

BÁO CÁO THỰC HÀNH

Môn học: Lập trình mạng căn bản

Buổi báo cáo: Lab 01

Tên chủ đề: Lập trình C# & Windows Forms cơ bản

GVHD: Nguyễn Văn Bảo

Ngày thực hiện: 20/03/2024

THÔNG TIN CHUNG:

Lớp: NT106.022.2

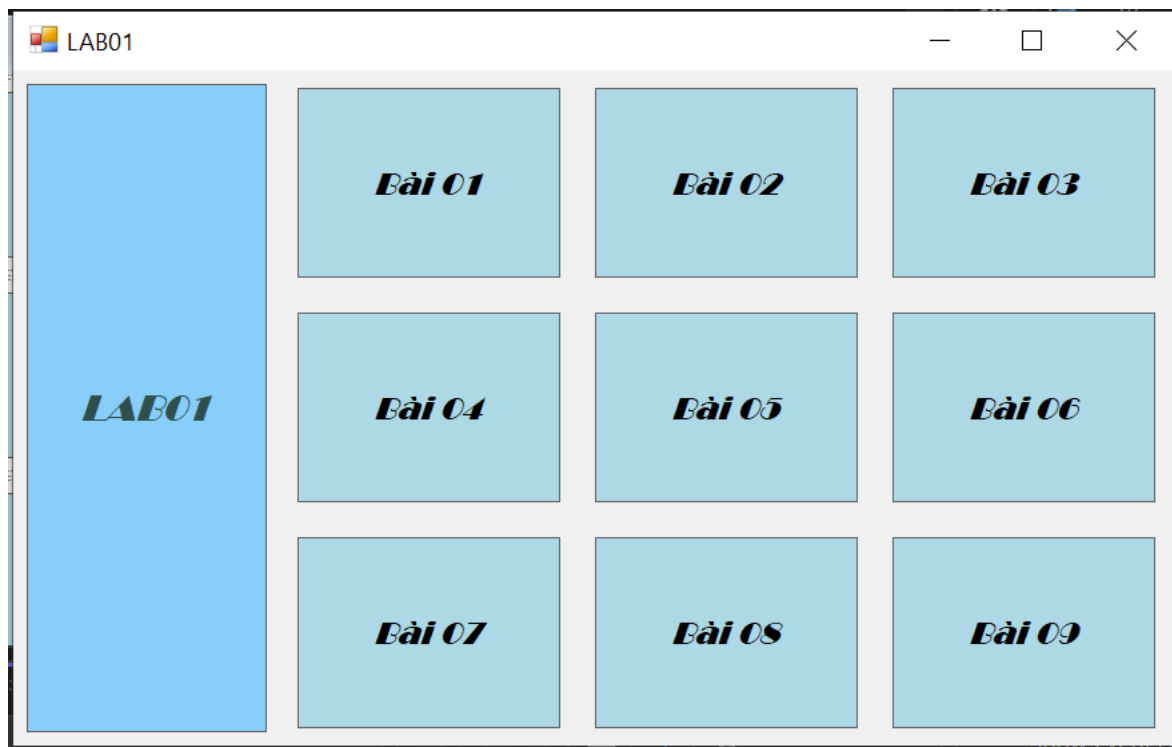
| STT | Họ và tên | MSSV | Email |
|-----|--------------------|----------|------------------------|
| 1 | Nguyễn Ngô Hải Nam | 22520919 | 22520919@gm.uit.edu.vn |

1. ĐÁNH GIÁ KHÁC:

| Nội dung | Kết quả |
|---|---------|
| Tổng thời gian thực hiện bài thực hành trung bình | 40 giờ |
| Link Video thực hiện (nếu có) | |
| Ý kiến (nếu có) + Khó khăn + Đề xuất ... | |
| Điểm tự đánh giá | 10 |

BÁO CÁO CHI TIẾT

- **Giao diện chính:** Tập hợp các nút bấm dẫn đến Form của từng bài tập.
- Các button Bài 01, Bài 02, ... được đặt trong một Table Layout có 3 hàng, 3 cột.
- Khi người dùng nhấn vào các button sẽ mở các Form tương ứng.



1) Bài 01 – Tính tổng hai số nguyên (có điều kiện dữ liệu vào)

GIAO DIỆN:

The screenshot shows a Windows Form titled "Bài 1". The form has a light gray background and contains the following elements:

- A label "Số thứ nhất:" followed by a text input box.
- A label "Số thứ hai:" followed by a text input box.
- A button labeled "Tính Tổng" (Calculate Sum).
- A label "Kết quả" (Result) followed by a text input box.

- Giao diện gồm các TextBox để người dùng nhập các số nguyên thứ nhất và thứ hai.
- TextBox kết quả được set thuộc tính ReadOnly để hiển thị kết quả.

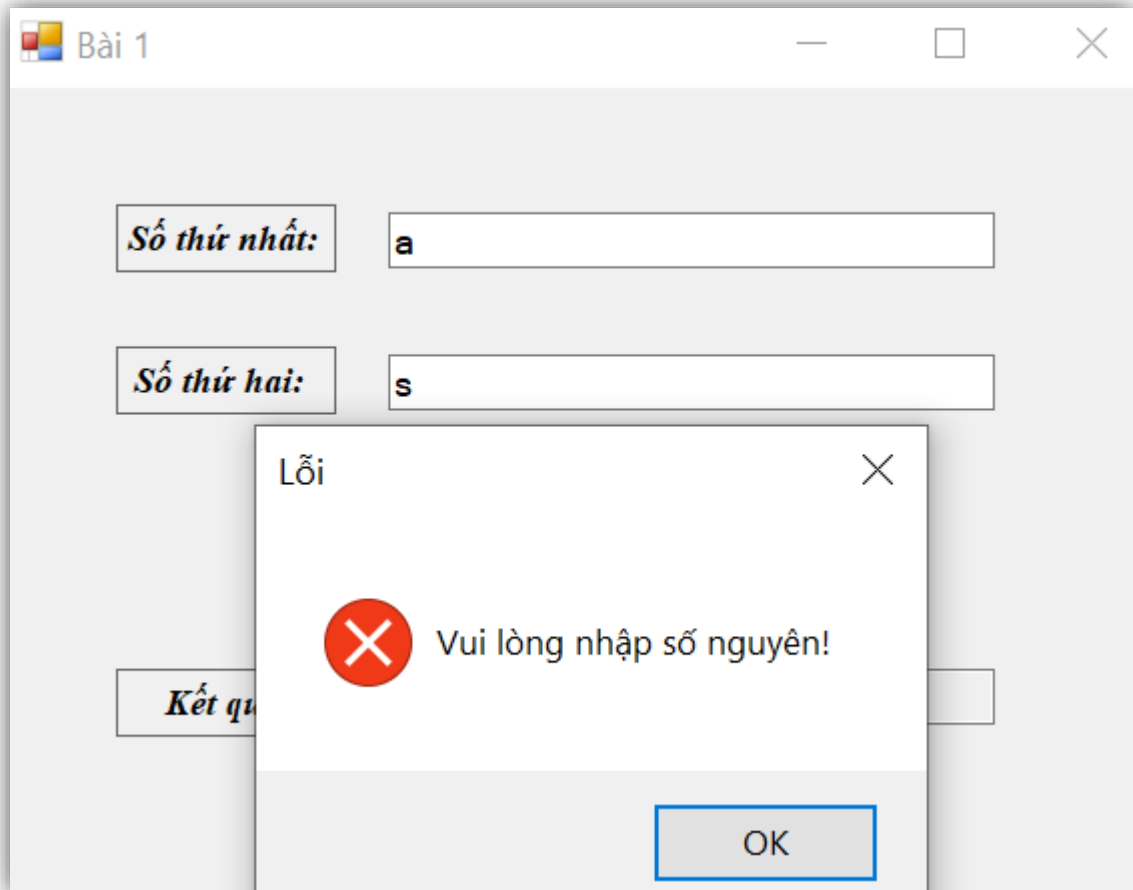
CODE:

```
1 reference
private void buttonTinhTong_Click(object sender, EventArgs e)
{
    int x = 0;
    int y = 0;

    if(!int.TryParse(textBoxSoThuNhat.Text, out x) || !int.TryParse(textBoxSoThuHai.Text, out y))
    {
        MessageBox.Show("Vui lòng nhập số nguyên!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        int tong = x + y;
        textBoxKetQua.Text = tong.ToString();
    }
}
```

- Cài đặt hàm buttonTinhTongClick(object sender,EventArgs e) để tính toán kết quả tính tổng.
- Nếu người dùng nhập sai dữ liệu đầu vào thì câu lệnh if sẽ được thực thi và in ra một MessageBox báo lỗi.

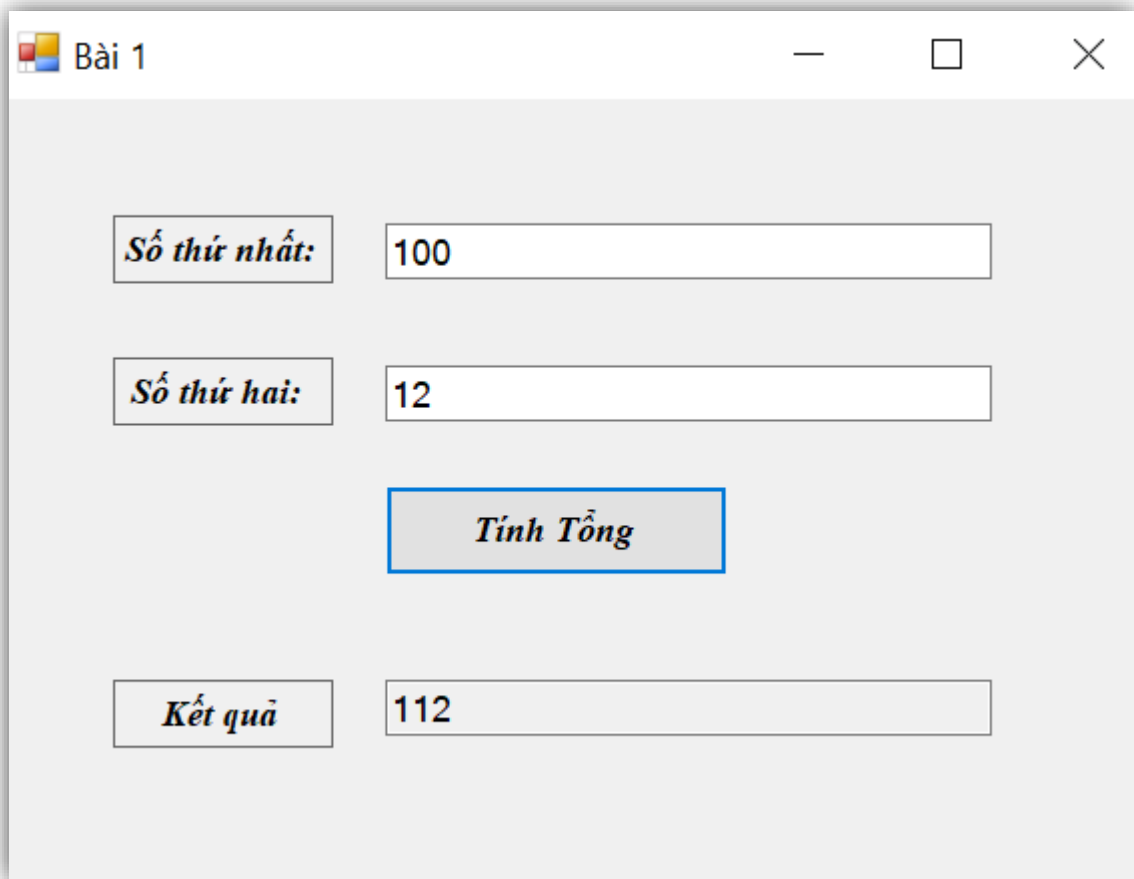
```
if(!int.TryParse(textBoxSoThuNhat.Text, out x) || !int.TryParse(textBoxSoThuHai.Text, out y))
{
    MessageBox.Show("Vui lòng nhập số nguyên!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```



Khi nhập dữ liệu sai

- Nếu người dùng nhập đúng dữ liệu đầu vào thì các câu lệnh trong khối else sẽ được thực hiện và in kết quả.

```
else
{
    int tong = x + y;
    textBoxKetQua.Text = tong.ToString();
}
```



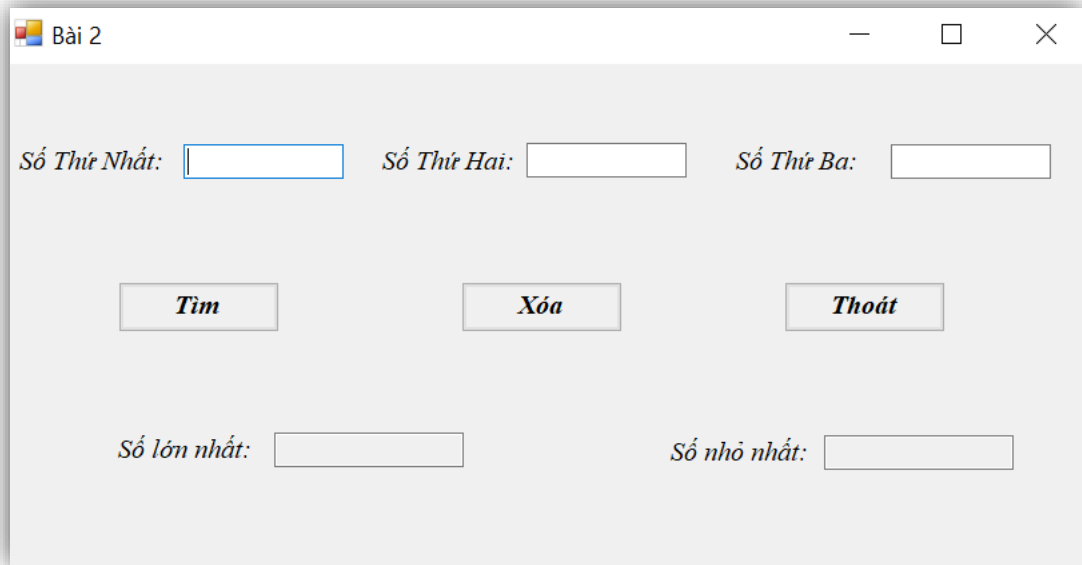
The screenshot shows a Windows Form titled "Bài 1". It contains the following elements:

- A label "Số thứ nhất:" next to a text box containing the value "100".
- A label "Số thứ hai:" next to a text box containing the value "12".
- A button labeled "Tính Tổng" (Calculate Sum) with a blue border.
- A label "Kết quả" (Result) next to a text box containing the value "112".

Khi nhập dữ liệu hợp lệ

2) Bài 02 – Số lớn nhất, nhỏ nhất.

GIAO DIỆN:



- Giao diện gồm các TextBox cho phép người dùng nhập các dữ liệu đầu vào như Số Thứ Nhất, Số Thứ Hai, Số Thứ Ba.
- Các TextBox có thuộc tính ReadOnly để hiển thị kết quả.
- Các button Tìm, Xóa và Thoát.

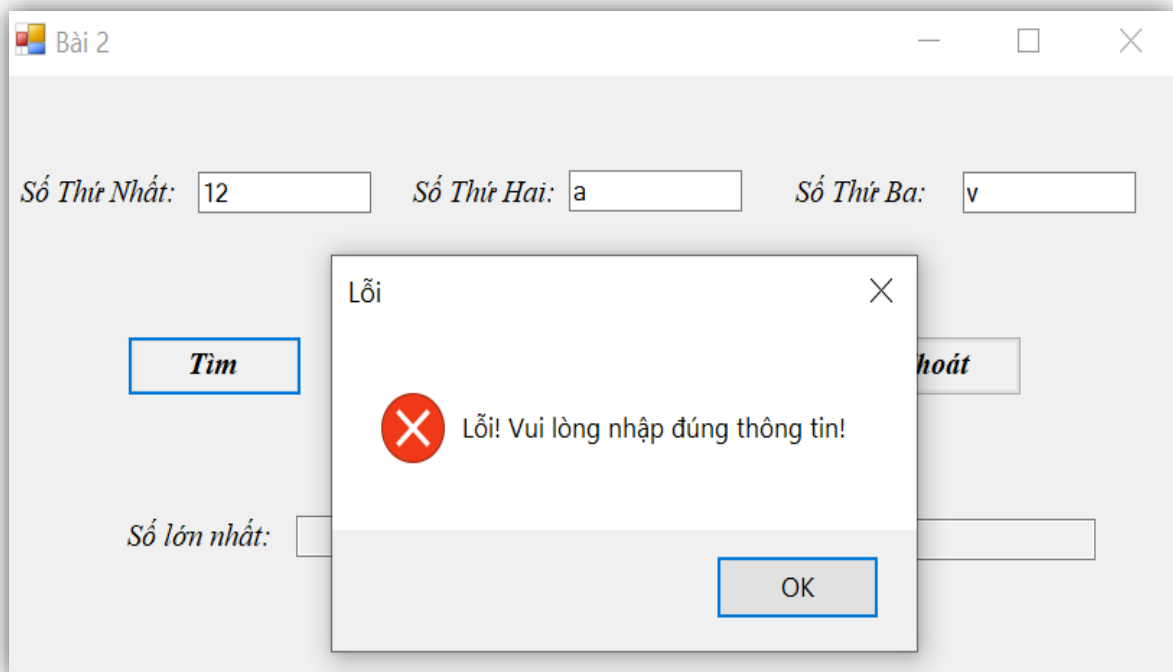
CODE:

Có các hàm sau:

- Hàm `buttonTim_Click(object sender, EventArgs e)`

```
private void buttonTim_Click(object sender, EventArgs e)
{
    try
    {
        double sothunhat = double.Parse(textBoxSoThuNhat.Text);
        double sothuhai = double.Parse(textBoxSoThuHai.Text);
        double sothuba = double.Parse(textBoxSoThuBa.Text);
        double solonnhat = Math.Max(sothunhat, (Math.Max(sothuhai, sothuba)));
        double sonhonhat = Math.Min(sothunhat, (Math.Min(sothuhai, sothuba)));
        textBoxMax.Text = solonnhat.ToString();
        textBoxMin.Text = sonhonhat.ToString();
    }
    catch
    {
        MessageBox.Show("Lỗi! Vui lòng nhập đúng thông tin!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

- Các câu lệnh lấy thông tin từ TextBox và ép sang kiểu double được đặt trong khối try.
- Trong khối try, có hàm Math.Max() đây là một phương thức tĩnh của lớp Math trong không gian tên System, được dùng để tìm số lớn nhất giữa hai số.
- Nếu ép kiểu không thành công thì khối catch sẽ được thực hiện và sẽ hiện thị MessageBox thông báo lỗi.



Khi nhập dữ liệu sai

Số Thứ Nhất: 120 Số Thứ Hai: 190 Số Thứ Ba: 30

Tìm Xóa Thoát

Số lớn nhất: 190 Số nhỏ nhất: 30

Khi nhập dữ liệu hợp lệ

- Hàm `buttonXoa_Click(object sender, EventArgs e)`

```
private void buttonXoa_Click(object sender, EventArgs e)
{
    textBoxSoThuNhat.Text = "";
    textBoxSoThuHai.Text = "";
    textBoxSoThuBa.Text = "";
    textBoxMax.Text = "";
    textBoxMin.Text = "";
}
```

- Hàm này có tác dụng đặt các Text của `TextBoxSoThuNhat`, `TextBoxSoThuHai`, ... thành chuỗi rỗng.

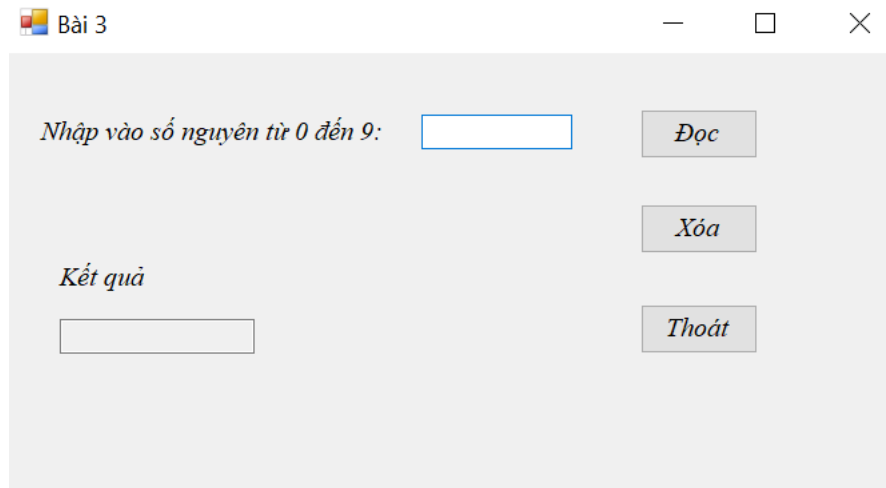
- Hàm `buttonThoat_Click(object sender, EventArgs e)`

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

- Hàm này được dùng để đóng Form Bài 2 lại. Quay về giao diện chính.

3) Bài 03 – Đọc số

GIAO DIỆN:



- Giao diện gồm TextBox để người dùng nhập số nguyên từ 0 đến 9, TextBox có thuộc tính ReadOnly để hiển thị kết quả.
- Các button Đọc, Xóa, Thoát.

CODE:

Có các hàm sau:

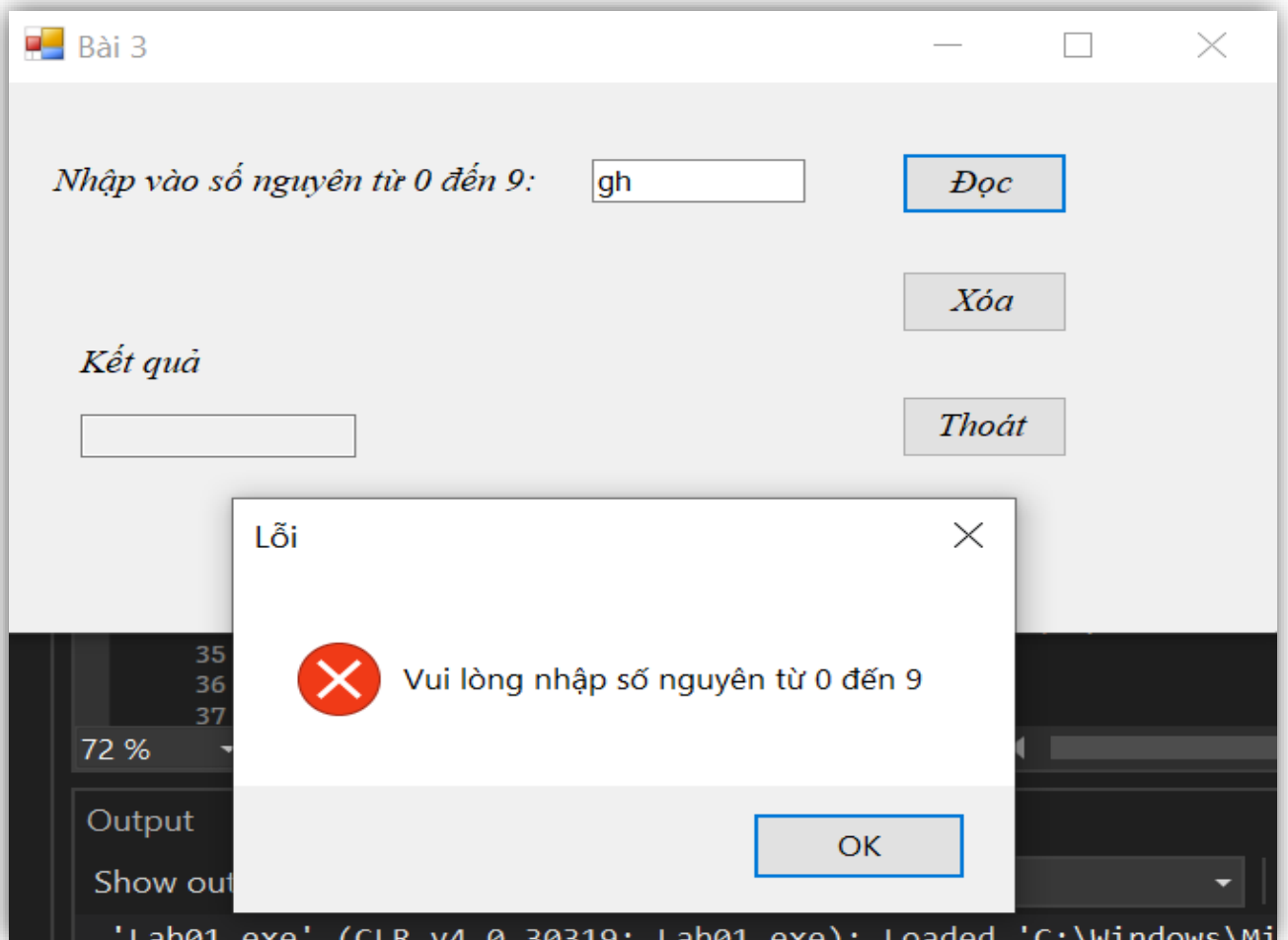
- Hàm `buttonDoc_Click(object sender, EventArgs e)`

```
private void buttonDoc_Click(object sender, EventArgs e)
{
    try
    {
        int x = int.Parse(textBox1.Text);
        switch (x)
        {
            case 0:
            {
                textBoxKetQua.Text = "Không";
                break;
            }
            case 1:
            {
                textBoxKetQua.Text = "Một";
                break;
            }
            case 2:
            {
                textBoxKetQua.Text = "Hai";
                break;
            }
            case 3:
            {
                textBoxKetQua.Text = "Ba";
                break;
            }
            case 4:
            {
                textBoxKetQua.Text = "Bốn";
                break;
            }
            case 5:
            {
                textBoxKetQua.Text = "Năm";
                break;
            }

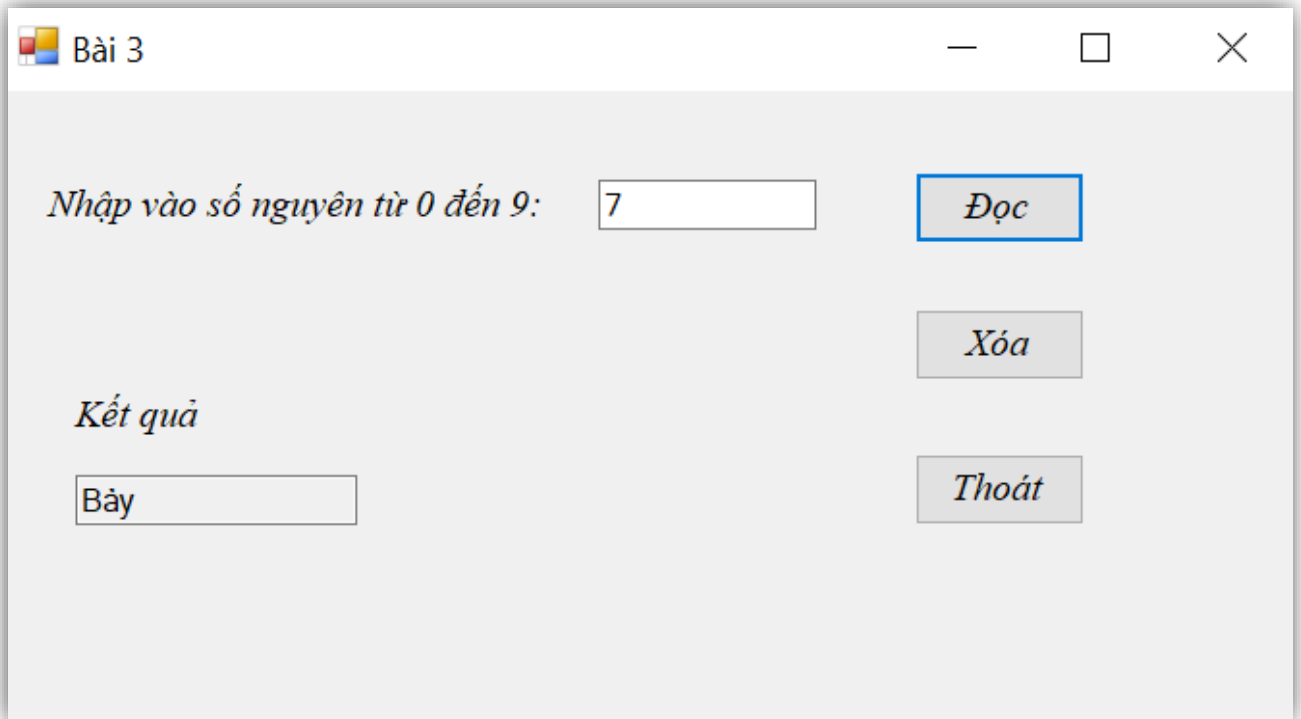
```

```
            case 7:
            {
                textBoxKetQua.Text = "Bảy";
                break;
            }
            case 8:
            {
                textBoxKetQua.Text = "Tám";
                break;
            }
            case 9:
            {
                textBoxKetQua.Text = "Chín";
                break;
            }
            default:
            {
                textBoxKetQua.Text = "Lỗi";
                MessageBox.Show("Vui lòng nhập số nguyên từ 0 đến 9", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
                break;
            }
        }
    }
    catch
    {
        MessageBox.Show("Vui lòng nhập số nguyên từ 0 đến 9", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
}
```

- Hàm này chứa các câu lệnh để lấy dữ liệu từ TextBox, chuyển nó sang kiểu int và dùng cấu trúc switch case để kiểm tra dữ liệu, đồng thời đặt kết quả cho textBoxKetQua.
- Các câu lệnh trên được đặt trong khối try để đảm bảo người dùng nhập số nguyên, nếu người dùng không nhập đúng sẽ hiện thị MessageBox thông báo lỗi.



Khi nhập dữ liệu sai



Khi nhập dữ liệu hợp lệ

- Hàm `buttonXoa_Click(object sender, EventArgs e)`

```
1 reference
private void buttonXoa_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    textBoxKetQua.Text = "";
}
```

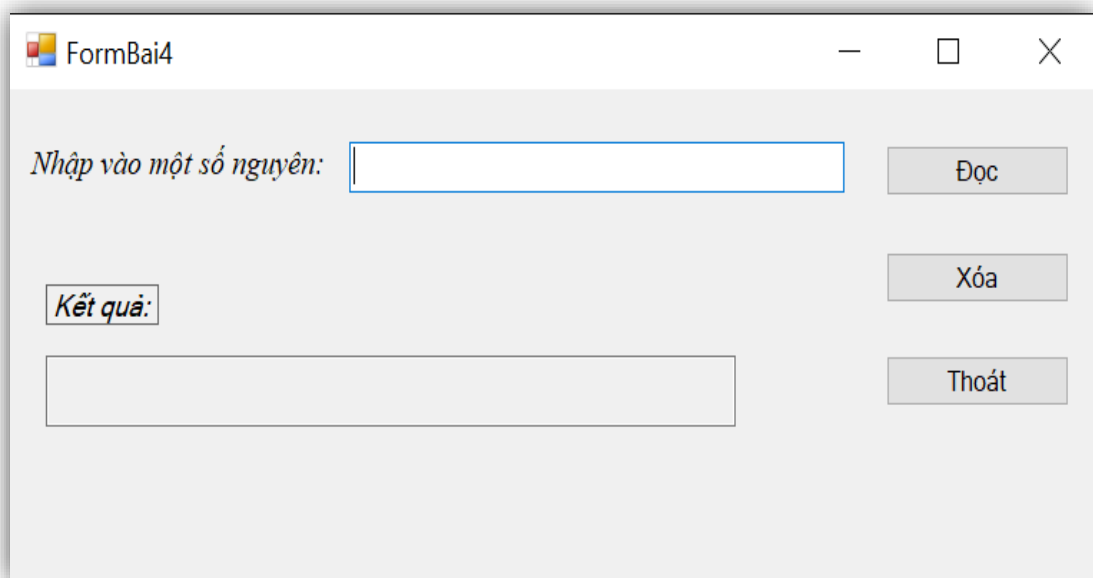
- Hàm này dùng để đặt Text trong ô `textBox1` (là ô người dùng nhập dữ liệu) và `textBoxKetQua` về rỗng.
- Hàm `buttonThoat_Click(object sender, EventArgs e)`

```
1 reference
private void buttonThoat_Click(object sender, EventArgs e)
{
    Close();
}
```

- Hàm này dùng để thoát Form Bài 3 trở về giao diện chính.

4) Bài 3.1 – Đọc số nâng cao

GIAO DIỆN:



- Giao diện gồm textBoxNhap để người dùng nhập một số nguyên có tối đa 12 chữ số
- TextBoxKetQua có thuộc tính ReadOnly để hiển thị kết quả
- Các button Đọc, Xóa, Thoát

Ý TƯỞNG:

- Để thực hiện được việc đọc số có tối đa 12 chữ số, ta sẽ tách số do người dùng nhập thành các lớp tỉ (trăm tỉ, chục tỉ, tỉ), lớp triệu (trăm triệu, chục triệu, triệu), lớp nghìn (trăm nghìn, chục nghìn, nghìn),..
- Sau khi tách ra, ta nhận thấy cách đọc của từng lớp là tương tự nhau chỉ thay mỗi đơn vị. Ví dụ như số 123 nếu ở lớp triệu thì sẽ đọc là “một trăm hai mươi ba triệu”, còn nếu như ở lớp nghìn sẽ được đọc là “một trăm hai mươi ba nghìn”,..

- Ta sẽ thiết kế một hàm với tham số thứ nhất truyền vào là các số của các lớp tỉ,triệu,nghìn,... tham số thứ hai là đơn vị.Trong hàm này ta sẽ xử lí một số ngoại lệ với số 0 vì khi có số 0 cách đọc trong Tiếng Việt sẽ khác như có thêm các chữ lẻ,mười,..
- Ví dụ: 101: “một trăm lẻ một” , “110” : “một trăm mười” ,100” “một trăm “

CODE:

- Hàm public static string Dem(char x)

```
public static string Dem(char x)
{
    switch (x)
    {
        case '0':
        {
            return "không";
        }
        case '1':
        {
            return "một";
        }
        case '2':
        {
            return "hai";
        }
        case '3':
        {
            return "ba";
        }
        case '4':
        {
            return "bốn";
        }
        case '5':
        {
            return "năm";
        }
    }
}
```

- Hàm này nhận một kí tự trong chuỗi người dùng nhập vào và trả về các đọc của nó. Hàm này tương tự như bài 03 Đọc Số.
- **Hàm public static string DocTheoDonVi (string number, string unit)**

```
public static string DocTheoDonVi(string number, string unit)
{
    string ketqua = "";
    if (number == "000")
    {
        return ketqua = "";
    }
    //nếu số truyền vào có 3 chữ số
    if (number.Length == 3)
    {
        //TH số truyền vào có dạng 00X với X là số bất kì khác 0
        if (number[0] == '0' && number[1] == '0')
        {
            ketqua = $"không trăm lẻ {Dem(number[2])} {unit}";
        }
        //TH số truyền vào có dạng 0X0
        else if (number[0] == '0' && number[2] == '0')
        {
            // TH số truyền vào có dạng 0X0 với
            if (number[1] == '1')
            {
                ketqua = $"không trăm mười {unit}";
            }
            // TH số truyền vào có dạng 0X0 với X!=1
            else
            {
                ketqua = $"không trăm {Dem(number[1])} mươi {unit}";
            }
        }
        //TH số truyền vào có dạng X00
        else if (number[1] == '0' && number[2] == '0')
        {
            ketqua = $"{Dem(number[0])} trăm {unit}";
        }
    }
}
```



```

}
//TH số truyền vào có dạng X00
else if (number[1] == '0' && number[2] == '0')
{
    ketqua = $"{Dem(number[0])} trăm {unit}";
}
//TH số truyền vào có dạng 0XX
else if (number[0] == '0')
{
    //TH số truyền vào có dạng 01X
    if (number[1] == '1')
    {
        ketqua = $"không trăm mười {Dem(number[2])} {unit}";
    }
    //TH số truyền vào có dạng 0xX với x khác 1
    else
    {
        ketqua = $"không trăm {Dem(number[1])} mươi {Dem(number[2])} {unit}";
    }
}
//TH số truyền vào có dạng X0X
else if (number[1] == '0')
{
    ketqua = $"{Dem(number[0])} trăm lẻ {Dem(number[2])} {unit}";
}
//TH số truyền vào có dạng XX0
else if (number[2] == '0')
{
    //TH số truyền vào có dạng X10
    if (number[1] == '1')
    {
        ketqua = $"{Dem(number[0])} trăm mười {unit}";
    }
    //TH số truyền vào có dạng Xx0 với x khác 1
    else
    {
        ketqua = $"{Dem(number[0])} trăm {Dem(number[1])} mươi {unit}";
    }
}

```

```
}  
//nếu số truyền vào có 2 chữ số  
if (number.Length == 2)  
{  
    if (number[1] == '0')  
    {  
        if (number[0] == '1')  
        {  
            ketqua = $"mười {unit}";  
        }  
        else  
        {  
            ketqua = $"{Dem(number[0])} mười {unit}";  
        }  
    }  
    else  
    {  
        if (number[0] == '1')  
        {  
            ketqua = $"mười {Dem(number[1])} {unit}";  
        }  
        else  
        {  
            ketqua = $"{Dem(number[0])} mười {Dem(number[1])} {unit}";  
        }  
    }  
}  
//nếu số truyền vào có một chữ số  
if (number.Length == 1)  
{  
    ketqua = $"{Dem(number[0])} {unit}";  
}  
  
return ketqua;  
}
```

- Đây là hàm quan trọng nhất ở bài này.
- Giải thích nguyên lý hoạt động:

```
if (number == "000")
{
    return ketqua = "";
}
```

- Nếu số ở lớp tương ứng truyền vào là 000 thì sẽ không được lớp đó. Ví dụ: 1000219 sẽ đọc là “một triệu hai trăm mười chín”, ta thấy rằng phần nghìn là 000 nên bị bỏ qua.
- Hàm public static string DocTheoDonVi (string number, string unit) sẽ có 3 câu lệnh if chính đó chính là:

```
//nếu số truyền vào có 3 chữ số
if (number.Length == 3)
```

```
//nếu số truyền vào có 2 chữ số
if (number.Length == 2)
```

```
//nếu số truyền vào có một chữ số
if (number.Length == 1)
```

- Ở khối lệnh if(number.Length==3), ta sẽ có các trường hợp sau:

❖ TH1: 00X với X là số bất kì khác 0

```
//TH số truyền vào có dạng 00X với X là số bất kì khác 0
if (number[0] == '0' && number[1] == '0')
{
    ketqua = $"không trăm lẻ {Dem(number[2])} {unit}";
}
```

❖ TH2: 0X0 với X là số bất kì khác 0

```
//TH số truyền vào có dạng 0X0
else if (number[0] == '0' && number[2] == '0')
{
    // TH số truyền vào có dạng 0X0 với X=1
    if (number[1] == '1')
    {
        ketqua = $"không trăm mười {unit}";
    }
    // TH số truyền vào có dạng 0X0 với X!=1
    else
    {
        ketqua = $"không trăm {Dem(number[1])} mười {unit}";
    }
}
```

❖ TH3: X00 với X là số bất kì khác 0

```
//TH số truyền vào có dạng X00
else if (number[1] == '0' && number[2] == '0')
{
    ketqua = $"{Dem(number[0])} trăm {unit}";
}
```

❖ TH4: 0XX với X là số bất kì khác 0

```
//TH số truyền vào có dạng 0XX
else if (number[0] == '0')
{
    //TH số truyền vào có dạng 01X
    if (number[1] == '1')
    {
        ketqua = $"không trăm mười {Dem(number[2])} {unit}";
    }
    //TH số truyền vào có dạng 0xX với x khác 1
    else
    {
        ketqua = $"không trăm {Dem(number[1])} mười {Dem(number[2])} {unit}";
    }
}
```

❖ TH5: X0X với X là số bất kì khác 0

```
//TH số truyền vào có dạng X0X
else if (number[1] == '0')
{
    ketqua = $"{Dem(number[0])} trăm lẻ {Dem(number[2])} {unit}";
}
```

- ❖ TH6: XX0 với X là số bất kì khác 0

```
//TH số truyền vào có dạng XX0
else if (number[2] == '0')
{
    //TH số truyền vào có dạng X10
    if (number[1] == '1')
    {
        ketqua = $"{Dem(number[0])} trăm mười {unit}";
    }
    //TH số truyền vào có dạng Xx0 với x khác 1
    else
    {
        ketqua = $"{Dem(number[0])} trăm {Dem(number[1])} mươi {unit}";
    }
}
```

- ❖ TH7: số truyền vào có 3 chữ số đều khác 0

```
//TH số truyền vào có ba chữ số đều khác 0
else
{
    if (number[1] == '1')
    {
        ketqua = $"{Dem(number[0])} trăm mười {Dem(number[2])} {unit}";
    }
    else
    {
        ketqua = $"{Dem(number[0])} trăm {Dem(number[1])} mươi {Dem(number[2])}";
    }
}
```

- Ở khối lệnh if(number.Length==2), ta sẽ có các trường hợp sau:

- ❖ TH1: số truyền vào có dạng X0 với X là số bất kì khác 0

```

if (number[1] == '0')
{
    if (number[0] == '1')
    {
        ketqua = $"mười {unit}";
    }

    else
    {
        ketqua = $"{Dem(number[0])} mười {unit}";
    }
}

```

❖ TH2: số truyền vào có dạng XX với X là số bất kì khác 0

```

else
{
    if (number[0] == '1')
    {
        ketqua = $"mười {Dem(number[1])} {unit}";
    }
    else
    {
        ketqua = $"{Dem(number[0])} mười {Dem(number[1])} {unit}";
    }
}

```

○ Ở khối lệnh if(number.Length==1), ta sẽ có:

```

//nếu số truyền vào có một chữ số
if (number.Length == 1)
{
    ketqua = $"{Dem(number[0])} {unit}";
}

```

- Hàm buttonDoc_Click(object sender, EventArgs e)

```
1 reference
private void buttonDoc_Click(object sender, EventArgs e)
{
    try
    {
        string number = textBoxNhap.Text;
        //dùng để tránh người dùng nhập dư các số 0 đằng trước
        long tmp = long.Parse(number);
        number = tmp.ToString();
        int length = number.Length;

        if(length >12)
        {
            MessageBox.Show("Vui lòng nhập số nguyên có tối đa 12 chữ số!", "Lỗi", MessageBoxButtons.OK, Me
            return;
        }

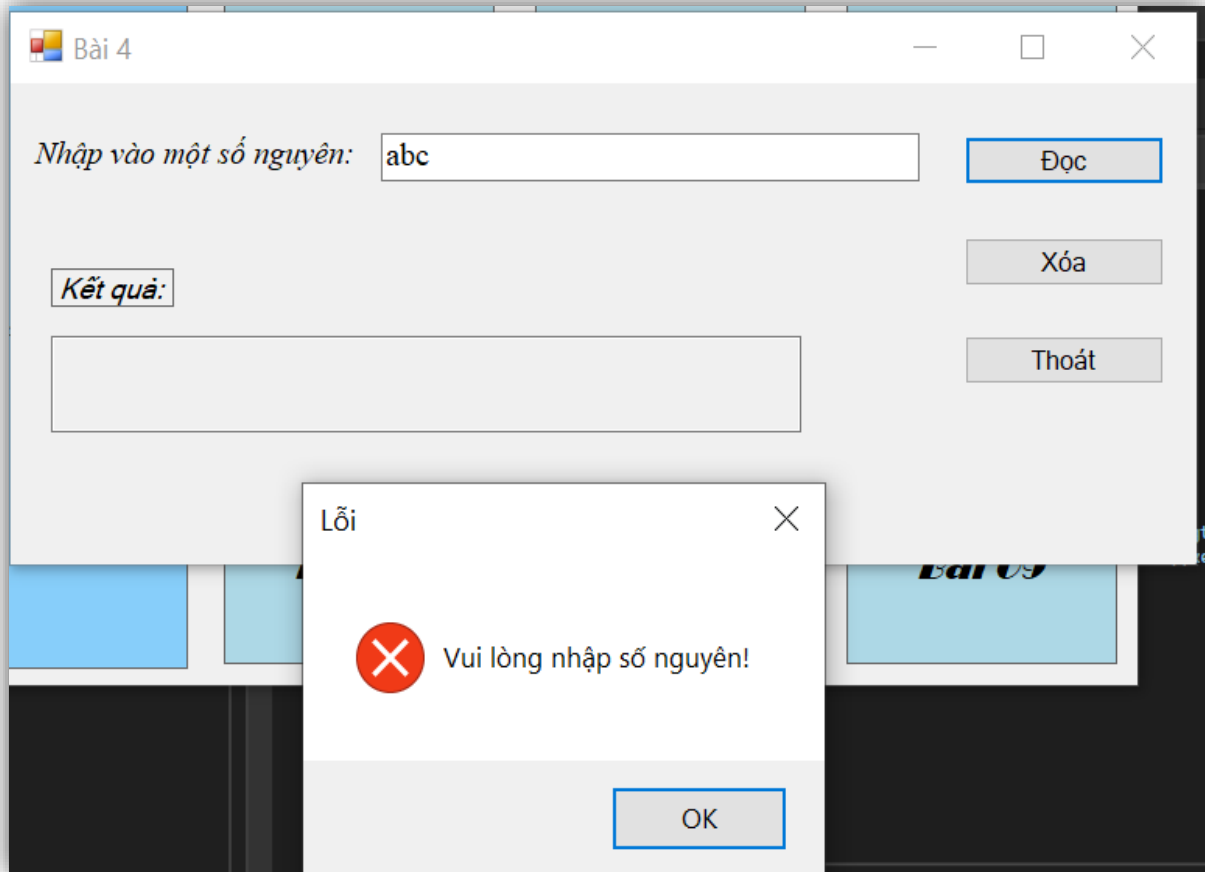
        if (length >= 1 && length <= 3)
        {
            string ketqua = DocTheoDonVi(number, "");
            while (ketqua.Contains("mười một") || ketqua.Contains("mười năm") || ketqua.Contains("mười năm")
            {
                ketqua = ketqua.Replace("mười một", "mười mốt");
                ketqua = ketqua.Replace("mười năm", "mười lăm");
                ketqua = ketqua.Replace("mười năm", "mười lăm");
            }
            textBoxKetQua.Text = ketqua;
        }
        else if (length >= 4 && length <= 6)
        {
            int batdauhangdonvi = length - 3;
            string donvi = number.Substring(batdauhangdonvi);
            string hangngan = number.Substring(0, batdauhangdonvi);
        }
    }
}
```

```

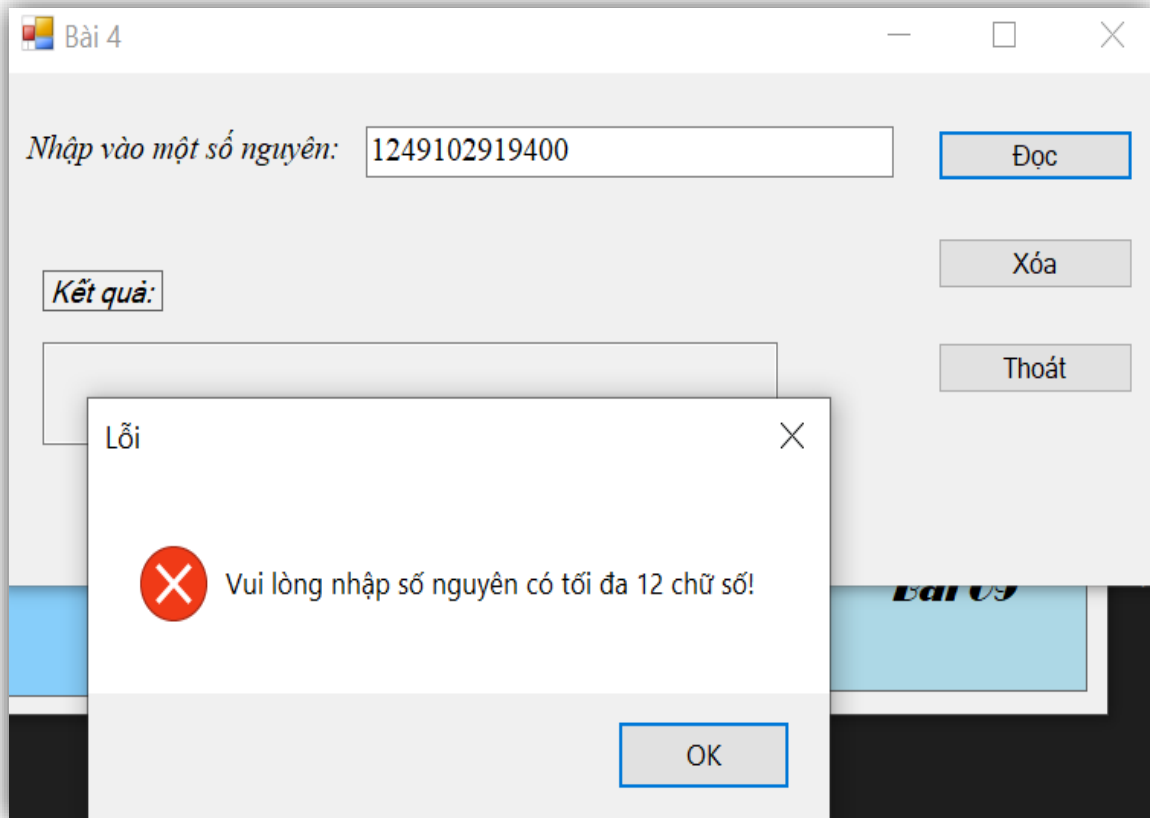
    }
    else if (length >= 7 && length <= 9)
    {
        int batdauhangdonvi = length - 3;
        string donvi = number.Substring(batdauhangdonvi);
        int batdauhangngan = length - 6;
        string hangngan = number.Substring(batdauhangngan, 3);
        string hangtrieu = number.Substring(0, batdauhangngan);
        string ketqua = DocTheoDonVi(hangtrieu, "triệu") + " " + DocTheoDonVi(hangngan, "ngàn") + " " + DocTheoDonVi(donvi, "");
        while (ketqua.Contains("mười một") || ketqua.Contains("mười năm") || ketqua.Contains("mười lăm"))
        {
            ketqua = ketqua.Replace("mười một", "mười mốt");
            ketqua = ketqua.Replace("mười năm", "mười lăm");
            ketqua = ketqua.Replace("mười năm", "mười lăm");
        }
        textBoxKetQua.Text = ketqua;
    }
    else
    {
        int batdauhangdonvi = length - 3;
        string donvi = number.Substring(batdauhangdonvi);
        int batdauhangngan = length - 6;
        string hangngan = number.Substring(batdauhangngan, 3);
        int batdauhangtrieu = length - 9;
        string hangtrieu = number.Substring(batdauhangtrieu, 3);
        string hangti = number.Substring(0, batdauhangtrieu);
        string ketqua = DocTheoDonVi(hangti, "tỉ") + " " + DocTheoDonVi(hangtrieu, "triệu") + " " + DocTheoDonVi(hangngan, "ngàn") + " " + DocTheoDonVi(donvi, "");
        while (ketqua.Contains("mười một") || ketqua.Contains("mười năm") || ketqua.Contains("mười lăm"))
        {
            ketqua = ketqua.Replace("mười một", "mười mốt");
            ketqua = ketqua.Replace("mười năm", "mười lăm");
            ketqua = ketqua.Replace("mười năm", "mười lăm");
        }
        textBoxKetQua.Text = ketqua;
    }
}

```

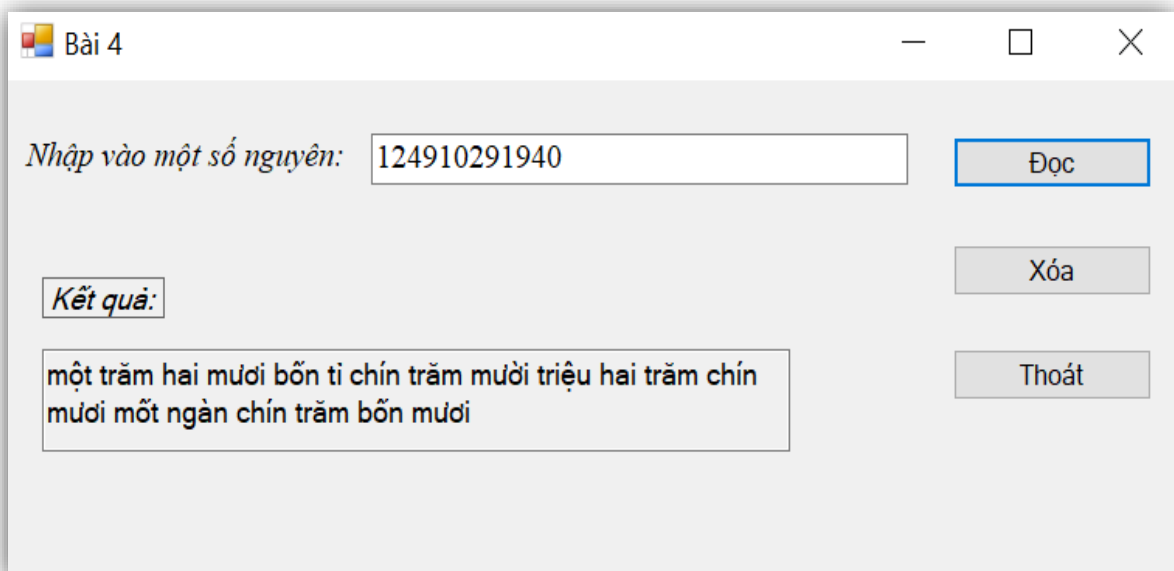
- Hàm buttonDoc_Click(object sender, EventArgs e) có nhiệm vụ kiểm tra độ dài của chuỗi số do người dùng nhập vào, tách chuỗi số ấy ra theo các lớp tỉ, triệu, nghìn, trăm để gọi hàm DocTheoDonVi và truyền theo đơn vị tương ứng.
- Nếu chuỗi do người dùng nhập có độ dài lớn hơn 12 hoặc không phải là số thì sẽ có MessageBox báo lỗi.
- Trong hàm sẽ có các khối lệnh while để thay thế một số từ trong chuỗi kết quả như “mười một” thành “mười mốt”, “mười năm” thành “mười lăm”, ... để giống với các đọc tiếng Việt hơn.



Khi nhập dữ liệu sai (không phải là số nguyên)



Khi nhập dữ liệu sai (số nguyên có số lượng chữ số lớn hơn 12)



Khi nhập dữ liệu hợp lệ

- Hàm `buttonXoa_Click(object sender, EventArgs e)`

```
1 reference
private void buttonXoa_Click(object sender, EventArgs e)
{
    textBoxKetQua.Text = "";
    textBoxNhap.Text = "";
}
```

- Hàm dùng để đặt Text trong các ô `textBoxKetQua` và `textBoxNhap` thành chuỗi rỗng.

- Hàm `buttonThoat_Click(object sender, EventArgs e)`

```
1 reference
private void buttonThoat_Click(object sender, EventArgs e)
{
    Close();
}
```

- Hàm dùng để đóng Form hiện tại, quay về giao diện chính.

5) Bài 04- Quản lí phòng vé

GIAO DIỆN:

Bài 5

HỆ THỐNG BÁN VÉ

MÀN HÌNH

| | | | | |
|----|----|----|----|----|
| A1 | A2 | A3 | A4 | A5 |
| B1 | B2 | B3 | B4 | B5 |
| C1 | C2 | C3 | C4 | C5 |

PHÒNG

Chú thích

- Ghế đã được chọn
- Ghế bạn đang chọn
- Ghế trống

Thông tin

Họ tên khách hàng:

Chọn bộ phim:

Phòng chiếu phim:

Xác nhận

Hóa đơn

Khách hàng:

Vé đã chọn:

Phim:

Phòng chiếu:

Số tiền thanh toán:

- Giao diện có bố cục gồm 3 phần chính:
 - ❖ Phần chứa danh sách các ghế: các ghế là các label, khi được chọn các label sẽ đổi màu BackColor thành màu vàng, những label có BackColor màu trắng là những ghế trống và những label có BackColor màu đỏ là những ghế đã được chọn.
 - ❖ Phần thông tin dùng để nhập các họ tên khách hàng, tên phim và phòng chiếu
 - ❖ Phần hóa đơn

Ý TƯỞNG:

- Ta sẽ sử dụng các Collections sau:
 - List<string> để lưu danh sách các ghế được chọn trong mỗi lần đăng kí xem phim.
 - Dictionary<string,double> dùng để lưu tên phim và giá vé chuẩn.
 - Dictionary<string,List<string>> với key để lưu tên phim, số phòng và value là một List các ghế đã được đặt.
 - Khi người dùng chọn ghế ta sẽ thay đổi màu của label ghế tương ứng thành màu vàng đồng thời đưa tên label vào trong List. Sau khi người dùng chọn

xong và nhấn xác nhận ta sẽ đưa List các ghế được chọn vào Dictionary<string, List<string>> để lưu trữ.

- Mỗi lần người dùng chọn xong tên xong và số phòng ta sẽ lấy các ghế trong Dictionary<string, List<string>> để tiến hành tô màu cho các label tương ứng.

CODE:

- Gồm các hàm sau:
 - **Hàm labelA1_Click(object sender, EventArgs e):** Hàm này dùng để đặt màu cho label các ghế khi được chọn đồng thời lưu tên ghế đang chọn vào trong List.

```
1 reference
private void labelA1_Click(object sender, EventArgs e)
{
    if(string.IsNullOrEmpty(textBoxNhapKhachHang.Text) || comboBoxPhim.SelectedIndex == -1 || comboBoxPhong.SelectedIndex == -1)
    {
        MessageBox.Show("Vui lòng điền đầy đủ thông tin khách hàng, tên phim, số phòng!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    if(labelA1.BackColor == Color.Red)
    {
        MessageBox.Show("Ghế đã được chọn! Vui lòng chọn ghế khác", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    if(labelA1.BackColor == Color.White)
    {
        labelA1.BackColor = Color.Yellow;
        list.Add(labelA1.Text);
    }
    else {
        labelA1.BackColor = Color.White;
        list.Remove(labelA1.Text);
    }
}
```

- **Hàm ResetColor()** : dùng để đặt lại toàn bộ màu cho các ghế lại ban đầu sau khi người dùng nhấn nút Xác nhận.

```
public void ResetColor()
{
    labelA1.BackColor = Color.White;
    labelA2.BackColor = Color.White;
    labelA3.BackColor = Color.White;
    labelA4.BackColor = Color.White;
    labelA5.BackColor = Color.White;
    labelB1.BackColor = Color.Red;
    labelB2.BackColor = Color.White;
    labelB3.BackColor = Color.White;
    labelB4.BackColor = Color.White;
    labelB5.BackColor = Color.Red;
    labelC1.BackColor = Color.White;
    labelC2.BackColor = Color.White;
    labelC3.BackColor = Color.White;
    labelC4.BackColor = Color.White;
    labelC5.BackColor = Color.White;
}
```

- **Hàm ToMau()**: dùng để tô màu các ghế đã được chọn sau khi người dùng chọn tên phim và số phòng

```
public void ToMau(string tenphim, string phong)
{
    string key = tenphim + phong;
    foreach (var item in dictGheDuocChon)
    {
        if(item.Key == key)
        {
            foreach (var ghe in item.Value)
            {
                switch (ghe)
                {
                    case "A1":
                    {
                        labelA1.BackColor = Color.Red;
                        break;
                    }
                    case "A2":
                    {
                        labelA2.BackColor = Color.Red;
                        break;
                    }
                    case "A3":
                    {
                        labelA3.BackColor = Color.Red;
                        break;
                    }
                    case "A4":
                    {
                        labelA4.BackColor = Color.Red;
                        break;
                    }
                    case "A5":
                    {
                        labelA5.BackColor = Color.Red;
                        break;
                    }
                }
            }
        }
    }
}
```

- **Hàm TinhTien(List<string>list)** : dùng để tính tiền dựa trên List các ghế đã chọn.

```

1 reference
public double TinhTien(List<string> list)
{
    double tien = 0;
    string tenphim=comboBoxPhim.SelectedItem.ToString();
    foreach (var item in list)
    {
        switch (item)
        {
            case "A1": case "A5": case "C1": case "C5":
            {
                tien += dictGiaChuan[tenphim] / 4;
                break;
            }
            case "A2": case "A3": case "A4": case "C2": case "C3": case "C4":
            {
                tien += dictGiaChuan[tenphim];
                break;
            }
            case "B2": case "B3": case "B4":
            {
                tien += dictGiaChuan[tenphim]*2;
                break;
            }
        }
    }

    return tien;
}

```

- **Hàm buttonXacNhan_Click(object sender, EventArgs e):** dùng để thiết lập Text cho các ô textBoxHienThiKhachHang, textBoxHienThiPhim, textBoxHienThiPhong,... đồng thời đưa danh sách các ghế được chọn vào trong Dictionary<string, List<string>>

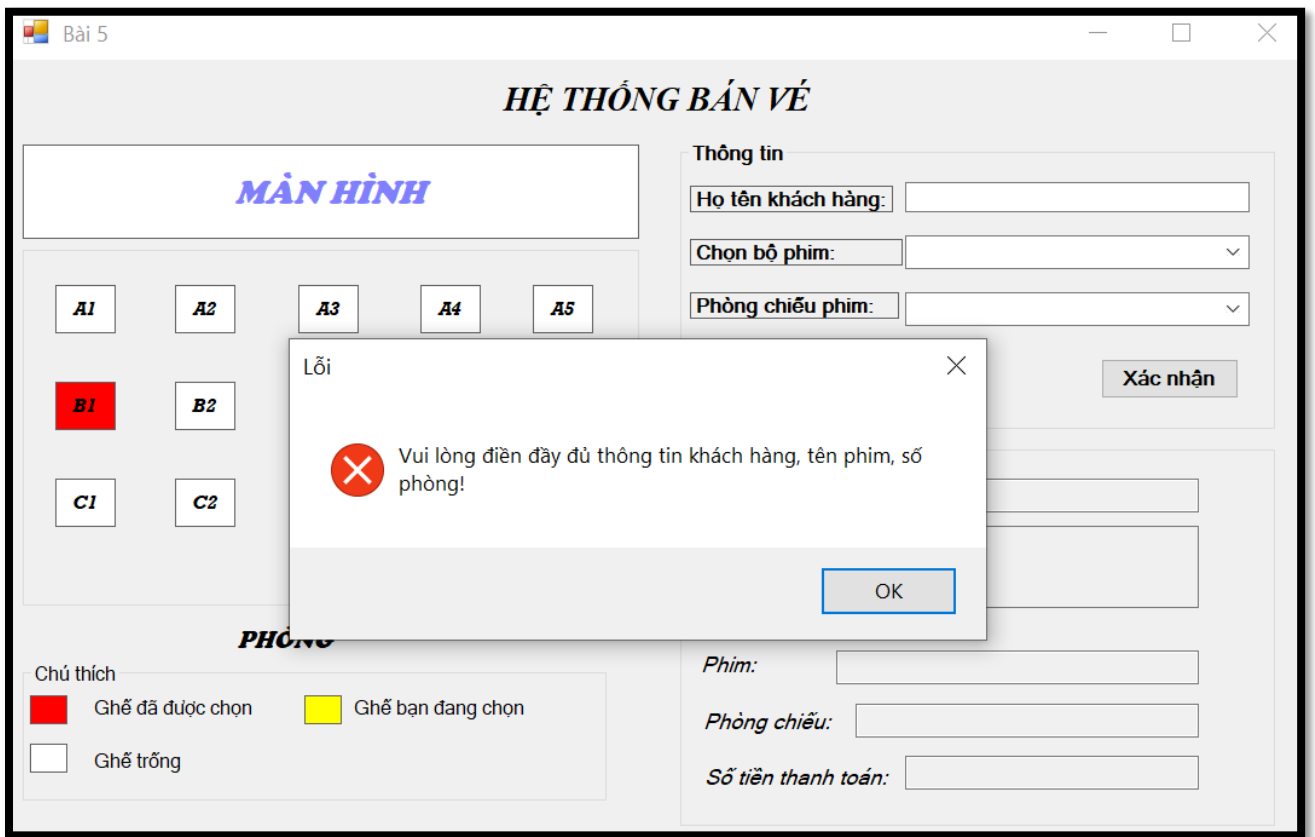

```
private void buttonXacNhan_Click(object sender, EventArgs e)
{
    listBoxHienThiVe.Items.Clear();
    if (!string.IsNullOrEmpty(textBoxNhapKhachHang.Text) && comboBoxPhim.SelectedIndex != -1
        && comboBoxPhong.SelectedIndex != -1 && list.Count != 0)
    {
        string tenkhachhang = textBoxNhapKhachHang.Text;
        string tenphim = (string)comboBoxPhim.SelectedItem;
        string phong = (string)comboBoxPhong.SelectedItem;
        textBoxHienThiKhachHang.Text = tenkhachhang;
        textBoxHienThiPhim.Text = tenphim;
        textBoxHienThiPhong.Text = phong;
        foreach (var item in list)
        {
            string tmp = CheckLoaiVeCuaChoNgon(item) + " " + item.ToString();
            listBoxHienThiVe.Items.Add(tmp);
        }
        double tien = TinhTien(list);
        string textTien = tien.ToString("N").Replace(".", " ");
        textBoxHienThiTien.Text = textTien + " VNĐ";

        string phimphong = tenphim + phong;
        if (dictGheDuocChon.ContainsKey(phimphong))
        {
            var x = dictGheDuocChon[phimphong];
            List<string> listnew = new List<string>();
            foreach (var item in x)
            {
                listnew.Add(item.ToString());
            }
            foreach (var item in list)
            {
                listnew.Add(item);
            }
            dictGheDuocChon.Remove(phimphong);
            dictGheDuocChon.Add(phimphong, listnew);
            list.Clear();
        }
    }
}
```

```
dictGheDuocChon.Add(phimphong, listnew);
list.Clear();
}
else
{
    List<string> tmp = new List<string>(list);
    dictGheDuocChon.Add(phimphong, tmp);
    list.Clear();
}
ResetColor();
textBoxNhapKhachHang.Text = "";
comboBoxPhim.Text = "";
comboBoxPhong.Text = "";
comboBoxPhim.SelectedIndex = -1;
comboBoxPhong.SelectedIndex = -1;
}
else
{
    if (string.IsNullOrEmpty(textBoxNhapKhachHang.Text) || comboBoxPhong.SelectedIndex == -1
        || comboBoxPhim.SelectedIndex == -1)
    {
        MessageBox.Show("Vui lòng điền đầy đủ thông tin!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    if (list.Count == 0)
    {
        MessageBox.Show("Vui lòng chọn chỗ ngồi!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
}
```

Nhập thông tin khách hàng và chọn ghế

Sau khi nhấn nút Xác nhận



Lỗi khi chưa điền đầy đủ thông tin

Bài 5

HỆ THỐNG BÁN VÉ

MÀN HÌNH

| | | | | |
|----|----|----|----|----|
| A1 | A2 | A3 | A4 | A5 |
| B1 | B2 | B3 | B4 | B5 |
| C1 | C2 | C3 | C4 | C5 |

PHÒNG 3

Chú thích

 Ghế đã được chọn  Ghế bạn đang chọn

 Ghế trống

Thông tin

Họ tên khách hàng:

Chọn bộ phim:

Phòng chiếu phim:

Lỗi

 Vui lòng chọn chỗ ngồi!

Phim:

Phòng chiếu:

Số tiền thanh toán:

Lỗi chưa chọn ghế

6) Bài 05: Tính toán

GIAO DIỆN:

The screenshot shows a Windows application window titled "Bài 6". Inside the window, there are two text boxes for input, labeled "Nhập A:" and "Nhập B:". Below these is a dropdown menu. There are three buttons: "Tính các giá trị", "Xóa", and "Thoát". At the bottom is a large text area labeled "Kết quả".

- Giao diện gồm các TextBox cho phép người dùng nhập các số nguyên.
- Combobox với các lựa chọn: “Bảng cửu chương”, “Tính toán giá trị”
- Các Button Tính các giá trị, Xóa, Thoát
- Group Box Kết quả để hiển thị kết quả tính toán

CODE:

- Hàm `buttonTinh_Click(object sender ,EventArgs e)`

```

1 reference
private void buttonTinh_Click(object sender, EventArgs e)
{
    try
    {
        if (comboBox1.SelectedItem == "Bảng cửu chương")
        {
            int a = int.Parse(textBoxA.Text);
            int b = int.Parse(textBoxB.Text);
            int kq = b - a;
            labelKetQua.Text = kq.ToString();
        }
        else if (comboBox1.SelectedItem == "Tính toán giá trị")
        {
            int a = int.Parse(textBoxA.Text);
            int b = int.Parse(textBoxB.Text);
            int giaithua = a - b;
            int kqgiaithua = 1;
            string kq = "";
            if (giaithua < 0)
            {
                kq = $"{a} - {b} )! : Lỗi\n";
            }
            else
            {
                for (int i = 1; i <= giaithua; i++)
                {
                    kqgiaithua *= i;
                }
                kq = $"{a} - {b} )! = {kqgiaithua}\n";
            }
        }
    }
}

```

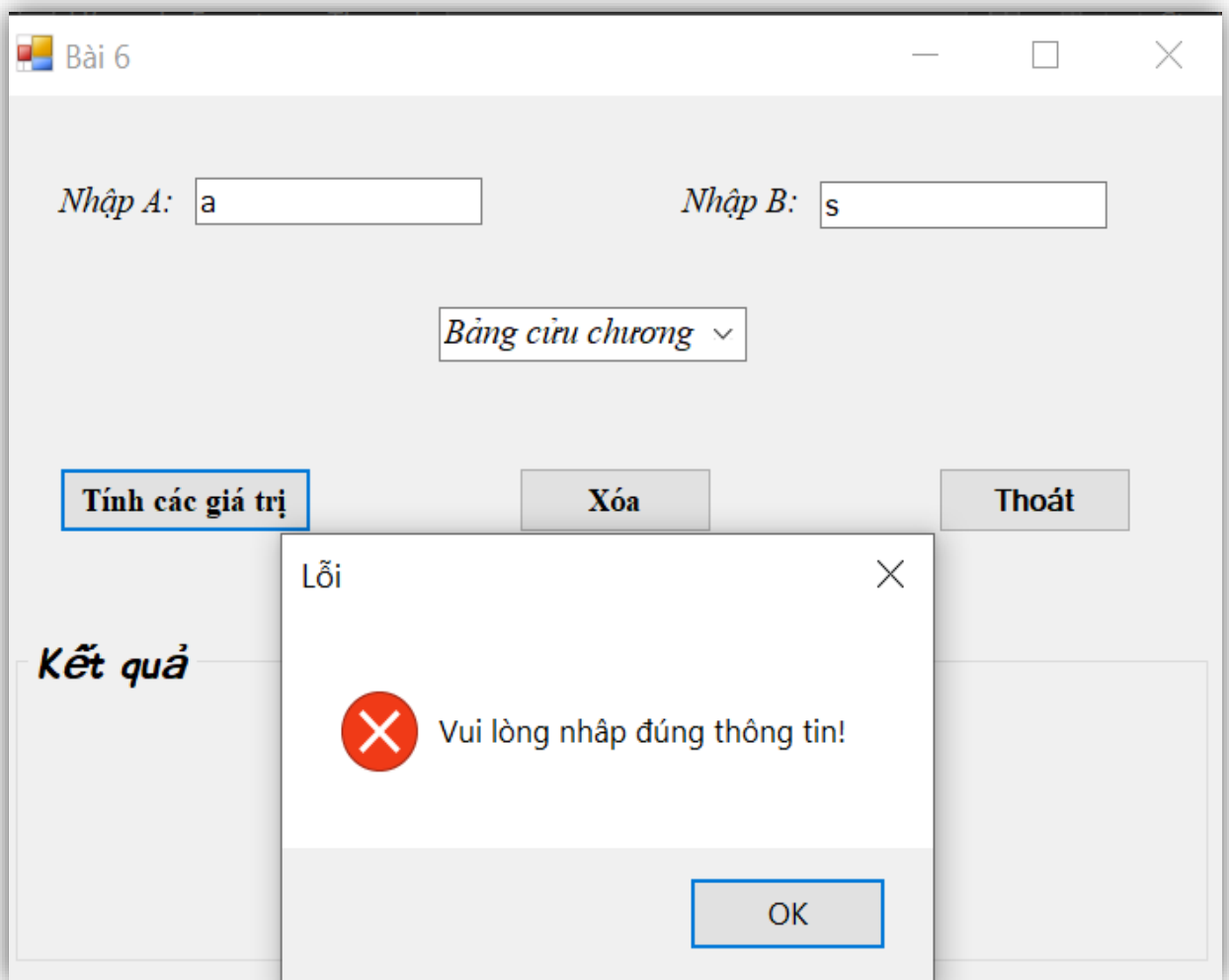
```

        if (b < 0)
        {
            kq += $"Tổng S: Lỗi";
        }
        else
        {
            double tong = 0;
            for (int i = 1; i <= b; i++)
            {
                tong += Math.Pow(a, i);
            }
            kq += $"Tổng S: {tong}";
        }

        labelKetQua.Text = kq.ToString();

```

- Hàm này được dùng để tính toán các giá trị theo đề bài dựa trên lựa chọn của combobox.
- Các câu lệnh tính toán được đặt trong khối try để đảm bảo người dùng nhập đúng input là 2 số nguyên.



Khi nhập dữ liệu sai

Nhập A: 4 Nhập B: 2

Tính toán giá trị ▾

Tính các giá trị Xóa Thoát

Kết quả
(4 - 2)! = 2
Tổng S: 20

Khi nhập dữ liệu hợp lệ

- Hàm `buttonXoa_Click(object sender, EventArgs e)`

```
1 reference
private void buttonXoa_Click(object sender, EventArgs e)
{
    textBoxA.Text = "";
    textBoxB.Text = "";
    labelKetQua.Text = "";
}
```

- Được dùng để đặt text của `textBoxA`, `textBoxB` và `labelKetQua` thành chuỗi rỗng.

- Hàm `buttonThoat_Click(object sender, EventArgs e)`

```
1 reference
private void buttonThoat_Click(object sender, EventArgs e)
{
    Close();
}
```

- Dùng để thoát Form hiện tại và quay lại giao diện chính.

7) Bài 06: Viết chương trình nhập vào ngày tháng năm sinh, xuất ra thông tin cung hoàng đạo của bạn là gì?

GIAO DIỆN:

The screenshot shows a Windows application window titled "Bài 7". Inside the window, there is a header box with the text "CUNG HOÀNG ĐẠO CỦA BẠN". Below the header, there are three labels with corresponding text boxes: "Nhập ngày sinh:", "Nhập tháng sinh:", and "Nhập năm sinh:". Below these is a button labeled "Xác nhận". At the bottom, there is a label "Kết quả:" followed by a large, empty text area for displaying the result.

- Giao diện gồm các TextBox để người dùng nhập ngày, tháng, năm sinh.
- Button Xác nhận và GroupBox Kết quả để hiển thị thông tin về cung hoàng đạo của người dùng.

CODE:

- Hàm `buttonXacNhan_Click(object sender, EventArgs e)`

```

1 reference
private void buttonXacNhan_Click(object sender, EventArgs e)
{
    DateTime dt;

    try
    {
        int nam = int.Parse(textBoxNam.Text);
        int thang = int.Parse(textBoxThang.Text);
        int ngay = int.Parse(textBoxNgay.Text);
        dt = new DateTime(nam, thang, ngay);
        if (dt >= new DateTime(nam, 03, 21) && dt <= new DateTime(nam, 04, 20))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Bạch Dương!";
        }
        else if (dt >= new DateTime(nam, 04, 21) && dt <= new DateTime(nam, 05, 21))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Kim Ngưu!";
        }
        else if (dt >= new DateTime(nam, 05, 22) && dt <= new DateTime(nam, 06, 21))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Song Tử!";
        }
        else if (dt >= new DateTime(nam, 06, 22) && dt <= new DateTime(nam, 07, 22))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Cự Giải!";
        }
        else if (dt >= new DateTime(nam, 07, 23) && dt <= new DateTime(nam, 08, 22))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Sư Tử!";
        }
        else if (dt >= new DateTime(nam, 08, 23) && dt <= new DateTime(nam, 09, 23))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Xử Nữ!";
        }
        else if (dt >= new DateTime(nam, 09, 24) && dt <= new DateTime(nam, 10, 23))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Thiên Bình!";
        }
        else if (dt >= new DateTime(nam, 10, 24) && dt <= new DateTime(nam, 11, 22))
    }

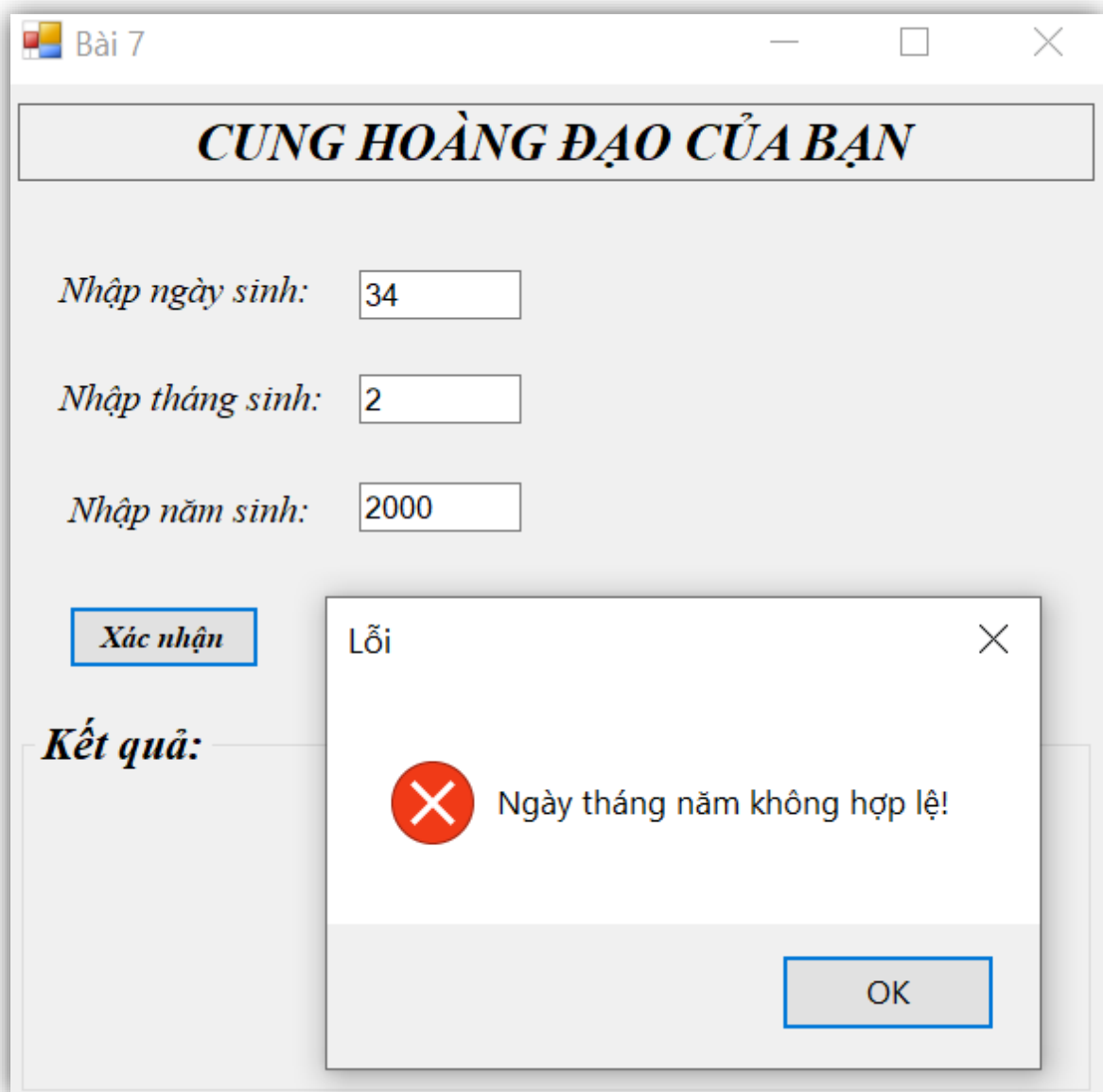
```

```

        else if (dt >= new DateTime(nam, 10, 24) && dt <= new DateTime(nam, 11, 22))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Thần Nông!";
        }
        else if (dt >= new DateTime(nam, 11, 23) && dt <= new DateTime(nam, 12, 21))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Nhân Mã!";
        }
        else if (dt >= new DateTime(nam, 12, 22) && dt <= new DateTime(nam+1, 01, 20))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Ma Kết!";
        }
        else if (dt >= new DateTime(nam, 01, 21) && dt <= new DateTime(nam, 02, 19))
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Bảo Bình!";
        }
        else
        {
            labelKetQua.Text = $"Ngày sinh nhật của bạn: {dt.ToString("dd/MM/yyyy")}. \nBạn thuộc cung Song Ngư!";
        }
    }
    catch
    {
        MessageBox.Show("Ngày tháng năm không hợp lệ!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

- Khởi tạo kiểu dữ liệu DateTime với các đối số truyền vào là ngày tháng năm sinh do người dùng nhập.
- Sau đó dùng cấu trúc if else để xác định cung hoàng đạo cho ngày tháng năm sinh đó.
- Các câu lệnh được đặt trong khối try để đảm bảo ngày tháng năm sinh của người dùng nhập vào là hợp lệ.



Khi nhập dữ liệu sai

Bài 7

CUNG HOÀNG ĐẠO CỦA BẠN

Nhập ngày sinh:

Nhập tháng sinh:

Nhập năm sinh:

Xác nhận

Kết quả:
*Ngày sinh nhật của bạn: 20/08/2004.
Bạn thuộc cung Sư Tử!*

Khi nhập dữ liệu hợp lệ

8) Bài 07: Xử lý mảng

GIAO DIỆN:

The screenshot shows a Windows application window titled "Bài 8". Inside the window, there is a label "Nhập thông tin điểm sinh viên:" followed by a single-line text input field. Below this input field, there are three buttons arranged horizontally: "Xác nhận", "Xóa", and "Thoát". At the bottom of the window, there is a section titled "Kết quả" which contains a large, empty multi-line text area for displaying results.

- Giao diện gồm TextBox để nhập thông tin điểm sinh viên và tên sinh viên.
- Group Box Kết quả gồm một ListBox để hiển thị họ tên sinh viên, điểm số của từng môn, điểm trung bình, điểm cao nhất, thấp nhất, số môn đạt, không đạt và xếp loại học lực của sinh viên đó.
- Các button Xác nhận, Xóa và Thoát.

Ý TƯỞNG:

- Ta dùng hàm Split() để tách chuỗi do người dùng nhập theo dấu “,” sau đó lưu trữ tên sinh viên và danh sách các điểm của sinh viên vào một mảng double[].

CODE:

- Gồm các hàm sau:

- Hàm bool CheckFormat (string format):

```

public bool CheckFormat(string format)
{
    var parts = format.Split(',');
    int length = parts.Length;
    if (length == 1)
    {
        MessageBox.Show("Format không phù hợp!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
    DsDiem = new double[length - 1];
    for (int i = 0; i < length; i++) {
        if (i == 0)
        {
            tensinhvien = parts[i];
        }
        else
        {
            double output;
            if (double.TryParse(parts[i], out output))
            {
                if (output >= 0 && output <= 10)
                {
                    DsDiem[i - 1] = output;
                }
                else
                {
                    MessageBox.Show("Điểm không phù hợp!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    return false;
                }
            }
            else
            {
                MessageBox.Show("Format không phù hợp!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
        }
    }
}

```

- Dùng để tách chuỗi người dùng nhập vào theo dấu “,” đồng thời lưu điểm sinh viên vào mảng bằng cách ép kiểu sang double. Nếu người dùng nhập đúng format thì sẽ trả về true, ngược lại sẽ trả về false.
- Hàm bool KiemTraRangBuocGioi (double[] tmp):

```
1 reference
public bool KiemTraRangBuocGioi(double[] tmp)
{
    foreach(var item in tmp)
    {
        if(item<6.5)
        {
            return false;
        }
    }
    return true;
}
```

- Dùng để kiểm tra ràng buộc không có môn nào dưới 6.5. Các hàm kiểm tra ràng buộc khác cũng có cấu trúc tương tự như hàm: KiemTraRangBuocKha(double[] tmp), KiemTraRangBuocTrungBinh(double[] tmp),... Nếu ràng buộc được thỏa mãn thì trả về true, ngược lại trả về false.
- **Hàm buttonXacNhan_Click(object sender, EventArgs e)**


```
private void buttonXacNhan_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    string s=textBoxNhap.Text;
    if(CheckFormat(s))
    {
        MessageBox.Show("Đã nhập đúng format!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        string ThôngTinTen = $"Họ và tên sinh viên: {tensinhvien}";

        double diemtrungbinh = DsDiem.Average();
        string ThôngTinDTB = $"Điểm trung bình: {diemtrungbinh}";
        double diemcaonhat = DsDiem.Max();
        string ThôngTinDiemCao = $"Điểm cao nhất: {diemcaonhat}";
        double diemthapnhat = DsDiem.Min();
        string ThôngTinDiemThap = $"Điểm thấp nhất: {diemthapnhat}";
        int SoMonDat = 0;
        int SoMonKhongDat = 0;
        for(int i=0;i<DsDiem.Length;i++)
        {
            if (DsDiem[i] >=5.0)
            {
                SoMonDat++;
            }
            else
            {
                SoMonKhongDat++;
            }
        }
        string ThôngTinSoMonDat=$"Số môn đạt: {SoMonDat}";
        string ThôngTinSoMonKhongDat= $"Số môn không đạt: {SoMonKhongDat}";
        //
    }
}
```

```

string ThôngTinXepLoai = "";
if(diemtrungbinh>=8&&KiemTraRangBuocGioi(DsDiem))
{
    ThôngTinXepLoai = $"Sinh viên: {tensinhvien} xếp loại giỏi";
}
else if(diemtrungbinh>=6.5&&KiemTraRangBuocKha(DsDiem))
{
    ThôngTinXepLoai = $"Sinh viên: {tensinhvien} xếp loại khá";
}
else if (diemtrungbinh >= 5 && KiemTraRangBuocTB(DsDiem))
{
    ThôngTinXepLoai = $"Sinh viên: {tensinhvien} xếp loại trung bình";
}
else if (diemtrungbinh >= 3.5 && KiemTraRangBuocYeu(DsDiem))
{
    ThôngTinXepLoai = $"Sinh viên: {tensinhvien} xếp loại yếu";
}
else
{
    ThôngTinXepLoai = $"Sinh viên: {tensinhvien} xếp loại kém";
}

listBox1.Items.Add(ThôngTinTen);
for(int i=0;i<DsDiem.Length;i++)
{
    listBox1.Items.Add($"Môn {i + 1}: {DsDiem[i]}");
}
listBox1.Items.Add(ThôngTinDTB);
listBox1.Items.Add(ThôngTinDiemCao);
listBox1.Items.Add(ThôngTinDiemThap);
listBox1.Items.Add(ThôngTinSoMonDat);
listBox1.Items.Add(ThôngTinSoMonKhongDat);
listBox1.Items.Add(ThôngTinXepLoai);
}

```

- Hàm này được dùng để tính điểm trung bình (bằng phương thức Average()), điểm cao nhất (bằng phương thức Max()) , điểm thấp nhất (bằng phương thức Min()) , tính số môn đạt, không đạt và xác định loại học lực của sinh viên(bằng cấu trúc if-else).
- Sau đó hiển thị các thông tin này lên listbox bằng câu lệnh listBox1.Items.Add().
- **Hàm buttonXoa_Click(object sender, EventArgs e)**

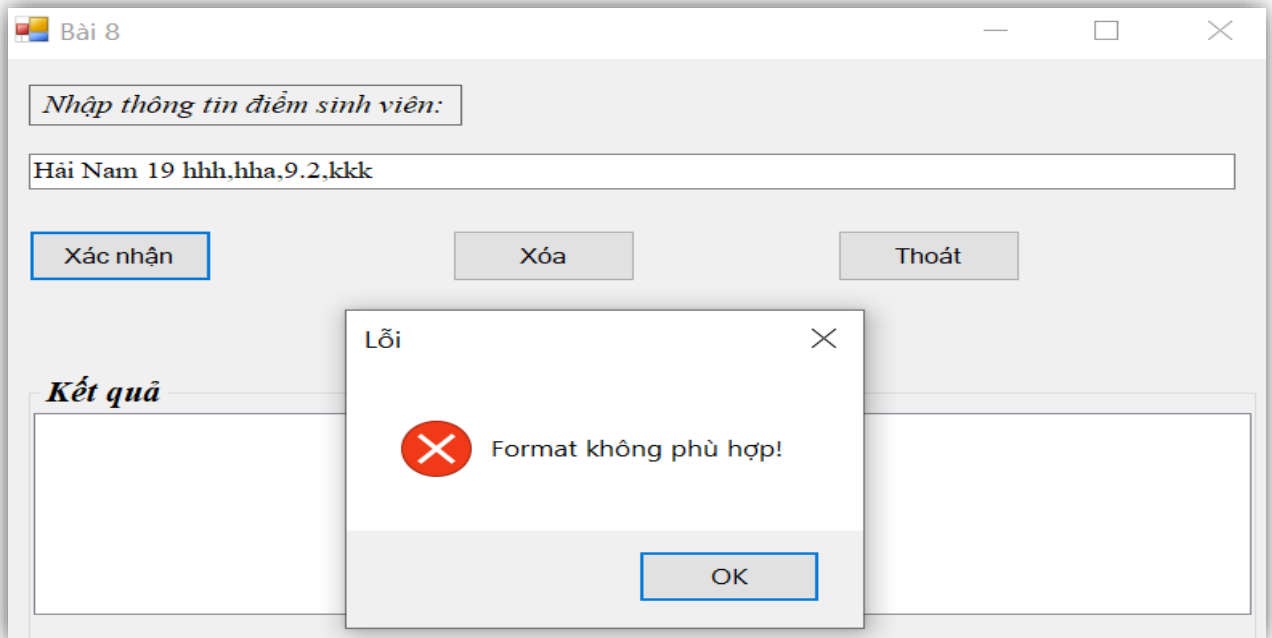
```
1 reference
private void buttonXoa_Click(object sender, EventArgs e)
{
    textBoxNhap.Text = "";
    listBox1.Items.Clear();
}
```

- Hàm này dùng để xóa tất cả các item trong listbox và đặt text trong textBoxNhap thành chuỗi rỗng.

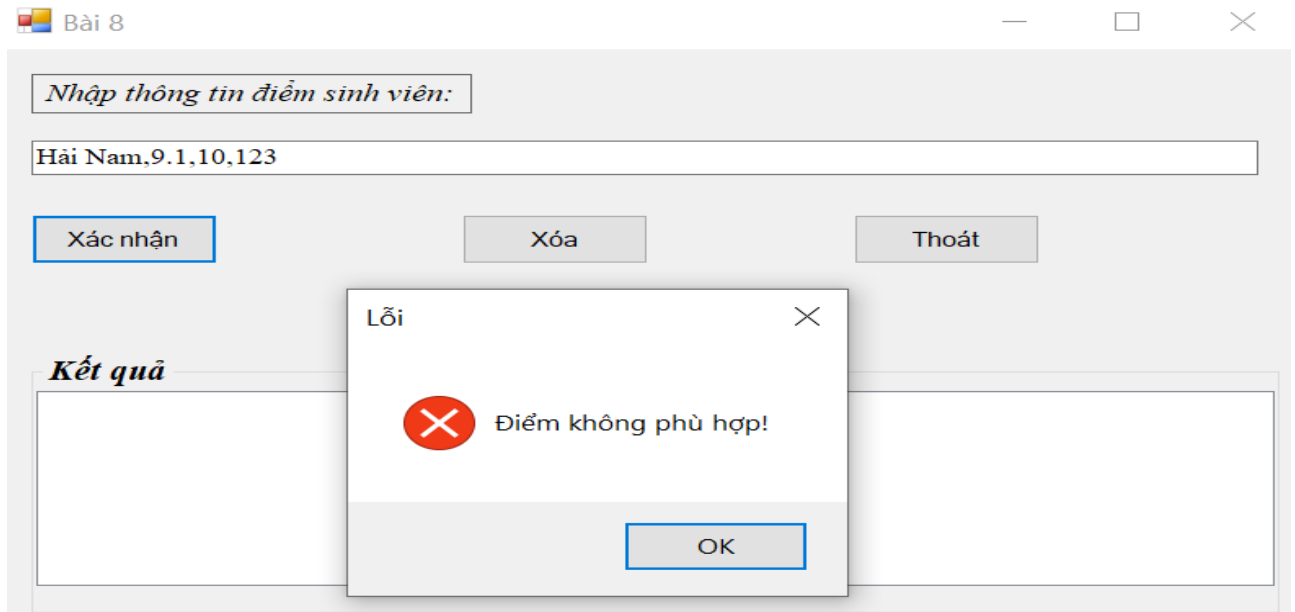
- **Hàm buttonThoat_Click(object sender, EventArgs e)**

```
1 reference
private void buttonThoat_Click(object sender, EventArgs e)
{
    Close();
}
```

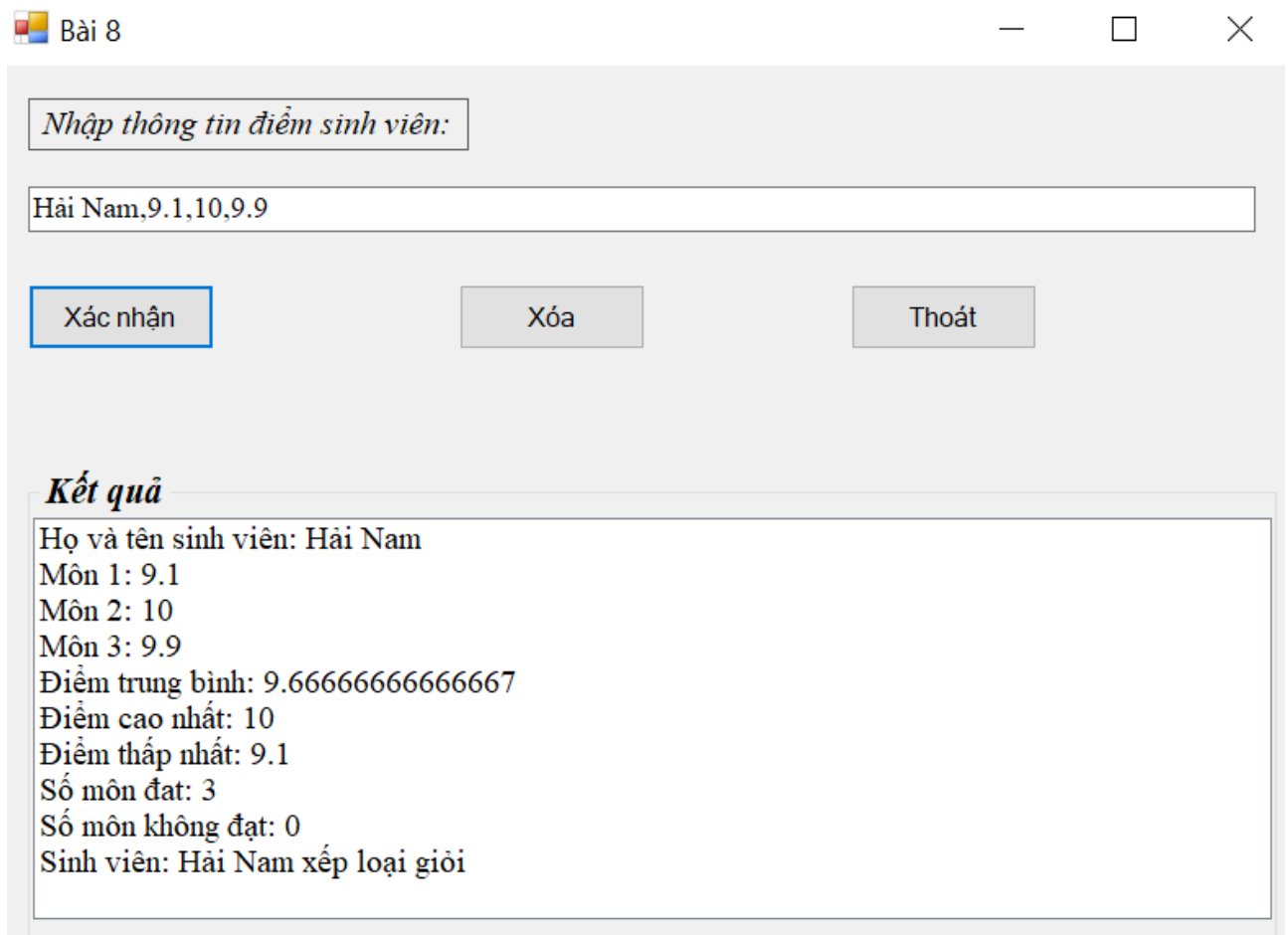
- Dùng để thoát Form hiện tại và quay về giao diện chính.



Khi nhập sai format



Đúng format nhưng điểm không hợp lệ



Khi nhập dữ liệu hợp lệ

9) Bài 08: Hôm nay ăn gì?

GIAO DIỆN:

The screenshot shows a Windows application window titled "Bài 9". Inside the window, there is a text box on the left labeled "Nhập món ăn:" with a button labeled "Thêm" positioned below it. To the right of the text box is a list box containing five items: "Bún Bò", "Bún riêu", "Phở", "Hủ Tiếu", and "Cơm Chiên". At the bottom of the window, there are three buttons: "Tìm món ăn", "Xóa", and "Thoát". Below these buttons is a text box labeled "Món ăn hôm nay là:".

- Giao diện gồm TextBox cho phép người dùng nhập món ăn và một TextBox hiển thị món ăn hôm nay.
- ListBox hiển thị danh sách các món ăn.
- Các button Thêm, Tìm món ăn, Xóa và Thoát Form.

Ý TƯỞNG:

- Ta sẽ dùng Random để sinh ra số ngẫu nhiên từ 0 đến độ dài của các Items trong listbox-1. Đây cũng chính là chỉ số index của các món ăn trong listbox.

CODE:

- Hàm `buttonThem_Click(object sender, EventArgs e)`

```
1 reference
private void buttonThem_Click(object sender, EventArgs e)
{
    string monan = textBoxNhap.Text;
    if(!string.IsNullOrEmpty(monan) )
    {
        listBoxDsMonAn.Items.Add(monan);
        textBoxNhap.Text = "";
    }
}
```

- Hàm này dùng để lấy tên món ăn do người dùng nhập vào đồng thời thêm món ăn đó vào trong listbox.

- **Hàm buttonTim_Click(object sender, EventArgs e)**

```
1 reference
private void buttonTim_Click(object sender, EventArgs e)
{
    Random random = new Random();
    int number=random.Next(0,listBoxDsMonAn.Items.Count);
    textBoxHienThi.Text = listBoxDsMonAn.Items[number].ToString();
}
```

- Hàm này dùng Random để sinh ra một số ngẫu nhiên từ 0 đến listBoxDsMonAn.Items.Count-1.
- Sau đó hiển thị tên món ăn này lên textBoxHienThi.

- **Hàm buttonXoa_Click(object sender,EventArgs e)**

```
1 reference
private void buttonXoa_Click(object sender, EventArgs e)
{
    textBoxHienThi.Text = "";
    textBoxNhap.Text = "";
}
```

- Hàm này được dùng để đặt lại Text trong các ô textBoxHienThi và textBoxNhap thành chuỗi rỗng.
- Hàm `buttonThoat_Click(object sender, EventArgs e)`

```
1 reference
private void buttonThoat_Click(object sender, EventArgs e)
{
    Close();
}
```

- Hàm này dùng để đóng Form hiện tại và quay về giao diện chính.

Bài 9

Nhập món ăn:

Thêm

- Bún Bò
- Bún riêu
- Phở
- Hủ Tiếu
- Cơm Chiên
- Mì Xào

Tìm món ăn **Xóa** **Thoát**

Món ăn hôm nay là:

Random một món ăn

YÊU CẦU CHUNG

1) Đánh giá

- Chuẩn bị tốt các yêu cầu đặt ra trong bài thực hành.
- Sinh viên hiểu và tự thực hiện được bài thực hành, trả lời đầy đủ các yêu cầu đặt ra.
- Nộp báo cáo kết quả chi tiết những đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (*nếu có*); giải thích cho quan sát (*nếu có*).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

2) Báo cáo

- File **.PDF** hoặc **.docx**. Tập trung vào nội dung, giải thích.
- Nội dung trình bày bằng Font chữ **Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Avo)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.**
- Đặt tên theo định dạng: LabX_MSSV1. (trong đó X là Thứ tự buổi Thực hành).
Ví dụ: Lab01_21520001
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT