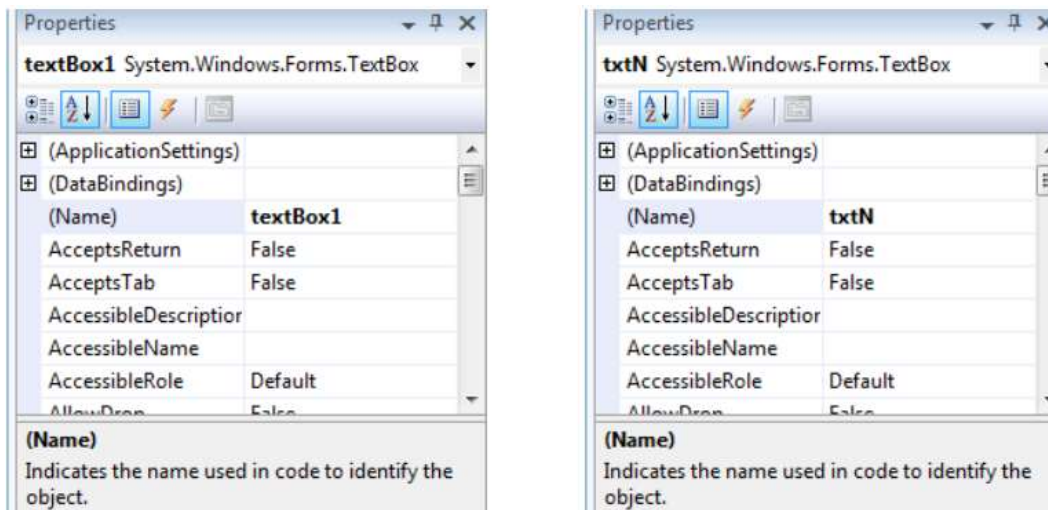
Từ thanh Toolbox, kéo thả một Label bỏ vào Form, sau đó di chuyển chuột đến cửa sổ Properties, chọn thuộc tính Text, đổi từ **label1** sang **Nhập n.**



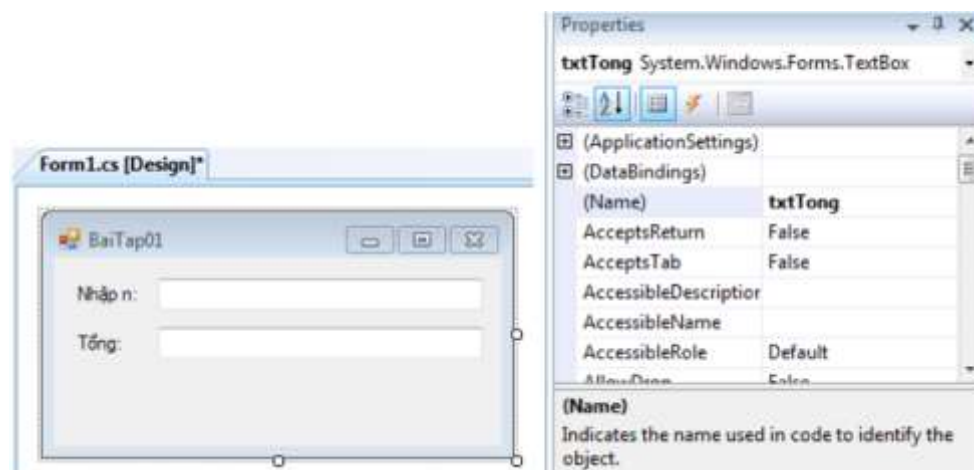
Tiếp theo kéo thả một Textbox bỏ vào form.



Trong cửa sổ Properties của textbox này, thuộc tính Name ta đổi từ **textBox1** thành **txtN**.

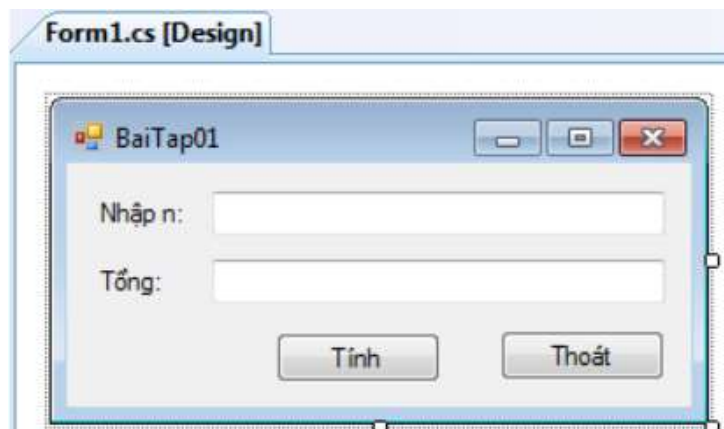


Tương tự thêm vào form một label có thuộc tính Text đổi thành **Tổng** và một textbox có thuộc tính Name đổi thành **txtTong**.



Tiếp theo, kéo một Button từ Toolbox và thả vào Form, đổi thuộc tính Name của button này là **btnTinh**, text là **Tính**.

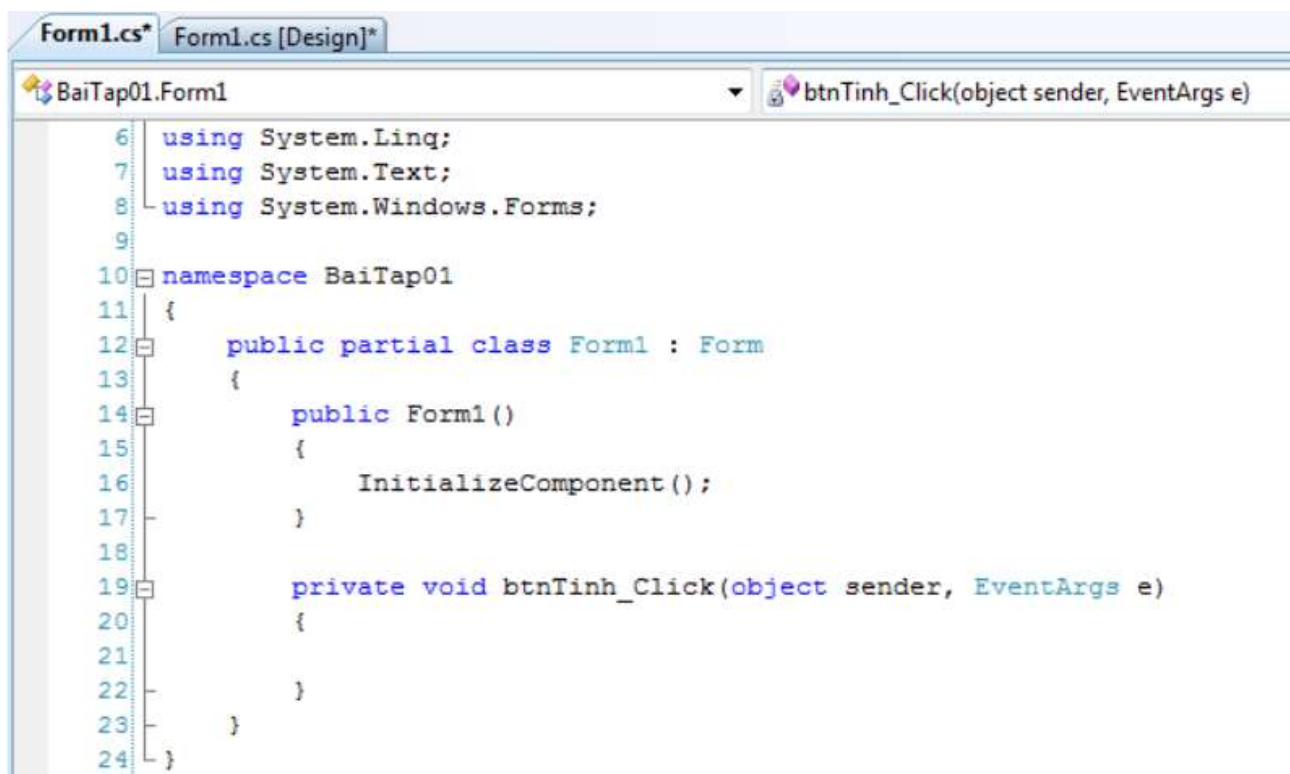
Kéo thả thêm một Button nữa tương tự trên, Name = **btnThoat**, Text = **Thoát**.



Đưa chuột về các góc của Form để thay đổi kích thước cửa sổ.

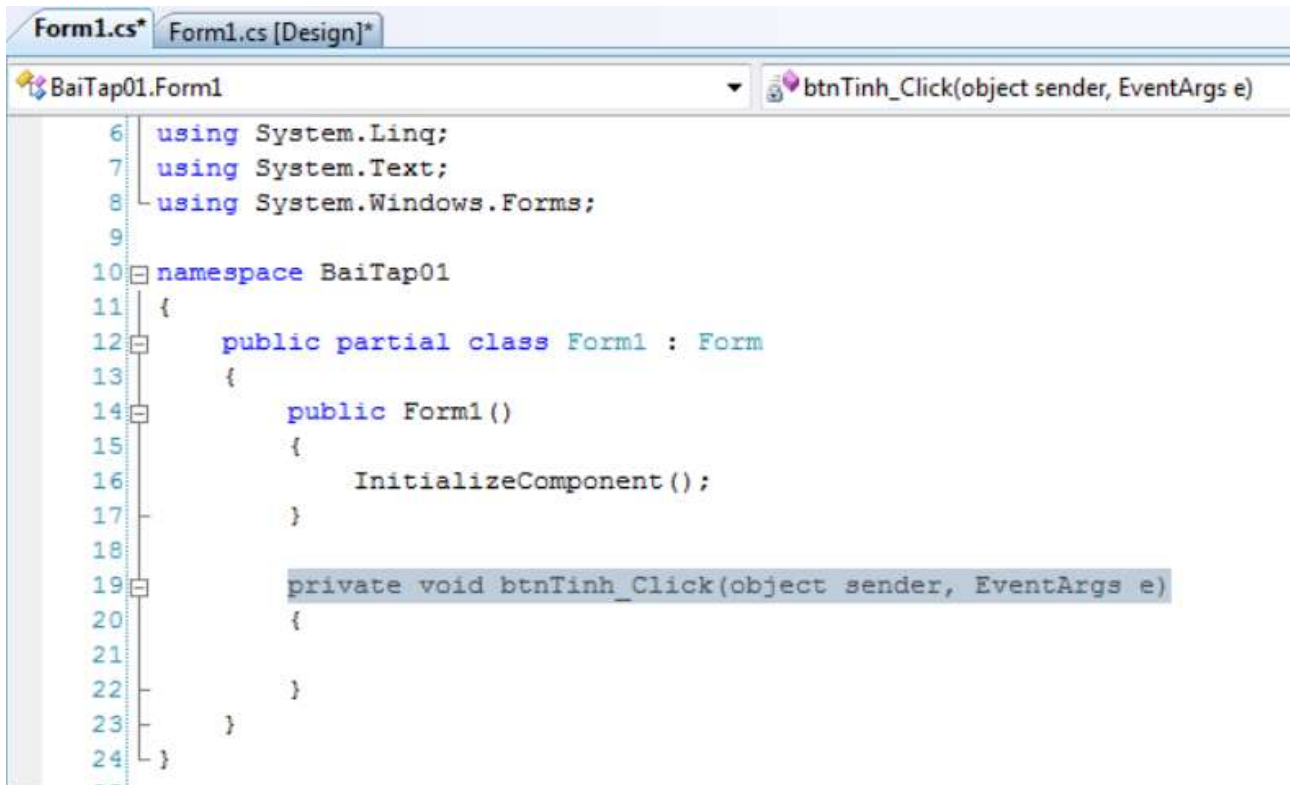
2.1.5 Xử lý sự kiện click chuột lên button

Double click vào button Tính (btnTinh), một cửa sổ soạn thảo xuất hiện với nội dung như sau:



Chú ý: Cửa sổ này là Form1.cs tương ứng với Form1.cs[Design].

Phương thức btnTinh_Click được phát sinh do ta vừa double click vào btnTinh.



Ta định nghĩa cho lớp này như sau:

```
namespace BaiTap01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnTinh_Click(object sender, EventArgs e)
        {
            string text = txtN.Text;
            int n = int.Parse(text);

            long s = 0;
            for (int i = 1; i <= n; i++)
            {
                s = s + i;
            }

            txtTong.Text = s.ToString();
        }
    }
}
```

Tiếp theo ta xử lý sự kiện click chuột cho button btnThoat.

Quay lại Form1.cs[Design], double click lên button Thoát, Form1.cs có nội dung thay đổi như sau:

```
namespace BaiTap01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnTinh_Click(object sender, EventArgs e)
        {
            string text = txtN.Text;
            int n = int.Parse(text);

            long s = 0;
            for (int i = 1; i <= n; i++)
            {
                s = s + i;
            }

            txtTong.Text = s.ToString();
        }

        private void btnThoat_Click(object sender, EventArgs e)
        {
        }
    }
}
```

Ta xử lý cho sự kiện này như sau:

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

Chạy thử chương trình (F5):

Nhập n = 4 và click vào button Tính. Kết quả xuất hiện ở txtTong:



2.2 Viết chương trình nhập vào 3 số a, b và c, cho biết số lớn nhất và nhỏ nhất trong 3 số a,b,c với giao diện như sau

Hướng dẫn gọi ý: sử dụng cấu trúc If ...Else

2.3 Nhập vào một số nguyên từ 0 đến 9, hiển thị bằng “chữ” các ký số trên, ví dụ:

- Nhập 1 → “Một”
- Nhập 2 → “Hai”
- Nhập 3 → “Ba”
- ..
- Nhập 9 → “Chín”

Hướng dẫn gọi ý: sử dụng cấu trúc switch ... case.

Mở rộng cho bài toán đọc số: 3 số, nhiều hơn 3 số...

2.4 Viết chương trình nhập vào giá trị nguyên dương N, tính tổng

$$S = 1 + 2 + 3 + \dots + N$$

với giao diện như sau

2.5 Viết chương trình nhập vào giá trị nguyên dương N, và số thực X, tính tổng:

$$S = X + X^2 + X^3 + X^4 + \dots + X^N$$

2.6 Thiết kế giao diện và cài đặt chương trình như sau:

3 Bài tập Hướng đối tượng – Quản lý Hình học

Tạo ứng dụng thực hiện các yêu cầu sau:

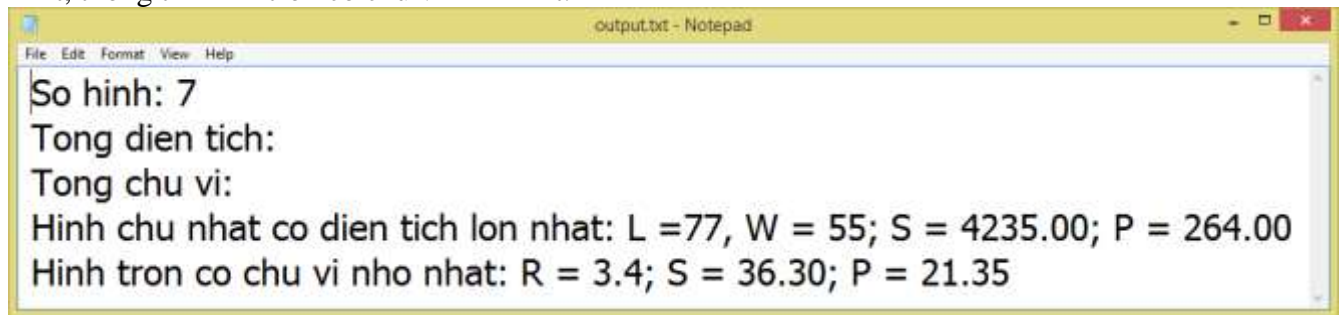
- Thêm vào 1 class **HinhHoc** gồm các thành phần biểu diễn diện tích (DienTich) và chu vi (ChuVi).
 - Khai báo và định nghĩa các properties để truy cập đến giá trị của các thành phần dữ liệu diện tích, chu vi.
 - Viết phương thức **tinhDienTichChuVi()** để tính diện tích, chu vi.
 - Override phương thức **ToString()** để xuất diện tích, chu vi hình.
- Thêm vào 1 class **HinhChuNhat** kế thừa từ lớp HinhHoc biểu diễn thông tin hình chữ nhật bao gồm thuộc tính riêng của nó là mChieuDai, mChieuRong.
 - Khai báo thành phần dữ liệu cần thiết để biểu diễn hình chữ nhật
 - Khai báo và định nghĩa các constructor cần thiết để khởi tạo giá trị cho các thành phần dữ liệu.
 - Khai báo và định nghĩa các properties để truy cập đến giá trị của các thành phần dữ liệu mChieuDai và mChieuRong (get, set) .
 - Override phương thức **tinhDienTichChuVi()** để tính diện tích hình chữ nhật ($mChieuDai \times mChieuRong$) và chu vi ($(mChieuDai + mChieuRong) \times 2$).
 - Override phương thức **ToString()** để xuất thông tin hình chữ nhật.
- Thêm 1 lớp tên **HinhTron** kế thừa từ lớp HinhHoc và viết thêm các thành phần sau :
 - Khai báo thêm thành phần dữ liệu bán kính: **mBanKinh**
 - Khai báo và định nghĩa các Constructor tham số và không tham số để khởi tạo các giá trị cho các thành phần dữ liệu.
 - Khai báo và định nghĩa các properties để truy cập đến giá trị của các thành phần dữ liệu (get, set).
 - Override phương thức **tinhDienTichChuVi()** để tính chu vi và diện tích hình tròn (cách viết giống như lớp hình chữ nhật).
 - Override phương thức **ToString()** để xuất thông tin hình tròn.
- Xây dựng Form để kiểm tra các constructor, các properties, các phương thức của các lớp trên.
- Mở rộng, tạo List<HinhHoc> để quản lý thông tin các hình học. Đọc thông tin từ file text theo định dạng cho trước để lấy thông tin các hình.

Cấu trúc file như sau:

```

#1: Hình chữ nhật, 2: Hình tròn; các thông số cách nhau bởi tab
1      5.3      3.2
2      9.1
1      11       2
1      12.9     10
2      4.5
2      3.4
1      77       55
  
```

Sau đó xuất ra file **output.txt** tổng diện tích, tổng chu vi, thông tin hình chữ nhật có diện tích lớn nhất, thông tin hình tròn có chu vi nhỏ nhất.



Gợi ý code:

```
//Lớp HìnhHoc
public class HìnhHoc
{
    #region AutomaticProperty
    public double DiệnTich { get; set; }
    public double ChuVi { get; set; }
    #endregion

    public virtual void tinhDiệnTichChuVi()
    {
    }
    public override string ToString()
    {
        return string.Format("S = {0}; P = {1}.", Math.Round(DiệnTich, 2), Math.Round(ChuVi, 2));
    }
}

//Lớp HìnhChuNhat
public class HìnhChuNhat: HìnhHoc
{
    //Fields
    private double mChieuDai;
    private double mChieuRong;

    //Property
    public double ChieuDai {
        get { return mChieuDai; }
        set
        {
            if (value > 0)
                mChieuDai = value;
            else
                throw new Exception("Chieu dai am");
        }
    }
    public double ChieuRong...
```

```

//constructor
public HìnhChuNhat(double d = 1, double r = 0.5)
{
    ChieuDai = d; ChieuRong = r;
}

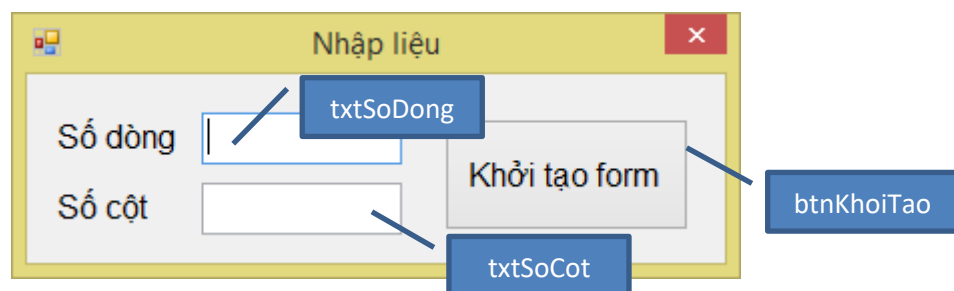
public override void tinhDienTichChuVi()
{
    DienTich = ChieuDai * ChieuRong;
    ChuVi = (ChieuDai + ChieuRong) * 2;
}

public override string ToString()
{
    return string.Format("L = {0}, W = {1}; {2}", ChieuDai, ChieuRong,
        base.ToString());
}
}
}

```

4 Truyền dữ liệu giữa các form

4.1 Tạo form nhập liệu (frmNhapLieu)



4.2 Tạo form xuất mảng (frmXuat)

4.3 Viết code cho form frmXuat

Để nhận giá trị số dòng, số cột cần 2 Property để giữ giá trị nó. Ngoài ra, có thể dùng các tham số (dạng Field) để lưu khoảng cách kích thước, mức canh lề, ...

```

public partial class frmXuat : Form
{
    //Automatic Property
    public int SoDong { get; set; }
    public int SoCot { get; set; }

    //Field
    int h = 35, margin = 5;

    public frmXuat(int dong = 1, int cot = 1)
    {
        InitializeComponent(); //khởi tạo Form & control

        //Gán số dòng, số cột
        this.SoDong = dong;
        this.SoCot = cot;
    }
}

```

```

//Thêm các control vào form
Button btn = null;
for(int i = 0; i < SoDong; i++)
{
    for(int j = 0; j < SoCot; j++)
    {
        btn = new Button();
        btn.Width = h; btn.Height = h;
        btn.Text = (i * SoCot + j + 1).ToString();
        btn.Left = margin * (j + 1) + h * j;
        btn.Top = margin * (i + 1) + h * i;
        //gắn sự kiện cho button
        btn.Click += btn_Click;
        btn.MouseHover += HoChuotLenButton;
        btn.Tag = string.Format("Dòng {0} cột {1}", i, j);
        this.Controls.Add(btn);
    }
}

//chỉnh lại kích thước form
this.ClientSize = new Size(SoCot * (h + margin + 1), SoDong * (h + margin +
1));
}

private void HoChuotLenButton(object sender, EventArgs e)
{
    this.Text = (sender as Button).Tag.ToString();
}

private void btn_Click(object sender, EventArgs e)
{
    MessageBox.Show((sender as Button).Text);
}

```

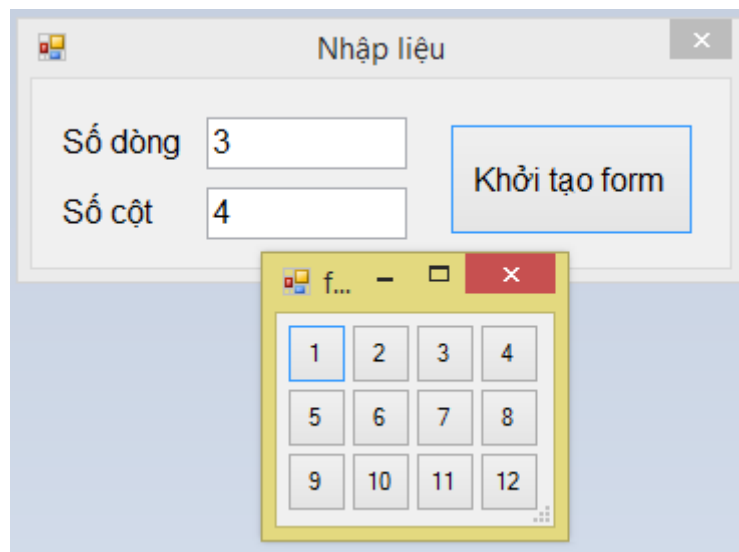
4.4 Xử lý nút khởi tạo ở form khởi tạo

```

private void btnKhoiTao_Click(object sender, EventArgs e)
{
    int sodong, socot;
    if (!int.TryParse(txtSoDong.Text, out sodong))
        sodong = 1;
    if (!int.TryParse(txtSoCot.Text, out socot))
        socot = 1;
    frmXuat frmX = new frmXuat(sodong, socot);
    //canh giữa
    frmX.StartPosition = FormStartPosition.CenterScreen;
    frmX.ShowDialog();//hiện
}

```

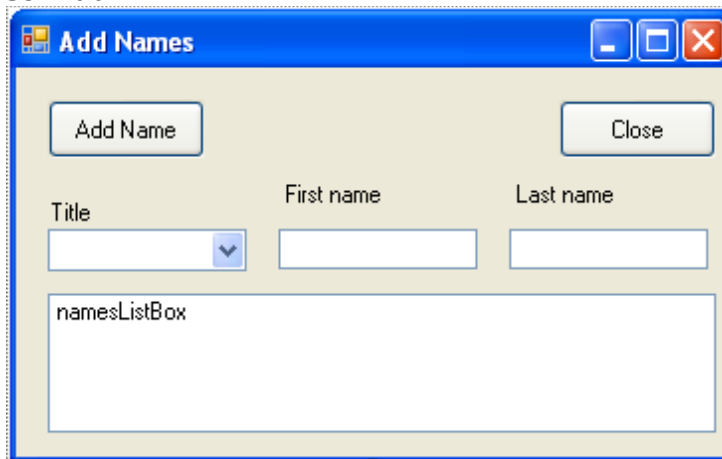
4.5 Demo



5 ComboBox, ListBox, NumericUpDown

5.1 Hiện thị Họ tên

5.1.1 Tạo Form theo mẫu:



5.1.2 Các thuộc tính:

5.1.2.1 Không cho phép resize form:

Property	New Value
FormBorderStyle	FixedSingle
Size	370, 225
Text	Add Names

5.1.2.2 Các Control:

Control Type	Property	New Value
Button	Name	closeButton
	Location	273, 16
	Text	Close
Button	Name	addButton
	Location	16, 16
	Text	Add
Label	Text	Title
Label	Text	First Name
	Location	132, 56

Label	Text	Last Name
	Location	248, 56
TextBox	Name	firstTextBox
	Location	132, 80
	Text	(blank)
TextBox	Name	lastTextBox
	Location	248, 80
	Text	(blank)
ComboBox	Name	titleComboBox
	Items	Mr.;Mrs.;Ms.;Miss.;Dr.
	Location	16, 80
	MaxDropDownItems	4
	Size	100,21
	Text	(blank)
ListBox	Name	namesListBox
	Location	16, 112
	Size	336, 69

5.1.3 Các sự kiện:

5.1.3.1 Sự kiện cho nút Close:

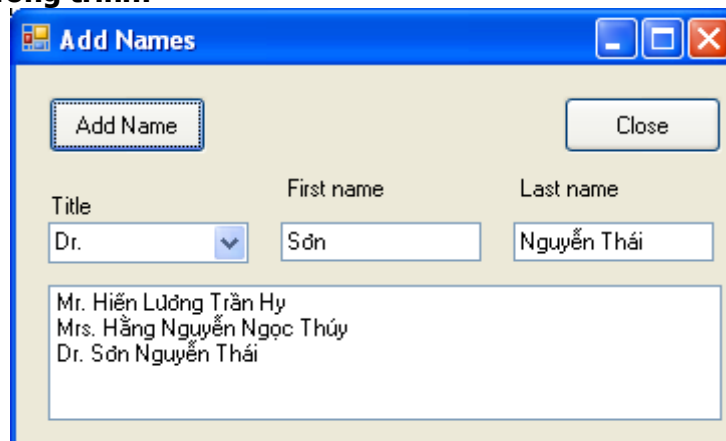
```
private void closeButton_Click(object sender, EventArgs e)
{
    this.Close(); //thêm code này
}
```

5.1.3.2 Sự kiện cho nút Add:

```
private void addButton_Click(object sender, EventArgs e)
{
    //add the following code
    string first = firstTextBox.Text;
    string title = titleComboBox.Text;
    string last = lastTextBox.Text;

    string nameToAdd = title + " " + first + " " + last;
    namesListBox.Items.Add(nameToAdd);
}
```

5.1.4 Kết quả chương trình:

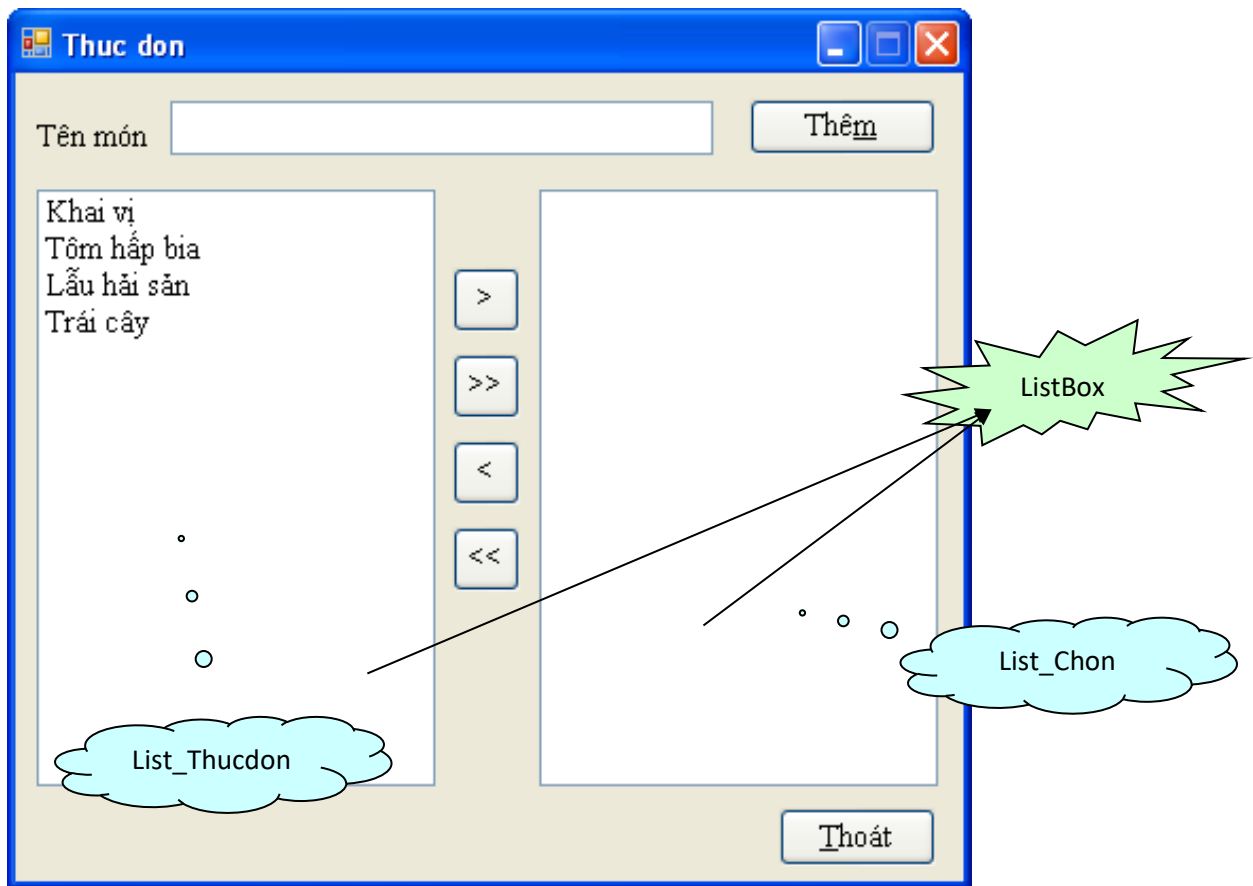


Ghi chú: Bài tập này sinh viên tự thực hiện

5.2 Danh mục món ăn

5.2.1 Tạo ứng dụng WinForm

Thiết kế giao diện như hình sau:



- Đặt lại thuộc tính và tên cho form.
- Đặt lại thuộc tính và tên cho các control trên form.

5.2.2 Đăng ký và xử lý sự kiện Button “Thêm”

- Thêm một item mới vào List_Thucdon:
`List_Thucdon.Items.Add(txt_Mon.Text);`
- Sau khi thêm xong thì clear txt_Mon
`txt_Mon.Text = "";`

5.2.3 Đăng ký và xử lý sự kiện cho Button

- Thêm một item được chọn trong List_Thucdon vào List_Chon
`List_Chon.Items.Add(List_Thucdon.SelectedItem);`
- Tuy nhiên, sẽ xảy ra trường hợp **một item được add nhiều lần**. Vì vậy, phải kiểm tra xem trong List_Chon đã tồn tại item đó hay chưa?

5.2.4 Tương tự, đăng ký và xử lý sự kiện cho Button

- Sử dụng vòng lặp để duyệt tất cả các item trong ListBox
- ???

5.2.5 Đăng ký và xử lý sự kiện cho Button

- Xóa một item ra khỏi ListBox
`List_Chon.Items.Remove(List_Chon.SelectedItem);`
- hoặc:
`List_Chon.Items.RemoveAt(List_Chon.SelectedIndex);`
- Xóa tất cả các item trong ListBox
`List_Chon.Items.Clear();`

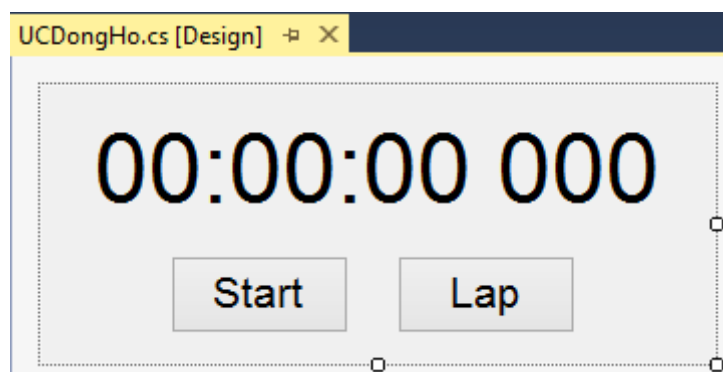
5.3 Dịch vụ Nha khoa

1. Khi chạy chương trình (F5) sẽ xuất hiện màn hình như trên, trong đó cấm thao tác trên tất cả các đối tượng trừ đối tượng textbox lưu trữ tên khách hàng và nút thoát.
2. Nếu người dùng nhập tên khách hàng thì mới cho phép thao tác trên các control khác (trừ textbox tổng cộng).
3. Nhấn vào nút **“Tính tiền”**, thực hiện tính tổng tiền theo lựa chọn của khách hàng và hiển thị thông báo dạng **“Khách hàng: XYZ cần trả số tiền ABC”** với XYZ là tên nhập vào, ABC là số tiền cần trả.
4. Nhấn vào nút **“Xóa”**: đưa màn hình trở lại trạng thái lúc khởi tạo.
5. Nhấn vào nút **“Thoát”**: Xuất hiện thông báo **“Bạn muốn có đóng chương trình không?”**
 - Nếu người dùng chọn **Yes**, đóng chương trình.
 - Nếu người dùng chọn **No**, trở lại chức năng trước đó.

6 User Control

6.1 UserControl Đồng hồ bấm giờ

6.1.1 Mô tả



- Bấm **Start** thì cho đồng hồ chạy, đồng thời đổi Text thành *Stop*.
- Bấm **Stop** thì dừng đồng hồ và đổi Text thành *Start*.
- Bấm **Lap** (ghi giờ) thực hiện ghi xuống file số lần cho trước.

6.1.2 Hướng dẫn code

Khai báo các biến cần thiết:

```
DateTime timestart;
int LapStep = 0;
const int LapCount = 5;
```

Xử lý cho nút Start/Stop:

```
private void btnStartSTop_Click(object sender, EventArgs e)
{
    if(timer1.Enabled)//true - đang chạy
    {
        timer1.Stop();
        btnStartSTop.Text = "Start";
    }
    else//đang đứng
    {
        timestart = DateTime.Now;
        timer1.Start();
        btnStartSTop.Text = "Stop";
    }
}
```

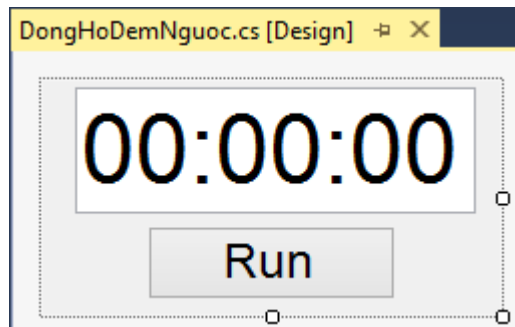
Xử lý cho sự kiện tick:

```
private void timer1_Tick(object sender, EventArgs e)
{
    TimeSpan timetick = DateTime.Now - timestart;
    lblDongHo.Text = string.Format("{0}:{1}:{2} {3}",
        timetick.Hours, timetick.Minutes, timetick.Seconds,
        timetick.Milliseconds);
}
```

Xử lý cho button **Lap**:

```
private void btnLap_Click(object sender, EventArgs e)
{
    lapstep++;
    StreamWriter sw = new StreamWriter("Data.txt", true);//ghi thêm
    sw.WriteLine(string.Format("Lap {0}: {1}", lapstep, lblDongHo.Text));
    if(lapstep == LapCount)
    {
        sw.WriteLine("FINISH");
        lapstep = 0;
        timer1.Stop();
        btnStartSTop.Text = "Start";
    }
    sw.Close();
}
```

6.2 UserControl đồng hồ đếm ngược



Thiết kế và coding đồng hồ đếm ngược cho phép người dùng nhập giá trị vào (nhớ kiểm tra hợp lệ), khi bấm “Run” thì tiến hành đếm ngược.

7 MyWordPad

7.1 Mục tiêu

- Phát triển ứng dụng Single Document Interface
- Làm quen với MenuStrip, ToolStrip, StatusStrip.
- Sử dụng các Common Dialog: OpenFileDialog, SaveFileDialog, ColorDialog, PrintDocument, ...
- Sử dụng control RichTextBox.

7.2 Nội dung

Xây dựng một ứng dụng soạn thảo văn bản có các chức năng sau:

Nhóm chức năng cơ bản:

- Cho phép soạn thảo trên hai định dạng
 - o Cơ bản: text File (*.txt)
 - o Nâng cao: Rich Text Format (*.rtf)
- Cho phép định dạng các đoạn text
 - o Align: left, right và center.
 - o Format: bold, italic, underline
 - o Chọn font chữ
 - o Chọn màu chữ
 - o Chọn màu nền của document
- Chức năng Cut, Copy và Paste.

Nhóm chức năng nâng cao:

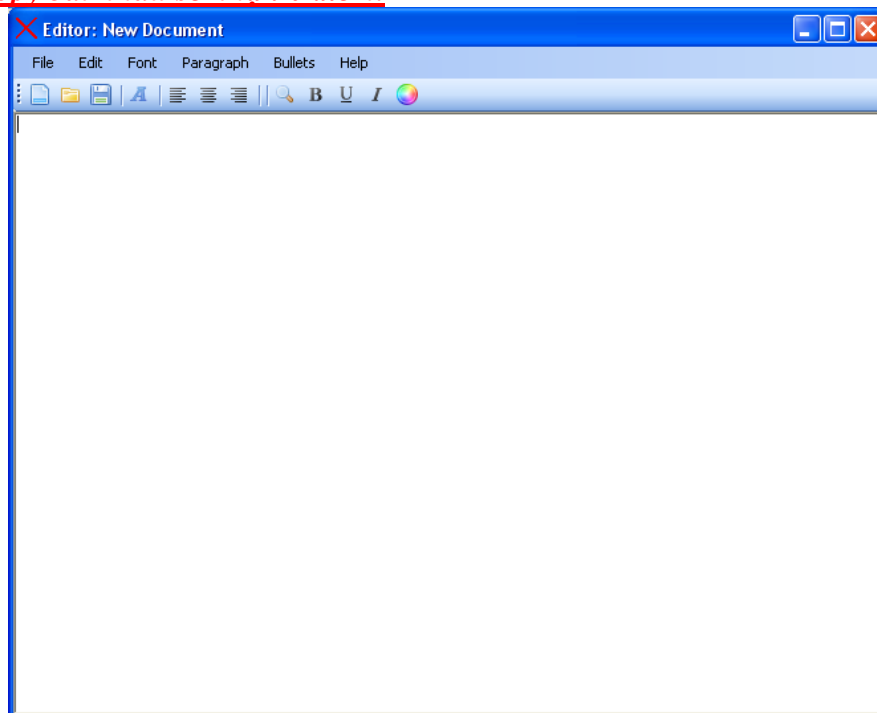
- Các chức năng như sau
 - o Open file: mở file RTF hoặc TXT
 - o Save file: lưu document
 - o New file: tạo mới document
 - o Chức năng chèn ảnh
 - o Chức năng chèn bullet
 - o Indent
 - o Zoom
 - o Tìm kiếm, thay thế
 - o In tài liệu ra máy in (chỉ phần nội dung RichTextBox)
 - o ...

Quy ước đặt tên Project: **STT_Nhom_WordPad**.

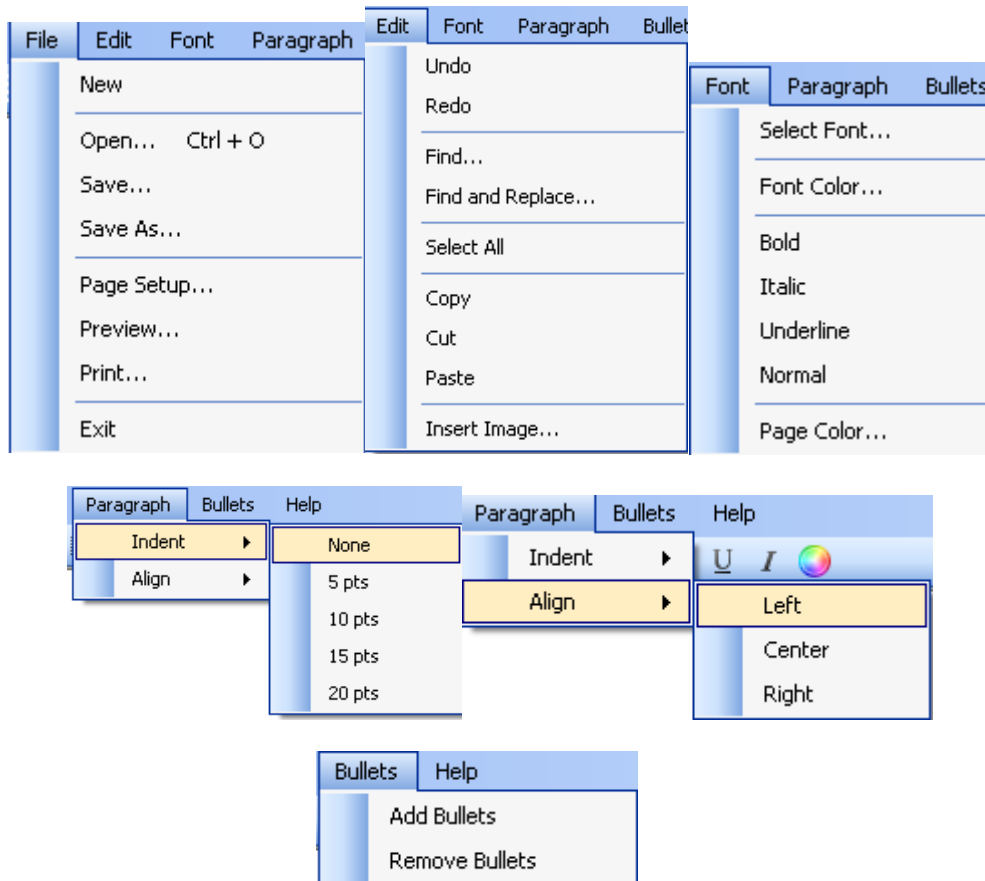
Nén project và upload lên host: <http://fit.hcmup.edu.vn/~hienlth/COMP1019/submit>

Hạn chót: **29/08/2015**.

Mọi hình thức copy bài nhau sẽ nhận 0 điểm.

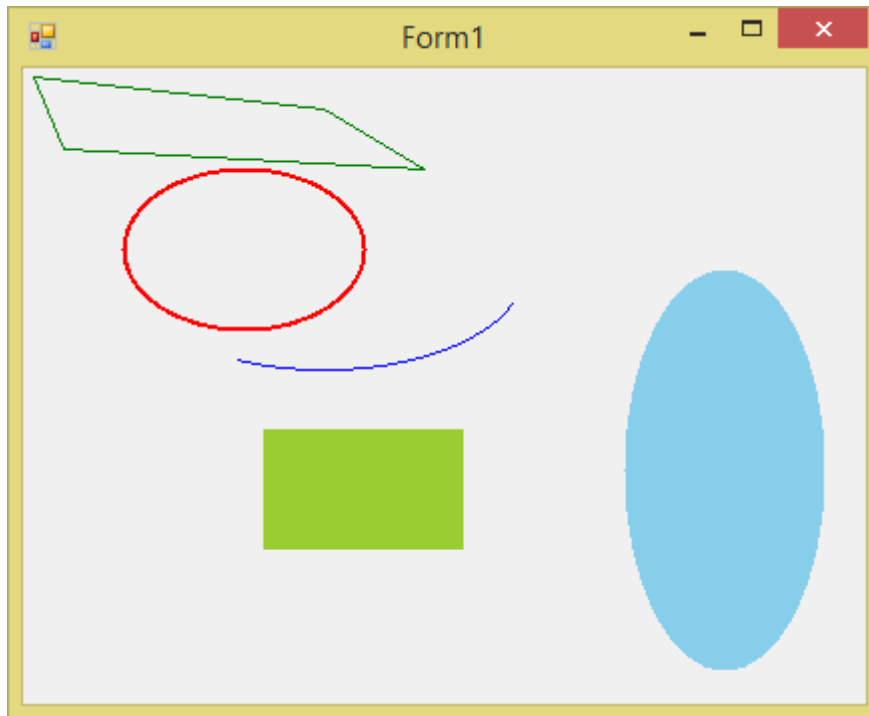


Chi tiết các Menu:



8 GDI+

8.1 Vẽ một số hình cơ bản lên Form



Để vẽ hình lên form có 2 cách:

8.1.1 Code cho sự kiện Paint của Form

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    //Lấy đối tượng Graphics của form
    Graphics g = e.Graphics;

    //Tạo bút vẽ
    Pen pen1 = new Pen(Color.Red, 2);

    //Vẽ ellipse
    g.DrawEllipse(pen1, 50, 50, 120, 80);

    //Vẽ cung
    Rectangle rect = new Rectangle(50, 50, 200, 100);
    g.DrawArc(new Pen(Color.Blue), rect, 10, 124);

    //Vẽ đa giác
    g.DrawPolygon(new Pen(Color.Green),
        new PointF[] {
            new PointF(20, 40), new PointF(5, 4),
            new PointF(150, 20), new PointF(200, 50)
        });

    //Tạo chổi tô
    SolidBrush brush1 = new SolidBrush(Color.YellowGreen);

    //Vẽ và tô hình chữ nhật
```

```
e.Graphics.FillRectangle(brush1, 120, 180, 100, 60);

//Vẽ và tô ellipse
e.Graphics.FillEllipse(new SolidBrush(Color.SkyBlue),
    300, 100, 100, 200);
// ...
}
```

8.1.2 Dùng hàm CreateGraphics() của form

```
private void btnDraw_Click(object sender, EventArgs e)
{
    // Lay doi tuong graphic cua form
    Graphics g = this.CreateGraphics();

    //Tạo bút vẽ
    Pen pen1 = new Pen(Color.Red, 2);

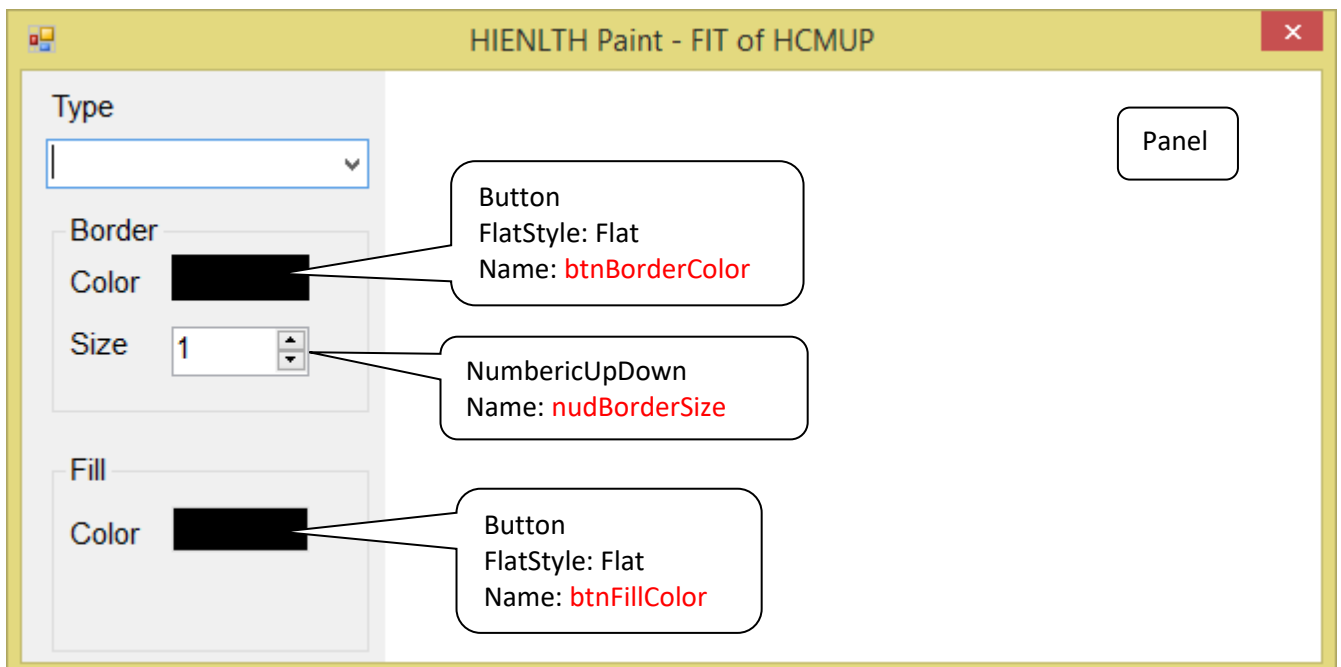
    //Vẽ ellipse
    g.DrawEllipse(pen1, 50, 50, 120, 80);

    //Vẽ cung
    Rectangle rect = new Rectangle(50, 50, 200, 100);
    g.DrawArc(new Pen(Color.Blue), rect, 10, 124);

    // ...
}
```

8.2 Bài tập Vẽ hình

8.2.1 Thiết kế Form theo mẫu:



Sự kiện vẽ: Ấn chuột trái để lấy điểm đầu tiên, nhả phím chuột để xác định điểm cuối cùng. Sau đó thực hiện vẽ và tô theo màu theo hình đã chọn trên khung vẽ.

8.2.2 Hướng dẫn:

//Các thuộc tính

```
private Color fill;           //lưu trữ màu tô
private Color border; //lưu trữ màu nét vẽ
private bool mouse_is_down = false;
private Point start;
private Graphics g;           //đối tượng đồ họa
```

//Sự kiện Form_Load

```
g = panel1.CreateGraphics();
border = Color.Black;
btnBorderColor.BackColor = border;
fill = Color.Red;
btnFillColor.BackColor = fill;
```

//Sự kiện Click chọn màu cho Border¹

```
ColorDialog dl = new ColorDialog();
dl.FullOpen = true;
dl.AnyColor = true;
if (dl.ShowDialog() == DialogResult.OK)
{
    btnBorderColor.BackColor = dl.Color;
    border = dl.Color;
}
```

//Sự kiện nhấn chuột vào vùng vẽ (Panel_MouseDown)

```
mouse_is_down = true;
start = new Point(e.X, e.Y);
```

//Sự kiện nhả chuột vào vùng vẽ (Panel_MouseUp)

```
mouse_is_down = false;
```

//Sự kiện rê (di chuyển) chuột trong vùng vẽ (Panel_MouseMove)

```
if (mouse_is_down)
{
    g.Clear(panel1.BackColor);
    Point end = new Point(e.X, e.Y);
    switch (cboType.SelectedIndex)
    {
        //line
        case 0:
            g.DrawLine(new Pen(border, (int)nudBorderSize.Value),
start, end);
            break;
        //empty ellipse.
        case 1:
```

¹ SV tự cài đặt cho sự kiện click vào nút chọn màu cho màu tô (fill)

```

        g.DrawEllipse(new Pen(border,
(int)nudBorderSize.Value), start.X, start.Y, e.X - start.X, e.Y - start.Y);
        break;
        //filled ellipse
        case 2:
        {
            g.DrawEllipse(new Pen(border,
(int)nudBorderSize.Value), start.X, start.Y, e.X - start.X, e.Y - start.Y);
            g.FillEllipse(new SolidBrush(fill), start.X,
start.Y, e.X - start.X, e.Y - start.Y);
            break;
        }
        //empty rectangle
        case 3:
            g.DrawRectangle(new Pen(border,
(int)nudBorderSize.Value), start.X, start.Y, e.X - start.X, e.Y - start.Y);
            break;
            //filled rectangle
            case 4:
                g.DrawRectangle(new Pen(border,
(int)nudBorderSize.Value), start.X, start.Y, e.X - start.X, e.Y - start.Y);
                g.FillRectangle(new SolidBrush(fill), start.X, start.Y,
e.X - start.X, e.Y - start.Y);
                break;
            default:
                break;
        }
    }

```

8.2.3 Mở rộng

- Thử xoay đối tượng vẽ?

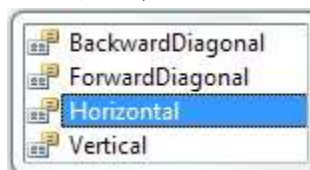
Sử dụng: **RotateTransform(góc) & ResetTransform();**

8.2.4 Bài tập nộp (Assignment 03 – My Paint)

8.2.4.1 Yêu cầu

Hoàn thành chương trình Paint thêm những yêu cầu sau:

- Tổ chức các loại hình học dạng Hướng đối tượng.
- Bổ sung thêm nhiều loại hình: Tam giác, Vuông, Bình hành, ...
- Cho phép vẽ nhiều (đối tượng) hình trên bàn vẽ.
- Cho phép chọn và đổi màu hình
- Cho phép vẽ chuỗi với Font, Size tùy chọn.
- Cho phép tô màu theo các dạng:
 - Solid,
 - LinearGradientMode (một trong các loại sau):



- PathGradientBrush
- TextureBrush

- HatchBrush

Brush Class	Description
SolidBrush	The simplest form of brush, which paints in a solid color.
HatchBrush	Similar to a SolidBrush , but it allows you to select from a large variety of preset patterns to paint with, rather than a solid color.
TextureBrush	Paints using a texture, such as an image.
LinearGradientBrush	Paints two colors blended along a gradient.
PathGradientBrush	Paints using a complex gradient of blended colors, based on a unique path defined by the developer.

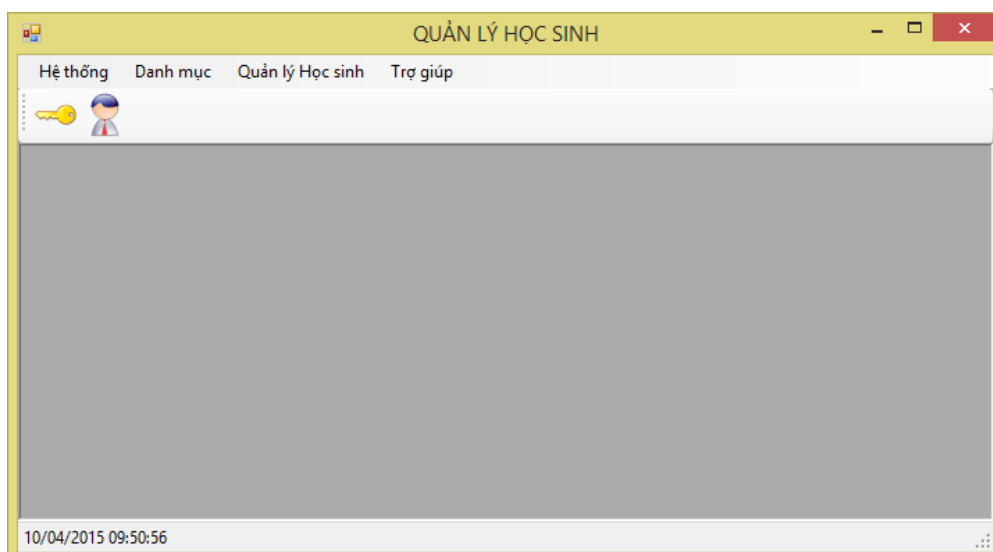
- Cho phép save hình từ khung vẽ xuống file bmp, png, jpeg
- Phải có About giới thiệu chương trình, tác giả.

8.2.4.2 Hình thức & Đánh giá

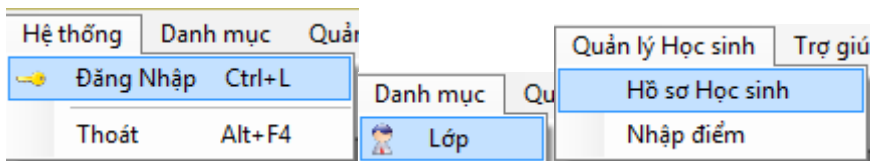
- Làm nhóm, đặt tên project dạng **STT_Nhom_Paint**. Sai tên – 01 điểm.
- Ngày nộp: Hạn chót 29/08/2015.
- Nộp tại: <http://fit.hcmup.edu.vn/~hienlth/COMP1019/submit/>
- Đánh giá:
 - Giao diện: 25%
 - Tổ chức dạng hướng đối tượng, có kế thừa: 25%
 - Vẽ hình dạng dùng Brush dạng: Hatch, Gradient, Texture : 20%
 - Vẽ chữ: 10%
 - Chọn hình: 10%
 - Save file ảnh: 10%

9 Quản lý Học sinh

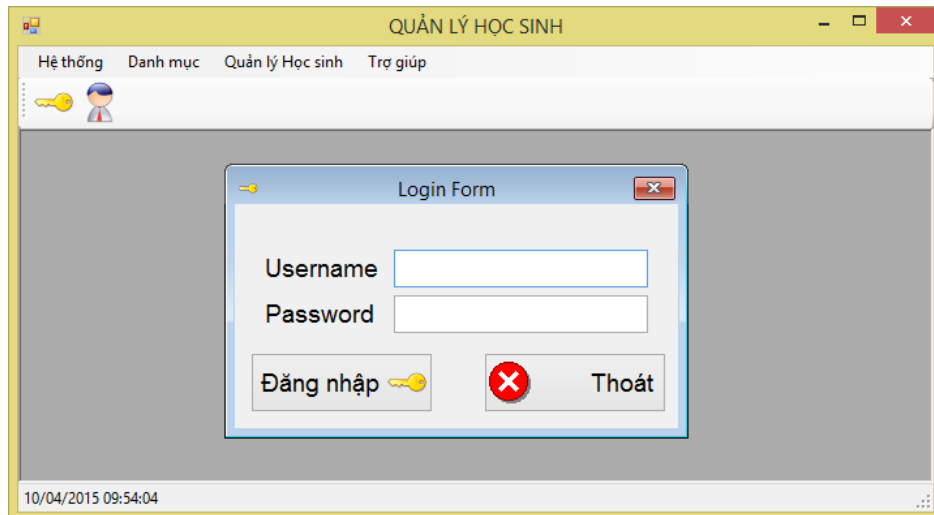
9.1 Thiết kế form chính:



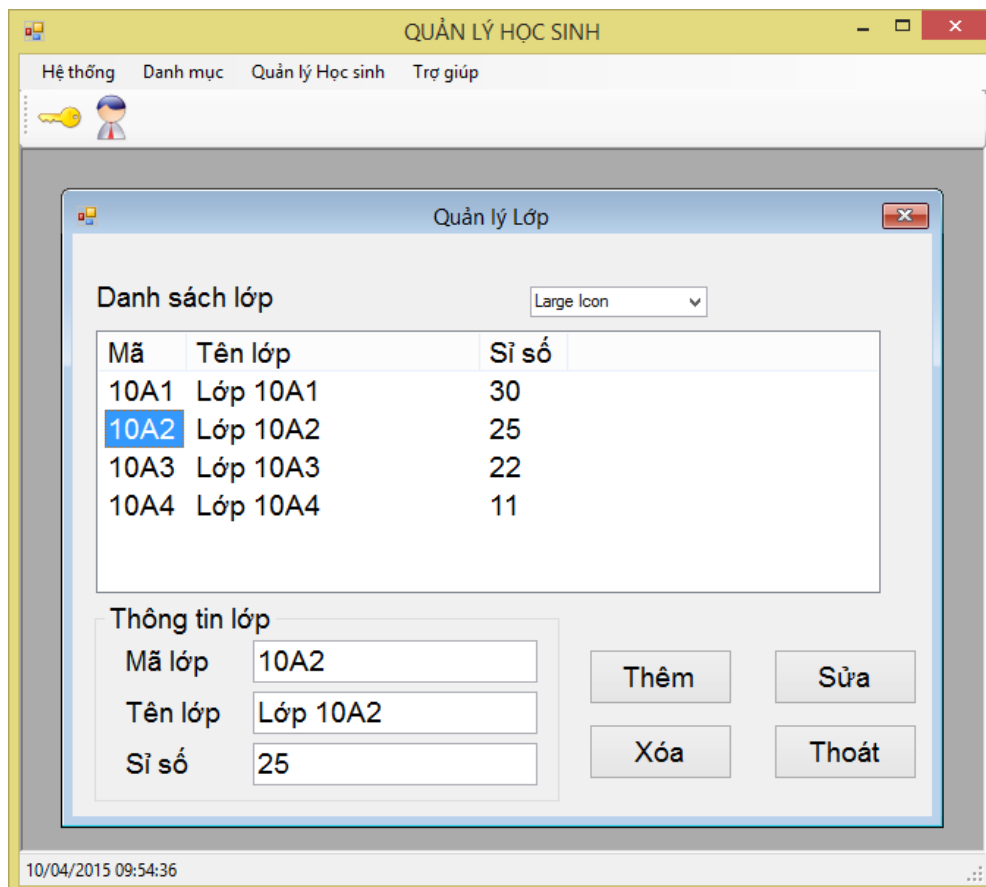
Một số menu của chương trình chính:



Màn hình đăng nhập:



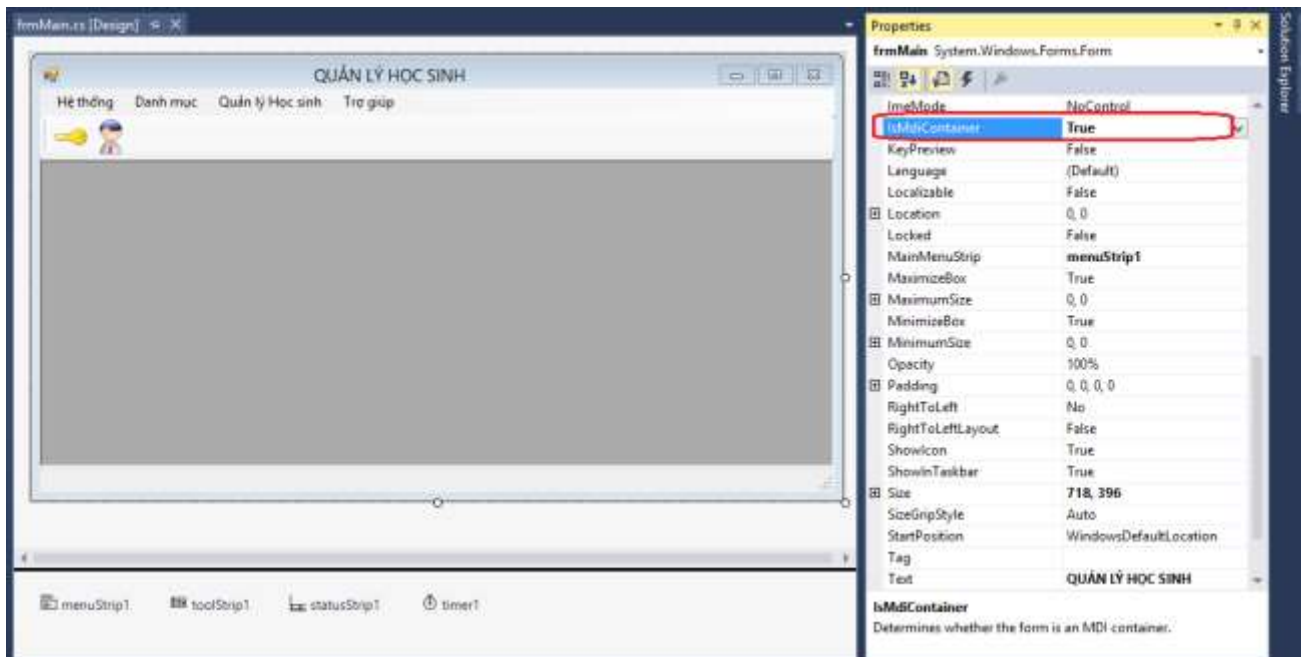
Màn hình Quản lý Lớp:



9.2 Hướng dẫn

9.2.1 Thiết kế Form chính

Thiết lập form chính: **IsMdiContainer = True**



9.2.2 Gọi form con

Gọi form đăng nhập có name=frmLogin:

```
private void mnuDangNhap_Click(object sender, EventArgs e)
{
    frmLogin f = new frmLogin();
    f.MdiParent = this; //đang code ở form cha
    f.Show();
}
```

Chú ý: Chỉ cho phép gọi method Show() ở form con.

9.3 Thực hiện sắp xếp các cửa sổ

Gọi hàm LayoutMdi để sắp xếp các form theo các kiểu **Cascade**, **TileVertical**, **TileHorizontal**, **Arrange Icon**.

```
this.LayoutMdi(MdiLayout.Cascade);
```

9.3.1 Thiết kế form Quản lý lớp

9.3.2 Tạo cơ sở dữ liệu

HIENLTH.DBHocSinh - LuocDo

HocSinh			
	Column Name	Data Type	Allow Nulls
🔑	MaHS	varchar(20)	<input type="checkbox"/>
	TenHS	nvarchar(100)	<input type="checkbox"/>
	NgaySinh	datetime	<input checked="" type="checkbox"/>
	DiaChi	nvarchar(150)	<input checked="" type="checkbox"/>
	DTB	real	<input checked="" type="checkbox"/>
	MaLop	varchar(20)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

🔑

Lop			
	Column Name	Data Type	Allow Nulls
🔑	MaLop	varchar(20)	<input type="checkbox"/>
	TenLop	nvarchar(100)	<input type="checkbox"/>
	SiSo	smallint	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

9.3.3 Viết code cho form Quản lý lớp

9.3.3.1 Khai báo chuỗi kết nối

```
private string chuoiketnoi = "Server=.; Database=DBHocSinh; Integrated Security=SSPI;";
```

9.3.3.2 Code hàm hiển thị danh sách lớp

```

void loadDanhSach()
{
    //xóa listview
    lvLop.Items.Clear();
    //Tạo kết nối tới CSDL
    SqlConnection con = new SqlConnection(chuoiiketnoi);
    con.Open();
    SqlCommand cmd = new SqlCommand("SELECT * FROM Lop", con);
    SqlDataReader dr = cmd.ExecuteReader();
    while(dr.Read())
    {
        string malop = dr.GetString(0);
        string tenlop = dr.GetString(1);
        int siso = dr.GetInt16(2);
        ListViewItem lvi = new ListViewItem();
        lvi.Tag = malop;
        lvi.Text = malop;
        lvi.SubItems.Add(tenlop);
        lvi.SubItems.Add(siso.ToString());
        lvLop.Items.Add(lvi);
    }
    //đóng kết nối
    con.Close();
}

```

9.3.3.3 Viết code cho sự kiện load form

```

private void frmLop_Load(object sender, EventArgs e)
{
    loadDanhSach();
    cboListType.SelectedIndex = 1; //hiển thị kiểu Large Icon
}

```

9.3.3.4 Code cho sự kiện thay đổi kiểu hiển thị của listview

```

private void cboListType_SelectedIndexChanged(object sender, EventArgs e)
{
    lvLop.View = (View)cboListType.SelectedIndex;
}

```

9.3.3.5 Xử lý sự kiện click chọn Item trên listview

```

private void lvLop_ItemSelectionChanged(object sender,
ListViewItemSelectionChangedEventArgs e)
{
    txtMaLop.Text = e.Item.Text;
    txtTenLop.Text = e.Item.SubItems[1].Text;
    txtSiSo.Text = e.Item.SubItems[2].Text;
}

```

9.3.3.6 Xử lý cho nút thêm

```

private void btnThem_Click(object sender, EventArgs e)
{
}

```

```

        string maLop = txtMaLop.Text;
        string tenLop = txtTenLop.Text;
        int siSo = int.Parse(txtSiSo.Text);

        SqlConnection con = new SqlConnection(chuoiketnoi);
        con.Open();

        string sql = "INSERT INTO Lop(MaLop, TenLop, SiSo) VALUES (@MaLop,
@TenLop, @SiSo)";
        SqlCommand cmd = new SqlCommand(sql, con);
        SqlParameter pa1 = new SqlParameter("@MaLop", maLop);
        cmd.Parameters.Add(pa1);
        SqlParameter pa2 = new SqlParameter("@TenLop", tenLop);
        cmd.Parameters.Add(pa2);
        SqlParameter pa3 = new SqlParameter("@SiSo", siSo);
        cmd.Parameters.Add(pa3);
        cmd.ExecuteNonQuery();
        //Đóng kết nối
        con.Close();

        //cập nhật danh sách
        loadDanhSach();
    }

```

9.3.3.7 Xử lý cho nút Xóa

```

private void btnXoa_Click(object sender, EventArgs e)
{
    //Lấy mã lớp
    string maLop = lvLop.SelectedItems[0].Tag.ToString();
    SqlConnection con = new SqlConnection(chuoiketnoi);
    con.Open();

    //Tạo lệnh sql xóa
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = con;
    cmd.CommandText = "delete from Lop where MaLop = @MaLop";
    SqlParameter pa = new SqlParameter("@MaLop", maLop);
    cmd.Parameters.Add(pa);
    cmd.ExecuteNonQuery();
    //Đóng kết nối
    con.Close();

    //cập nhật danh sách
    loadDanhSach();
}

```

9.3.3.8 Xử lý cho nút Sửa

```

private void btnSua_Click(object sender, EventArgs e)
{
    // Lay ma lop
    string maLop = lvLop.SelectedItems[0].Tag.ToString();
    string tenLop = txtTenLop.Text;
    int siSo = int.Parse(txtSiSo.Text);

```

```

SqlConnection con = new SqlConnection(chuoiketnoi);
con.Open();

//Tạo lệnh sql xóa
SqlCommand cmd = new SqlCommand();
cmd.Connection = con;
cmd.CommandText = "UPDATE Lop SET TenLop = @TenLop, SiSo = @SiSo
where MaLop = @MaLop";
SqlParameter pa1 = new SqlParameter("@MaLop", maLop);
cmd.Parameters.Add(pa1);
SqlParameter pa2 = new SqlParameter("@TenLop", tenLop);
cmd.Parameters.Add(pa2);
SqlParameter pa3 = new SqlParameter("@SiSo", siSo);
cmd.Parameters.Add(pa3);
cmd.ExecuteNonQuery();
//Đóng kết nối
con.Close();

//cập nhật danh sách
loadDanhSach();
}

```

9.4 Thiết kế Form Quản lý Học sinh

The application window is titled "QUẢN LÝ HỌC SINH". It contains a section titled "THÔNG TIN HỌC SINH" (Student Information) with the following fields:

- Mã học sinh (Student ID): HS0001
- Lớp (Class): Lóp 10A3 (dropdown menu)
- Tên học sinh (Student Name): Nguyễn Thanh Tùng
- Ngày sinh (Date of Birth): 11/02/1995 (dropdown menu)
- Địa chỉ (Address): 123 Ký Con, Q12
- Điểm trung bình (Average Score): 7.77

On the right side, there is a "Chức năng" (Function) section with four buttons: Thêm (Add), Sửa (Edit), Xóa (Delete), and Tìm (Find).

Below the form fields is a section titled "Danh sách Học sinh" (Student List) containing a table with the following data:

	MaHS	TenHS	DTB	MaLop
▶	HS0006	Nguyễn Thanh Bình		10A3

9.4.1 Các hàm truy vấn trong LINQ to SQL

STT	Tên hàm	Ý nghĩa
1	Where()	Chọn thỏa mãn điều kiện
2	SingleOrDefault()	Tìm trả về đối tượng hoặc null
3	Take(n)	Lấy n đối tượng đầu tiên
4	Skip(n)	Bỏ n đối tượng đầu tiên
5	Count()	Trả về số lượng
6	Sum()	Trả về tổng
7	Min()	Trả về số nhỏ nhất
8	Max()	Trả về số lớn nhất
9	Average()	Trả về giá trị trung bình
10	OrderBy()	Sắp xếp tăng dần theo điều kiện
11	OrderByDescending()	Sắp xếp giảm dần theo điều kiện

9.4.2 Khai báo & Cài đặt các thuộc tính

```

public partial class frmQLHocSinh : Form
{
    QLHSDDataContext db = new QLHSDDataContext();
    public frmQLHocSinh()
    {
        InitializeComponent();

        /// <summary>
        /// sự kiện load form
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void frmQLHocSinh_Load(object sender, EventArgs e)
        {
            //đổ dữ liệu vào combobox Lớp
            cboLop.DataSource = db.Lops;
            cboLop.DisplayMember = "TenLop";
            cboLop.ValueMember = "MaLop";

            //lấy học sinh theo lớp (đang chọn)
            fillHocSinhByLop();
        }

        /// <summary>
        /// Lấy danh sách học sinh theo lớp
        /// </summary>
        private void fillHocSinhByLop()
        {
            var dsHocSinh = db.HocSinhs.Where(p =>
                p.MaLop == cboLop.SelectedValue.ToString() &&
                p.TenHS.Contains(txtTenHS.Text.Trim()));
        }
    }
}

```


e)

```

        dgvHocSinh.DataSource = dsHocSinh.Select(p => new {
            p.MaHS, p.TenHS, p.Lop.TenLop, p.DTB });
    }

    /// <summary>
    /// Sự kiện thay đổi thứ tự chọn trong ComboBox lớp
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void cboLop_SelectedIndexChanged(object sender, EventArgs
    {
        fillHocSinhByLop();
    }

    /// <summary>
    /// Tìm học sinh theo lớp, gần đúng theo tên
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void btnTim_Click(object sender, EventArgs e)
    {
        fillHocSinhByLop();
    }

    /// <summary>
    /// Tạo mới đối tượng học sinh
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void btnThem_Click(object sender, EventArgs e)
    {
        //1. Tạo đối tượng
        HocSinh hs = new HocSinh
        {
            MaHS = txtMaHS.Text,
            TenHS = txtTenHS.Text,
            NgaySinh = dtpNgaySinh.Value,
            DiaChi = txtDiaChi.Text,
            MaLop = cboLop.SelectedValue.ToString(),
            DTB = float.Parse(txtDiemTB.Text)
        };
        //2. Thêm
        db.HocSinhs.InsertOnSubmit(hs);
        //3. Cập nhật
        db.SubmitChanges();

        fillHocSinhByLop();
    }

```

```

    /// <summary>
    /// Xóa học sinh theo mã
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void btnXoa_Click(object sender, EventArgs e)
    {
        //1. Tìm đối tượng cần xóa
        HocSinh hs = db.HocSinhs.SingleOrDefault(p => p.MaHS ==
txtMaHS.Text);
        //nếu có
        if(hs != null)
        {
            //2. xóa
            db.HocSinhs.DeleteOnSubmit(hs);
            //3. cập nhật
            db.SubmitChanges();

            fillHocSinhByLop();
        }
    }

    /// <summary>
    /// Sửa học sinh theo mã
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void btnSua_Click(object sender, EventArgs e)
    {
        //1. Tìm đối tượng cần sửa
        HocSinh hs = db.HocSinhs.SingleOrDefault(p => p.MaHS ==
txtMaHS.Text);
        if (hs != null) //nếu có
        {
            //2. Tiến hành sửa
            hs.MaLop = cboLop.SelectedValue.ToString();
            hs.TenHS = txtTenHS.Text;
            hs.NgaySinh = dtpNgaySinh.Value;

            //3. Cập nhật
            db.SubmitChanges();

            fillHocSinhByLop();
        }
    }
}

```

---Hết---