

MỤC LỤC

BÀI 1. TỔNG QUAN NGÔN NGỮ LẬP TRÌNH C#.....	2
BÀI 2. MẢNG, MẢNG CHUỖI	16
BÀI 3. GIAO DIỆN.....	25
BÀI 4. GIAO DIỆN (tiếp theo)	36
BÀI 5. GIAO DIỆN (tiếp theo)	45
BÀI 6. GIAO DIỆN (tiếp theo)	58
BÀI 7. LẬP TRÌNH TƯƠNG TÁC CƠ SỞ DỮ LIỆU	67
BÀI 8. LẬP TRÌNH TƯƠNG TÁC CƠ SỞ DỮ LIỆU (tiếp theo).....	75
BÀI 9. LẬP TRÌNH TƯƠNG TÁC CƠ SỞ DỮ LIỆU (tiếp theo).....	88
BÀI 10. LẬP TRÌNH TƯƠNG TÁC CƠ SỞ DỮ LIỆU (tiếp theo).....	97
BÀI 11. LẬP TRÌNH TƯƠNG TÁC CƠ SỞ DỮ LIỆU (tiếp theo).....	107
BÀI 12. CRYSTAL REPORT.....	115
BÀI 13. LẬP TRÌNH THEO DỰ ÁN	126
BÀI 14. LẬP TRÌNH THEO DỰ ÁN	129
BÀI 15. LẬP TRÌNH THEO DỰ ÁN	132

A. MỤC TIÊU:

- Xử lý các thao tác cơ bản trong ngôn ngữ C#.
- Sử dụng được các cấu trúc điều khiển trong C#.
- Sử dụng các vòng lặp thực hiện các bài toán cơ bản.
- Xử lý các ngoại lệ phát sinh.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

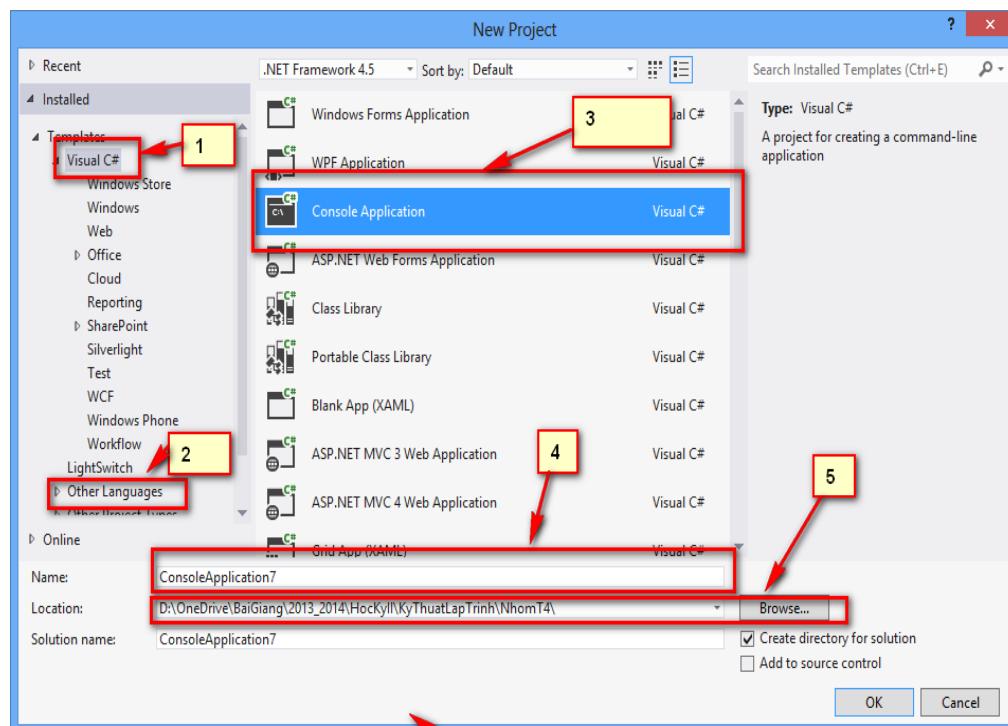
1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

❖ Tạo một project mới

Bước 1: Khởi động Microsoft Visual Studio 2012.

Bước 2: Chọn File → New → Project



Hình 1. Tạo mới project trong Microsoft Visual Studio 2012

Trong đó:

[1][2]: Chọn loại ngôn ngữ phù hợp

[3]: Chọn loại project

[4]: Tên project

[5]: Đường dẫn lưu project

Bước 3:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Demo
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13         }
14     }
15 }
16
```

Hình 2. Cấu trúc một chương trình C#

Trong đó:

[1]: Vùng khai báo các thư viện cần thiết.

[2]: Vùng viết các code xử lý của chương trình.

1.2 Giới thiệu bài tập mẫu

Bài 1: Viết chương trình nhập vào số nguyên, kiểm tra số đã nhập là số âm hay số dương.

Bước 1: Xử lý nhập vào số nguyên.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
```

```

        static void Main(string[] args)
    {
        int n; //Khai báo biến số nguyên n
        string s; //Khai báo biến s kiểu string
        //Thông báo nhập n
        Console.WriteLine("Nhập vào số nguyên n=");
        //Đọc dữ liệu nhập từ bàn phím vào chuỗi s
        s = Console.ReadLine();
        //Chuyển chuỗi s thành số
        n = int.Parse(s);
    }
}

```

Hoặc có thể xử lý ngắn gọn như sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai báo biến số nguyên n
            int n;
            //Thông báo nhập n
            Console.WriteLine("Nhập vào số nguyên n=");
            //Đọc dữ liệu nhập từ bàn phím vào chuỗi s
            //Chuyển chuỗi nhập vào thành số
            n = int.Parse(Console.ReadLine());
        }
    }
}

```

Bước 2: Kiểm tra số đã nhập là số âm hay số dương

```

.....static void Main(string[] args)
{
    .....// Khai báo biến số nguyên n
    int n;
    //B1
    Console.Write("Nhập vào số nguyên n=");
    //B2 đọc dữ liệu nhập từ bàn phím vào chuỗi s
    //B3 Chuyển chuỗi s thành số
    n = int.Parse(Console.ReadLine());
    // Sử dụng cấu trúc điều kiện if
    if (n == 0)
    {
        Console.Write("Không âm không dương");
    }
    if (n > 0)
    {
        Console.Write("Số dương");
    }
    if (n < 0)
    {
        Console.Write("Số âm");
    }
    Console.ReadLine(); 2
}

```

Trong đó:

[1]: Sử dụng cấu trúc điều khiển if.

[2]: Dừng màn hình lại xem kết quả (tương đương getch() trong C++).

Ngoài ra chúng ta có thể sử dụng cấu trúc điều khiển đầy đủ của if như sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            // Khai báo biến số nguyên n
            int n;
            // thông báo nhập n
            Console.Write("Nhập vào số nguyên n=");
            // đọc dữ liệu nhập từ bàn phím vào chuỗi s
        }
    }
}

```

```

//Chuyển chuỗi nhập vào thành số
n = int.Parse(Console.ReadLine());
//Sử dụng cấu trúc điều kiện if
if (n == 0)
{
    Console.WriteLine("Không âm không dương");
}
else
{
    if (n > 0)
    {
        Console.WriteLine("Số dương");
    }
    else
    {
        Console.WriteLine("Số âm");
    }
}
Console.ReadLine();
}
}

```

Hướng dẫn sử dụng try...catch...

Trong một lệnh hoặc đoạn lệnh có khả năng phát sinh lỗi chúng ta nên dùng mệnh đề **try..catch..** để tránh phát sinh lỗi không mong muốn khi đang thực hiện chương trình. Ví dụ khi nhập vào số nguyên n người dùng có thể vô tình nhập một ký tự không là số khi đó lệnh:

```
n = int.Parse(Console.ReadLine());
```

sẽ xảy ra lỗi. Khi đó chúng ta có thể sử dụng mệnh đề **try..catch..** với cú pháp như sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo

```

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            //Khai báo biến số nguyên n  
            int n;  
            //Thông báo nhập n  
            Console.WriteLine("Nhập vào số nguyên n=");  
            //Đọc dữ liệu nhập từ bàn phím vào chuỗi s  
            try  
            {  
                //Chuyển chuỗi nhập vào thành số  
                n = int.Parse(Console.ReadLine());  
            }  
            catch  
            {  
                Console.WriteLine("Bạn nhập n không phải số");  
                return;  
            }  
            // Sử dụng cấu trúc điều kiện if  
            if (n == 0)  
            {  
                Console.WriteLine("Không âm không dương");  
            }  
            else  
            {  
                if (n > 0)  
                {  
                    Console.WriteLine("Số dương");  
                }  
                else  
                {  
                    Console.WriteLine("Số âm");  
                }  
            }  
            Console.ReadLine();  
        }  
    }  
}
```

```

        }
    }
}
```

Nếu người dùng vô tình nhập một ký tự không là số thì những lệnh trong `catch{ }` sẽ được thực thi.

Bài 2: Viết chương trình tính tích: $1*2*3*4*5*....*n$, trong đó n nhập từ phím.

Bộ giá trị kiểm tra:

$n = 1: S_1 = 1;$

$n = 2: S_1 = 2;$

$n = 3: S_1 = 6$

Bước 1: Xử lý nhập vào số nguyên

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai báo biến số nguyên n
            int n;
            //Thông báo nhập n
            Console.Write("Nhập vào số nguyên n=");
            //Đọc dữ liệu nhập từ bàn phím vào chuỗi s
            //Chuyển chuỗi nhập vào thành số
            n = int.Parse(Console.ReadLine());
        }
    }
}
```

Bước 2: Xác định biểu thức 1 (bt1), biểu thức 2 (bt2), biểu thức 3 (bt3) cho vòng lặp for. Công việc cần làm là gì?

bt1 là: `int i=1; // khởi tạo gán giá trị`

bt2 là: `i<=n; // điều kiện dừng`

bt3 là: `i++; // tăng bước nhảy`

Công việc cần làm là tính tích giá trị từ 1 đến n. Khai báo khởi tạo biến giữ giá trị tích.

Bước 3: Code xử lý yêu cầu đề bài:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai báo biến số nguyên n
            int n;
            //Khởi tạo giá trị cho biến tích
            int tich = 1;
            //Thông báo nhập n
            Console.WriteLine("Nhập vào số nguyên n=");
            //Nhập n từ bàn phím
            try
            {
                //Chuyển chuỗi nhập vào thành số
                n = int.Parse(Console.ReadLine());
            }
            catch
            {
                Console.WriteLine("Bạn nhập n không phải số");
                return;
            }
            //Xử lý tính tích
            for (int i = 1; i <= n; i++)
            {
                tich = tich * i;
            }
            //In giá trị ra màn hình theo 1 trong 2 cách
            Console.WriteLine("Tích là: {0}",tich);
        }
    }
}
```

```

        //Console.WriteLine("Tich la: " + tich);
        Console.ReadLine();
    }
}
}

```

Chúng ta có thể chuyển ví dụ trên thành dạng phương thức gọi khi cần. Lưu ý có thể đặt phương thức phía trên hoặc phía dưới void main() mà không cần phải khai báo prototype (nguyên mẫu hàm).

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            // Gọi phương thức BaiMau()
            BaiMau();
            Console.ReadLine();
        }
        // Phương thức BaiMau() được định nghĩa như sau
        static void BaiMau()
        {
            //Khai báo biến số nguyên n
            int n;
            //Khởi tạo giá trị cho biến tích
            int tich = 1;
            // thông báo nhập n
            Console.WriteLine("Nhập vào số nguyên n=");
            //Nhập từ bàn phím
            try
            {
                // Chuyển chuỗi nhập vào thành số

```

```

        n = int.Parse(Console.ReadLine());
    }
    catch
    {
        Console.WriteLine("Bạn nhập n không phải số");
        return;
    }
    // Xử lý tính tích
    for (int i = 1; i <= n; i++)
    {
        tich = tich * i;
    }
    // In giá trị ra màn hình
    Console.WriteLine("Tích la: {0}", tich);
    //Console.WriteLine("Tích la: " + tich);
}
}

```

Bài 3: Viết chương trình tìm ước chung lớn nhất (UCLN) của hai số nguyên a và b.

Bước 1: Xử lý nhập vào 2 số nguyên a và b

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai báo
            int a, b;
            //Nhập a, b
            Console.WriteLine("Mời nhập a: ");
            a = int.Parse(Console.ReadLine());

```

```

        Console.WriteLine("Mời nhập b: ");
        b = int.Parse(Console.ReadLine());
        Console.ReadLine();
    }
}
}

```

Bước 2: Xác định biểu thức 1, biểu thức 2, biểu thức 3 cho bài toán. Công việc cần làm là gì?

Bước 3: Công việc cần làm là tìm UCLN của 2 số a và b, thực hiện như sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            // Khai báo
            int a, b;
            // Nhập a, b
            Console.WriteLine("Mời nhập a: ");
            a = int.Parse(Console.ReadLine());
            Console.WriteLine("Mời nhập b: ");
            b = int.Parse(Console.ReadLine());
            while (a != b) //bt2
            {
                if (a > b)
                {
                    a = a - b; //bt2
                }
                else
                {
                    b = b - a; //bt2
                }
            }
        }
    }
}

```

```

        }
    }

    // In kết quả
    Console.WriteLine("UCLN là: ", a);
    Console.ReadLine();
}

}
}

```

Viết dạng phương thức bài trên chúng ta làm như sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            // Khai báo
            int a, b;
            // Nhập giá trị cho biến a và b
            Console.Write("Mời nhập a: ");
            a = int.Parse(Console.ReadLine());
            Console.Write("Mời nhập b: ");
            b = int.Parse(Console.ReadLine());
            //In kết quả
            Console.WriteLine("UCLN là: ", UCLN(a, b));
            Console.ReadLine();
        }

        static int UCLN(int a, int b)
        {
            while (a != b) //bt2
            {

```

```

        if (a > b)
    {
        a = a - b; //bt2
    }
    else
    {
        b = b - a; //bt2
    }
}
return a;
}
}
}

```

2 . Bài tập tại lớp

Bài 1. Viết chương trình nhập vào đơn giá một mặt hàng, và số lượng bán của mặt hàng. Tính tiền khách phải trả, với thông tin như sau:

- Thành tiền: đơn giá * số lượng
- Giảm giá: Nếu thành tiền > 100, thì giảm 3% thành tiền, ngược lại không giảm
- Tổng tiền phải trả: thành tiền – giảm giá.

Bài 2. Viết chương trình tính tiền điện phải trả trong tháng:

- Từ 01 – 100KW: 5\$
- Từ 101 – 150KW: 7\$
- Từ 151 – 200KW: 10\$
- Từ 201 – 300KW: 15\$
- Từ 301KW trở lên: 20\$

Ví dụ: Nếu điện tiêu thụ 50KW thì tiền điện phải trả là: $50 \times 5 = 250$ \$.

Nếu điện tiêu thụ 101KW thì tiền điện phải trả là: $100 \times 5 + 1 \times 7 = 507$ \$.

Bài 3. Viết chương trình in bảng cửu chương từ 1 đến 9 theo hàng dọc.

Bài 4. Viết chương trình in bảng cửu chương từ 1 đến 9 theo hàng ngang.

Bài 5. Viết chương trình vẽ hình chữ nhật có kích thước d x r, trong đó d là chiều dài, và r là chiều rộng được nhập từ phím.

```

*   *   *   *   *
*   *   *   *   *
*   *   *   *   *

```

Bài 6. Viết chương trình vẽ hình chữ nhật có kích thước $d \times r$, trong đó d là chiều dài, và r là chiều rộng được nhập từ phím.

```
* * * * *\n*           *\n* * * * *
```

Bài 7. Viết chương trình tính tổng: $1^2 + 2^2 + \dots + n^2$ với $n > 0$ nhập từ bàn phím.

Bộ giá trị kiểm tra:

$$n = 1: S_1 = 1; \quad n = 2: S_1 = 5; \quad n = 3: S_1 = 14$$

Bài 8. Kiểm tra n có phải là số nguyên tố hay không?

Bài 9. Liệt kê các số nguyên tố $< n$

Bài 10. Nhập dãy số nguyên (nhập số 0 thì dừng)

- Tìm tổng dãy số vừa nhập
- Tìm max của dãy số đó
- Tìm min của dãy số đó

3 . Bài tập nâng cao

Bài 1. Viết chương trình hỗ trợ cách giải phương trình bậc 2 ($ax^2 + bx + c = 0$).

Bài 2. In tam giác cân rỗng.

4 . Bài tập về nhà

Bài 1. Tính giá trị hàm

a. $f(x) = \begin{cases} 3x + \sqrt{x} & , x > 0 \\ e^x + 4 & , x \leq 0 \end{cases}$

b. $f(x) = \begin{cases} \sqrt{x^2 + 1} & , x \geq 1 \\ 3x + 5 & , -1 < x < 1 \\ x^2 + 2x - 1 & , x \leq -1 \end{cases}$

Gợi ý dùng thư viện `Math.Sqrt`, `Math.Pow`, `Math.E`

Bài 2. Viết chương trình xuất số có 2 chữ số sao cho các chữ số khác nhau đôi một.

Ví dụ: danh sách các số thỏa yêu cầu:

Danh sach cac so co 2 chu so doi mot khac nhau la: 10, 12, 13, ..., 20, 21, 23, ..., 98

Bài 3. Viết chương trình in ra tất cả các số lẻ nhỏ hơn n , trong đó n nhập từ bàn phím.

Bài 4. Viết chương trình đếm số lượng số chẵn trong $[n, m]$, trong đó n, m nhập từ bàn phím.

Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: CNPM Môn: Lập trình .NET	BÀI 2. MẢNG, MẢNG CHUỖI	
--	--	--

A. MỤC TIÊU:

- Sử dụng cấu trúc mảng một chiều thực hiện các bài toán cơ bản.
- Biết cách xử lý dùng mảng một chiều trong C#.
- Sử dụng cấu trúc mảng hai chiều thực hiện các bài toán cơ bản.
- Biết cách xử lý dùng mảng hai chiều trong C#.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

a. Mảng một chiều:

Cú pháp khai báo:

`<Kiểu dữ liệu>[] <Tên_mảng>;`

Ví dụ: khai báo mảng số nguyên A (không xác định số phần tử tối đa)

`int[] A;`

Khởi tạo mảng A bằng từ khóa “`new`”. Ưu điểm là tiết kiệm vùng nhớ vì chỉ khai báo đủ nhu cầu sử dụng.

`A = new int[n];`

b. Mảng hai chiều:

1[0,0]	2[0,1]	4[0,2]	9[0,3]	3[0,4]	...[0,...]	m (cột) [0,m]
6[0,1]	4	3	6
...
n (dòng) [0,n]

Cú pháp khai báo:

`<Kiểu dữ liệu>[][] <Tên_mảng>;`

Ví dụ: khai báo mảng 2 chiều số nguyên A (không xác định số phần tử tối đa)

```
int[][] A;
```

Khởi tạo mảng A bằng từ khóa “new” với n dòng và m là cột:

```
A = new int[n][m];
```

c. Chuỗi

Khái niệm: *Chuỗi là mảng một chiều chứa các ký tự.*

Ví dụ:

‘N’	‘g’	‘u’	‘y’	‘e’	‘n’
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Cú pháp khai báo:

```
string s;
```

Một số phương thức thao tác với chuỗi:

Phương thức	Ý nghĩa
Length	Chiều dài chuỗi
IndexOf (s1)	Vị trí xuất hiện đầu tiên của chuỗi s1 trong chuỗi ban đầu
LastIndexOf (s1)	Giống IndexOf, nhưng tìm lần xuất hiện cuối cùng
Replace (s1, s2)	Thay thế chuỗi s1 thành s2 trong chuỗi ban đầu
Split (<ký tự ngắt>)	Tách chuỗi thành mảng chuỗi bởi <ký tự ngắt>
Substring (vt, l)	Trả về chuỗi con bắt đầu ở một vị trí vt và chiều dài l.
ToLower ()	Chuyển chuỗi thành chữ thường
ToUpper ()	Chuyển chuỗi thành chữ in
Trim ()	Xóa khoảng trắng ở đầu và cuối chuỗi

1.2 Giới thiệu bài tập mẫu

Bài 1: Viết phương thức nhập mảng a gồm n phần tử, xuất mảng a.

Công việc 1: Phương thức xử lý nhập vào mảng a.

Phương thức có thể có hoặc không có giá trị trả về, tham số truyền vào là mảng 1 chiều a và số phần tử mảng.

Bước 1: Cấu trúc phương thức nhập mảng 1 chiều A:

```
static int[] NhapMang(int[] A, int n) {
    // Xử lý
    return A; }
```

Bước 2: Viết code xử lý công việc nhập mảng

```
static int[] NhapMang(int[] A, int n)
{
    // Xử lý
    for (int i = 0; i < n; i++)
    {
        Console.WriteLine("Mời nhập phần tử A[{0}]=", i);
        A[i] = int.Parse(Console.ReadLine());
    }
    return A;
}
```

Công việc 2: Phương thức xử lý xuất mảng A.

Bước 1: Cấu trúc phương thức xuất mảng 1 chiều A

```
static void XuatMang(int[] A)
{
    // Xử lý
}
```

Bước 2: Viết code xử lý công việc xuất mảng

```
static void XuatMang(int[] A)
{
    // Xử lý
    for (int i = 0; i < A.Length; i++)
    {
        Console.WriteLine("\t {0}", A[i]);
    }
}
```

Viết phương thức tổng quát gọi kiểm tra kết quả 2 phương thức trên:

```
public static void Mang1Chieu()
{
    // Khai báo mảng A
    int[] A;
    Console.WriteLine("Mời bạn nhập vào số phần tử: ");
    // Nhập n
    int n = int.Parse(Console.ReadLine());
    // Khởi tạo mảng A đúng n phần tử
```

```

A = new int[n];
//Gọi phương thức nhập mảng
A = NhapMang(A,n);
//Gọi phương thức xuất mảng
XuatMang(A);
}

```

Một số kỹ thuật xử lý mảng 1 chiều:

- **Sắp xếp:**

```

static void SapXep(int[] A)
{
    // Sắp xếp tăng
    Array.Sort(A);
}

```

- **Đảo mảng:** `Array.Reverse();`
- **Thay đổi kích thước mảng:** `Array.Resize();`
- **Chuyển mảng từ kiểu string sang mảng kiểu int:**
`string[] B= new[] {"1","2"};`
`int[] intArray = Array.ConvertAll(B, int.Parse);`

Ngoài ra mảng còn hỗ trợ trực tiếp các phương thức static thực hiện các công việc:

- Tính tổng các phần tử trong mảng: `A.Sum();`
- Tính trung bình các phần tử trong mảng: `A.Average();`
- Đếm số phần tử có trong mảng: `A.Count();`
- Tìm phần tử nhỏ nhất trong mảng: `A.Min();`
- Tìm phần tử lớn nhất trong mảng `A.Max();`
- Tính tổng phần tử trong mảng theo điều kiện:
`A.Where(t => t % 2 == 0).Sum();`

Bài 2: Viết phương thức nhập mảng 2 chiều a gồm n phần tử, xuất mảng 2 chiều.

Công việc 1: Phương thức xử lý nhập vào mảng 2 chiều a.

Phương thức có thể có hoặc không có giá trị trả về, tham số truyền vào là mảng 2 chiều a và số phần tử mảng.

Bước 1: Cấu trúc phương thức nhập mảng 2 chiều

```

static void NhapMang2Chieu(int[][] A, int n, int m) {
    // Xử lý
}

```

Bước 2: Viết code xử lý công việc nhập mảng

```
static void NhapMang2Chieu(int[][] A, int n, int m)
{
    // Xử lý
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            Console.WriteLine("Nhập phần tử A[{0}][{1}] = ", i, j);
            A[i][j] = int.Parse(Console.ReadLine());
        }
    }
}
```

Công việc 2: Phương thức xử lý xuất mảng a.

Bước 1: Cấu trúc phương thức xuất mảng

```
static void XuatMang2Chieu(int[][] A)
{
    // Xử lý
}
```

Bước 2: Viết code xử lý công việc xuất mảng

```
static void XuatMang2Chieu(int[][] A)
{
    // Xử lý
    for (int i = 0; i < A.GetLength(0); i++)
    {
        for (int j = 0; j < A.GetLength(1); j++)
        {
            Console.Write("{0} ", A[i][j]);
        }
        Console.WriteLine();
    }
}
```

Chúng ta có thể thay thế n, m bằng phương thức `GetLength()`.

Viết phương thức tổng quát gọi kiểm tra kết quả 2 phương thức trên

```
static void Mang2Chieu()
{
    int n, m;
    int[][] A;
    Console.Write("Mời bạn nhập số dòng: ");
    n = int.Parse(Console.ReadLine());
    Console.Write("Mời bạn nhập số cột: ");
    m = int.Parse(Console.ReadLine());
    // Khởi tạo
    A = new int[n][];
    for (int i = 0; i < n; i++)
    {
        A[i] = new int[m];
    }
    NhapMang2Chieu(A, n, m);
    XuatMang2Chieu(A);
}
```

✚ Một số kỹ thuật xử lý mảng 2 chiều:

☞ Lưu ý: Tận dụng các kỹ thuật của mảng 1 chiều. Vì mảng 2 chiều là sự kết hợp của nhiều mảng 1 chiều.

Ví dụ: Tổng các số dương trong mảng 2 chiều:

```
static int TongDuong2Chieu(int[][] A)
{
    int tong = 0;
    for (int i = 0; i < A.GetLength(0); i++)
    {
        tong = A[i].Where(t => t > 0).Sum();
    }
    return tong;
}
```

Bài 3: Nhập chuỗi S. Kiểm tra S có là chuỗi đối xứng không ?

Bước 1: Nhập chuỗi S.

```
static void Main(string[] args)
{
```

```

        string S;
        S = Console.ReadLine();
        Console.ReadLine();
    }

```

Bước 2: Chuyển chuỗi S thành mảng chuỗi.

```

static void Main(string[] args)
{
    string S;
    S = Console.ReadLine();
    char[] mangChuoi = S.ToCharArray();
    Console.ReadLine();
}

```

Bước 3: Viết phương thức kiểm tra đối xứng

```

static bool ChuoiDoiXung(char[] A)
{
    for (int i = 0; i < A.Length; i++)
    {
        if (A[i] != A[A.Length - 1 - i])
            return false;
    }
    return true;
}

```

Bước 4: Hàm Main gọi thực thi các phương thức trên

```

static void Main(string[] args)
{
    string S;
    S = Console.ReadLine();
    char[] mangChuoi = S.ToCharArray();
    if (ChuoiDoiXung(mangChuoi))
    {
        Console.WriteLine("Chuoi doi xung");
    }
    else
    {
        Console.WriteLine("Chuoi khong doi xung");
    }
}

```

```

    }
    Console.ReadLine();
}

```

2. Bài tập tại lớp

Bài 1. Mảng một chiều

Cho mảng 1 chiều chứa các số nguyên. Viết phương thức thực hiện:

- Nhập mảng a gồm n phần tử. Xuất mảng a.
- Liệt kê các phần tử lẻ ở vị trí chẵn.
- Liệt kê các số nguyên tố trong mảng.
- Tìm phần tử âm đầu tiên trong mảng.
- Tìm max, min của dãy.
- Tính tổng các phần tử trong mảng.

Bài 2. Mảng hai chiều

Cho một ma trận nguyên kích thước $m \times n$. Viết phương thức thực hiện:

- Nhập các giá trị cho mảng 2 chiều, sau đó xuất mảng vừa nhập.
- Tổng tất cả các phần tử dương của ma trận.
- Tổng các phần tử trên đường chéo chính.
- Tổng các phần tử trong tam giác trên.
- Tổng tất cả các phần tử chẵn của ma trận.
- Tính tổng các phần tử ở dòng thứ i.

Bài 3. Nhập chuỗi s (là họ tên). Viết phương thức thực hiện các yêu cầu sau:

- S có phải là chuỗi đối xứng không?
- Đổi ký tự đầu từ thành chữ hoa
- Đổi ký tự thường thành chuỗi hoa và ngược lại
- Có bao nhiêu nguyên âm, phụ âm, khoảng trắng trong chuỗi s.

3. Bài tập nâng cao

Bài 1. Thực hiện phép cộng 2 đa thức trên mảng một chiều. Với đa thức cho trước như sau: DaThuc {int Bac, int ChiSoDaThuc[]};

Bài 2. Cho một ma trận nguyên kích thước $m \times n$. Viết phương thức tìm:

- Số nhỏ nhất, lớn nhất (kèm chỉ số) của ma trận.
- Số nhỏ nhất, lớn nhất (kèm chỉ số) của từng hàng của ma trận.
- Số nhỏ nhất, lớn nhất (kèm chỉ số) của đường chéo chính của ma trận.
- Viết phương thức tìm giá trị phần tử xuất hiện nhiều nhất trong dòng i, nếu có nhiều phần tử có số lần xuất hiện bằng nhau thì xuất phần tử đầu tiên.

Bài 3. Nhập chuỗi s (là họ tên). Viết phương thức thực hiện các yêu cầu sau:

- Xóa khoảng trắng thừa trong chuỗi s.
- Có bao nhiêu từ trong chuỗi s
- Lấy ra họ
- Lấy ra tên
- Lấy ra họ lót (nếu có)

4. Bài tập về nhà

Bài 1. Viết phương thức tính tổng các phần tử cực đại trong mảng các số nguyên (phần tử cực đại là phần tử lớn hơn các phần tử xung quanh nó).

Ví dụ: 1 **5** 2 **6** 3 **5** 1 **8** 6

Bài 2. Cho ma trận vuông cấp n chứa các số nguyên. Viết phương thức thực hiện các yêu cầu sau:

- Liệt đường chéo chính, phụ.
- Liệt kê tam giác trên, dưới.
- Tổng đường chéo chính, phụ.
- Tổng tam giác trên, dưới.

Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: CNPM Môn: Lập trình .NET	BÀI 3. GIAO DIỆN	
--	-----------------------------------	--

A. MỤC TIÊU:

- Thiết kế được giao diện Windows Form từ các control trong C#.
- Sử dụng được các Controls Textbox, Label, Button.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1. Kiến thức cần nhớ

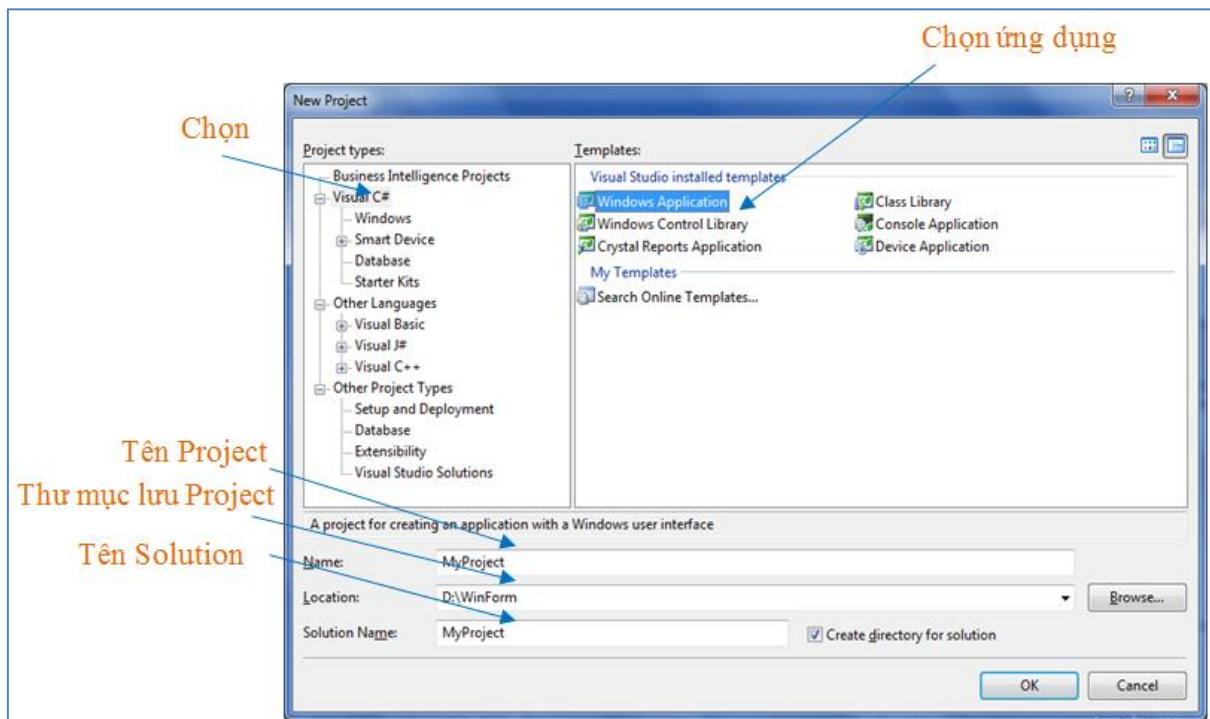
Mỗi Solution hoặc Project có các loại tập tin sau:

File Name	Diễn giải
MyName.sln	Đây là file solution (file text) lưu trữ thông tin về solution và project. Nếu project đã có sẵn thì ta sẽ mở file này để tiếp tục project.
frmMyName.cs	Lưu trữ code trong form.
frmMyName.Designer.cs	Lưu trữ code design trong form.
frmMyName.rexs	Đây là file resource, định nghĩa tất cả system resources sử dụng trong form.
MyName.csproj	File project này mô tả thông tin project và danh sách các file có trong project.
Program.cs	Là tập tin tự phát sinh khi tạo project và cũng là tập tin bắt đầu một ứng dụng.
Các files khác	Sau khi thực thi một project thì một số file khác sẽ tự động phát sinh trong thư mục solution.

Tạo một ứng dụng Windows Form

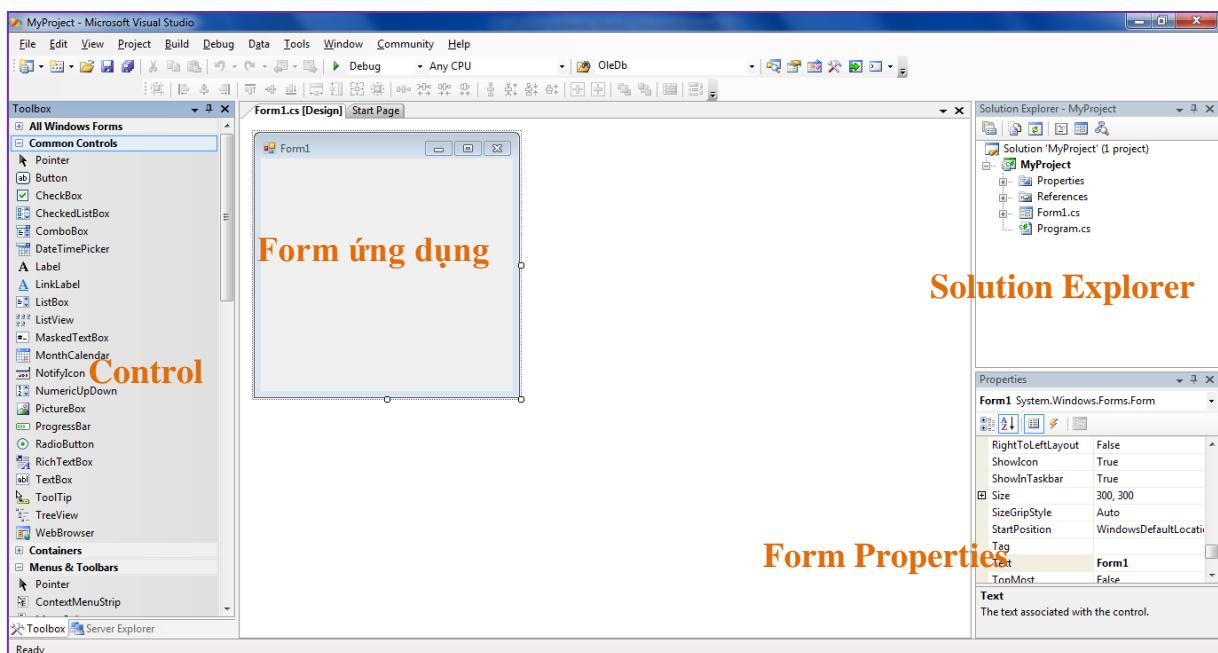
Khởi động chương trình: Start → All program → Microsoft Visual Studio 2012 → Microsoft Visual Studio 2012.

Tạo một project mới: File → New → Project



Hình 3. Cửa sổ tạo projec mới

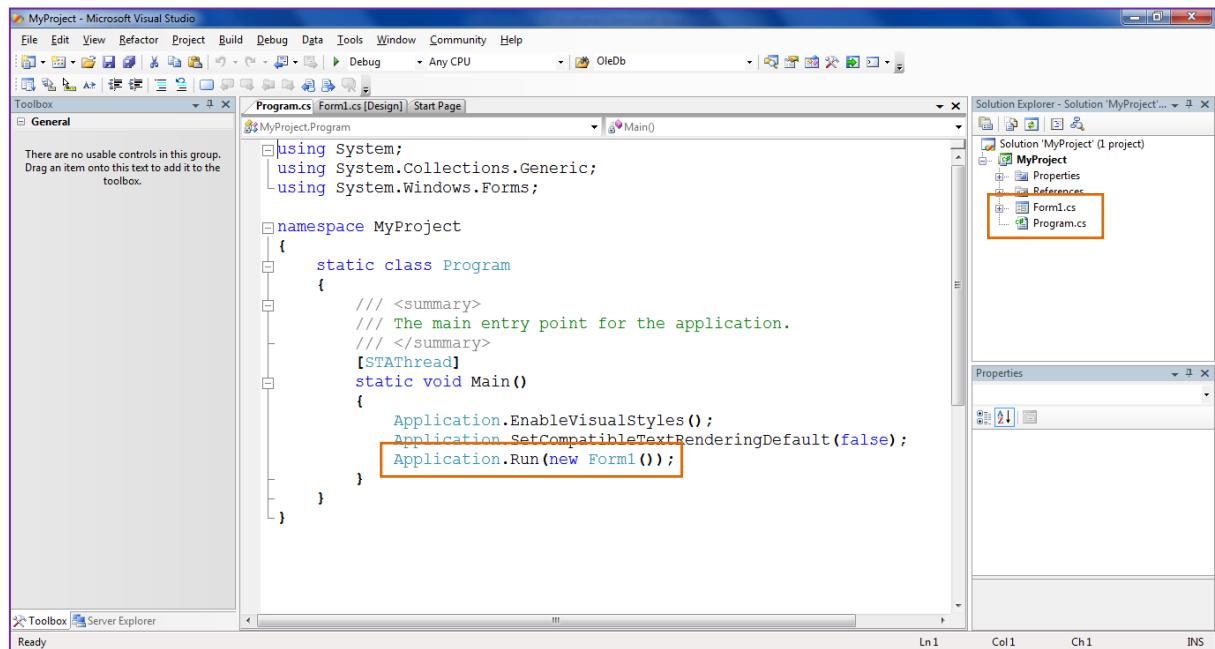
Sau khi chọn ngôn ngữ, ứng dụng, đặt tên cho project, ... click OK, cửa sổ thiết kế giao diện (Form Design) sẽ hiện ra:



Hình 4. Cửa sổ thiết kế giao diện

☞ Lưu ý:

- Trong cùng một project có thể có nhiều form, muốn chọn form khởi động đầu tiên ta mở file Program.cs và thay tên form cần mở trong lệnh `Application.Run(new Ten_form());`

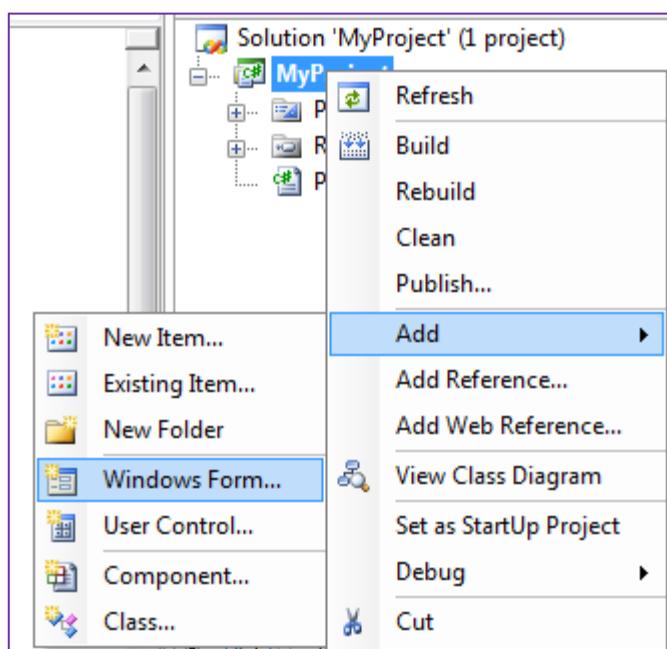


Hình 5. Tập tin Program.cs

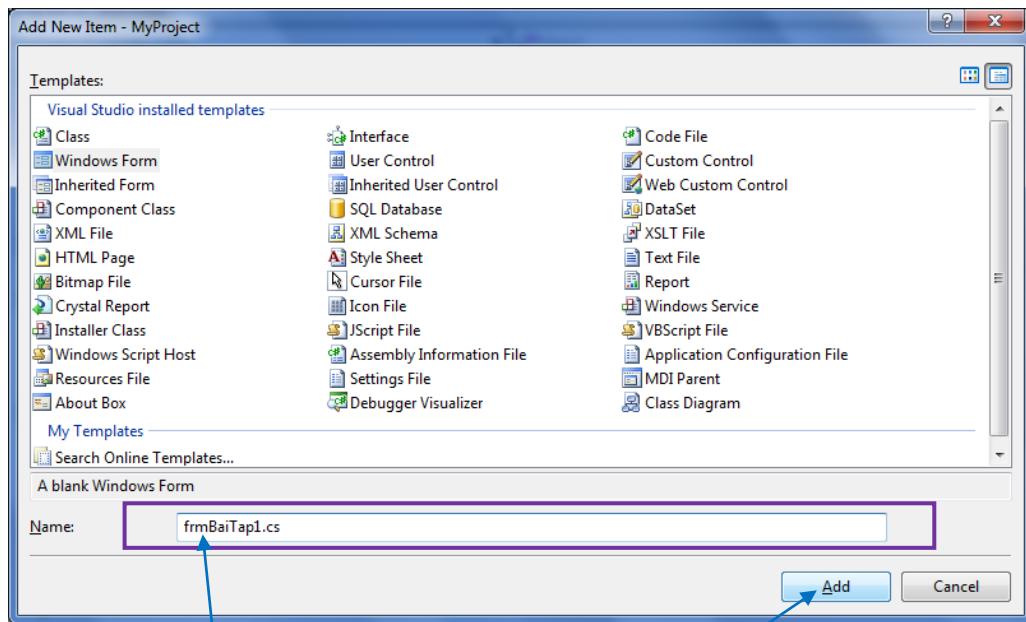
- Trước khi làm bài tập thì nên xóa Form1 (form được tạo mặc định), tạo Form mới và đặt tên theo yêu cầu. Cách thực hiện như sau:

Trong cửa sổ Solution Explorer, Click chuột phải vào Form1 → Delete

Sau đó Click chuột phải trên MyProject → Add → Windows Form...



Cửa sổ Add New Item xuất hiện, đặt tên cho form mới và chọn Add.



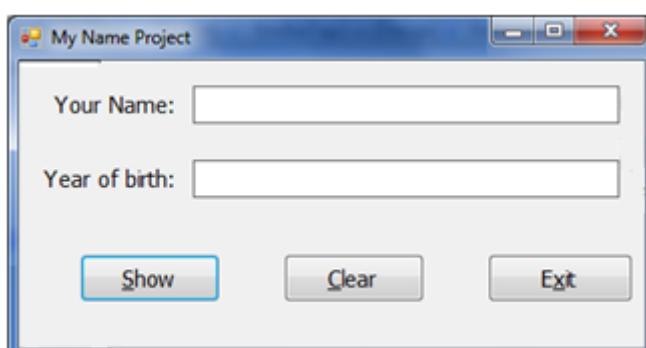
Đặt tên Form

Click Add Form

Hình 6. Cửa sổ Add New Item vào My Project

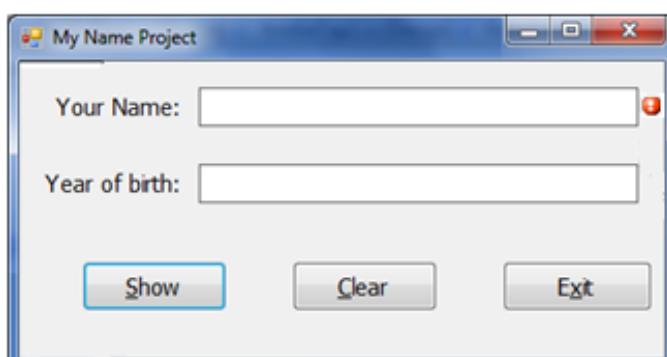
1.2. Giới thiệu bài tập mẫu

Thiết kế form sau:



Yêu cầu:

- Khi Form khởi động thực hiện nhập tên, năm sinh vào TextBox YourName và Year of Birth tương ứng. Nếu YourName không được nhập dữ liệu, năm sinh không phải là số thì thông báo lỗi (dùng **ErrorProvider**).
- **ErrorProvider**: Cung cấp thông tin lỗi của điều khiển trên Form, thường được dùng với control input (ví dụ: TextBox) ràng buộc với một điều kiện nhập.



- Khi người dùng nhấn button Show sẽ hiển thị thông tin nhập vào Messagebox bao gồm: tên, tuổi (năm hiện tại – năm sinh)



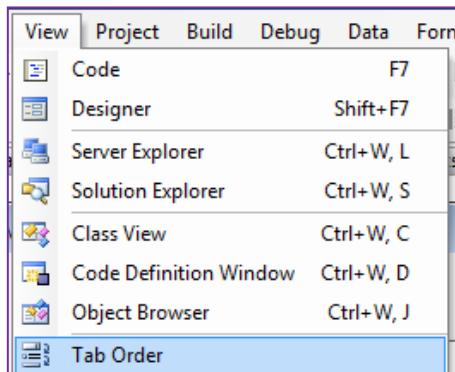
- Khi người dùng nhấn button Clear sẽ xóa thông tin đã nhập trên các TextBox, đặt con trỏ văn bản vào Textbox YourName.
- Khi nhấn button Exit xác nhận người dùng có thực sự muốn thoát khỏi chương trình không? Yes: thoát, No: không.

Hướng dẫn:

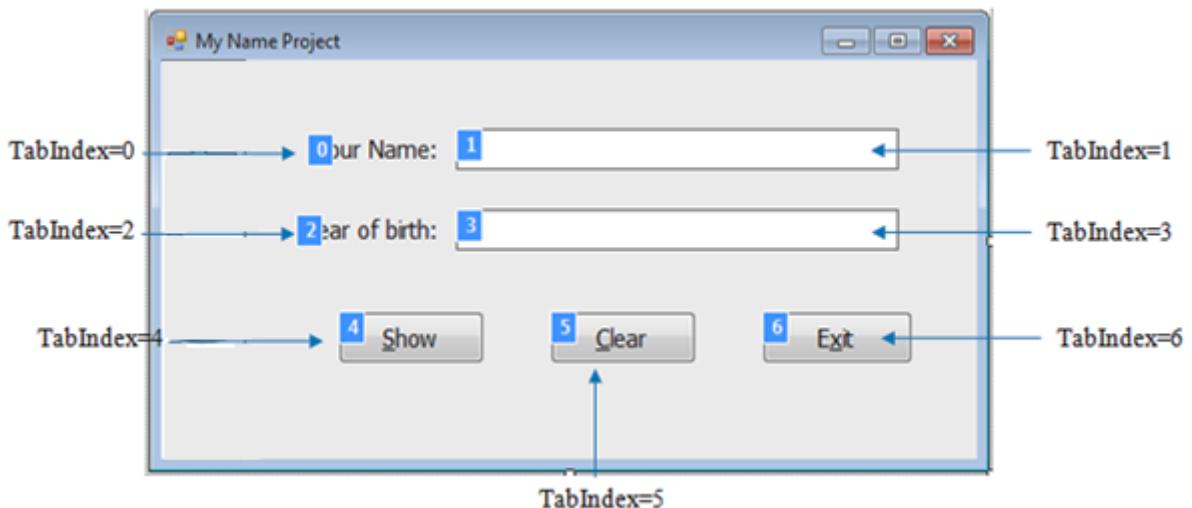
- **Danh sách các thuộc tính của các đối tượng (Object) cần thiết:**

Object	Properties	Events
frmBaiTap1	Name: frmBaiTap1 Text: My name Project FontName: Tahoma FontSize: 11 AcceptButton: btnShow (nhận sự kiện Click chuột khi nhấn Enter) CancelButton: btnExit (nhận sự kiện Click chuột khi nhấn Esc)	FormClosing
txtYourName	Name: txtYourName BorderStyle: FixSingle	Leave (mất tiêu điểm)
txtYear	Name: txtYear BorderStyle: FixSingle	TextChanged
btnShow	Name: btnShow Text: &Show	Click
btnClear	Name: btnClear Text: &Clear	Click
btnExit	Name: btnExit Text: E&xit	Click
errorProvider	Name: errorProvider1	

- Thay đổi thứ tự nhận tiêu điểm trên Form: chọn menu View → Tab Order



Lần lượt thực hiện click chọn từng phần tử trên Form theo thứ tự nhận tiêu điểm:



- Viết các sự kiện tương ứng với các yêu cầu:

Sự kiện đóng xác nhận khi form được đóng:

```
private void frmBT1_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult r;
    r = MessageBox.Show("Bạn có muốn thoát?", "Thoát",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question,
        MessageBoxDefaultButton.Button1);
    if (r == DialogResult.No)
        e.Cancel = true;
}
```

Kiểm tra Textbox YourName đã được nhập nội dung chưa:

```
private void txtYourName_Leave(object sender, EventArgs e)
{
    Control ctr = (Control)sender;
    if (ctr.Text.Trim().Length == 0)
        this.errorProvider1.SetError(ctr, "You must enter Your Name");
```

```

    else
        this.errorProvider1.Clear();
}

```

Yêu cầu Textbox Year phải nhập số, nhập sai dùng ErrorProvider thông báo lỗi:

```

private void txtYear_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, "This is not a valid number");
    else
        this.errorProvider1.Clear();
}

```

Hiển thị thông tin theo yêu cầu khi click vào Button Show:

```

private void btnShow_Click(object sender, EventArgs e)
{
    int age;
    string s;
    s = "My name is: " + txtYourName.Text + "\n";
    age = DateTime.Now.Year - Convert.ToInt32(txtYear.Text);
    s = s + "Age: " + age.ToString();
    MessageBox.Show(s);
}

```

Button Clear thực hiện lệnh xóa các TextBox chuẩn bị cho lần nhập tiếp theo:

```

private void btnShow_Click(object sender, EventArgs e)
{
    //clear Text
    txtYourName.Clear();
    txtYear.Clear();
    //Textbox YourName reset focus
    txtYourName.Focus();
}

```

Thoát chương trình khi click vào Button Exit:

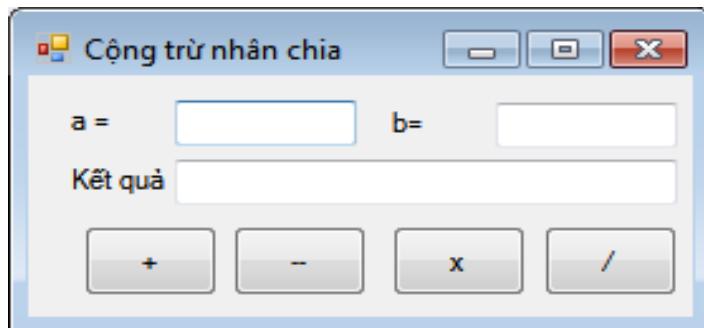
```

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

```

2. Bài tập tại lớp

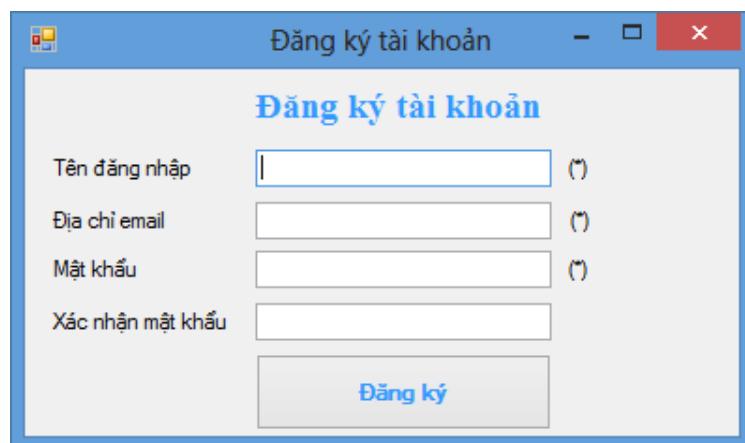
Bài 1. Thiết kế Form thực hiện phép tính +, -, ×, / của hai số a, b với giao diện như sau:



Yêu cầu: Khi chọn các phép toán +, -, *, / tương ứng trên các button thì hiển thị kết quả lên textbox kết quả.

- Kiểm tra tính hợp lệ của các textbox theo hai mức độ:
 - Mức 1: Nếu dữ liệu nhập vào textbox a, b không hợp lệ thì dùng ErrorProvider thông báo lỗi.
 - Mức 2: Chặn không cho người dùng nhập giá trị khác số vào textbox a, b.
- Thông báo lỗi bằng MessageBox nếu dữ liệu nhập không phù hợp
- Hỏi xác nhận trước khi đóng Form.

Bài 2. Thiết kế form đăng ký tài khoản



Yêu cầu:

- Kiểm tra định dạng của email sau khi nhập và ra khỏi textbox Địa chỉ email.
- Bắt buộc nhập dữ liệu trên những textbox có (*) .
- Khi chọn button Đăng ký hoặc nhấn phím Enter sau khi nhập xong dữ liệu vào textbox Xác nhận mật khẩu thì hiển thị thông tin của tất cả các textbox lên MessageBox.
- Hỏi xác nhận trước khi đóng Form.

Bài 3. Thiết kế giao diện thực hiện yêu cầu nhập vào 2 số a, b. Xuất ra UCLN và BCNN của 2 số a và b (kiểm tra dữ liệu nhập và xuất các thông báo lỗi nếu có).

Ước Số Chung - Bội Số Chung

Nhập số a :

Nhập số b :

Ước số chung lớn nhất :

Bội số chung nhỏ nhất :

Thực Hiện Tiếp Tục Thoát

Bài 4. Thiết kế giao diện sau:

Dãy số và Tính Tổng

Nhập số : Nhập

Dãy vừa nhập :

Tổng các phần tử trong dãy :

Tổng Chẵn : Tổng Lẻ :

Tiếp Tục Thoát

Yêu cầu:

- Nhập vào một dãy số nguyên bất kỳ (nhập số nguyên vào textbox Nhập số, nhấn button Nhập), xuất các số vừa nhập ra textbox Dãy vừa nhập.
- Bổ sung thêm button Tính tổng, Tổng chẵn, Tổng lẻ các số vừa nhập và xuất ra các textbox tương ứng.
- Tính tổng các số chẵn và tổng các số lẻ.
- Button Tiếp tục trả lại trạng thái ban đầu của form.
- Button Thoát: xác nhận và đóng form.

Bài 5. Viết chương trình đọc số thành chữ với giao diện sau:



Yêu cầu:

- Nhập vào một số nguyên dương bất kỳ từ 1 đến 999.
- Đọc số vừa nhập thành chữ số khi chọn button Thực hiện.
- Button Xóa: trả form lại trạng thái ban đầu
- Button Thoát: xác nhận và đóng form.

3. Bài tập nâng cao

Bài 1. Thiết kế giao diện sau:



Yêu cầu: Tạo một project giúp cho rạp chiếu phim quản lý việc bán vé của mình. Rạp có 3 hàng ghế, mỗi hàng có 5 ghế, các ghế được đánh số từ 1 đến 15 và được phân thành 3 lô như (hình trên). Giá vé theo lô như sau:

Giá vé lô A 1000/vé

Giá vé lô B 1500/vé

Giá vé lô C 2000/vé

Form giao diện trình bày là sơ đồ các chỗ ngồi tương ứng trong rạp phim để người sử dụng chọn vị trí muốn mua. Trên sơ đồ này cũng thể hiện vị trí vé đã bán và những vị

trí vé chưa bán bằng các thẻ hiện màu khác nhau (vé chưa bán màu trắng, vé đã bán màu vàng, vé đang chọn màu xanh). Khi người sử dụng click chuột tại một vị trí trên sơ đồ thì:

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

Sau khi đã chọn các vị trí, người sử dụng có thể click chuột vào button **CHỌN** để thanh toán hoặc **HỦY BỎ**:

Nếu click vào button **CHỌN** thì:

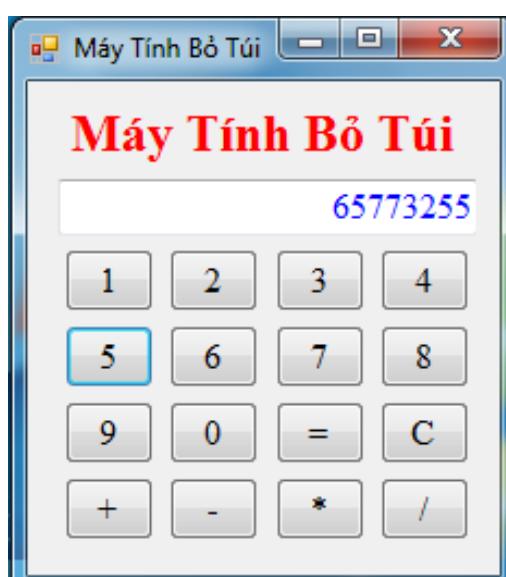
- Đổi màu các vị trí đang chọn (màu xanh) trên sơ đồ sang màu vàng (đây là vị trí đã bán).
- Xuất tổng số tiền phải trả cho số vé đã mua lên label Thành Tiền.

Nếu click vào button **HỦY BỎ** thì:

- Đổi màu các vị trí đang chọn (màu xanh) trên sơ đồ sang màu trắng.
- Xuất lên label Thành Tiền giá trị 0.

4. Bài tập về nhà

Bài 1: Viết chương trình dành cho máy tính bỏ túi đơn giản thực hiện các phép tính cộng, trừ, nhân, chia và xóa đơn giản.





A. MỤC TIÊU:

- Thiết kế được giao diện Windows Form từ các control trong C#.
- Sử dụng được các Controls Textbox, Label, Button, RadioButton, Checkbox.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

Thực hiện tạo project loại WindowsForms Application như bài thực hành 3.

Control được sử dụng trong bài: RadioButton, Checkbox với thuộc tính và sự kiện cần nhớ là:

- + Thuộc tính: **Checked, Text**
- + Sự kiện: **CheckedChanged**

1.2 Giới thiệu bài tập mẫu

Bài 1. Thiết kế giao diện thực hiện tính +, -, x, / của hai số:



Yêu cầu:

- Tạo giao diện như hình: nhập vào 2 số và 4 radio button tương ứng với 4 phép toán.
- Thực hiện các phép toán tương ứng của 2 số khi người dùng click chọn các radio button và xuất kết quả vào textbox Kết quả.
- Nếu dữ liệu nhập vào textbox a, b không hợp lệ, cảnh báo theo hai mức:
 - + Dùng ErrorProvider thông báo lỗi.

- + Chặn không cho nhập giá trị khác số vào textbox.
- Xây dựng class TinhToan (khai báo các thuộc tính, property set/get, phương thức khởi tạo và các phương thức tính toán cần thiết phục vụ cho yêu cầu của giao diện trên).
- Khi click button Tính gọi thực hiện các phương thức tương ứng trong class TinhToan.
- Hỏi xác nhận trước khi đóng Form.

Hướng dẫn:

- Khai báo class TinhToan

```
public class TinhToan
{
    float _a, _b;
    public float b
    {
        get { return _b; }
        set { _b = value; }
    }
    public float a
    {
        get { return _a; }
        set { _a = value; }
    }
    public TinhToan() { a = b = 0; }
    public TinhToan(float a, float b)
    {
        _a = a;
        _b = b;
    }
    public float Cong() { return _a + _b; }
    public float Tru() { return _a - _b; }
    public float Nhan() { return _a * _b; }
    public float Chia() { return _a / _b; }
}
```

- Viết sự kiện click cho button Tính toán

```
private void btn_Tinh_Click(object sender, EventArgs e)
{
```

```

string s = "Kết quả là: \n";
float a = float.Parse(txt_a.Text);
float b = float.Parse(txt_b.Text);
TinhToan dt = new TinhToan(a, b);
if (rdo_cong.Checked)
    MessageBox.Show(s + a + "+" + b + " = " + dt.Cong());
else if (rdo_tru.Checked)
    MessageBox.Show(s + a + "-" + b + " = " + dt.Tru());
else if (rdo_nhan.Checked)
    MessageBox.Show(s + a + "*" + b + " = " + dt.Nhan());
else if (b!=0)
    MessageBox.Show(s + a + "/" + b + " = " + dt.Chia());
else
    MessageBox.Show("Phép chia bị lỗi!");
}

```

Bài 2. Thiết kế giao diện sau:



Yêu cầu:

- Định dạng nội dung của label theo các checkbox và radio button được chọn tương ứng về Font Style và Color.
- Khi chọn button Exit: xác nhận và thoát khỏi Form.

Hướng dẫn:

- Thiết lập thuộc tính màu chữ cho label:

```
Tên_Label.ForeColor = Color.Blue ;
```

- Thiết lập thuộc tính Font Style là in nghiêng:

```
Tên_Label.Font = new Font(TênLabel.Font, FontStyle.Italic);
```

- Thiết lập thuộc tính Font Style vừa nghiêng, vừa đậm:

```
Tên_Label.Font = new Font(TênLabel.Font,
                           (FontStyle)(FontStyle.Bold | FontStyle.Italic));
```

- Sự kiện CheckedChanged của radio button Red

```
private void rdo_red_CheckedChanged(object sender, EventArgs e)
{
    if (rdo_red.Checked == true)
        label1.ForeColor = Color.Red;
}
```

2. Bài tập tại lớp

Bài 1. Thiết kế giao diện hỗ trợ giải phương trình bậc nhất và phương trình bậc hai theo những yêu cầu sau:

- Khai báo class PhuongTrinhBacHai với những thuộc tính và phương thức phục vụ cho yêu cầu của giao diện bên dưới và được gọi thực thi khi cần thiết.
- Khi form được load lên button “Giải” mờ đi.



- Khi chọn radio button Giải phương trình Bậc nhất thì TextBox Nhập c bị mờ/ ẩn đi. Khi nhập thông tin đầy đủ vào Textbox cần thiết thì Chức năng “Giải” có tác dụng để thực thi. Giải phương trình kết quả hiện lên trong Textbox Kết quả, sau đó Button Giải mờ đi như hình sau:

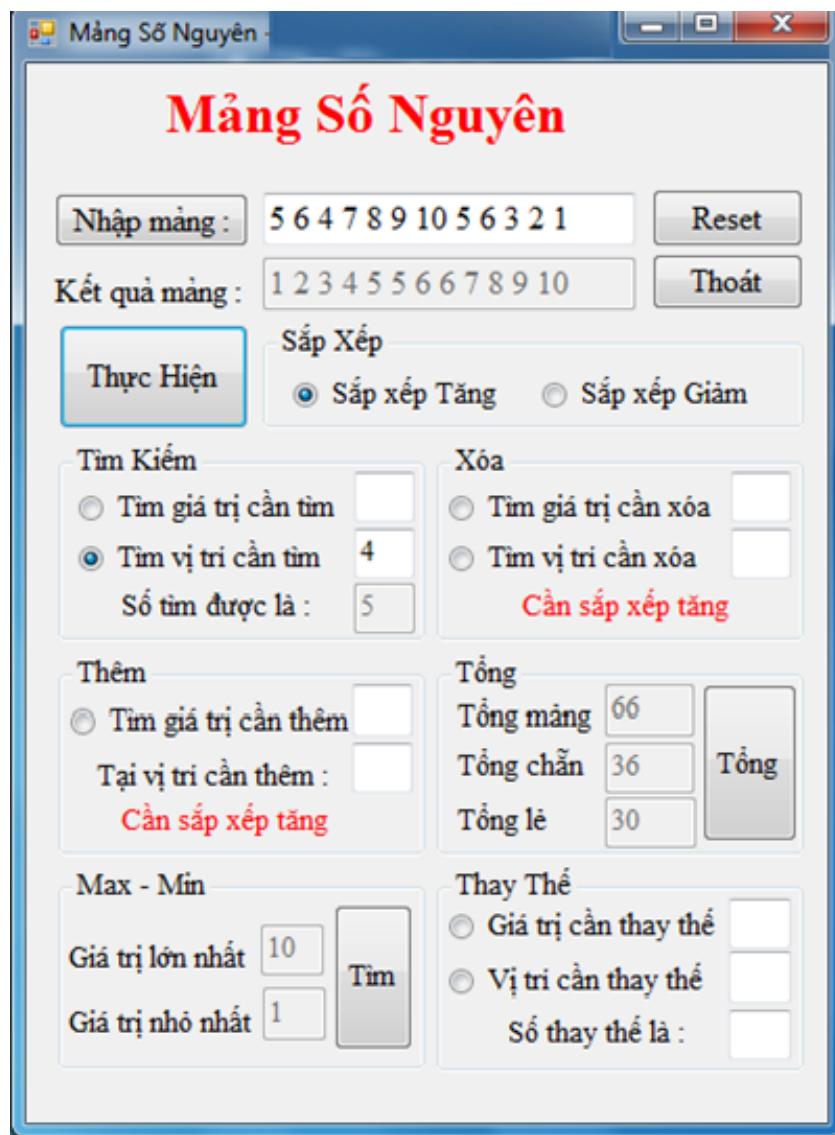


- Khi chọn radio button Giải phương trình Bậc hai thì hiển thị đầy đủ 3 Textbox để nhập liệu và thực hiện yêu cầu giống như giải phương trình bậc nhất, kết quả như hình sau:



- Kiểm tra dữ liệu nhập vào và hiện thông báo lỗi khi cần.
- Khi Form đóng cần có xác nhận từ người dùng.

Bài 2. Thiết kế giao diện sau và thực hiện các yêu cầu về mảng một chiều ở bên dưới:

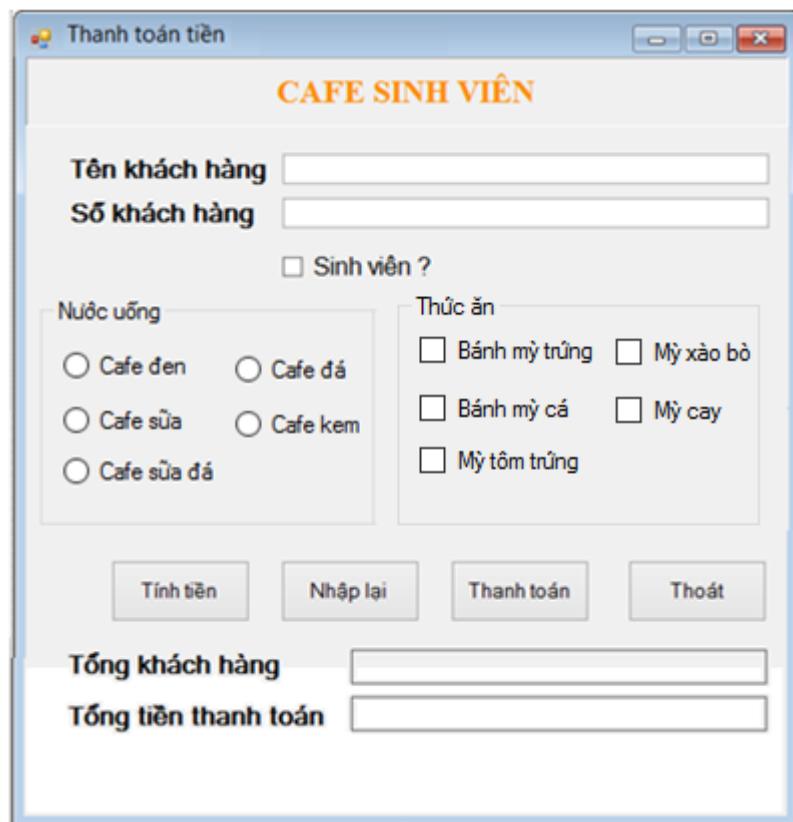


Yêu cầu:

- Viết class lưu trữ mảng một chiều số nguyên với các thuộc tính và phương thức thực hiện theo yêu cầu của giao diện: sắp xếp mảng tăng/ giảm, tìm kiếm giá trị và vị trí phần tử bất kỳ trong mảng, thêm giá trị mới tại một vị trí được chỉ định trong mảng, tính tổng mảng, tổng chẵn và tổng lẻ trong mảng, tìm giá trị lớn nhất và nhỏ nhất của mảng, thay thế giá trị và vị trí với số thay thế bất kỳ trong mảng.
- Thực hiện kiểm tra các lỗi có khả năng xảy ra.
- Đóng form phải có xác nhận từ người dùng.

3. Bài tập nâng cao

Bài 1. Viết chương trình hỗ trợ việc quản lý thu tiền cho quán Cafe Sinh Viên như sau:



Yêu cầu:

- Khi nhóm khách hàng vào quán gọi cafe sẽ được chọn một trong các loại cafe, chọn thức ăn sẽ chọn một trong các loại thức ăn. Giá cho từng loại tương ứng như sau:

Cafe đen:	20.000đ	Bánh mỳ trứng:	15.000đ
Cafe đá:	25.000đ	Bánh mỳ cá:	15.000đ
Cafe sữa:	25.000đ	Mỳ tôm trứng:	20.000đ
Cafe sữa đá:	30.000đ	Mỳ xào bò:	30.000đ
Cafe kem:	35.000đ	Mỳ cay:	50.000đ

- Khi khách hàng gọi tính tiền, thu ngân sẽ nhập thông tin tên khách hàng và tùy vào từng loại thì sẽ có giá tương ứng (**Lưu ý**: nếu khách hàng là Sinh Viên – check vào checkbox Sinh Viên – thì sẽ được giảm giá 20%).

☞ **Lưu ý:** Textbox số khách hàng chỉ cho nhập số.

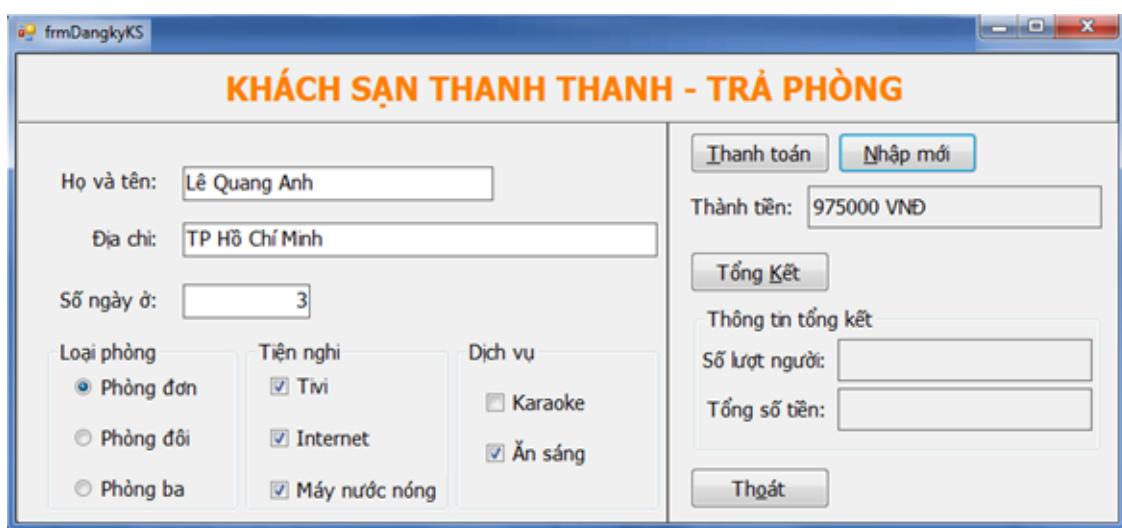
- Chương trình thực hiện:

- + **Form_Load:** con trỏ văn bản đặt vào ô tên khách hàng, các button TinhTien, NhapLai, ThanhToan bị mờ (Enabled = false). Thực hiện nhập tên khách hàng (tên khách hàng không được để trống). Check vào checkbox Sinh Viên nếu khách hàng là Sinh Viên, chọn loại nước uống và thức ăn mà khách hàng đã gọi. Khi nhập đầy đủ thông tin thì btnTinhTien có tác dụng.

- + **btnTinhTien_Click:** thực hiện tính tiền cho khách hàng vừa nhập và hiển thị lên Messagebox, **btnNhaphLai**, **btnThanhToan** sáng lên.
- + **btnNhaphLai_Click:** quay về trạng thái ban đầu của Form, btnNhaphLai mờ.
- + **btnThanhToan_Click:** Ghi lại thông tin tổng số khách và tổng tiền Thanh toán vào các label tương ứng, sẵn sàng cho việc nhập nhóm khách hàng mới. btnThanhToan bị mờ.
- + **btnThoat_Click:** hỏi người dùng có chắc chắn thoát khỏi chương trình hay không? Yes: thoát, No: không.

4. Bài tập về nhà

Bài 1. Thiết kế giao diện Quản lý Thanh toán tiền phòng cho Khách sạn Thành Thành như sau:



Khi khách trả phòng, nhân viên kế toán sẽ thực hiện kiểm tra tên, địa chỉ, số ngày ở, loại phòng, tiện nghi và dịch vụ của khách. Sau đây là bảng giá:

- Loại phòng:

Phòng đơn:	300.000đ/1 ngày
Phòng đôi:	350.000đ/1 ngày
Phòng ba:	400.000đ/1 ngày
- Tiện nghi: mỗi loại cộng thêm 10.000đ
- Dịch vụ:

Karaoke:	50.000đ
Ăn sáng:	15.000đ/1 ngày

Cuối ngày Nhân viên kế toán sẽ thống kê tổng số lượt người trả phòng, tổng số tiền thu được trong ngày.

Chương trình thực hiện các chức năng:

- **Form_Load:** con trỏ văn bản đặt vào ô tên khách hàng, các button TongKet, NhaphMoi, ThanhToan bị mờ (enabled=false). Thực hiện nhập tên (tên khách không

được để trống), địa chỉ, số ngày ở (number), check vào loại phòng, tiện nghi, dịch vụ mà khách đã thuê. Khi nhập đầy đủ thông tin thì btnThanhToan có tác dụng.

- **btnThanhToan:** thực hiện tính tiền cho khách vừa nhập và hiển thị lên label Thành Tiền, đồng thời lưu lại thông tin tổng số tiền và tổng số lượt khách. btnNhapMoi, btnTongKet sáng lên sẵn sàng cho việc nhập khách mới.
- **btnNhapMoi:** khởi tạo lại trạng thái ban đầu của Form, btnNhapMoi bị mờ.
- **btnTongKet:** Ghi lại thông tin tổng số khách và tổng tiền Thanh toán vào các label tương ứng, đồng thời khởi tạo lại giá trị tổng số khách hàng =0, tổng tiền thanh toán =0. btnTongKet bị mờ.
- **btnThoat_Click:** hỏi người dùng có chắc chắn thoát khỏi chương trình hay không? Yes: thoát, No: không.

Trường ĐH CNTP TP.HCM
 Khoa: CNTT
 Bộ môn: CNPM
 Môn: Lập trình .NET

BÀI 5.
GIAO DIỆN
(tiếp theo)



A. MỤC TIÊU:

- Thiết kế được giao diện Windows Form từ các control trong C#.
- Sử dụng được các Controls Textbox, Label, Button, Listbox, Combobox, TreeView, TabControl.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

- **Listbox:**
 - + Thuộc tính: Items, SelectionMode, SelectedIndex, SelectedItem.
 - + Sự kiện: SelectedIndexChanged.
 - + Phương thức: Add, AddRange, RemoveAt, Remove, IndexOf.
- **ComboBox:**
 - + Thuộc tính: Items, DropDownStyle, SelectedIndex, SelectedItems, Text.
 - + Sự kiện: SelectedIndexChanged.
- **TreeView:**
 - + Thuộc tính: Nodes, SelectedNode, ShowRootLine, ImageList, ShowLine, ImageIndex, SelectedImageIndex, FirstName, LastName, NextNode, PrevNode, Text.
 - + Sự kiện: AfterSelect, Click
 - + Phương thức: CollapseAll, ExpandAll.

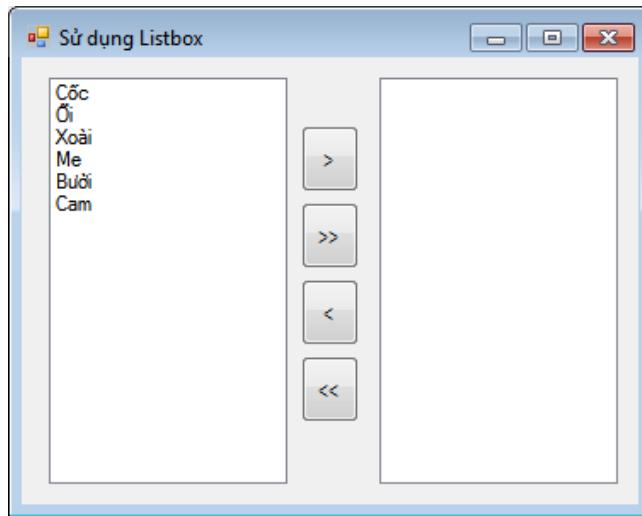
1.2 Giới thiệu bài tập mẫu

Bài 1: Thiết kế giao diện sau với các yêu cầu:

- Button >: Chuyển phần tử đang chọn qua listbox phải.
- Button >>: Chuyển tất cả phần tử qua listbox phải.
- Button <: Chuyển phần tử đang chọn qua listbox trái.
- Button <<: Chuyển tất cả phần tử qua listbox trái.
- Thêm button “Chuyển tùy ý” cho listbox trái: chọn item nào thì chuyển item đó

qua listbox phải.

- Đóng Form có xác nhận từ người dùng.



Hướng dẫn:

- **Danh sách các đối tượng được sử dụng trong form:**

Object	Properties	Events
frmListBox	Name: frmListBox Text: “Sử dụng listbox” FontName: Tahoma FontSize: 11	FormClosing
lst_Trai	Name: lst_Trai	
lst_Phai	Name: lst_Phai	

- **Các sự kiện tương ứng với các yêu cầu:**

- Chuyển một item từ listbox trái sang listbox phải:

```
private void btnQuaPhai_Click(object sender, EventArgs e)
{
    // Chuyển giá trị chọn bên trái sang phải
    lst_Phai.Items.Add(lst_Trai.SelectedItem);
    // Xóa giá trị chọn bên trái
    lst_Trai.Items.Remove(lst_Trai.SelectedItem);
}
```

- Chuyển tất cả item từ listbox trái sang listbox phải

```
private void btnQuaPhaiAll_Click(object sender, EventArgs e)
{
```

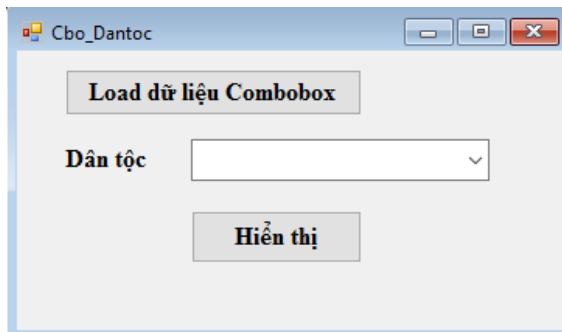
```

    // Chuyển giá trị items bên trái sang phải
    lst_Phai.Items.AddRange(lst_Trai.Items);
    // Xóa giá trị bên trái
    lst_Trai.Items.Clear();
}

```

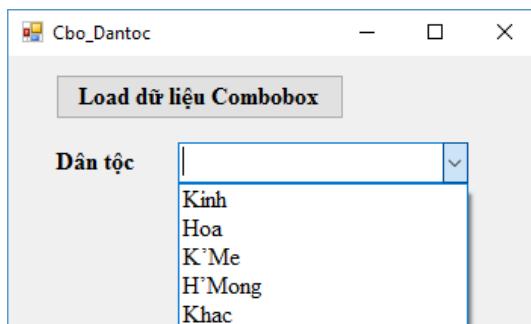
- Sinh viên tự viết code thực hiện cho các yêu cầu còn lại.

Bài 2: Thiết kế giao diện sau:

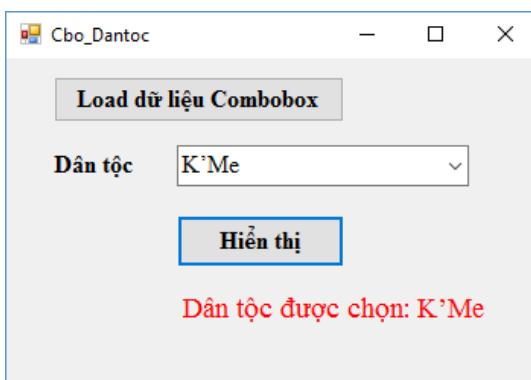


Yêu cầu:

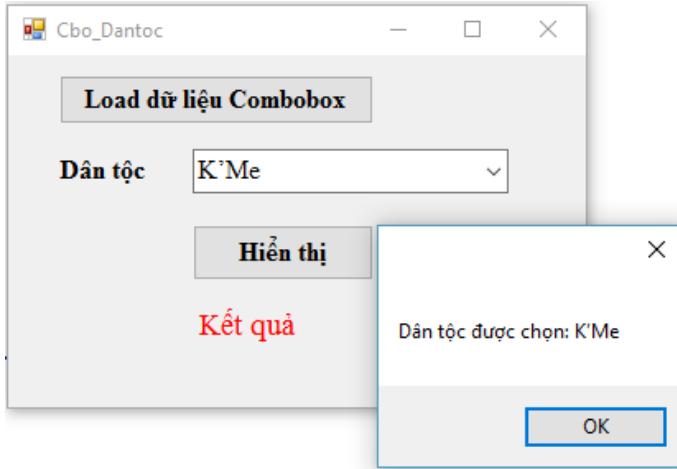
- Click chọn button Load dữ liệu Combobox, 5 dân tộc: Kinh, Hoa, K'Me, H'Mong, Khác sẽ được đưa vào combobox Dân tộc.



- Chọn một dân tộc trong combobox, hiển thị dân tộc được chọn theo 2 cách:
 - Cách 1: click button Hiển thị, dân tộc được chọn sẽ xuất hiện trong một Label hoặc Textbox.



- Cách 2: Dân tộc được chọn sẽ xuất hiện trên MessageBox ngay khi chọn.



Hướng dẫn:

- Sự kiện click của button Load dữ liệu Combobox

```
private void btn_loadDL_Click(object sender, EventArgs e)
{
    string[] dt = { "Kinh", "Hoa", "K'Me", "H'Mong", "Khac" };
    foreach (string s in dt)
    {
        cbo_dt.Items.Add(s);
    }
}
```

- Sự kiện click của button Hiển thị

```
private void btn_hienthi_Click(object sender, EventArgs e)
{
    string s = "Dân tộc được chọn: ";
    if (cbo_dt.SelectedIndex >= 0)
        lbl_kq.Text = s + cbo_dt.SelectedItem.ToString();
    else
        lbl_kq.Text = "Bạn chưa chọn dân tộc";
}
```

- Sự kiện SelectedIndexChanged của Combobox

```
private void cbo_dt_SelectedIndexChanged(object sender, EventArgs e)
{
    MessageBox.Show("Dân tộc được chọn: " +
                    cbo_dt.SelectedItem.ToString());
}
```

Bài 3: Thiết kế giao diện sau:

HỒ SƠ NHÂN VIÊN

Mã số Họ tên
Địa chỉ
Phòng ban
Phòng ban
Thêm Thoát
Xóa phòng ban

Yêu cầu:

- Khi Form hiển thị lên, TreeView phòng ban và ComboBox phòng ban có sẵn những phòng ban sau: Giám đốc, Tổ chức hành chính, Kế hoạch, Kế Toán.
- Thực hiện thêm mới phòng ban và xóa phòng ban được chọn (Lưu ý: kiểm tra trùng khi thêm và có xác nhận khi xóa).
- Khi một phòng ban mới được thêm thì phải thêm phòng đó vào ComboBox phòng ban.
- Thực hiện thêm mới nhân viên vào phòng ban được chọn lên TreeView.

HỒ SƠ NHÂN VIÊN

Mã số N001 Họ tên Nguyễn Bình
Địa chỉ TPHCM
Phòng ban Giám đốc
Thêm Thoát
Phòng ban
Thêm phòng ban
Xóa phòng ban

Hướng dẫn:

- **Danh sách các thuộc tính của các Object:**

Object	Properties	Events
Form	Name: frmTreeView Text: “Phòng ban –Nhân viên” FontName: Times New Roman FontSize: 11	FormClosing
Label		
TextBox	Name: txt_phongban, txt_maso, txt_hoten, txt_diachi	
Combobox	Name: cbo_phongban	
Button	Name: btn_ThemPB, btn_XoaPB, btn_ThemNV, btn_Thoat	Click
TreeView	Name: trv_DS	

– Xử lý sự kiện theo yêu cầu:

- Form Load

```
private void TreeView_Load(object sender, EventArgs e)
{
    string[] pb = {"Giám đốc", "Tổ chức hành chính", "Kế hoạch",
                   "Kế Toán"};
    foreach (string s in pb)
    {
        trv_DS.Nodes.Add(s); //thêm node vào treeview
        cbo_phongban.Items.Add(s); //thêm item vào combobox
    }
    cbo_phongban.SelectedIndex = 0; //item đầu tiên trên được chọn
}
```

- Thêm phòng ban mới vào treeview và combobox phòng ban

```
private void btn_ThemPB_Click(object sender, EventArgs e)
{
    // Kiểm tra txt_Phongban có tồn tại trong TreeView chưa
    if (kiemtra(txt_Phongban.Text))
    {
```

```

        trv_DS.Nodes.Add(txt_Phongban.Text);
        cbo_phongban.Items.Add(txt_Phongban.Text);
    }
    else
        MessageBox.Show("Phòng ban đã tồn tại!");
    txt_Phongban.Text = "";
    txt_Phongban.Focus();
}

```

☞ **Lưu ý:** Sinh viên tự viết phương thức kiểm tra trùng phòng ban: `bool kiemtra(string s)` và sử dụng phương thức so sánh chuỗi không phân biệt chữ hoa chữ thường:

`string.Compare(string, string, bool)`

- **Xóa phòng ban được chọn trong treeview**

```

private void btn_XoaPB_Click(object sender, EventArgs e)
{
    if(MessageBox.Show("Bạn có chắc muốn xóa?", "Thông báo",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question,
        MessageBoxDefaultButton.Button1)== DialogResult.Yes)
    {
        // Một phòng ban trong treeView được chọn
        if (trv_DS.SelectedNode != null)
        {
            cbo_phongban.Items.Remove(trv_DS.SelectedNode.Text);
            trv_DS.Nodes.Remove(trv_DS.SelectedNode);
        }
    }
}

```

- **Thêm nhân viên của phòng ban vào treeview**

```

private void btn_ThemNV_Click(object sender, EventArgs e)
{
    //tìm index của node có nội dung là item được chọn trong
    //combobox phòng ban
    int index = -1;
    foreach (TreeNode node in trv_DS.Nodes)
        if (node.Text == cbo_phongban.Text)
    {

```

```

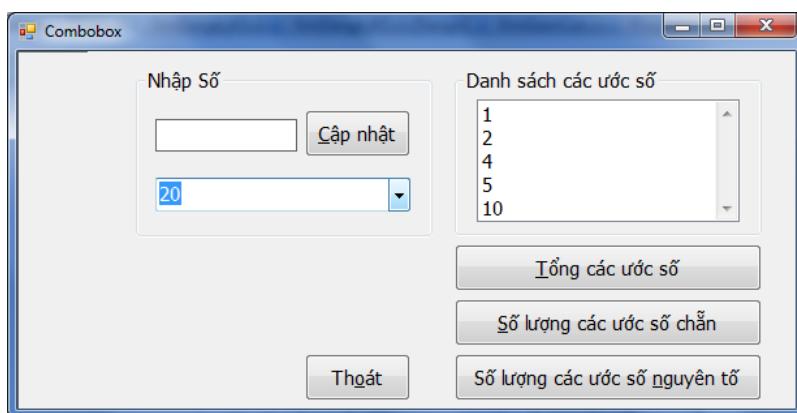
        index = node.Index;
        break;
    }
    trv_DS.Nodes[index].Nodes.Add(txt_hoten.Text+
        ("+" + txt_maso.Text + "));

    trv_DS.ExpandAll(); //mở rộng treeview
}

```

2. Bài tập tại lớp

Bài 1. Thiết kế giao diện sau:



Yêu cầu:

- Nhập số vào textbox Nhập số (có kiểm tra dữ liệu nhập).
- btnCapNhat: thêm số vừa nhập vào combobox, đồng thời xóa dữ liệu trong textbox Nhập số, đặt con trỏ lại textbox (có kiểm tra số vừa nhập đã tồn tại trong combobox chưa trước khi thêm).
- Khi người dùng chọn một số trong combobox thì danh sách các ước số của số này sẽ hiển thị vào listbox bên phải tương ứng.
- Khi nhấn các button Tính: “Tổng các ước số”, “Số lượng các ước số chẵn”, “Số lượng các ước số nguyên tố” thì hiển thị thông tin tương ứng vào MessageBox dựa vào các ước số trên listbox.
- btnThoat: thoát khỏi ứng dụng có xác nhận từ người dùng.
- Tạo HotKey cho các button như trên giao diện

Bài 2. Thiết kế giao diện như sau:

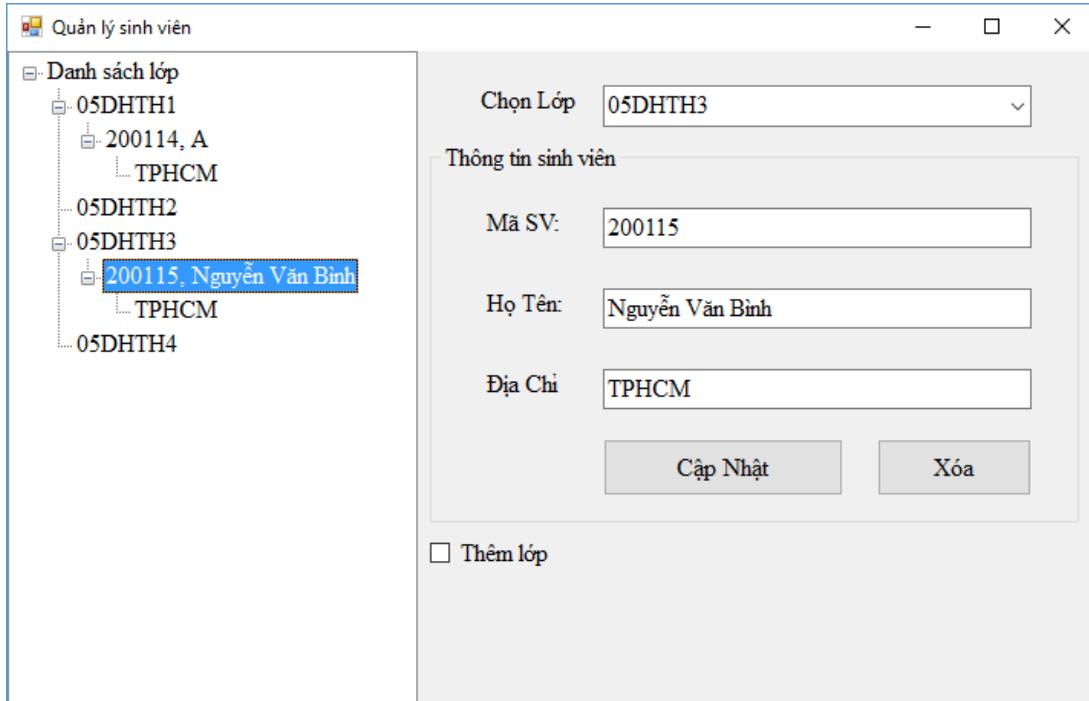
The screenshot shows a Windows application window titled "Quản lý sinh viên". On the left, there is a tree view control labeled "Danh sách lớp" with four nodes: 05DHTH1, 05DHTH2, 05DHTH3, and 05DHTH4. The main panel contains several controls: a dropdown menu "Chọn Lớp" set to "05DHTH1", a group box "Thông tin sinh viên" with three text input fields for "Mã SV", "Họ Tên", and "Địa Chỉ", and two buttons "Cập Nhật" and "Xóa". Below these is a checked checkbox "Thêm lớp" and another group box "Thông tin lớp" with a text input field "Tên lớp" and a button "Thêm lớp".

Yêu cầu:

- Khi Form hiện lên, danh sách lớp ở treeView đã tồn tại một số lớp; checkbox Thêm lớp là false (groupbox Thông tin lớp và các Control bên trong ẩn đi). Khi checkbox Thêm lớp là true groupBox Thông tin lớp và các control bên trong hiện lên.
- Button Thêm lớp: Thực hiện thêm một lớp mới vào Danh sách lớp của TreeView (Kiểm tra tên lớp không được trùng trước khi thêm), đồng thời thêm lớp đó vào ComboBox chọn lớp.
- Button Cập Nhật: Thêm một sinh viên vào danh sách lớp đang chọn trên ComboBox với nội dung các node như hình. Trước khi thêm kiểm tra thông tin nhập bao gồm: các textbox nhập không được để trống, không trùng mã sinh viên.

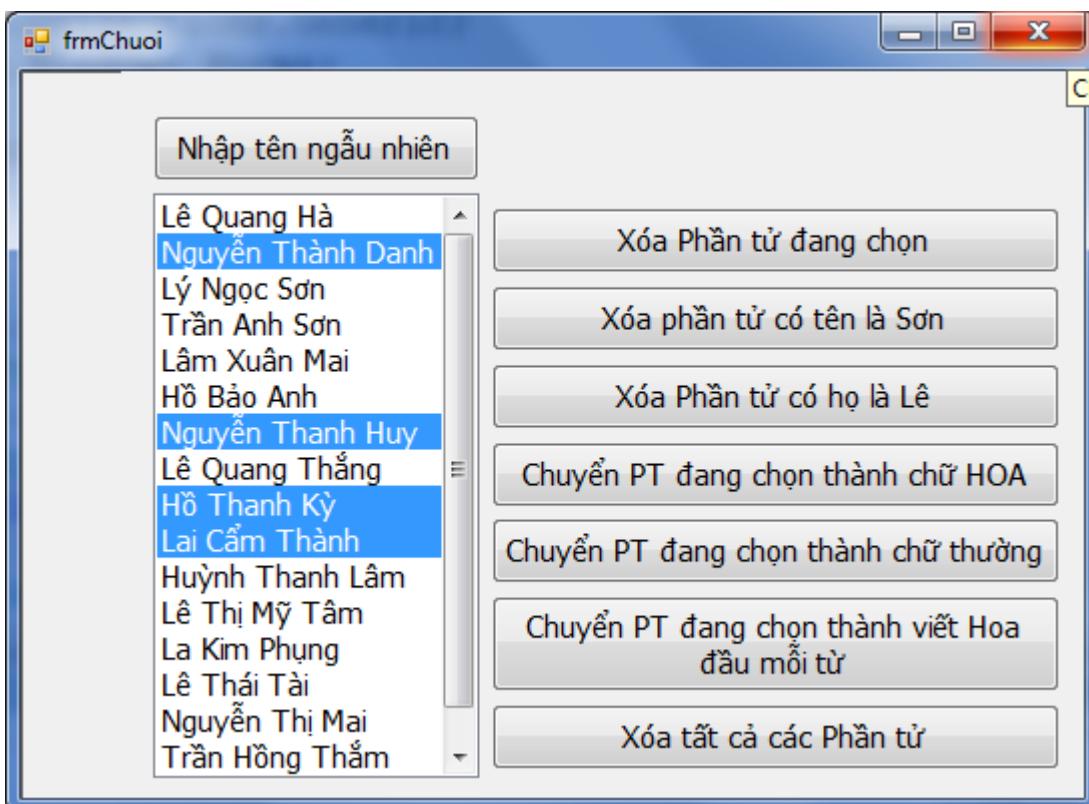
The screenshot shows the same Windows application window after some interaction. The "Chọn Lớp" dropdown is now set to "05DHTH2". The "Thông tin sinh viên" group box is visible, showing empty text input fields for "Mã SV", "Họ Tên", and "Địa Chỉ". The "Cập Nhật" and "Xóa" buttons are also visible. The "Thêm lớp" checkbox is unchecked. The tree view on the left shows that "05DHTH2" is now expanded, revealing a child node: "2001140001, Nguyễn Văn A" under "TP.HCM".

- Button Xóa: Chỉ cho phép xóa một node là sinh viên đang chọn trên TreeView (xác nhận trước khi xóa).
- Khi click chọn một node là mã sinh viên trong TreeView, hiển thị thông tin của sinh viên đó lên textbox tương ứng.



3. Bài tập nâng cao

Bài 1. Thực hiện form xử lý chuỗi có giao diện như sau:

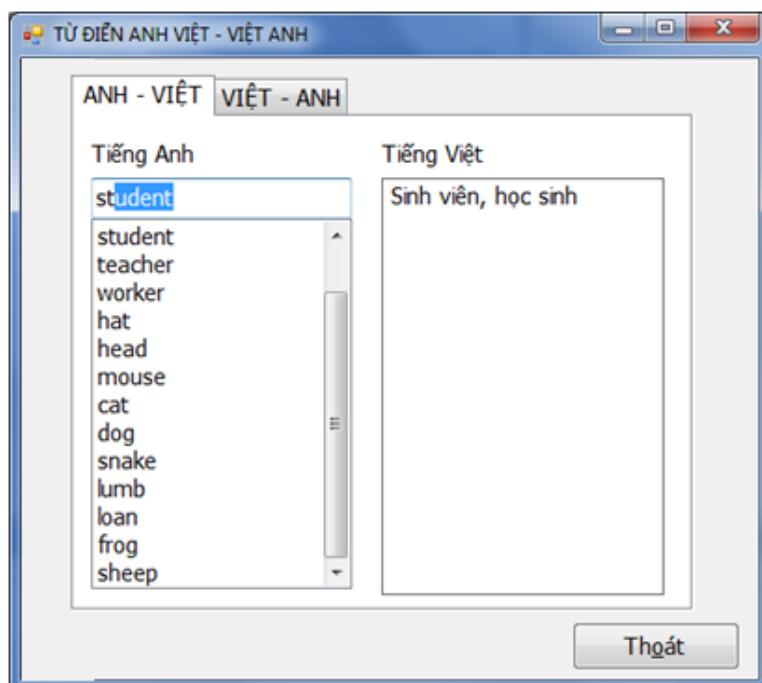


Yêu cầu:

- btnNgauNhien: thực hiện thêm vào listbox 50 tên dựa vào mảng chuỗi HO, TENLOT, TEN được kết hợp ngẫu nhiên
 - HO = {"Lê", "Nguyễn", "Lý", "Trần", "Lâm", "Hồ", "Lai", "Huỳnh", "La"}
 - TENLOT = {"Quang", "Thành", "Ngọc", "Anh", "Xuân", "Bảo", "Cẩm", "Thị", "Kim", "Thái", "Hồng"}
 - TEN = {"Ha", "Danh", "Sơn", "Mai", "Thắng", "Kỳ", "Thành", "Lâm", "Tâm", "Phụng", "Thắm"}
- Thực hiện các button tương ứng trên giao diện.
- Khi người dùng double click vào tên đang chọn thì sẽ mở Inputbox, cho phép người dùng thay đổi tên mới vào vị trí tên này.

4. Bài tập về nhà

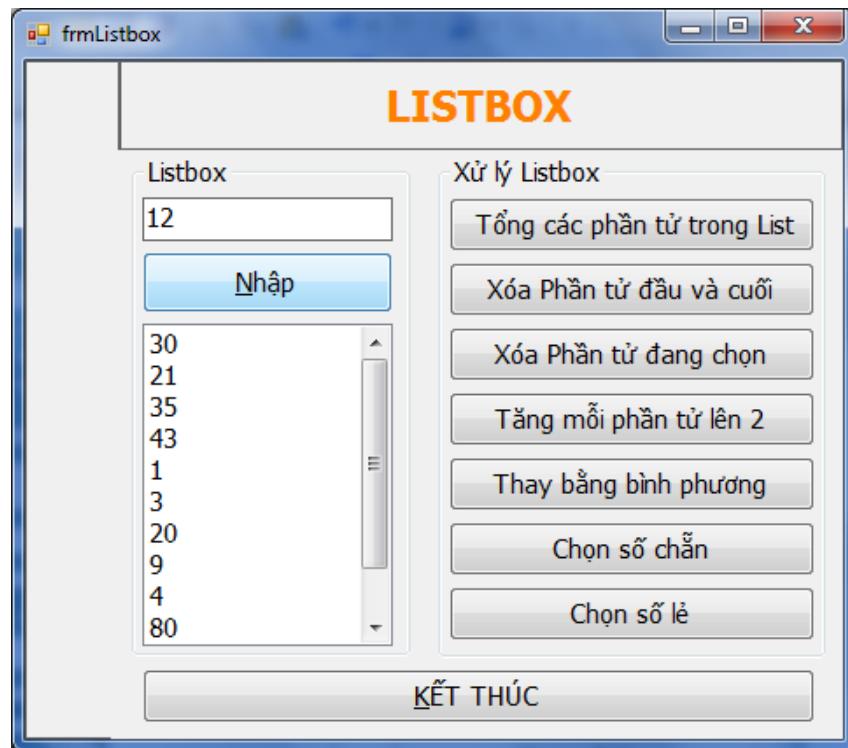
Bài 1. Viết chương trình từ điển Anh – Việt và Việt – Anh



Yêu cầu:

- Khi nhập vào combobox từ cần tra thì chương trình sẽ dò tìm đến chữ nào khớp với ký tự gần nhất.
- Khi nhấn button Enter hoặc double click vào từ cần tra thì nghĩa tương ứng của từ sẽ hiển thị vào textbox bên phải tương ứng.
- Danh sách các từ lưu sẵn vào List

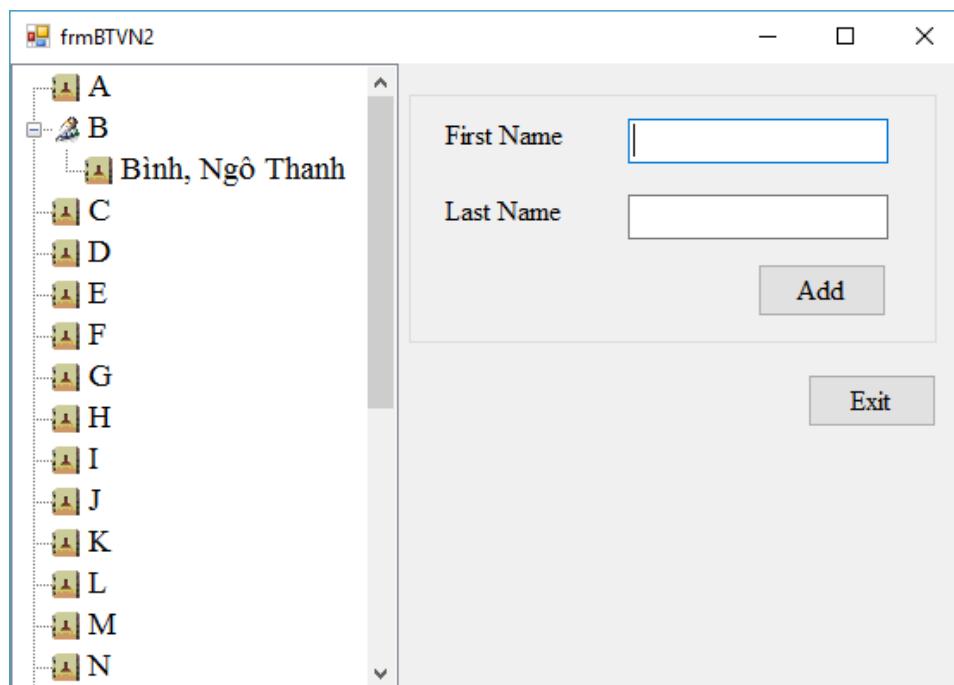
Bài 2. Viết chương trình thêm các phần tử vào listbox (listbox được chọn nhiều phần tử) các số tự nhiên N được nhập từ textbox.



Yêu cầu: các button lệnh thực hiện các công việc sau:

- Tính tổng các phần tử trong Listbox, hiển thị lên Messagebox.
- Xóa phần tử đầu và cuối của listbox.
- Xóa các phần tử đang chọn trong listbox.
- Tăng giá trị mỗi phần tử lên 2.
- Thay mỗi giá trị của mỗi phần tử bằng bình phương của chính nó
- Thực hiện chọn các phần tử trong listbox là số chẵn.
- Thực hiện chọn các phần tử trong listbox là số lẻ.

Bài 3. Thiết kế giao diện nhập danh bạ với yêu cầu như sau:



Yêu cầu:

- Khi form được load lên treeView chứa tất cả các ký tự từ A→ Z.
- Nhằm mục đích tiện lợi cho người sử dụng, khi muốn thêm một người mới vào danh bạ, chương trình sẽ đưa người này vào treeView ở vị trí node tương ứng với chữ cái đầu của tên (xem hình).



A. MỤC TIÊU:

- Thiết kế được giao diện Windows Form từ các control trong C#.
- Sử dụng được các Controls Textbox, Label, Button, ListView, Menu, ContextMenuStrip, Timer, Common Dialog.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

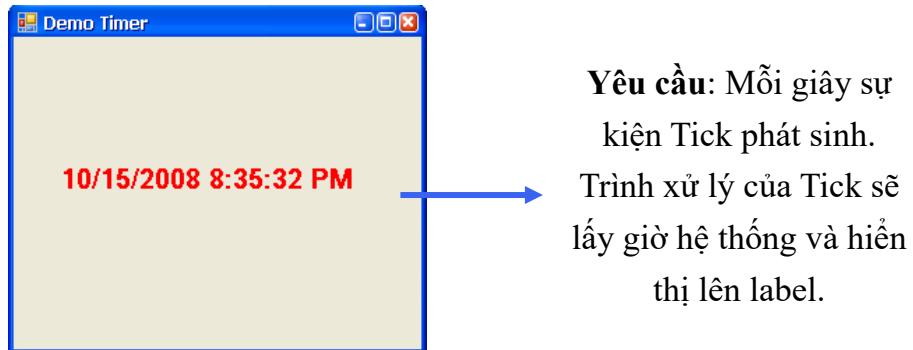
1.1 Kiến thức cần nhớ

- **ListView:**
 - + Thuộc tính: Columns, Items, View, FullRowSelect, MultiSelect, SelectedItems
 - + Sự kiện: SelectedIndexChanged, Click
 - + Phương thức: Add, AddRange, RemoveAt, Remove, IndexOf, Contains
 - **MenuStrip:** cung cấp nhóm lệnh có quan hệ với nhau cho các ứng dụng Windows Form
 - + Khai báo trình xử lý sự kiện Click cho ToolStrip:
 - Kích đúp vào menu item của ToolStrip để tạo sự kiện.
 - Hoặc trong cửa sổ Properties → chọn Event, double click vào sự kiện Click.
 - **ContextMenuStrip:** Khi user kích chuột phải lên control thì sẽ hiển thị Context Menu đã cài đặt sẵn. Cách tạo và sử dụng tương tự ToolStrip.
 - **Timer (bộ định thời gian):**
 - + Thuộc tính: Enabled, Interval.
 - + Sự kiện: Tick.
 - + Phương thức: Start, Stop.
 - **Dialog:** SaveDialog, OpenFileDialog, ColorDialog, FontDialog
- Các bước tạo một Dialog:**
- 1. Kéo thả dialog vào form

- Visual Studio sẽ tạo thẻ hiện chứa trong lớp form
- 2. Khai báo đối tượng và tạo thẻ hiện của lớp CD
 - Ví dụ: `OpenFileDialog oFile = new OpenFileDialog();`

1.2 Giới thiệu bài tập mẫu

Bài 1: Thực hiện form sau:

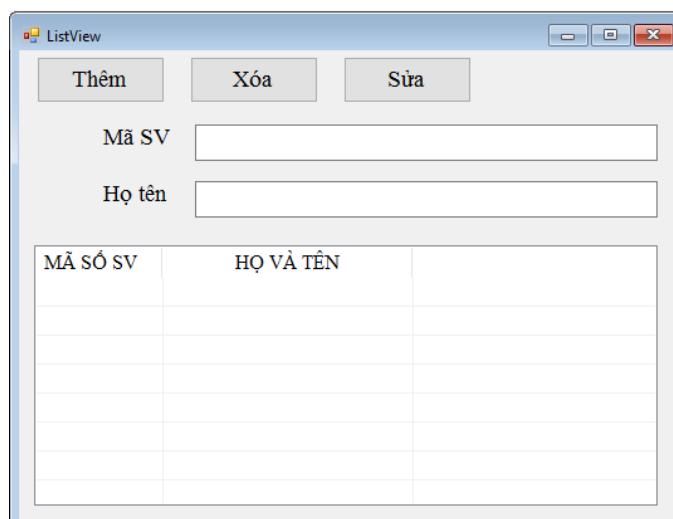


Hướng dẫn:

- Các control được sử dụng trong form gồm: một Label (label1) (dùng để hiển thị giờ lấy từ hệ thống), một Timer (timer1). Timer được khai báo với các thuộc tính sau:
 - Enabled: Enable sự kiện Tick.
 - Interval: khoảng thời gian chờ giữa 2 lần gọi Tick (đơn vị mili giây)
- **Các sự kiện được xử lý:**

```
private void timer1_Tick(object sender, EventArgs e)
{
    // Lấy thời gian hệ thống
    DateTime now = new DateTime();
    label1.Text = now.ToString();
}
```

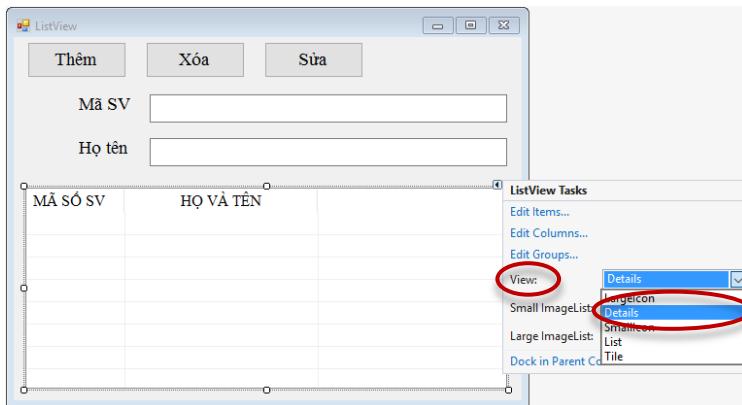
Bài 2: Thiết kế form sau:



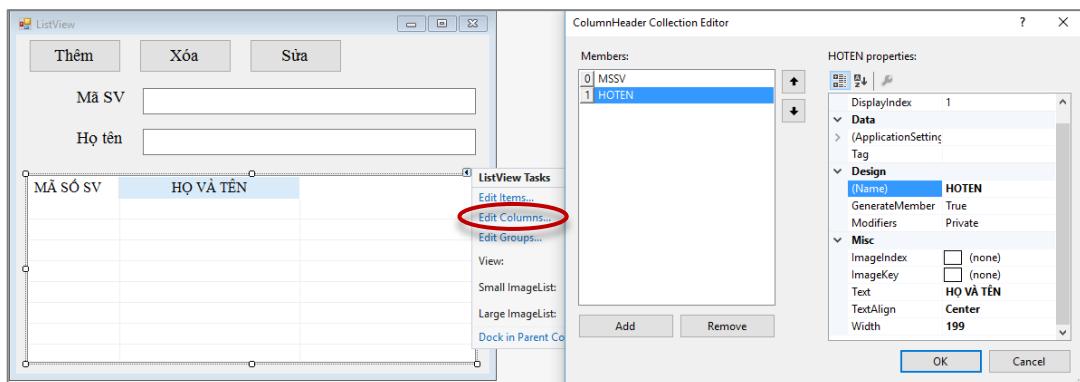
Hướng dẫn:

– Thiết kế giao diện

Bước 1: Chọn chế độ hiển thị View cho listview:



Bước 2: Thiết kế cột (Columns) cho listview:



Bước 3: Thiết lập thuộc tính GridLines, FullRowSelect (nếu cần) cho listview:

– Các sự kiện xử lý cho button:

+ Button Thêm:

```
private void btnthem_Click(object sender, EventArgs e)
{
    //cách 1
    //lstv1.Items.Add(txtmssv.Text).SubItems.Add(txthoten.Text);
    //cách 2
    ListViewItem item = new ListViewItem();
    ListViewItem.ListViewSubItem subitem = new
        ListViewItem.ListViewSubItem();
    item.Text = txtmssv.Text;
    subitem.Text = txthoten.Text;
    item.SubItems.Add(subitem);
    lstv1.Items.Add(item);
}
```

```

//cách 3
/*ListViewItem item1 = new ListViewItem(new[] { txtmssv.Text,
                                              txthoten.Text });
lstv1.Items.Add(item1); */
}

```

☞ **Lưu ý:** Sinh viên tự viết phương thức kiểm tra trùng mã số sinh viên trước khi thêm vào danh sách.

+ Button **Xóa:**

```

private void btnxoa_Click(object sender, EventArgs e)
{
    foreach (ListViewItem l in lstv1.SelectedItems)
    {
        l.Remove();
    }
}

```

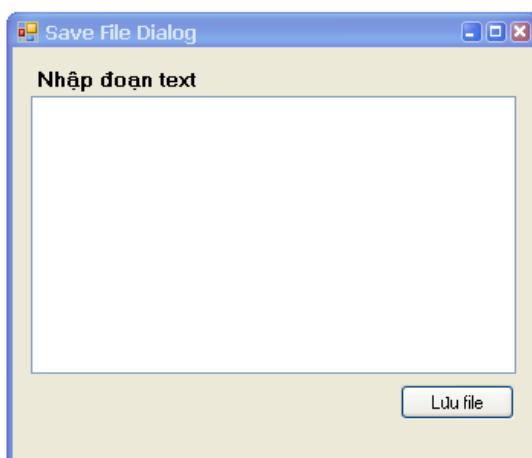
+ Button **Sửa:**

```

private void btnsua_Click(object sender, EventArgs e)
{
    if (lstv1.FocusedItem != null && lstv1.FocusedItem.Index >= 0)
    {
        ListViewItem item = lstv1.FocusedItem;
        item.SubItems[0].Text = txtmssv.Text;
        item.SubItems[1].Text = txthoten.Text;
    }
}

```

Bài 3. Thực hiện Form sau:

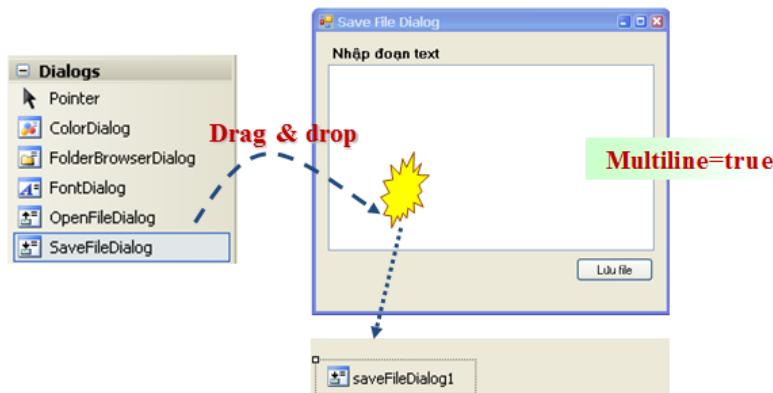


Yêu cầu:

- Nhập văn bản vào textbox, sau đó lưu nội dung đã nhập thành file: *.txt
- Bổ sung phần đọc file text từ ví dụ trên: mở file *.txt và hiển thị nội dung lên TextBox.

Hướng dẫn:

- Thiết kế giao diện



- Các sự kiện:

- + Button Lưu File:

```
SaveFileDialog svg = new SaveFileDialog();
svg.Filter = "Text files | *.txt";
if (svg.ShowDialog() == DialogResult.OK)
{
    string filename;
    filename = svg.FileName;
    textBox1.Text = filename;
    StreamWriter sw = File.CreateText(filename);
    sw.WriteLine(textBox1.Text);
    sw.Close();
}
```

- + Button Mở File

```
//Lọc loại đuôi file theo mong muốn
openFileDialog1.Filter = "All Text File (*.txt)|*.txt|All Document (*.pdf)|*.pdf";
//Nếu dialog openFileDialog1 chọn OK
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
    //Gán đường dẫn lên Textbox
    textBox1.Text = openFileDialog1.FileName;
```

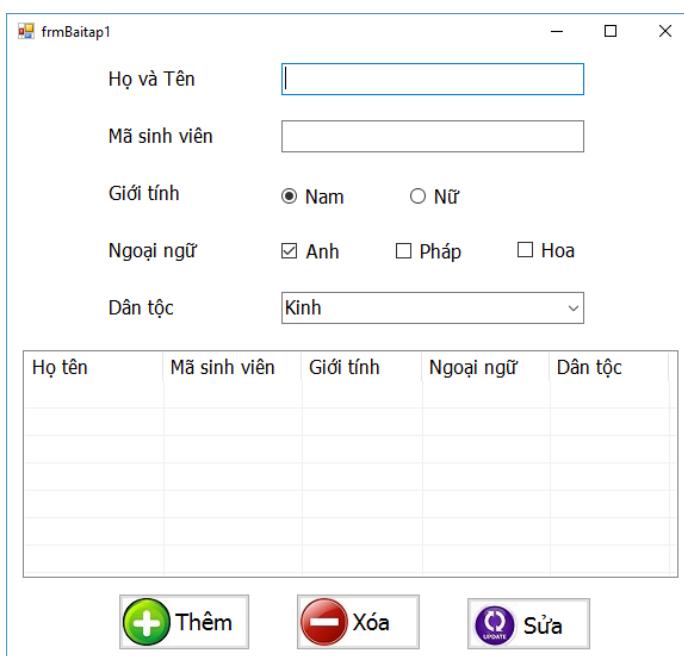
```

//Đọc dữ liệu từ file txt dùng
StreamReader f = new StreamReader(openFileDialog1.FileName);
//Đọc dữ liệu text và gán vào richTextBox
richTextBox1.Text = f.ReadToEnd();
// Đóng luồng đọc
f.Close();
}

```

2. Bài tập tại lớp

Bài 1. Thiết kế giao diện sau:



Yêu cầu:

- Khi Load form lên thực hiện thêm dữ liệu vào Combobox dân tộc.
- Thực hiện Thêm, Xóa, Sửa từ các control lên Listview. Lưu ý “Thêm không cho trùng mã sinh viên”, không cho sửa Mã sinh viên.
- Sử dụng ContextMenuStrip cho thực hiện Xóa khi kích chọn phải chuột vào Listview.
- Hỏi xác nhận trước khi đóng Form.

Bài 2. Thiết kế chương trình quản lý thông tin tài khoản cho một ngân hàng với giao diện sau:

Yêu cầu:

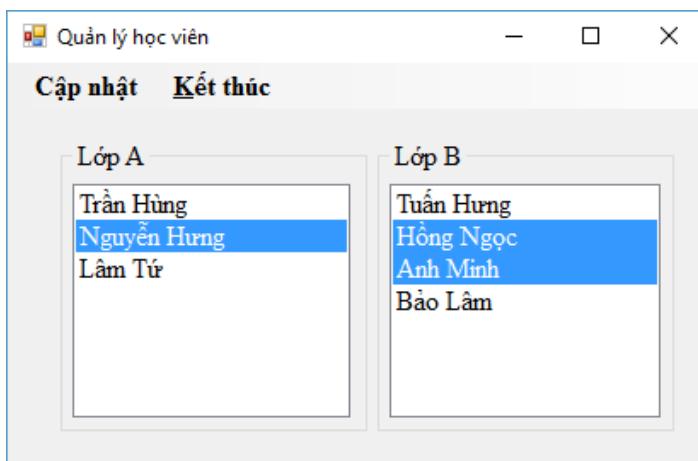
- Form load lên có hiện trạng như hình trên.
 - Khi nhấn vào button “Thêm”, text của button chuyển thành “Hủy” và Enabled button “Lưu”, các control textbox là rỗng, đặt con trỏ vào textbox Số tài khoản.
 - Button “Lưu” có tác dụng lưu dữ liệu từ các textbox vào trong listview.
 - Button “Xóa” có tác dụng xóa mẫu tin hiện hành trong listview (Enabled = true khi một mẫu tin được chọn trong listview).
 - Khi click một khách hàng trong listview, hiển thị thông tin lên các textbox tương ứng.
 - Thực hiện tính Tổng tiền, STT được đánh tự động khi thực hiện Thêm, Xóa.

QUẢN LÝ THÔNG TIN TÀI KHOẢN

Số tài khoản	0160901610			
Tên khách hàng	Trần Văn Anh			
Địa chỉ khách hàng	Tây Ninh			
Số tiền trong tài khoản	1200000			
<input type="button" value="Thêm"/> <input type="button" value="Lưu"/> <input type="button" value="Xóa"/> <input style="outline: none; border: 2px solid blue; padding: 2px;" type="button" value="Thoát"/>				
STT	Số tài khoản	Tên khách hàng	Địa chỉ	Số tiền
1	0160901609	Nguyễn Văn An	TPHCM	5,200,000
2	0160901610	Trần Văn Anh	Tây Ninh	1,200,000
				Tổng tiền
				6,400,000

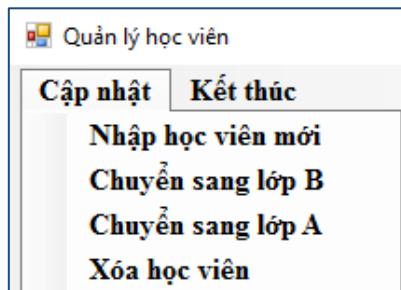
3. Bài tập nâng cao

Bài 1. Thiết kế form có giao diện sau:



Yêu cầu:

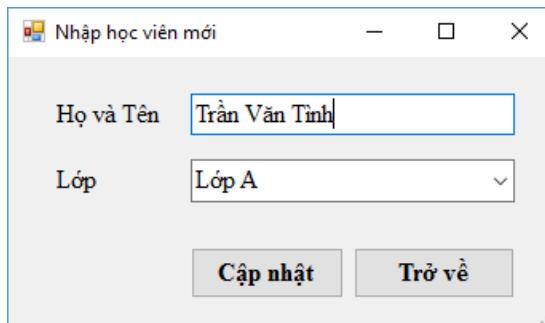
- Khi người sử dụng click vào mục kết thúc thì xuất hiện một Messagebox hỏi người sử dụng sau đó mới quyết định có dừng chương trình hay không.
- Trong menu Cập Nhật có các mục:



- + Xóa học viên: Xóa tất cả các học viên đang chọn trong Lớp A và Lớp B. (các listbox Lớp A và Lớp B cho phép chọn cùng lúc nhiều mục).
- + Chuyển sang lớp A: Chuyển các mục đang chọn trong Lớp B sang Lớp A
- + Chuyển sang lớp B: Chuyển các mục đang chọn trong Lớp A sang Lớp B

☞ Lưu ý: Khi hai mục trên được chọn nếu không có học viên nào được chọn trong listbox thì xuất hiện một MessageBox yêu cầu người sử dụng phải chọn học viên trước.

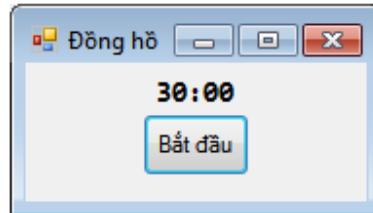
- + Nhập học viên mới: Khi mục này được chọn thì xuất hiện form cho phép nhập thông tin như sau (combobox Lớp: chứa Lớp A và Lớp B).



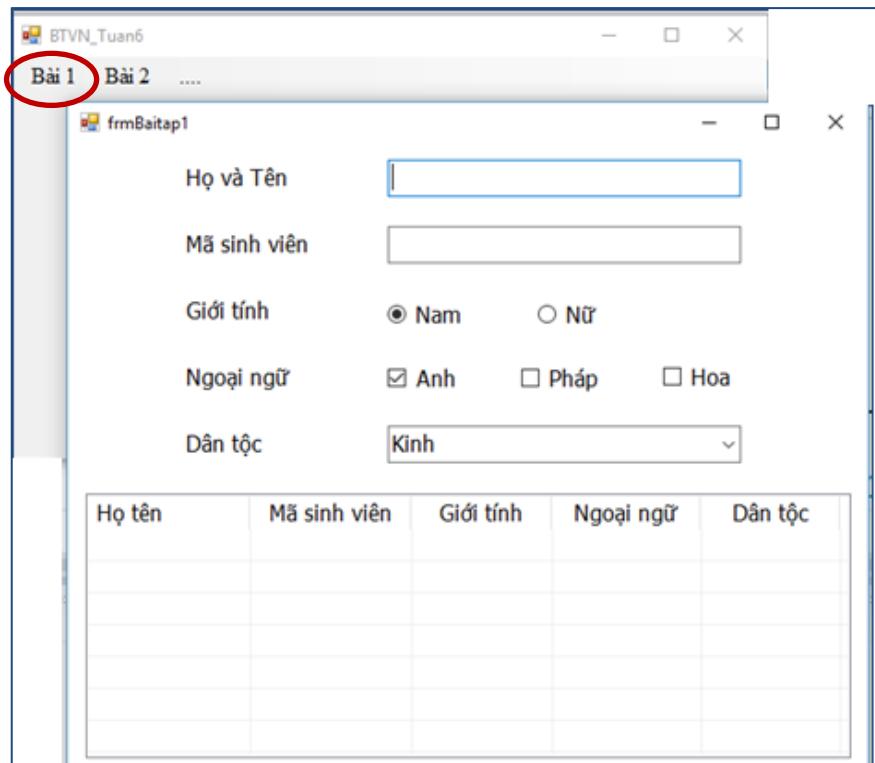
- Khi người sử dụng click vào button Cập Nhật:
 - o Nếu Họ và Tên khác rỗng thì:
 - Phụ thuộc vào lớp đã chọn trong combobox cập nhật họ tên vào listbox tương ứng trên form chính.
 - o Nếu Họ và Tên rỗng:
 - Xuất hiện một MessageBox yêu cầu phải nhập họ tên
 - Đặt con trỏ vào textbox Họ và Tên.
- Khi người sử dụng click vào button Trở về thì đóng form nhập thông tin và trở về form chính.

4. Bài tập về nhà

Bài 1. Thiết kế form có giao diện thực hiện đếm ngược thời gian:



Bài 2. Thiết kế form main có menustrip, gắn tắt cả bài tập trong tuần này và hiển thị các form khi chọn menu tương ứng.



Trường ĐH CNTP TP.HCM
 Khoa: CNTT
 Bộ môn: CNPM
 Môn: Lập trình .NET

BÀI 7. **LẬP TRÌNH TƯƠNG TÁC CƠ SỞ DỮ LIỆU**



A. MỤC TIÊU:

- Khai báo được đối tượng SqlConnection, SqlCommand.
- Sử dụng được đối tượng SqlConnection để thực hiện kết nối cơ sở dữ liệu.
- Sử dụng SqlCommand thực hiện thêm, xóa, sửa dữ liệu trên bảng.
- Lưu trữ dữ liệu dùng SQL Server 2012.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

✚ Giới thiệu ADO.NET

- ADO.NET (ActiveX Data Object.NET)
- ADO.NET là một tập các lớp nằm trong bộ thư viện lớp cơ sở của .NET Framework, cho phép các ứng dụng Windows (C#, VB.NET) hay ứng dụng Web (ASP.NET) thao tác dễ dàng với các nguồn dữ liệu.

Các thành phần của ADO.NET

– Connection:

- + Sử dụng đối tượng SqlConnection
- + Thuộc tính:

Thuộc tính	Mô tả
State	Tình trạng kết nối của Connection với các giá trị.
Closed	Kết nối đã đóng.
Connecting	Đang kết nối với nguồn dữ liệu.
Open	Kết nối đang mở.

- + Khai báo và viết chuỗi kết nối:

Cách 1: Đăng nhập với quyền Windows Authentication

```
SqlConnection conn = new SqlConnection("Data Source = <Tên_Server>;  
Initial Catalog = <Tên_Database>; Integrated Security = True");
```

Cách 2: Đăng nhập với quyền SQL Server Authentication

```
SqlConnection conn = new SqlConnection("Data Source =  
<Tên_Server>; Initial Catalog = <Tên_Database>; User ID =  
<Tên_Dăng_nhập>; Password = <Mật_khẩu_Dăng_nhập>");
```

- **Command:** Đối tượng SqlCommand cho phép chúng ta chọn kiểu tương tác ứng với thao tác thực hiện trên cơ sở dữ liệu.

- + Khai báo:

```
SqlCommand cmd = new SqlCommand();  
string strSQL = "select MaSinhVien from SinhVien";
```

- + Thuộc tính:

```
cmd.CommandText = <chuỗi tương tác>;  
cmd.Connection = <tên biến kết nối>;
```

- + Phương thức:

- ExecuteReader(): dạng truy vấn dữ liệu
 - Chuỗi tương tác: câu truy vấn Select
 - Đối tượng lưu trữ:

```
SqlDataReader rdr = cmd.ExecuteReader();
```

- ExecuteNonQuery(): dạng hành động (thêm/ sửa/ xóa dữ liệu)
 - Chuỗi tương tác: câu truy vấn Insert/ Update/ Delete
 - Đối tượng lưu trữ:

```
bool kq = cmd.ExecuteNonQuery();
```

- ExecuteScalar(): dạng truy vấn lấy một giá trị (cột đầu tiên, dòng đầu tiên tập kết quả)
 - Chuỗi tương tác: câu truy vấn Select với Count()/ Sum()/ Max()/ Min()
 - Đối tượng lưu trữ:

```
int kq = (int)cmd.ExecuteScalar();
```

Cơ sở dữ liệu mẫu sử dụng trong bài thực hành:

Cơ sở dữ liệu QLSinhVien có các bảng như sau:

Khoa (MaKhoa, TenKhoa)

Lop (MaLop, TenLop, *MaKhoa*)

SinhVien (MaSinhVien, HoTen, NgaySinh, *MaLop*)

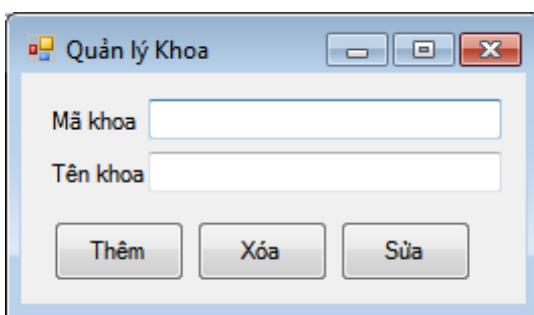
MonHoc (MaMonHoc, TenMonHoc)

Diem (MaSinhVien, MaMonHoc, Diem)

☞ **Lưu ý:** sinh viên tùy chọn nhập dữ liệu mẫu cho mỗi bảng (ít nhất 5 mẫu tin)

1.2. Giới thiệu bài tập mẫu

Thiết kế Form quản lý thông tin Khoa như hình mẫu sau:



Yêu cầu:

- Chức năng thêm dữ liệu: nhập thông tin vào textbox Mã khoa, Tên khoa sau đó nhấn chọn button Thêm thì hai thông tin Mã khoa, Tên khoa vừa nhập sẽ được lưu vào bảng Khoa trong cơ sở dữ liệu QLSinhVien.
- Chức năng xóa dữ liệu: nhập thông tin vào textbox Mã khoa, sau đó nhấn chọn button Xóa thì thông tin của Khoa có mã khoa vừa nhập sẽ bị xóa khỏi bảng Khoa trong cơ sở dữ liệu.
- Chức năng sửa dữ liệu: nhập thông tin vào textbox Mã khoa, Tên khoa sau đó nhấn button Sửa thì thông tin của bảng Khoa trong cơ sở dữ liệu sẽ thay đổi (Tên khoa thay đổi tương ứng Mã khoa vừa nhập).

Hướng dẫn:

Bước 1: Khai báo thư viện dùng cho việc thực hiện kết nối cơ sở dữ liệu:

```
//...  
using System.Data.SqlClient;
```

Bước 2: Khai báo đối tượng SqlConection

```

namespace AD01
{
    public partial class QLKhoa : Form
    {
        SqlConnection conssql;
        public QLKhoa()
        {
            InitializeComponent();
        }
}

```

Bước 3: Khởi tạo các thông số cấu hình cho đối tượng kết nối theo 2 cách:

- (1): Chứng thực quyền Windows.
- (2): Chứng thực quyền SQL Server.

Thực hiện việc Thêm dữ liệu từ textbox Mã khoa và Tên khoa.

```

namespace AD01
{
    public partial class QLKhoa : Form
    {
        SqlConnection conssql;
        SqlConnection conn;
        public QLKhoa()
        {
            InitializeComponent();
            conn = new SqlConnection("Data Source=<Tên_Server>;" +
                "Initial Catalog=QLSINHVIEN;Integrated Security=True");
            conssql = new SqlConnection("Data Source=<Tên_Server>;" +
                "Initial Catalog=QLSINHVIEN;User ID=...; Password=...");
        }
}

```

Trong sự kiện của button Thêm chúng ta xử lý như sau:

```

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

```

```

//Kiem tra ket noi truoc khi mo
if (connsql.State == ConnectionState.Closed)
{
    connsql.Open();
}
//Xac dinh chuoi truy van
string insertString;
insertString = "insert into Khoa values('" +
    txtMaKhoa.Text + "',N'" + txtTenKhoa.Text + "')";
//Khai bao commamnd moi
SqlCommand cmd = new SqlCommand(insertString, connsql);
//Goi ExecuteNonQuery de gui command
cmd.ExecuteNonQuery();
//Kiem tra ket noi truoc khi dong
if (connsql.State == ConnectionState.Open)
{
    connsql.Close();
}
//Hien thi thong bao
MessageBox.Show("Thanh cong");
}
catch (Exception ex)
{
    MessageBox.Show("That bai");
}
}

```

Tương tự như việc thêm dữ liệu để thực hiện xóa và sửa chúng ta chỉ cần thực hiện thay đổi chuỗi câu lệnh điều khiển ở câu lệnh điều khiển trong quy trình trên tương ứng với sự kiện của việc xóa hay sửa như sau:

```

//Chuoi ket noi ung voi yeu cau sua du lieu
string updateString = "update Khoa set TenKhoa=''" +
txtTenKhoa.Text + "' where MaKhoa='"+ txtMaKhoa.Text + "'";
//Chuoi ket noi ung voi yeu cau xoa du lieu
string deleteString = "delete Khoa where MaKhoa=''" +
txtMaKhoa.Text + "'";

```

2. Bài tập tại lớp

Bài 1: Thiết kế form Quản lý môn học như hình mẫu sau:

Yêu cầu:

- Chức năng thêm dữ liệu:
 - + Nhập thông tin cho trường Mã môn học, Tên môn học sau đó nhấn vào button Thêm thì hai thông tin Mã môn học, Tên môn học vừa nhập sẽ được lưu vào bảng MonHoc trong cơ sở dữ liệu QLSinhVien.
 - + Mã môn học, Tên môn học không được để trống.
- Chức năng xóa dữ liệu:
 - + Nhập thông tin cho trường Mã môn học, sau đó nhấn button Xóa thì thông tin của môn học có mã môn học vừa nhập sẽ bị xóa khỏi bảng MonHoc trong cơ sở dữ liệu.
 - + Mã môn học không được để trống, Tên môn học không nhập dữ liệu.
- Chức năng sửa dữ liệu:
 - + Nhập thông tin cho trường Mã môn học, Tên môn học sau đó nhấn button Sửa thì thông tin của Môn học trong cơ sở dữ liệu sẽ thay đổi (tên môn học thay đổi tương ứng mã môn học vừa nhập vào).
 - + Mã môn học, Tên môn học không được để trống.

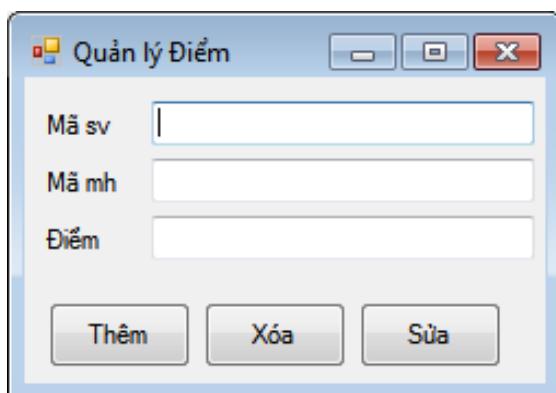
3. Bài tập nâng cao

Bài 1: Thiết kế form Quản lý sinh viên như hình mẫu sau:

Yêu cầu:

- Chức năng thêm dữ liệu:
 - + Nhập thông tin cho trường Mã lớp, Mã sinh viên, Tên sinh viên, Ngày sinh sau đó nhấn vào button Thêm thì 4 thông tin vừa nhập sẽ được lưu vào bảng SinhVien trong cơ sở dữ liệu QLSinhVien.
 - + Mã lớp, Mã sinh viên, Tên sinh viên, ngày sinh không được để trống.
- Chức năng xóa dữ liệu:
 - + Nhập thông tin cho trường Mã sinh viên, Tên sinh viên, sau đó nhấn button Xóa thì thông tin của sinh viên có mã sinh viên và tên trùng với thông tin vừa nhập sẽ bị xóa khỏi bảng SinhVien trong cơ sở dữ liệu.
 - + Mã sinh viên, Tên sinh viên không được để trống, Mã lớp, Ngày sinh không nhập dữ liệu.
- Chức năng sửa dữ liệu:
 - + Nhập thông tin cho trường Mã sinh viên, Mã lớp, Tên sinh viên, Ngày sinh sau đó nhấn button Sửa thì thông tin của sinh viên trong cơ sở dữ liệu sẽ thay đổi (Mã lớp, Tên sinh viên, Ngày sinh thay đổi nếu được nhập liệu, trường nào không nhập liệu thì không thay đổi tương ứng với mã sinh viên đã nhập).
 - + Mã sinh viên không được để trống, Mã lớp, Tên sinh viên, Ngày sinh phải có ít nhất một trường có dữ liệu.

Bài 2: Thiết kế form Quản lý Điểm như hình mẫu sau:



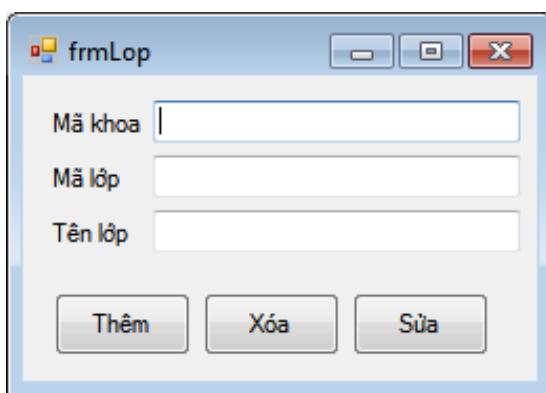
Yêu cầu:

- Chức năng thêm dữ liệu:
 - + Nhập thông tin cho trường Mã sinh viên, Mã môn học, Điểm sau đó nhấn vào button Thêm thì 3 thông tin vừa nhập sẽ được lưu vào bảng Diem trong cơ sở dữ liệu QLSinhVien.
 - + Mã môn học, Mã sinh viên, Điểm không được để trống.
- Chức năng xóa dữ liệu:

- + Nhập thông tin cho trường Mã sinh viên, Mã môn học, sau đó nhấn button Xóa thì một dòng có mã sinh viên và mã môn học tương ứng với thông tin vừa nhập sẽ bị xóa khỏi bảng Diem trong cơ sở dữ liệu.
- + Mã sinh viên, Mã môn học không được để trống, Điểm không nhập dữ liệu.
- Chức năng sửa dữ liệu:
 - + Nhập thông tin cho trường Mã sinh viên, Mã môn học, Điểm sau đó nhấn button Sửa thì điểm của sinh viên theo môn học trong cơ sở dữ liệu sẽ thay đổi (điểm thay đổi tương ứng Mã sinh viên, Mã môn học vừa nhập vào).
 - + Mã sinh viên, Mã môn học không được để trống, Điểm phải được nhập dữ liệu và chỉ được nhập số.

4. Bài tập về nhà

Bài 1: Thiết kế form Quản lý Lớp như hình mẫu sau:



Yêu cầu:

- Chức năng thêm dữ liệu:
 - + Nhập thông tin cho trường Mã khoa, Mã lớp, Tên lớp, sau đó nhấn vào button Thêm thì 3 thông tin Mã khoa, Mã lớp, Tên lớp vừa nhập sẽ được lưu vào bảng Lop trong cơ sở dữ liệu QLSinhVien.
 - + Mã khoa, Mã lớp, Tên lớp không được để trống.
- Chức năng xóa dữ liệu:
 - + Nhập thông tin cho trường Mã lớp, sau đó nhấn button Xóa thì thông tin của Lớp có mã lớp vừa nhập sẽ bị xóa khỏi bảng Lop trong cơ sở dữ liệu.
 - + Mã lớp không được để trống, Mã khoa, Tên lớp không nhập dữ liệu.
- Chức năng sửa dữ liệu:
 - + Nhập thông tin cho trường Mã lớp, Mã khoa, Tên lớp, sau đó nhấn button Sửa thì thông tin của Lớp trong cơ sở dữ liệu sẽ thay đổi (Mã khoa, Tên lớp thay đổi tương ứng Mã lớp vừa nhập vào).
 - + Mã lớp không để trống, Mã khoa, Tên lớp phải có ít nhất 1 trường có dữ liệu.

Trường ĐH CNTP TP.HCM
 Khoa: CNTT
 Bộ môn: CNPM
 Môn: Lập trình .NET

BÀI 8.

LẬP TRÌNH TƯƠNG TÁC

CƠ SỞ DỮ LIỆU

(tiếp theo)



A. MỤC TIÊU:

- Sử dụng được đối tượng SqlCommand.
- Sử dụng SqlCommand thực hiện thêm, xóa, sửa cho một bảng dữ liệu.
- Sử dụng SqlDataReader hiển thị khóa ngoại cho combobox.
- Sử dụng SqlDataReader kiểm tra khóa chính/ khóa ngoại trước khi Thêm/ Xóa/ Sửa dữ liệu.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

Các thành phần của ADO.NET

- **Command:** Đối tượng SqlCommand cho phép chúng ta chọn kiểu tương tác mà chúng ta muốn thực hiện với cơ sở dữ liệu.

- + Khai báo biến:

```
SqlCommand cmd = new SqlCommand();
string strSQL = "select MaSinhVien from SinhVien";
```

- + Thuộc tính:

```
cmd.CommandText = <chuỗi tương tác>;
cmd.Connection = <tên biến kết nối>;
```

- + Phương thức:

- ExecuteReader(): dạng truy vấn dữ liệu
 - Chuỗi tương tác: câu truy vấn Select
 - Đối tượng lưu trữ:

```
SqlDataReader rdr = cmd.ExecuteReader();
```

- ExecuteNonQuery(): dạng hành động (thêm/ sửa/ xóa dữ liệu)

- Chuỗi tương tác: câu truy vấn Insert/ Update/ Delete
- Đối tượng lưu trữ:

```
bool kq = cmd.ExecuteNonQuery();
```

- ExecuteScalar(): dạng truy vấn lấy một giá trị (cột đầu tiên, dòng đầu tiên tập kết quả)
 - Chuỗi tương tác: truy vấn Select với Count()/ Sum()/ Max()/ Min()
 - Đối tượng lưu trữ:

```
int kq = (int)cmd.ExecuteScalar();
```

Trước khi thực hành sinh viên lưu ý:

- Cơ sở dữ liệu mẫu: Sử dụng cơ sở dữ liệu QLSinhVien trong bài thực hành 7
 - Khoa (MaKhoa, TenKhoa)
 - Lop (MaLop, TenLop, MaKhoa)
 - SinhVien (MaSinhVien, HoTen, NgaySinh, MaLop)
 - MonHoc (MaMonHoc, TenMonHoc)
 - Diem (MaSinhVien, MaMonHoc, Diem)

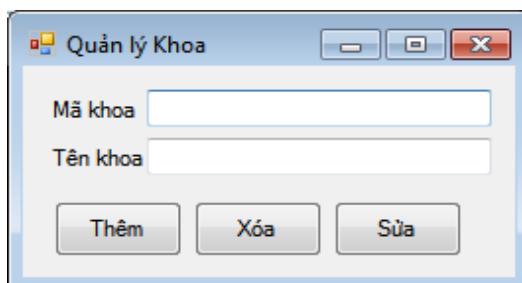
- Nhận diện các tham số kết nối SQL Server cần thiết như:

Server name/Data Source;
 Database/Initial Catalog;
 Login/User ID;
 Password/ Password;

- Thực hiện lưu cơ sở dữ liệu theo tùy chọn Generate Script (có dữ liệu/ hoặc không có dữ liệu kèm theo).

1.2 Giới thiệu bài tập mẫu

Bài 1: Thiết kế form Quản lý Khoa như hình mẫu sau:



Yêu cầu:

- Chức năng thêm dữ liệu:
 - + Nhập thông tin cho trường Mã khoa, Tên khoa, sau đó nhấn vào button Thêm thì 2 thông tin Mã khoa, Tên khoa vừa nhập sẽ được lưu vào bảng Khoa trong cơ sở dữ

liệu QLSinhVien.

- + Kiểm tra Mã khoa không được trùng khi thêm mới (Kiểm tra khóa chính).
- Chức năng xóa dữ liệu: nhập thông tin cho trường Mã khoa, sau đó nhấn button Xóa thì thông tin của Khoa có mã khoa vừa nhập sẽ bị xóa khỏi bảng Khoa trong cơ sở dữ liệu.
- Chức năng sửa dữ liệu: nhập thông tin cho trường Mã khoa, Tên khoa sau đó nhấn button Sửa thì thông tin của Khoa trong cơ sở dữ liệu sẽ thay đổi (Tên khoa thay đổi tương ứng Mã khoa vừa nhập vào).

Hướng dẫn:

Công việc 1: Viết phương thức thêm, xóa, sửa vào bảng Khoa (bài thực hành số 7)

Công việc 2:

- Định nghĩa phương thức kiểm tra trùng Mã khoa

```
bool KT_MaKhoa(string ma)
{
    try
    {
        //Mo ket noi
        connsql.Open();

        //Xay dung chuoi truy van
        string selectString = "select count(*) from Khoa
                               where MaKhoa = '" + ma + "'";

        //Khai bao command voi chuoi truy van va bien ket noi
        SqlCommand cmd = new SqlCommand(selectString, connsql);
        //goi ham thuc thi truy van
        int count = (int)cmd.ExecuteScalar();

        //Dong ket noi
        connsql.Close();
        //Xu ly ket qua truy van
        if (count >= 1)
            return false;
        return true;
    }
    catch (Exception ex) { return false; }
}
```

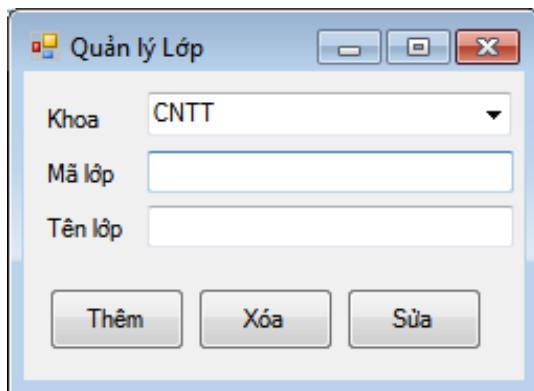
- Bổ sung việc gọi phương thức kiểm tra trùng mã khoa thực thi

```

private void btnThem_Click(object sender, EventArgs e)
{
    try
    {
        if (KT_MaKhoa(txtMaKhoa.Text) == true)
        {
            connsql.Open();
            string insertString;
            insertString = "insert into Khoa values('" +
                txtMaKhoa.Text + "',N'" + txtTenKhoa.Text + "')";
            SqlCommand cmd = new SqlCommand(insertString, connsql);
            cmd.ExecuteNonQuery();
            connsql.Close();
            MessageBox.Show("Thanh cong");
        }
        else { MessageBox.Show("Trung ma khoa"); }
    }
    catch (Exception ex)
    {
        MessageBox.Show("That bai");
    }
}

```

Bài 2: Thiết kế form Quản lý Lớp như hình mẫu sau:



Yêu cầu:

- Hiển thị mã khoa trong bảng Khoa lên comboBox Khoa.
- Chức năng thêm dữ liệu:
 - Chọn Mã khoa trong comboBox, Nhập mã lớp, Tên lớp, sau đó nhấn vào button Thêm thì các thông tin Mã khoa, Mã lớp, Tên lớp vừa nhập sẽ được lưu vào bảng Lop trong cơ sở dữ liệu QLSinhVien.

- + Kiểm tra Mã lớp không được trùng khi thêm mới (Kiểm tra khóa chính).
- Chức năng xóa dữ liệu: nhập thông tin cho trường Mã lớp, sau đó nhấn button Xóa thì thông tin của Lớp có Mã lớp vừa nhập sẽ bị xóa khỏi bảng Lop trong cơ sở dữ liệu.
- Chức năng sửa dữ liệu: chọn Mã khoa, nhập Mã lớp, Tên khoa, sau đó nhấn button Sửa thì thông tin của Lop trong cơ sở dữ liệu sẽ thay đổi (Tên lớp, Mã khoa thay đổi tương ứng Mã lớp vừa nhập vào).

Hướng dẫn:

Công việc 1: Dùng SqlDataReader hiển thị khóa ngoại cho combobox Khoa.

- Bước 1: Khai báo thư viện thực hiện kết nối cơ sở dữ liệu. Thực hiện việc kết nối CSDL thông qua đối tượng SqlConnection chúng ta có đối tượng connsql.
- Bước 2: Viết phương thức load dữ liệu vào combobox Khoa như sau:

```
private void LoadKhoa_ComboBox()
{
    //Mo ket noi CSDL
    connsql.Open();

    //Thiet lap chuoi truy van dang hien thi du lieu
    string selectString = "select * from Khoa";
    //Xay dung command truy van
    SqlCommand cmd = new SqlCommand(selectString, connsql);
    //Goi phuong thuc truy van dang hien thi du lieu
    SqlDataReader rd = cmd.ExecuteReader();
    //Hien thi ket qua truy van
    while (rd.Read())
    {
        cboKhoa.Items.Add(rd["MaKhoa"].ToString());
    }
    rd.Close();

    //Dong ket noi
    connsql.Close();
}
```

- Bước 3: sự kiện Form Load gọi thực thi phương thức LoadKhoa_ComboBox()

```

private void frmLop_Load(object sender, EventArgs e)
{
    LoadKhoa_ComboBox();
}

```

Công việc 2: Dùng SqlDataReader kiểm tra khóa chính trước khi thêm dữ liệu

– Bước 1: viết phương thức kiểm tra khóa chính cho bảng Lop dùng SqlDataReader. Tham số truyền vào là mã lớp, kết quả trả về là true khi mã lớp không tồn tại trong bảng Lop (thêm được dữ liệu), trả về là false khi tồn tại mã lớp trong bảng lớp (không thêm được dữ liệu và thông báo lỗi trùng khóa chính):

```

public bool KT_KhoaChinh(string pMa)
{
    //Mo ket noi CSDL
    connsql.Open();

    //Thiet lap chuoi truy van dang hien thi du lieu
    string selectString = "select * from Lop where MaLop =
        '" + pMa + "'";

    //Xay dung SqlCommand truy van
    SqlCommand cmd = new SqlCommand(selectString, connsql);
    //Goi phuong thuc truy van dang hien thi du lieu
    SqlDataReader rd = cmd.ExecuteReader();

    //Xu ly ket qua
    if (rd.HasRows)
    {
        rd.Close();
        connsql.Close();
        return false;
    }
    else
    {
        rd.Close();
        connsql.Close();
        return true;
    }
}

```

– Bước 2: Gọi phương thức KT_KhoaChinh() kiểm tra khóa chính trước khi thêm như sau:

```

private void btnThem_Click(object sender, EventArgs e)
{
    try
    {
        if (KT_KhoaChinh(txtMaLop.Text) == true)
        {
            connsql.Open();
            string insertString;
            insString = "insert into Lop values('" + txtMaLop.Text
            + "', '" + txtMaMaKhoa.SelectedItem.ToString() + "', N'"
            + txtTenKhoa.Text + ")";
            SqlCommand cmd = new SqlCommand(insertString, connsql);
            cmd.ExecuteNonQuery();
            connsql.Close();
            MessageBox.Show("Thanh cong");
        }
        else { MessageBox.Show("Trung ma khoa"); }
    }
    catch (Exception ex)
    {
        MessageBox.Show("That bai");
    }
}

```

Công việc 3: Dùng SqlDataReader kiểm tra khóa ngoại trước khi Xóa/Sửa dữ liệu.
Thực hiện tương tự *Công việc 2*.

☞ **Lưu ý:** Kiểm tra ràng buộc giao diện cho các trường hợp không cho rỗng. Ví dụ không cho rỗng mã lớp khi thêm:

```

private void btnThem_Click(object sender, EventArgs e)
{
    try
    {
        if (txtMaLop.Text == string.Empty)
        {
            MessageBox.Show("Bạn phải nhập " + lblMaLop.Text);
            txtMaLop.Focus();
            return;
        }
    }
}

```

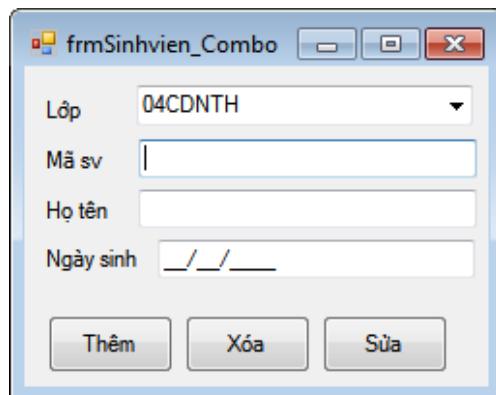
```

        if (KT_KhoaChinh(txtMaLop.Text) == true)
        {
            connsql.Open();
            string insertString;
            insertString = "insert into Lop values('" +
txtMaLop.Text + "','" + txtMaMaKhoa.SelectedItem.ToString() +
"',N'" + txtTenKhoa.Text + "')";
            SqlCommand cmd = new SqlCommand(insertString, connsql);
            cmd.ExecuteNonQuery();
            connsql.Close();
            MessageBox.Show("Thanh cong");
        }
        else
        {
            MessageBox.Show("Trung ma khoa");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("That bai");
    }
}

```

2. Bài tập tại lớp

Bài 1: Thiết kế form SinhVien theo mẫu sau:



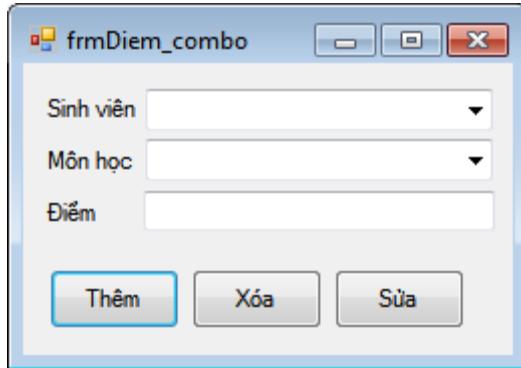
Yêu cầu:

- Hiển thị mã lớp trên comboBox Lớp.
- Khi nhấn button Thêm:
 - + Mã sinh viên mới nhập vào không được trùng mã sinh viên có trong bảng

SinhVien (thông báo nếu trùng).

- + Tất cả thông tin phải được nhập dữ liệu.
- + Sinh viên phải có tuổi từ 17 trở lên (chỉ tính theo năm sinh).
- + Thêm sinh viên vào bảng SinhVien nếu thỏa tất cả yêu cầu trên.
- Khi nhấn button Xóa/ Sửa: kiểm tra khóa chính/khóa ngoại khi thao tác.

Bài 2: Cập nhật dữ liệu trên bảng Diem

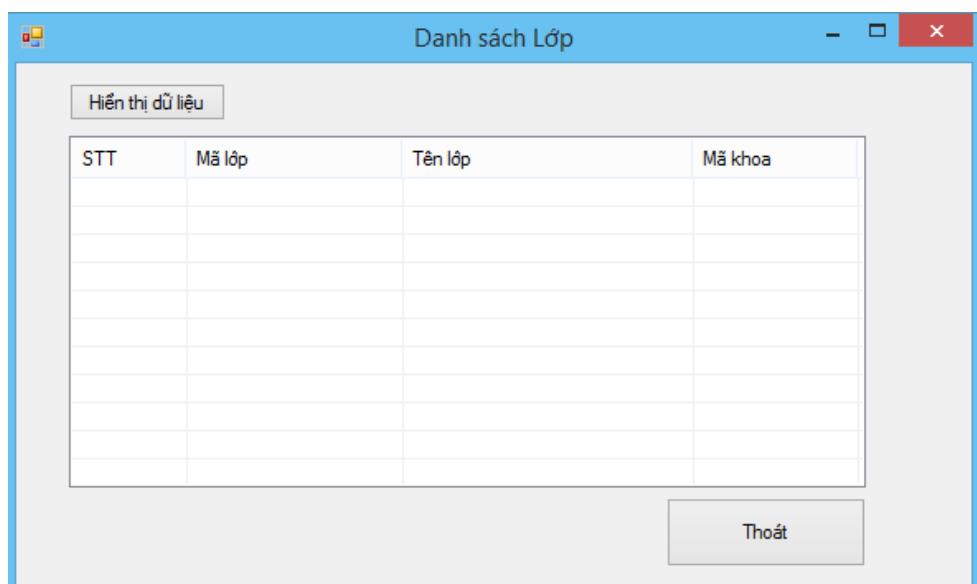


Yêu cầu:

- Hiển thị mã sinh viên trên combobox Sinh viên
- Hiển thị mã môn học lên combobox Môn học.
- Tất cả thông tin phải được nhập dữ liệu, Điểm phải nằm trong khoảng (0-10)
- Khi nhấn button Thêm: chỉ được phép thêm điểm mới nếu mã sinh viên và mã môn học đã tồn tại trong cơ sở dữ liệu, và dữ liệu mới nhập vào không được trùng khóa chính có trong bảng Điểm (thông báo nếu trùng).
- Khi nhấn button Xóa/ Sửa: kiểm tra khóa chính/khóa ngoại khi thao tác.

3. Bài tập nâng cao

Bài 1: Thiết kế form hiển thị danh sách Lớp trên listview:



Yêu cầu:

- Khi nhấn vào button Hiển thị dữ liệu thì thông tin lớp học trong bảng Lop sẽ được hiển thị như hình sau:

Danh sách Lớp

Hiển thị dữ liệu			
STT	Mã lớp	Tên lớp	Mã khoa
1	05DHKT2	Kế toán ngân hàng khóa 05	KT
2	05DHTH1	Công nghệ phần mềm khóa 05	CNTT
3	05DHTH2	Lập trình mạng khóa 05	CNTT
4	06DHHH1	Hóa phân tích khóa 06	HH
5	06DHKT1	Thuế doanh nghiệp khóa 06	KT
6	06DHTH1	Công nghệ phần mềm khóa 06	CNTT

Thoát

- Khi nhấn vào button Thoát thì thực hiện chức năng đóng form.

Bài 2: Thiết kế form hiển thị danh sách điểm của sinh viên theo môn học trên listview như hình sau:

Danh sách sinh viên

Môn học		Lọc dữ liệu		
STT	Tên SV	Tên môn học	Điểm	
1	Trần Thanh Tùng	12/10/1993 12:00:0...	Toán Cao cấp	4
2	Trần Thanh Tùng	12/10/1993 12:00:0...	Mac - Lenin	7
3	Nguyễn Thanh Bình	2/4/1993 12:00:0 ...	Toán Cao cấp	6
4	Nguyễn Thanh Bình	2/4/1993 12:00:0 ...	Mac - Lenin	3
5	Nguyễn Thanh Bình	2/4/1993 12:00:00 ...	Tin học đại cương	5
6	Nguyễn Đạt Nhân	3/15/1992 12:00:00...	Toán Cao cấp	9
7	Nguyễn Đạt Nhân	3/15/1992 12:00:00...	Mac - Lenin	10
8	Nguyễn Đạt Nhân	3/15/1992 12:00:00...	Tin học đại cương	9
9	Nguyễn Gia Bảo	5/11/1993 12:00:00...	Toán Cao cấp	3
10	Nguyễn Gia Bảo	5/11/1993 12:00:00...	Mac - Lenin	6
11	Nguyễn Thanh Lan	6/12/1993 12:00:00...	Toán Cao cấp	5
12	Nguyễn Thanh Lan	6/12/1993 12:00:00...	Tin học đại cương	7
13	Lâm Văn Tuấn	8/3/1993 12:00:00 ...	Mac - Lenin	8
14	Lâm Văn Tuấn	8/3/1993 12:00:00 ...	Tin học đại cương	8

Tổng điểm

Yêu cầu:

- Trong sự kiện Load của Form:
 - + Hiển thị mã môn học trong bảng dữ liệu MonHoc lên combobox Môn học, thêm một item là khoảng trắng (hiểu là tất cả môn học).

- + Hiển thị dữ liệu lên listview tất cả các thông tin có trong cơ sở dữ liệu (gồm Số thứ tự, Tên sinh viên, Ngày sinh, Tên môn học, Điểm).
- + Tính tổng điểm của cột Điểm hiển thị trên listview và gán vào textbox Tổng điểm.
- Nếu chọn một mã môn học cụ thể (khác khoảng trắng) trong combobox Môn học và nhấn button lọc dữ liệu, dữ liệu trên listview sẽ trở thành:

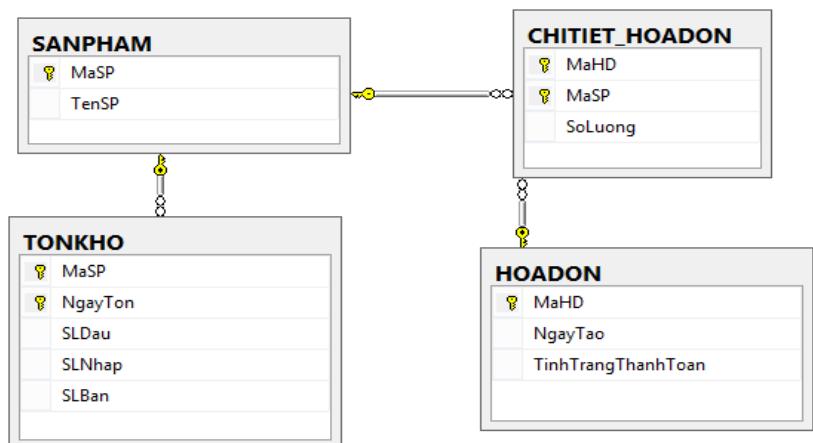
STT	Tên SV	Ngày sinh	Tên môn học	Điểm
1	Trần Thanh Tùng	12/10/1993 12:00:0...	Mac - Lênin	7
2	Nguyễn Thành Bình	2/4/1993 12:00:00 ...	Mac - Lênin	3
3	Nguyễn Đạt Nhân	3/15/1992 12:00:00...	Mac - Lênin	10
4	NGuyễn Gia Bảo	5/11/1993 12:00:00...	Mac - Lênin	6
5	Lâm Văn Tuấn	8/3/1993 12:00:00 ...	Mac - Lênin	8

Tổng điểm: 34

- + Chỉ hiển thị trên listview thông tin Điểm của những sinh viên có Mã môn học được chọn.
- + Tổng điểm cũng được cập nhập theo kết quả hiển thị trên ListView
- + Nếu chọn khoảng trắng trong combobox và nhấn lọc dữ liệu thì:
 - Hiển thị thông tin điểm của tất cả môn học của từng sinh viên.
 - Cập nhập tổng điểm theo kết quả hiển thị trên listview

4. Bài tập về nhà

Bài 1: Cho cơ sở dữ liệu QLCuaHang như sau :



Ghi chú:

- Cột MaSP trong bảng SANPHAM, MaHD trong bảng HOADON là trường dữ liệu tự động tăng.
- Dữ liệu ban đầu bảng như sau:

MaSP	TenSP	MaSP	NgayTon	SLDau	SLNhap	SLBan
1	Trà xanh	1	2016-10-29	50	0	0
2	Mức dâu	2	2016-10-29	50	0	0
3	Cafe	3	2016-10-29	50	0	0

Yêu cầu:

Thiết kế form chính chứa thanh menu gồm các chức năng:

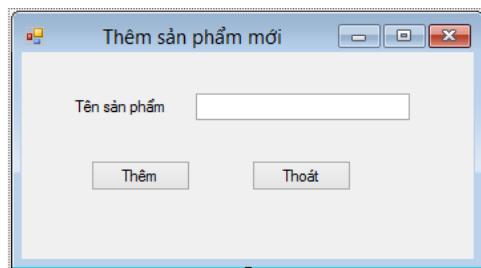
- Tạo tồn kho (thực hiện 1 lần mỗi ngày, trước khi bán hàng/ nhập hàng)
- Thêm sản phẩm mới
- Bán hàng

Thiết kế chức năng tồn kho:

Tạo dữ liệu cho bảng tồn kho cho tất cả sản phẩm trong theo ngày hiện tại, biết rằng:

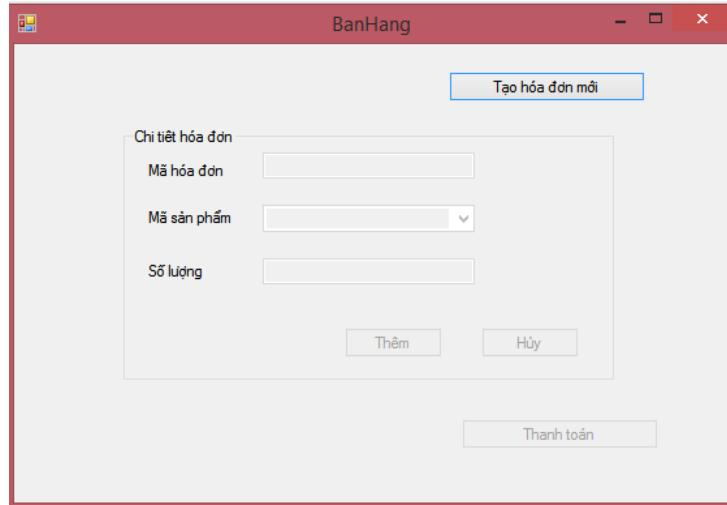
- SLTon = SLTon (ngày trước) + Tổng SL nhập (ngày trước) – Tổng SL xuất (ngày trước)
- Tổng số lượng nhập = 0
- Tổng số lượng xuất = 0

Tạo form nhập dữ liệu cho bảng SANPHAM theo mẫu sau:



- Thêm: Thêm dòng tin mới vào bảng Sản phẩm
- Thêm 1 dòng tin mới vào bảng Tồn kho với SLDau = 0, Tổng số lượng nhập = 0, Tổng số lượng bán = 0.

Thiết kế chức năng bán hàng:



- Textbox mã hóa đơn, mã sản phẩm, số lượng, button Thêm, Hủy, thanh toán bị mờ
- Tạo hóa đơn mới:
 - + Thêm một hóa đơn mới vào bảng hóa đơn với mã hóa đơn tự động tăng, ngày tạo hóa đơn là ngày hiện tại, tình trạng hóa đơn = 0 (chưa thanh toán)
 - + Lấy mã hóa đơn vừa tạo ghi vào textbox mã hóa đơn
 - + Hiển thị textbox mã sản phẩm, số lượng, button Thêm, Hủy, Thanh toán
- Button Thêm:
 - + Thêm một dòng tin vào bảng chi tiết hóa đơn
 - Hiển thị mã sản phẩm trên comboBox Sản phẩm
 - Kiểm tra khóa chính/ khóa ngoại trong bảng chi tiết hóa đơn
 - Số lượng bán phải lớn hơn 0 và nhỏ hơn số lượng tồn trong bảng tồn kho
 - + Cập nhật lại tổng số lượng bán trong bảng tồn kho ($Tổng SLBan = Tổng SLBan - số lượng$)

Gợi ý: (có thể sử dụng một trong hai cách)

Cách 1. Sử dụng SqlCommand Update để cập nhật.

Cách 2. Sử dụng Trigger để cập nhật.

Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: CNPM Môn: Lập trình .NET	BÀI 9. LẬP TRÌNH TƯƠNG TÁC CƠ SỞ DỮ LIỆU (tiếp theo)	
--	---	--

A. MỤC TIÊU:

- Sử dụng các đối tượng DataSet, DataTable, DataColumn, DataRow hiển thị dữ liệu từ cơ sở dữ liệu lên các đối tượng combobox, datagridview.
- Sử dụng đối tượng SqlDataAdapter, DataSet thực hiện thêm, xóa, sửa dữ liệu.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

- **DataSet**: nhận dữ liệu từ DataAdapter, và cung cấp các phương thức xử lý dữ liệu. DataSet có nhiều đối tượng đi kèm theo như sau:

- + DataTable (tương đương với một bảng trong database)
- + DataRow (tương đương với một dòng)
- + DataColumn (tương đương với một cột)

▪ Khởi tạo

```
DataSet ds_QLSinhVien = new DataSet();
```

▪ Sử dụng

- + Hiển thị dữ liệu lên đối tượng comboBox: lấy đối tượng DataTable trong DataSet làm bảng dữ liệu nguồn tương ứng thuộc tính DataSource của ComboBox.

Ví dụ:

```
cboKhoa.DataSource = ds_QKhoa.Tables[0];
cboKhoa.DisplayMember = "TenKhoa";
cboKhoa.ValueMember = "MaKhoa";
```

- + Hiển thị dữ liệu lên đối tượng datagridview: sử dụng DataTable trong DataSet làm bảng dữ liệu nguồn tương ứng thuộc tính DataSource của datagridview.

Ví dụ:

```
gdvKhoa.DataSource= ds_QLSinhVien.Tables[0];
```

- **SqlDataAdapter**: là cầu nối của database và dataset. SqlDataAdapter chứa các lệnh SQL (dạng Select) và đối tượng connection để đọc và ghi dữ liệu.

- **Khởi tạo:**

```
string strSelect = "select MaSinhVien, TenSinhVien  
                  from SinhVien";  
SqlDataAdapter daSinhVien = new SqlDataAdapter(strSelect, cnn);
```

- **Đỗ dữ liệu vào DataSet**

Sử dụng phương thức Fill của SqlDataAdapter để đỗ dữ liệu vào DataSet.

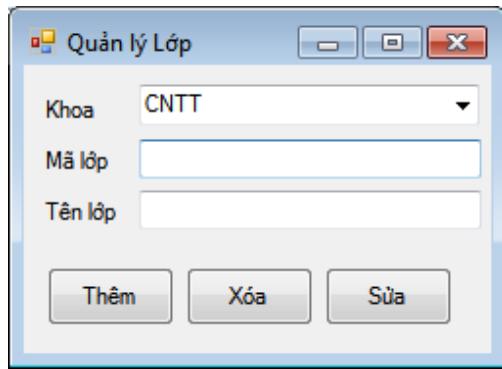
Ví dụ:

```
daSinhVien.Fill(ds_QLSinhVien, "SinhVien");
```

- **Tạo một command dạng insert vào cơ sở dữ liệu sử dụng SqlDataAdapter**
 - Khai báo và khởi tạo một SqlDataAdapter mới.
 - Khai báo và khởi tạo một DataRow là dòng tin mới trong DataTable của DataSet.
 - Thêm dòng DataRow vừa tạo vào DataSet.
 - Khai báo và khởi tạo một SqlCommandBuilder mới tương ứng với SqlDataAdapter.
 - Gọi phương thức Update của SqlDataAdapter.
- **Tạo một command dạng update vào cơ sở dữ liệu sử dụng SqlDataAdapter**
 - Tạo trường khóa chính (Primary Key) cho DataTable trong DataSet.
 - Khai báo và khởi tạo một SqlDataAdapter mới.
 - Tìm một DataRow là 1 dòng tin trong DataTable của DataSet.
 - Sửa/Xóa thông tin dòng DataRow vừa tìm được trong DataSet.
 - Khai báo và khởi tạo SqlCommandBuilder mới tương ứng với SqlDataAdapter
 - Gọi phương thức Update của SqlDataAdapter.

1.2 Giới thiệu bài tập mẫu (Sử dụng cơ sở dữ liệu QLSinhVien)

Bài 1: Xây dựng form cho phép thêm, xóa, sửa dữ liệu trên bảng Lop có giao diện như sau:



Yêu cầu:

- Đỗ dữ liệu từ bảng Khoa lên combobox Khoa, với mỗi item con thỏa tính chất nhãn hiển thị là tên khoa (DisplayMember), còn giá trị (ValueMember) là mã khoa khi chọn item.
- Kiểm tra ràng buộc khóa ngoại của bảng lớp dựa vào danh mục khoa trong bảng khoa.
- Kiểm tra mã lớp không trùng khóa chính trong bảng lớp khi thêm.

Hướng dẫn:

Công việc 1: Đỗ dữ liệu từ cơ sở dữ liệu lên combobox (sử dụng SqlDataAdapter)

```
void Load_ComboBox()
{
    //Khai bao DataSet
    DataSet ds = new DataSet();
    //Tao SqlDataAdapter
    string strselect = "Select * from Khoa";
    SqlDataAdapter da = new SqlDataAdapter(strselect, cn);
    //cn la doi tuong SqlConnection da duoc khai bao
    //Do du lieu tu SqlDataAdapter vao DataSet
    da.Fill(ds, "Khoa");
    //Gan du lieu nguon cho comboBox
    cboKhoa.DataSource = ds.Tables[0];
    cboKhoa.DisplayMember = "TenKhoa";
    cboKhoa.ValueMember = "MaKhoa";
}
```

Công việc 2: Viết phương thức kiểm tra trùng khóa, kiểm tra ràng buộc khóa ngoại (TH tuần 8).

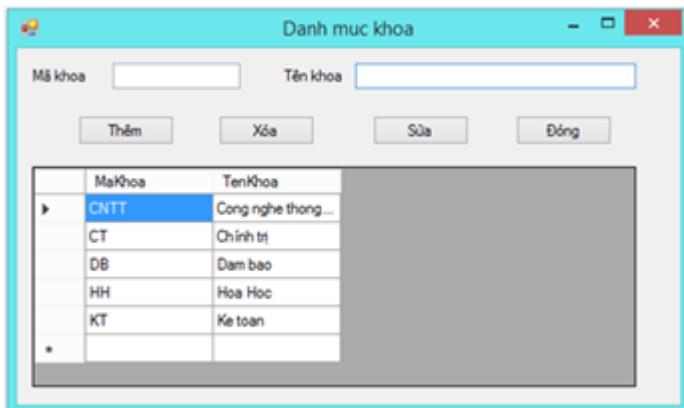
Công việc 3: Viết sự kiện thêm/xóa/sửa trên bảng Lớp (TH tuần 8).

Bài 2: Hiển thị danh sách lớp học lên DataGridView

Hướng dẫn:

```
void Load_DgvLophoc()
{
    //Khai bao DataSet
    DataSet ds = new DataSet();
    //Tạo SqlDataAdapter
    SqlDataAdapter da = new SqlDataAdapter("Select MaLop, TenLop,
                                                MaKhoa, TenKhoa from Khoa, Lop
                                                where Lop.MaKhoa = Khoa.MaKhoa", cn);
    //Đoạn code để lấy dữ liệu từ DataAdapter vào dataSet
    da.Fill(ds, "Lop_Khoa");
    //Gán dữ liệu nguồn cho dataGridView
    dgvLop_Khoa.DataSource = ds.Tables["Lop_Khoa"];
}
```

Bài 3: Hiển thị, cập nhật dữ liệu cho bảng Khoa



Yêu cầu:

- Hiển thị dữ liệu bảng Khoa trên lưới DataGridView khi form được load lên.
- Chức năng Thêm/Xóa/Sửa dữ liệu sử dụng SqlCommandBuilder.

Hướng dẫn:

```
public partial class Form3 : Form
{
    SqlConnection cn;
    SqlDataAdapter da_khoa;
    DataSet ds_khoa;
    DataColumn[] key = new DataColumn[1];
```

```

public Form3()
{
    InitializeComponent();
    cn = new SqlConnection("Data Source=PC24;Initial Catalog
= QLSINHVIEN; User ID=sa; Password=sa2012");
    string strSelect = "select * from Khoa";
    da_khoa = new SqlDataAdapter(strSelect, cn);
    ds_khoa = new DataSet();
    da_khoa.Fill(ds_khoa, "Khoa");
    //them khoa chinh
    key[0] = ds_khoa.Tables["Khoa"].Columns[0];
    ds_khoa.Tables["Khoa"].PrimaryKey = key;
}
void Databingding(DataTable pDT)
{
    txtMaKhoa.DataBindings.Clear();
    txtTenKhoa.DataBindings.Clear();
    //lien ket du lieu tren textbox voi truong du lieu tuong
    //ung trong dataTable
    txtMaKhoa.DataBindings.Add("Text", pDT, "MaKhoa");
    txtTenKhoa.DataBindings.Add("Text", pDT, "TenKhoa");
}
void load_grid()
{
    dataGridView1.DataSource = ds_khoa.Tables[0];
    Databingding(ds_khoa.Tables[0]);
}
private void Form3_Load(object sender, EventArgs e)
{
    load_grid();
}
private void btnThem_Click(object sender, EventArgs e)
{
    //Tao 1 dong du lieu moi
    DataRow newrow = ds_khoa.Tables[0].NewRow();
    newrow["MaKhoa"] = txtMaKhoa.Text;
    newrow["TenKhoa"] = txtTenKhoa.Text;
}

```

```

// Them dong du lieu vua tao vao DataSet
ds_khoa.Tables[0].Rows.Add(newrow);
// Cap nhat trong CSDL
SqlCommandBuilder cB = new SqlCommandBuilder(da_khoa);
// Cap nhat trong dataSet
da_khoa.Update(ds_khoa, "Khoa");
//Xoa textbox
txtMaKhoa.Clear();
txtTenKhoa.Clear();
}
private void btnXoa_Click(object sender, EventArgs e)
{
    //Tim dong du lieu co ma khoa trung voi ma khoa nhap
    //trong textbox
    DataRow dr = ds_khoa.Tables["Khoa"].Rows.Find(txtMaKhoa.Text);
    // Xoa dong du lieu vua tim duoc
    if (dr != null)
    {
        dr.Delete();
    }
    //Cap nhat trong CSDL
    SqlCommandBuilder cB = new SqlCommandBuilder(da_khoa);
    // Cap nhat trong dataSet hien thi
    da_khoa.Update(ds_khoa, "Khoa");
}
private void btnSua_Click(object sender, EventArgs e)
{
    //Tim dong du lieu co ma khoa trung voi ma khoa nhap
    //trong textbox
    DataRow dr = ds_khoa.Tables["Khoa"].Rows.Find(txtMaKhoa.Text);
    // Hieu chinh thong tin dong tin moi vua tim duoc
    if (dr != null)
    {
        dr["TenKhoa"] = txtTenKhoa.Text;
    }
    //Cap nhat trong CSDL
    SqlCommandBuilder cB = new SqlCommandBuilder(da_khoa);
}

```

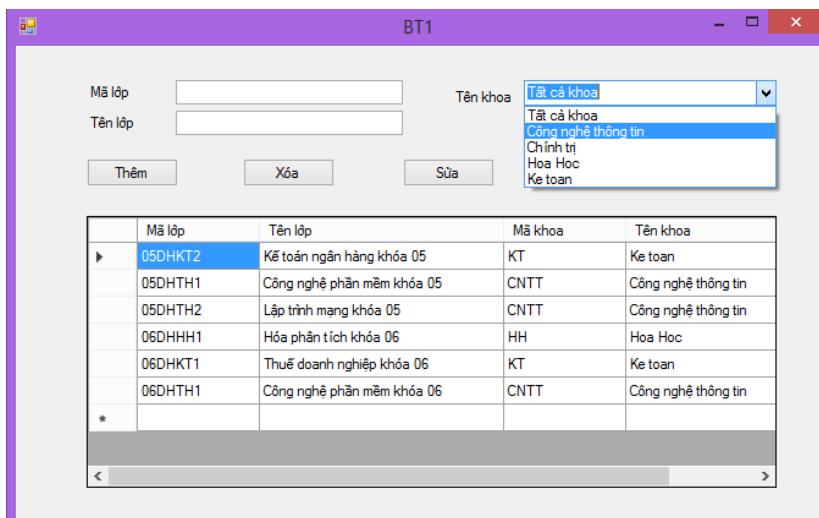
```

    // cap nhat trong dataSet hien thi
    da_khoa.Update(ds_khoa, "Khoa");
}
}

```

2. Bài tập tại lớp

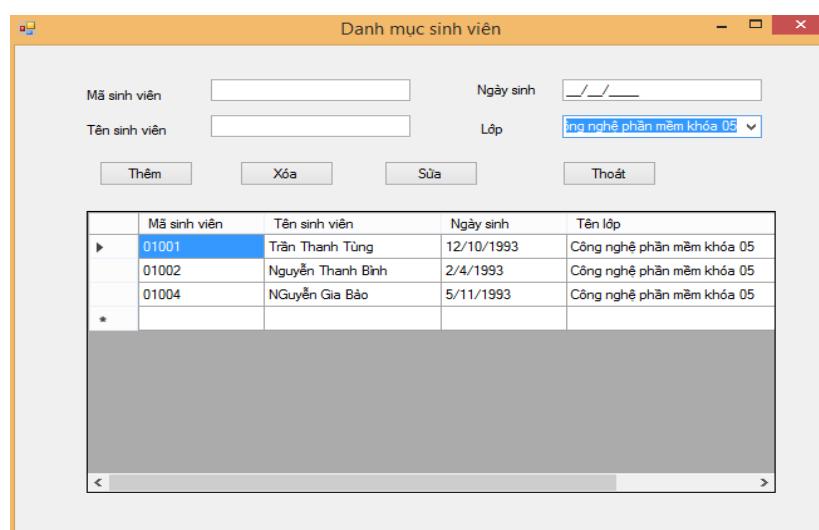
Bài 1: Hiển thị, cập nhật lớp học theo danh mục Khoa



Yêu cầu:

- Hiển thị thông tin Lớp lên datagridview.
- Đỗ dữ liệu lên combobox Khoa, thông tin khoa lấy từ cơ sở dữ liệu với tên khoa là nhãn hiển thị, mã khoa là giá trị khi nhấp chọn item, trong đó dòng đầu tiên là “Tất cả khoa”. Khi chọn một khoa trên combobox, datagridview sẽ được lọc dữ liệu theo.
- Chọn một dòng trên datagridview hiển thị thông tin dòng đang chọn lên các control (textbox, combobox).
- Thực hiện thao tác thêm, xóa, sửa trên bảng lớp (kiểm tra khóa chính, khóa ngoại).

Bài 2: Hiển thị, cập nhật dữ liệu bảng Sinh viên



Yêu cầu:

- Hiển thị thông tin sinh viên trên datagridview.
- Đỗ dữ liệu lên combobox Lớp, thông tin lớp lấy từ cơ sở dữ liệu với tên lớp là nhãn hiển thị, mã lớp là giá trị khi nhấp chọn item, trong đó dòng đầu tiên là “Tất cả lớp”. Khi chọn tên lớp trên combobox Lớp, datagridview sẽ được lọc dữ liệu theo.
- Chọn một dòng trên datagridview hiển thị thông tin dòng đang chọn lên các control (textbox, comboBox).
- Thực hiện thao tác thêm, xóa, sửa trên bảng sinh viên (có kiểm tra khóa chính, khóa ngoại).

3. Bài tập nâng cao

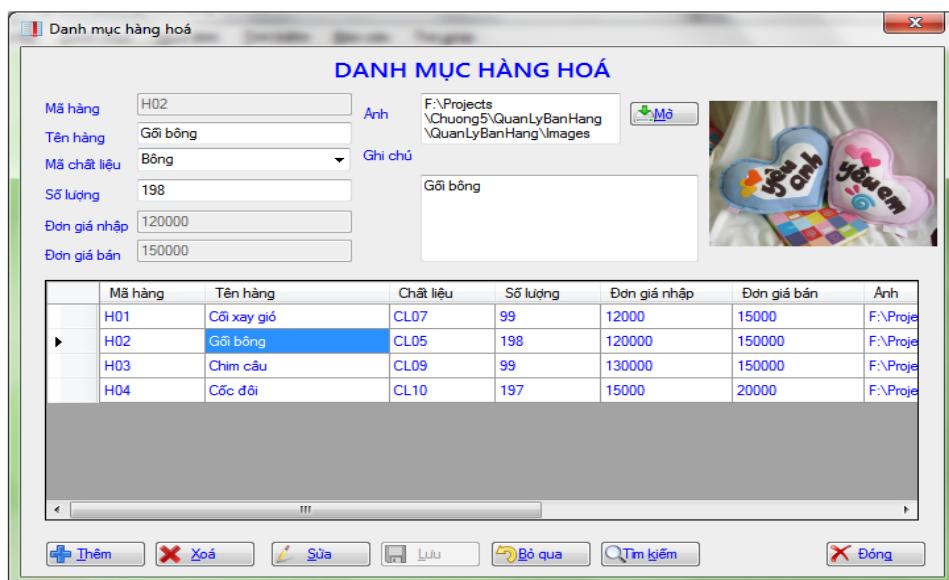
Cho cơ sở dữ liệu gồm các bảng dữ liệu:

DMChatLieu(MaChatLieu, TenChatLieu)

DMHang (MaHang, TenHang, MaChatLieu, SoLuong, DonGiaNhap, DonGiaBan, Hinhanh, GhiChu)

Yêu cầu:

- Thiết kế form theo mẫu sau:



- Hiển thị thông tin mặt hàng trên datagridview
- Đỗ dữ liệu trên combobox Mã chất liệu, thông tin lấy từ cơ sở dữ liệu với tên chất liệu là nhãn hiển thị, mã chất liệu là giá trị khi nhấp chọn item.
- Chọn một dòng trên datagridview hiển thị thông tin dòng đang chọn lên các control (textbox, comboBox).
- Button Mở: sao chép hình ảnh vào thư mục chỉ định, hiển thị tên file hình ảnh đính kèm lên textbox tương ứng.

- Thực hiện thao tác thêm, xóa, sửa trên bảng mặt hàng (có kiểm tra khóa chính, khóa ngoại).
- Tìm kiếm: lọc dữ liệu trên datagridview theo các trường dữ liệu có nhập dữ liệu.

4. Bài tập về nhà

Bài 1: Hiển thị, cập nhật dữ liệu bảng Điểm

	Mã sinh viên	Tên sinh viên	Tên lớp	Mã môn học	Tên môn học	Điểm
▶	01001	Trần Thanh Tùng	Công nghệ phần ...	1	Toán Cao cấp	4
	01001	Trần Thanh Tùng	Công nghệ phần ...	2	Mac - Lê nin	7
	01002	Nguyễn Thanh Bì...	Công nghệ phần ...	1	Toán Cao cấp	6
	01002	Nguyễn Thanh Bì...	Công nghệ phần ...	2	Mac - Lê nin	3
	01002	Nguyễn Thanh Bì...	Công nghệ phần ...	3	Tin học đại cương	5
	01003	Nguyễn Đạt Nhân	Lập trình mạng k...	1	Toán Cao cấp	9
	01003	Nguyễn Đạt Nhân	Lập trình mạng k...	2	Mac - Lê nin	10
	01003	Nguyễn Đạt Nhân	Lập trình mạng k...	3	Tin học đại cương	9
	01004	NGuyễn Gia Bảo	Công nghệ phần ...	1	Toán Cao cấp	3
	01004	NGuyễn Gia Bảo	Công nghệ phần ...	2	Mac - Lê nin	6
	02001	NGuyễn Thanh L...	Kế toán ngân hà...	1	Toán Cao cấp	5
	02001	NGuyễn Thanh L...	Kế toán ngân hà...	3	Tin học đại cương	7
	02002	Lâm Văn Tuấn	Thuế doanh nghiệp	2	Mac - Lê nin	8

Yêu cầu:

- Hiển thị thông tin Điểm của sinh viên theo môn học trên datagridview.
- Đỗ dữ liệu trên combobox Sinh viên, thông tin sinh viên lấy từ cơ sở dữ liệu với tên sinh viên là nhãn hiển thị, mã sinh viên là giá trị khi nhấn chọn item. Khi chọn tên sinh viên trên combobox Sinh viên, datagridview sẽ lọc dữ liệu theo.
- Đỗ dữ liệu trên combobox Môn học, thông tin môn học lấy từ cơ sở dữ liệu với tên môn học là nhãn hiển thị, mã môn học là giá trị khi nhấn chọn item. Khi chọn tên môn học trên combobox Môn học, datagridview sẽ lọc dữ liệu theo.
- Chọn một dòng trên datagridview hiển thị thông tin dòng đang chọn lên các control (textbox, comboBox).
- Thực hiện thao tác thêm, xóa, sửa trên bảng điểm (có kiểm tra khóa chính, khóa ngoại).

Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: CNPM Môn: Lập trình .NET	BÀI 10. LẬP TRÌNH TƯƠNG TÁC CƠ SỞ DỮ LIỆU (tiếp theo)	
--	--	--

A. MỤC TIÊU:

- Sử dụng được đối tượng SqlDataAdapter, DataSet.
- Sử dụng SqlCommand thực hiện thêm, xóa, sửa cho một bảng dữ liệu.
- Sử dụng DataReader hiển thị khóa ngoại cho combobox.
- Sử dụng SqlCommand thực hiện thêm, xóa, sửa cho một bảng dữ liệu.
- Sử dụng DataReader kiểm tra khóa chính/ khóa ngoại trước khi cập nhật dữ liệu.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

- **SqlDataAdapter**: là cầu nối của Database và Dataset. SqlDataAdapter chứa các lệnh SQL (dạng Select) và đối tượng Connection để đọc và ghi dữ liệu.
- **DataSet**: nhận dữ liệu từ SqlDataAdapter, và cung cấp các phương thức xử lý dữ liệu. DataSet có nhiều đối tượng đi kèm theo như sau:
 - DataTable (tương đương với 1 bảng trong database)
 - DataRow (tương đương với 1 dòng trong bảng)
 - DataColumn(tương đương với 1 cột trong bảng)
- **SqlCommandBuilder**: là đối tượng cập nhật dữ liệu từ SqlDataAdapter vào cơ sở dữ liệu.

1.2 Giới thiệu bài tập mẫu

Thiết kế form Quản lý Khoa hoàn chỉnh (nhập trên Textbox), các control giãn theo tỷ lệ tùy ý (Sử dụng cơ sở dữ liệu QLSinhVien)



Yêu cầu:

- FormLoad:
 - + Datagridview: hiển thị tất cả khoa trong bảng Khoa
 - + Tất cả textbox bị vô hiệu hóa
 - + Các button Sửa, Xóa, Lưu bị vô hiệu hóa
- Khi chọn vào button lệnh Thêm:
 - + Các Textbox có hiệu lực
 - + Button Lưu có hiệu lực
 - + Dấu nháy xuất hiện ở textbox Mã khoa.
- Khi chọn vào DataGridView
 - + Hiển thị thông tin tương ứng lên các textbox
 - + Button Sửa và Xóa có hiệu lực
- Chọn button Sửa
 - + Button Lưu có hiệu lực
 - + Các textbox có hiệu lực trừ textbox Mã khoa
 - + Cho phép sửa các thông tin còn lại
- Khi chọn button Lưu
 - + Kiểm tra thông tin vừa nhập hoặc sửa cho phù hợp
 - + Lưu vào Cơ sở dữ liệu (cần xác định là đang lưu cho hành động Thêm hay Sửa).
 - + Thông báo thành công hoặc báo lỗi nếu có.
 - + Button Lưu bị vô hiệu hóa.

- Khi nhấn button Xóa.
 - + Hiển thị thông báo xác nhận
 - + Nếu đồng ý
 - Kiểm tra mã khoa định xóa có tồn tại trong bảng lớp hay không. Nếu có thì thông báo “Dữ liệu đang sử dụng không thể xóa ?”.
 - Nếu mã khoa không tồn tại bên bảng lớp thì xóa bên bảng khoa.
 - + Hiển thị thông báo nếu xóa thành công hoặc báo lỗi (nếu có)

Công việc 1: Thiết kế giao diện hoàn chỉnh như hình trên

Công việc 2: Xử lý yêu cầu của đề bài

```

SqlConnection _Connsql = new SqlConnection("Data Source =
<Tên_server>; Initial Catalog = QLSINHVIEN; User ID =
<Tài_khoản_ĐN>; Password = <Mật_khẩu>");

DataSet ds_QLSV = new DataSet();
public frmQuanLyKhoa()
{
    InitializeComponent();
}
//Phương thức load dữ liệu cho Datagridview
void LoadDuLieuKhoa()
{
    string strsel = "select * from Khoa";
    SqlDataAdapter da_Khoa=new SqlDataAdapter(strsel, _Connsql);
    da_Khoa.Fill(ds_QLSV, "Khoa");
    dataGridView1.DataSource = ds_QLSV.Tables["Khoa"];
    DataColumn[] key = new DataColumn[1];
    key[0] = ds_QLSV.Tables["Khoa"].Columns[0];
    ds_QLSV.Tables["Khoa"].PrimaryKey = key;
}
//Sự kiện Load của form
private void frmQuanLyKhoa_Load(object sender, EventArgs e)
{
    LoadDuLieuKhoa();
    //Tắt cả textbox bị vô hiệu hóa
    txtMaKhoa.Enabled = txtTenKhoa.Enabled = false;
    //Các button Sửa, xóa, Lưu bị vô hiệu hóa
    btnSua.Enabled = btnXoa.Enabled = btnLuu.Enabled = false;
}

```

```

//Sự kiện click của button Thêm
private void btnThem_Click(object sender, EventArgs e)
{
    //Các textbox có hiệu lực
    txtMaKhoa.Enabled = txtTenKhoa.Enabled = true;
    //button Lưu có hiệu lực
    btnLuu.Enabled = true;
    //Đầu nháy xuất hiện ở textbox Mã khoa.
    txtMaKhoa.Focus();
    //Xóa dữ liệu hiện tại trên các textbox.
    txtMaKhoa.Clear();
    txtTenKhoa.Clear();
}

//Sự kiện SelectionChanged của datagridview
private void dgv_Khoa_SelectionChanged(object sender, EventArgs e)
{
    //Button Sửa và Xóa có hiệu lực
    btnSua.Enabled = btnXoa.Enabled = true;
}

//Sự kiện click của button Sửa
private void btnSua_Click(object sender, EventArgs e)
{
    //Button Lưu có hiệu lực
    btnLuu.Enabled = true;
    //Các textbox có hiệu lực trừ textbox Mã khoa
    txtTenKhoa.Enabled = true;
    txtMaKhoa.Enabled = false;
}

//Sự kiện click của button Lưu
private void btnLuu_Click(object sender, EventArgs e)
{
    //Kiểm tra thông tin vừa nhập hoặc sửa cho phù hợp
    if (txtMaKhoa.Text == string.Empty)
    {
        MessageBox.Show("Chưa nhập Mã Khoa");
        txtMaKhoa.Focus();
        return;
    }
}

```

```

if (txtTenKhoa.Text == string.Empty)
{
    MessageBox.Show("Chưa nhập Tên Khoa");
    txtTenKhoa.Focus();
    return;
}
if (txtMaKhoa.Enabled == true) // Thêm
{
    DataRow insert_New = dsQLSV.Tables["Khoa"].NewRow();
    insert_New["MaKhoa"] = txtMaKhoa.Text;
    insert_New["TenKhoa"] = txtTenKhoa.Text;
    dsQLSV.Tables["Khoa"].Rows.Add(insert_New);
}
else // Sửa
{
    DataRow update_New = dsQLSV.Tables["Khoa"].Rows.Find
(txtMaKhoa.Text);
    if (update_New != null)
    {
        update_New["TenKhoa"] = txtTenKhoa.Text;
    }
}
SqlCommandBuilder cmb = new SqlCommandBuilder(da_Khoa);
da_Khoa.Update(dsQLSV, "Khoa")
MessageBox.Show("Thành công");
btnLuu.Enabled = false;
}

//Sự kiện click của button Xóa
private void btnXoa_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Bạn muốn xóa?", "Thông báo",
    MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
    MessageBoxDefaultButton.Button2) ==
    System.Windows.Forms.DialogResult.Yes)
    {
        //Nếu đồng ý xóa
        DataTable dt_Lop=new DataTable();

```

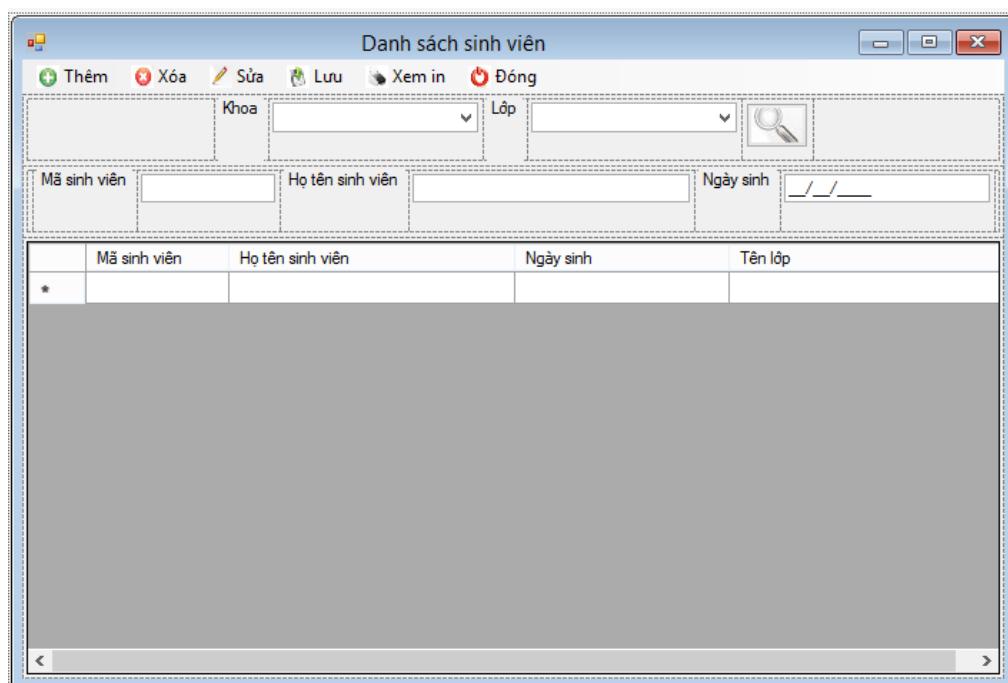
```

//Kiểm tra khóa ngoại trong bảng lớp
SqlDataAdapter da_lop = new SqlDataAdapter("select * from
Lop where makhoa= '" + txtmakhoa.Text + "'", _Connsql);
da_lop.Fill(dt_Lop);
if (dt_Lop.Rows.Count > 0)
{
    MessageBox.Show("Dữ liệu đang sử dụng không thể xóa");
    return;
}
// Nếu cập nhật dữ liệu
DataRow upd_New=ds_QLSV.Tables["Khoa"].Rows.Find (txtMaKhoa.Text);
if (update_New != null)
{
    update_New.Delete();
}
SqlCommandBuilder cmb = new SqlCommandBuilder(da_Khoa);
da_Khoa.Update(ds_QLSV, "Khoa");
MessageBox.Show("Thành công");
}
}

```

2. Bài tập tại lớp

Thiết kế giao diện theo mẫu sau:



Yêu cầu:

- Thiết kế:
 - + Sử dụng `tableLayoutPanel` để thiết kế giao diện như hình vẽ, sao cho nội dung luôn hiển thị ở giữa khi phóng to hay thu nhỏ cửa sổ.
- FormLoad:
 - + Combobox Khoa: hiển thị tất cả khoa trong Bảng Khoa
 - + Combobox Lớp: hiển thị tất cả lớp theo khoa
 - + Datagridview: Hiển thị tất cả sinh viên trong bảng sinh viên
 - + Tất cả textbox bị vô hiệu hóa
 - + Các button Sửa, Xóa, Lưu bị vô hiệu hóa
- Button Tìm kiếm:
 - + Hiển thị danh sách sinh viên theo lớp đã chọn trên combobox lớp
- Khi chọn vào button Thêm:
 - + Các textbox có hiệu lực
 - + Button Lưu có hiệu lực
 - + Textbox Mã sinh viên chờ nhập liệu
- Khi chọn vào Datagridview
 - + Hiển thị thông tin tương ứng lên các textbox
 - + Button lệnh Sửa và Xóa có hiệu lực
- Chọn button Sửa
 - + Button Lưu có hiệu lực
 - + Các textbox có hiệu lực trừ textbox Mã sinh viên
 - + Cho phép sửa các thông tin còn lại
- Khi chọn button Lưu
 - + Kiểm tra thông tin vừa thêm hoặc sửa cho phù hợp
 - + Lưu vào Cơ sở dữ liệu (Lưu ý đang lưu Thêm hay Sửa)
 - + Thông báo thành công hoặc báo lỗi nếu có.
 - + Button Sửa, Lưu bị vô hiệu hóa
- Khi nhấn button Xóa
 - + Hiển thị thông báo xác nhận
 - + Nếu đồng ý
 - Kiểm tra mã sinh viên xóa có tồn tại trong bảng điểm hay không. Nếu có

thì thông báo “Dữ liệu đang sử dụng không thể xóa ?”.

- Nếu mã sinh viên không tồn tại bên bảng lớp thì xóa bên bảng sinh viên.
- + Hiển thị thông báo nếu xóa thành công hoặc báo lỗi (nếu có)

3. Bài tập về nhà

Thiết kế form Phiếu Nhập hàng như sau:

The 'Phiếu nhập' section contains fields for: Mã phiếu nhập (PN10112015001), Nhà cung cấp (Điện máy Chợ lớn), Ngày nhập (10/11/2015), Thành tiền (13000000), and buttons for Tạo phiếu nhập and Lưu phiếu nhập.

The 'Chi tiết phiếu nhập' section contains fields for: Mã phiếu nhập (PN10112015001), Mã sản phẩm (TIVI SONY 32inch), Đơn giá (5400000), Số lượng (1), and buttons for Thêm sản phẩm and In phiếu nhập.

A grid table displays purchase details:

	Mã phiếu nhập	Mã sản phẩm	Đơn giá	Số lượng
▶	PN10112015001	SP001	5400000	1
	PN10112015001	SP002	Xóa	1
	PN10112015002	SP001	5400000	2
	PN10112015002	SP002	7600000	2
	PN10112015003	SP001	5400000	1

Yêu cầu

- Formload:

Groupbox Phiếu nhập:

- + Textbox Mã phiếu nhập để trống, textbox ngày nhập là ngày hiện tại
- + Con trỏ nhập liệu tại textbox Mã phiếu nhập.
- + Combobox Nhà cung cấp: Chứa tất cả Mã nhà cung cấp của bảng NCC
- + Textbox thành tiền: Bị vô hiệu hóa, hiển thị số 0.

Groupbox Chi tiết phiếu nhập

- + Combobox mã phiếu nhập: Hiển thị tất cả Mã phiếu nhập trong bảng Phieunhap.
- + Combobox mã sản phẩm: Hiển thị tất cả Mã sản phẩm trong bảng Sanpham.
- + Textbox đơn giá: bị vô hiệu hóa.
- + Số lượng: Hiển thị số 0.

- + Datagridview: Hiển thị thông tin trong bảng chitetphieunhap.
- Thực hiện các chức năng còn lại tương ứng.
- Chọn button “Tạo phiếu nhập”:
 - + Lưu thông tin trong groupbox Phiếu nhập vào table PhieuNhap lưu ý kiểm tra các dữ liệu đầu vào.
 - + Load lại Combobox Mã phiếu nhập trong groupbox Chi tiết phiếu nhập.
 - + Lưu ý: Không cho thao tác trong textbox thành tiền, lúc này đang là số 0.
 - Chọn “combobox Mã phiếu nhập” trong groupbox “Chi tiết Phiếu nhập”
 - + Thông tin về phiếu nhập vừa chọn sẽ được hiển thị lên groupbox “Phiếu nhập”.
 - + Thông tin về Chi tiết phiếu nhập tương ứng hiển thị lên datagridview ChiTietPhieuNhap.
 - Chọn “Combobox Mã sản phẩm”:
 - + Tự động hiển thị đơn giá tương ứng với mã sản phẩm vừa chọn.
 - Gõ số lượng vào “textbox số lượng”:
 - + Tự động cập nhật thành tiền trên “textbox thành tiền” ứng với Phiếu nhập đang chọn.
 - Chọn “Thêm sản phẩm”
 - + Kiểm tra Mã sản phẩm trong combobox “Mã sản phẩm” đã có trong “ChiTietPhieuNhap” đang chọn hay chưa. Nếu chưa có thì lưu thông tin trong groupbox “Chi tiết phiếu nhập” vào bảng Chitetphieunhap. Ngược lại cập nhật số lượng.
 - + Cập nhật Thành tiền vào trong bảng PhieuNhap.
 - Khi chọn datagridview Chi tiết phiếu nhập:
 - + Hiển thị thông tin vừa chọn lên các control trong groupbox Chi tiết phiếu nhập.
 - + DataGridView Chi tiết phiếu nhập cho hiển thị ContextMenuStrip có các chức năng Sửa, Xóa Chi tiết phiếu nhập.

Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: CNPM Môn: Lập trình .NET	BÀI 11. LẬP TRÌNH TƯƠNG TÁC CƠ SỞ DỮ LIỆU (tiếp theo)	
--	--	--

A. MỤC TIÊU:

- Sử dụng được đối tượng SqlDataAdapter, DataSet.
- Sử dụng SqlCommand thực hiện thêm, xóa, sửa cho một bảng dữ liệu.
- Sử dụng DataReader hiển thị khóa ngoại cho combobox.
- Sử dụng SqlCommand thực hiện thêm, xóa, sửa cho một bảng dữ liệu.
- Sử dụng DataReader kiểm tra khóa chính/khóa ngoại trước khi cập nhật dữ liệu.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

- **SqlDataAdapter**: là cầu nối của database và dataset. SqlDataAdapter chứa các lệnh SQL (dạng Select) và đối tượng connection để đọc và ghi dữ liệu.
- **DataSet**: nhận dữ liệu từ SqlDataAdapter, và cung cấp các phương thức xử lý dữ liệu. DataSet có nhiều đối tượng đi kèm theo như sau:
 - DataTable (tương đương với 1 bảng trong database)
 - DataRow (tương đương với 1 dòng trong bảng)
 - DataColumn(tương đương với 1 cột trong bảng)
- **SqlCommandBuilder**: là đối tượng cập nhật dữ liệu từ SqlDataAdapter vào cơ sở dữ liệu.

1.2 Giới thiệu bài tập mẫu

Thiết kế form Quản lý sinh viên hoàn chỉnh (nhập trên Lưới), các control giãn theo tỷ lệ tùy ý (Sử dụng cơ sở dữ liệu QLSinhVien):

MaSinhVien	HoTen	NgaySinh	MaLop
01001	Trần Thanh Tùng	12/10/1993	05DHTH1
01002	Nguyễn Thành Bình	2/4/1993	05DHTH1
01003	Nguyễn Đạt Nhân	3/15/1992	05DHTH2
01004	Nguyễn Gia Bảo	5/11/1993	05DHTH1
02001	Nguyễn Thành Lan	6/12/1993	05DHKT2
02002	Lâm Văn Tuấn	8/3/1993	06DHKT1

Yêu cầu:

- Formload:
 - + Combobox mã lớp: hiển thị tên lớp trong bảng lop
 - + Datagridview Sinh viên: hiển thị tất cả sinh viên trong bảng SinhVien và có thuộc tính chỉ đọc.
 - + Tất cả textbox, combobox bị vô hiệu hóa
 - + Các button Sửa, Xóa, Lưu bị vô hiệu hóa
 - + Thiết lập liên kết giữa control nhập liệu với chọn thông tin trên datagridview
- Khi chọn vào button Thêm:
 - + Button Lưu có hiệu lực
 - + Cho phép thêm các dòng tiếp theo trên datagridview

Lưu ý: không được sửa đổi các dòng trên datagridview đã có dữ liệu.

- Khi chọn vào datagridview
 - + Hiển thị thông tin tương ứng lên các textbox, combobox
 - + Button Sửa và Xóa có hiệu lực
- Chọn button Sửa
 - + Button Lưu có hiệu lực
 - + Cho phép sửa các thông tin trên Datagrid

Lưu ý: không cho phép nhập thêm các dòng mới.

- Khi chọn button Lưu
 - + Kiểm tra thông tin vừa nhập hoặc sửa trên lưới cho phù hợp
 - + Lưu vào cơ sở dữ liệu

- + Thông báo thành công hoặc báo lỗi nếu có
- + Button Lưu bị vô hiệu hóa
- Khi nhấn button Xóa
 - + Hiển thị thông báo xác nhận
 - + Nếu đồng ý
 - Kiểm tra dữ liệu trong table Diem tương ứng với mã số sinh viên đang chọn (nếu có) nếu tồn tại sinh viên đang cần xóa thì cảnh báo nhắc nhở dữ liệu đang sử dụng không thể xóa. Ngược lại xóa dữ liệu trong table sinh viên tương ứng với mã số sinh viên đang chọn.
 - Hiển thị thông báo nếu xóa thành công hoặc báo lỗi (nếu có)

Hướng dẫn:

```

SqlConnection _Connsql = new SqlConnection("Data Source = < > ;
Initial Catalog = QLSINHVIEN; User ID = <>;Password = <>");
DataSet ds_QLSV = new DataSet();
public frmQuanLySinhVien()
{
    InitializeComponent();
}
void LoadDuLieuKhoa()
{
    string strsel = "select * from Lop";
    SqlDataAdapter da_Khoa = new SqlDataAdapter(strsel, _Connsql);
    da_Khoa.Fill(ds_QLSV, "Lop");
    cboLop.DataSource=ds_QLSV.Tables["Lop"];
    cboLop.DisplayMember = "TenLop";
    cboLop.ValueMember = "MaLop";
}
void LoadDuLieuDataG()
{
    string strsel = "select * from SinhVien ";
    SqlDataAdapter da_Khoa = new SqlDataAdapter(strsel, _Connsql);
    da_Khoa.Fill(ds_QLSV, "SinhVien");
    dataGridView1.DataSource = ds_QLSV.Tables["SinhVien"];
}
void Databingding(DataTable pDT)
{

```

```

        cboLop.DataBindings.Clear();
        txtHoTen.DataBindings.Clear();
        txtMaSv.DataBindings.Clear();
        maskedTextBox1.DataBindings.Clear();

        cboLop.DataBindings.Add("Text", pDT, "MaLop");
        txtHoTen.DataBindings.Add("Text", pDT, "HoTen");
        txtMaSv.DataBindings.Add("Text", pDT, "MaSinhVien");
        maskedTextBox1.DataBindings.Add("Text", pDT, "NgaySinh");
    }

    private void frmQuanLySinhVien_Load(object sender, EventArgs e)
    {
        //+Combobox mã lớp: Chứa tên lớp trong bảng lop
        LoadDuLieuKhoa();
        //+Datagridview sinh viên:
        LoadDuLieuDataG();
        //và chỉ đọc
        dataGridView1.ReadOnly = true;
        //Ẩn dòng cuối cùng Trên dataGrd
        dataGridView1.AllowUserToAddRows = false;
        //+Tắt cả textbox, combobox bị vô hiệu hóa
        foreach (Control item in tableLayoutPanel1.Controls)
        {
            if (item.GetType() == typeof(TextBox) || item.GetType() == typeof(ComboBox) || item.GetType() == typeof(MaskedTextBox))
                item.Enabled = false;
        }
        //+ Các button Sửa, xóa, Lưu bị vô hiệu hóa
        btnSua.Enabled = btnXoa.Enabled = btnLuu.Enabled = false;
        Databingding(ds_QLSV.Tables["SinhVien"]);
    }

    private void btnThem_Click(object sender, EventArgs e)
    {
        btnLuu.Enabled = true;
        //+ Cho phép thêm các dòng tiếp theo trên datagridview
        dataGridView1.AllowUserToAddRows = true;
        dataGridView1.ReadOnly = false;
    }
}

```

```

//Không được sửa các dòng trên datagridview đã có dữ liệu
for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
{
    dataGridView1.Rows[i].ReadOnly = true;
}
dataGridView1.FirstDisplayedScrollingRowIndex =
    dataGridView1.Rows.Count - 1;
}

private void dataGridView1_SelectionChanged(object sender,
EventArgs e)
{
    try
    {
        maskedTextBox1.Text =
DateTime.Parse(dataGridView1.CurrentRow.Cells[2].Value.ToString())
            .ToString("dd/MM/yyyy");
    }
    catch
    {
        maskedTextBox1.Text = string.Empty;
    }
    //+Button Sửa và Xóa có hiệu lực
    btnSua.Enabled = btnXoa.Enabled = true;
}

private void btnSua_Click(object sender, EventArgs e)
{
    //+Button Lưu có hiệu lực
    btnLuu.Enabled = true;
    //+Cho phép sửa các thông tin trên Datagrid
    dataGridView1.ReadOnly = false;
    for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
        dataGridView1.Rows[i].ReadOnly = false;
    dataGridView1.Columns[0].ReadOnly = true;
    //+Lưu ý: không cho phép gõ thêm các dòng mới
    dataGridView1.AllowUserToAddRows = false;
}

```

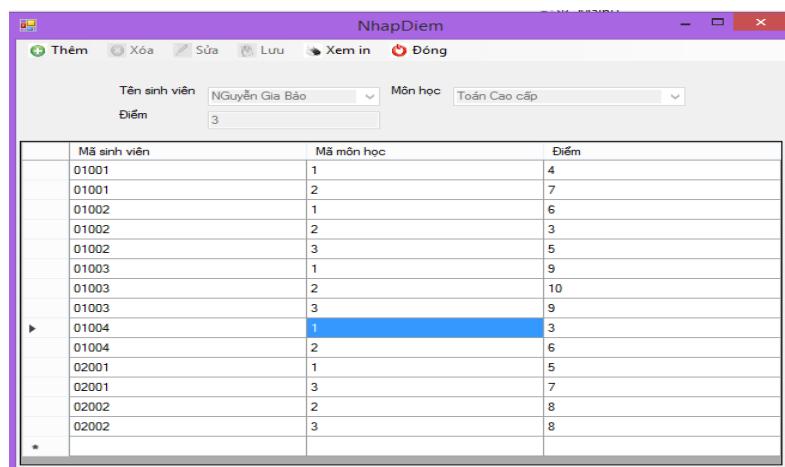
```

private void btnLuu_Click(object sender, EventArgs e)
{
    SqlDataAdapter da_Khoa = new SqlDataAdapter("select * from SinhVien", _Connsql);
    SqlCommandBuilder cmb = new SqlCommandBuilder(da_Khoa);
    da_Khoa.Update(ds_QLSV, "SinhVien");
    Databingding(ds_QLSV.Tables["SinhVien"]);
    MessageBox.Show("Thành công");
    btnLuu.Enabled = false;
}

```

2. Bài tập tại lớp

Thiết kế giao diện theo mẫu sau:



Yêu cầu:

- Formload:
 - + Combobox tên sinh viên: hiển thị tên sinh viên trong bảng SinhVien.
 - + Combobox Môn học: hiển thị tên môn học trong bảng MonHoc.
 - + Datagrid: Hiển thị tất cả điểm trong bảng Điểm và chỉ cho phép đọc.
 - + Tất cả textbox, combobox bị vô hiệu hóa.
 - + Các button Sửa, xóa, Lưu bị vô hiệu hóa.
 - + Thiết lập liên kết giữa control nhập liệu với chọn thông tin trên datagridview.
- Khi chọn vào button Thêm:
 - + Button Lưu có hiệu lực
 - + Cho phép thêm các dòng tiếp theo trên datagridview

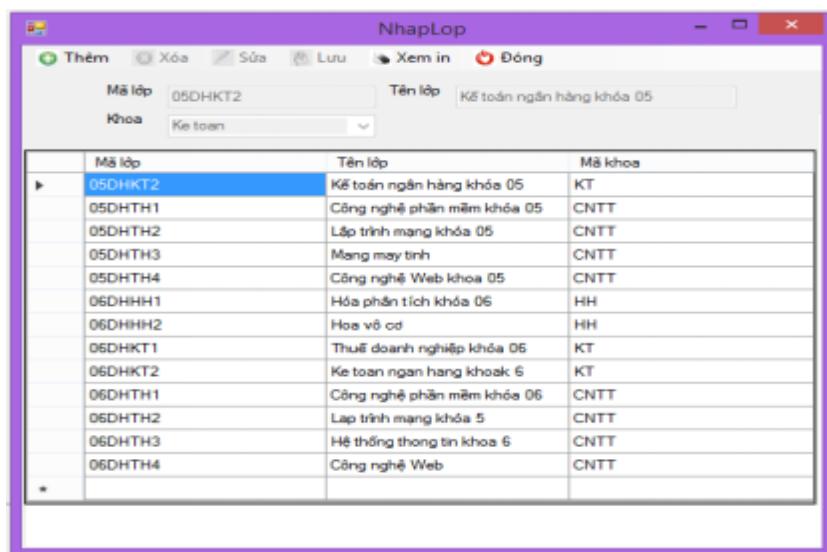
Lưu ý: không được sửa đổi các dòng trên datagridview đã có dữ liệu

- Khi chọn vào datagridview

- + Hiển thị thông tin tương ứng lên các textbox, combobox
- + Button Sửa và Xóa có hiệu lực
- Chọn button Sửa
 - + Button Lưu có hiệu lực
 - + Cho phép sửa các thông tin trên datagridview
- Lưu ý:** không cho phép gõ thêm các dòng mới
- Khi chọn button Lưu
 - + Kiểm tra thông tin vừa nhập hoặc sửa (không trùng khóa chính) trên lưới cho phù hợp
 - + Lưu vào cơ sở dữ liệu
 - + Thông báo thành công hoặc báo lỗi nếu có
 - + Button Lưu bị vô hiệu hóa
- **Khi nhấn button Xóa**
 - + Hiển thị thông báo xác nhận
 - + Nếu đồng ý: hiển thị thông báo nếu xóa thành công hoặc báo lỗi (nếu có)

3. Bài tập về nhà

Thiết kế giao diện sau:



Yêu cầu:

- Formload:
 - + Combobox tên Khoa: hiển thị tên Khoa trong bảng Khoa.
 - + Datagridview: hiển thị tất cả Lớp trong bảng Lop và chỉ cho phép đọc.
 - + Tất cả textbox, combobox bị vô hiệu hóa.
 - + Các button Sửa, Xóa, Lưu bị vô hiệu hóa.

- + Thiết lập liên kết giữa control nhập liệu với chọn thông tin trên datagridview.
- Khi chọn vào button Thêm:
 - + Button Lưu có hiệu lực
 - + Cho phép thêm các dòng tiếp theo trên datagridview

Lưu ý: không được sửa đổi các dòng trên datagridview đã có dữ liệu

- Khi chọn vào Datagrid
 - + Hiển thị thông tin tương ứng lên các textbox, combobox
 - + Button Sửa và Xóa có hiệu lực
- Chọn button Sửa
 - + Button Lưu có hiệu lực
 - + Cho phép sửa các thông tin trên datagridview

Lưu ý: không cho phép gõ thêm các dòng mới

- Khi chọn button Lưu
 - + Kiểm tra thông tin vừa nhập hoặc sửa (không trùng khóa chính) trên lưới cho phù hợp
 - + Lưu vào cơ sở dữ liệu
 - + Thông báo thành công hoặc báo lỗi nếu có
 - + Button Lưu bị vô hiệu hóa
- Khi nhấn button Xóa
 - + Hiển thị thông báo xác nhận
 - + Nếu đồng ý
 - Kiểm tra dữ liệu trong bảng Sinh viên tương ứng với mã lớp đang chọn (nếu có) nếu tồn tại mã lớp đang cần xóa thì cảnh báo nhắc nhớ dữ liệu đang sử dụng không thể xóa. Ngược lại xóa dữ liệu trong bảng Lớp tương ứng với mã số lớp đang chọn.
 - Hiển thị thông báo nếu xóa thành công hoặc báo lỗi (nếu có).

BÀI 12. CRYSTAL REPORT



A. MỤC TIÊU:

- Sử dụng đối tượng Crystal Report tạo báo cáo cho phần mềm ứng dụng

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

✚ Giới thiệu:

- Reports là các báo cáo có thể truy xuất và định dạng thông tin hiển thị của dữ liệu từ các nguồn dữ liệu khác nhau như cơ sở dữ liệu hoặc file ...
- Điểm khác nhau cơ bản nhất của report và form đó là: form dùng để hiển thị và cập nhật thông tin, report dùng để hiển thị và định dạng thông tin và có chức năng in ra giấy.
- Crystal Report là công cụ để tạo các báo cáo cho các ứng dụng Windows Application hoặc Web Application.
- Crystal Report có thể hiển thị thông tin dưới dạng bảng, đồ họa, biểu đồ, ... có khả năng tính toán như tính tổng, trung bình, ...
- Crystal Report có thể làm việc trên môi trường .NET hoặc Java.

✚ Thiết kế report với Crystal Report: để tạo một report, chúng ta cần xác định các giai đoạn sau:

- **Giai đoạn chuẩn bị:** cần xác định
 - Mục đích tạo report ?
 - Các field (các cột) nào sẽ có trong report ?
 - Dữ liệu được lấy từ bảng hay view nào ?
 - Hiển thị tất cả các record hay chỉ 1 phần ?
 - Dữ liệu có được group (gom nhóm) hay sort (sắp xếp)?
 - Những summary, calculations nào có trong report ?
- **Giai đoạn tạo report:** thực hiện 3 bước sau
 - Bước 1: Right click vào Project→Add New Item→Đặt tên MyReport.rpt

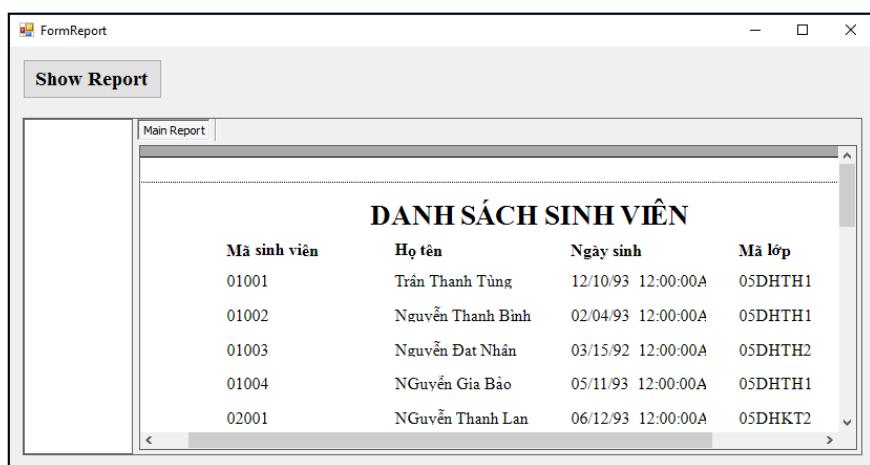
- Bước 2: Configure (Cấu hình) và Design (Thiết kế) MyReport.rpt
- Bước 3: Kết nối control CrystalReportViewer với MyReport.rpt

 **Các Section trong màn hình thiết kế:**

- Report Header: hiển thị ở phần đầu trang 1, không hiển thị ở các trang sau
- Report Footer: hiển thị ở phần cuối trang cuối cùng
- Page Header: hiển thị ở phần đầu tất cả các trang
- Page Footer: hiển thị ở phần cuối tất cả các trang
- Group Header: hiển thị ở phần đầu của mỗi group
- Group Footer: hiển thị ở phần cuối của mỗi group
- Detail: hiển thị phần thông tin dữ liệu, từng record sẽ được lặp lại.

1.2 Giới thiệu bài tập mẫu

Thiết kế một report theo mẫu sau (sử dụng cơ sở dữ liệu QLSinhVien)

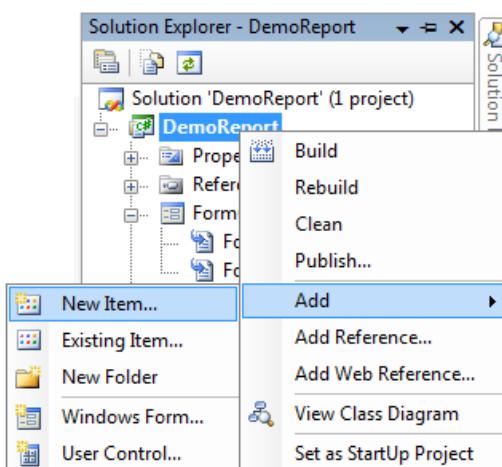


Yêu cầu: Click vào button Show Report hiển thị danh sách tất cả sinh viên trong bảng SinhVien

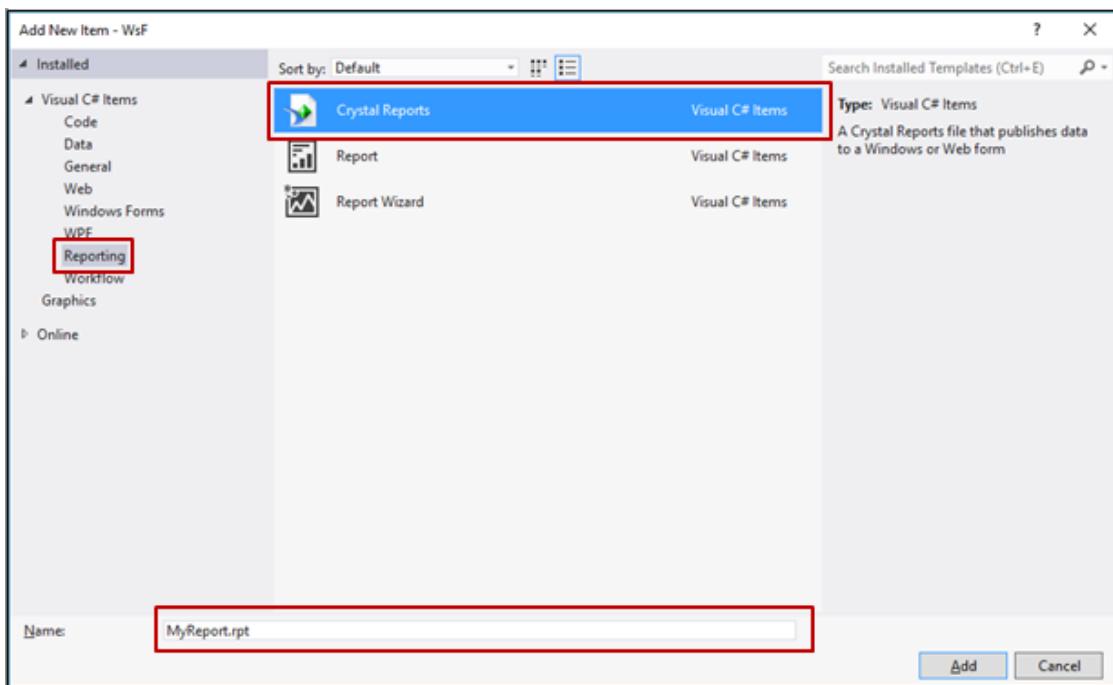
Hướng dẫn:

Bước 1: Tạo file report mới có tên là MyReport.rpt

Right click vào project → Add → New Item ...



Chọn **Crystal Report** → **Add**

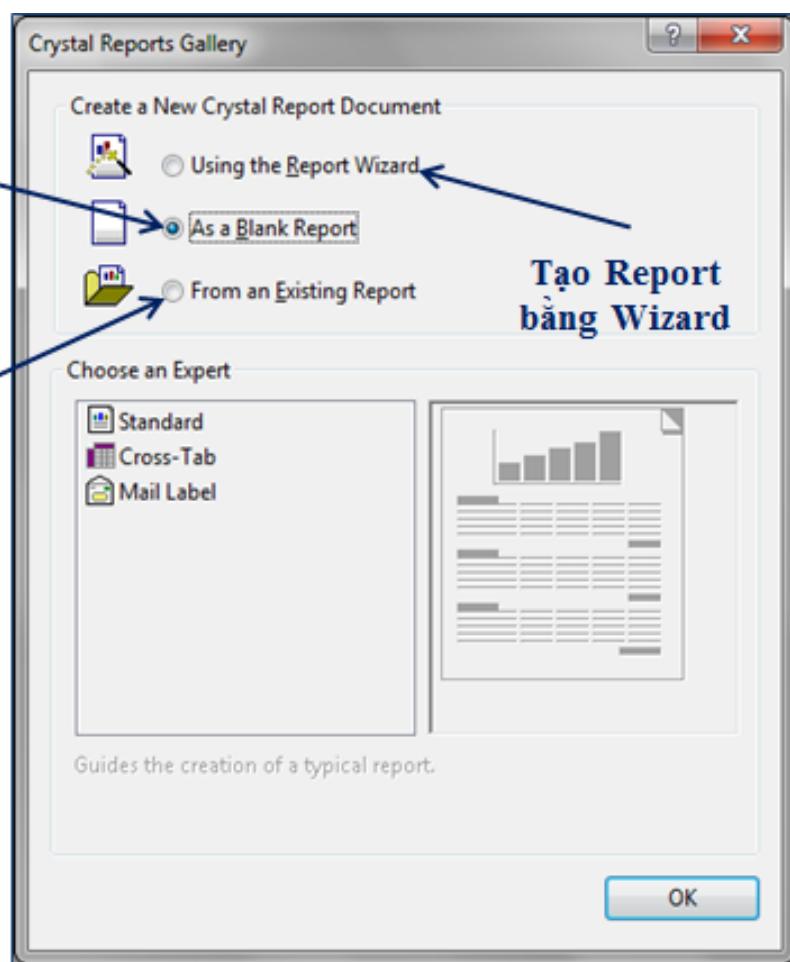


Chọn kiểu Report

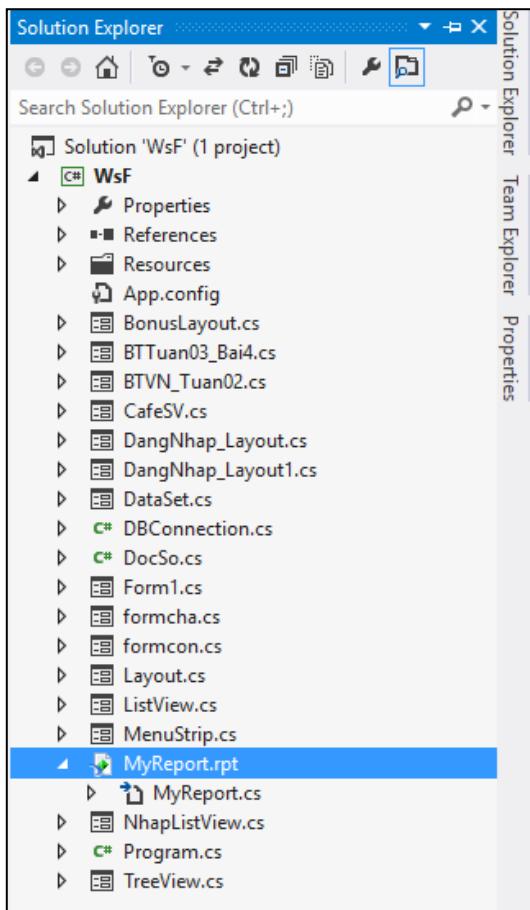
Tạo 1 Report
trống

Tạo Report
từ 1 file
report

Tạo Report
bằng Wizard



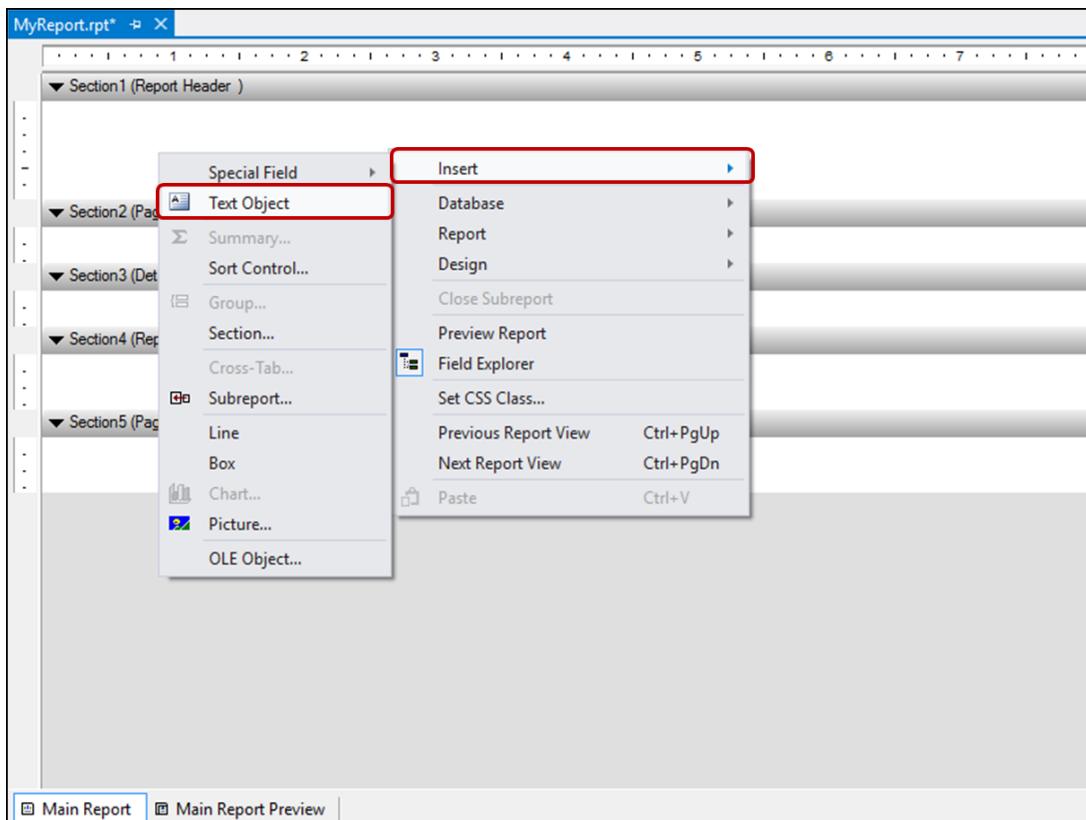
Trong Solution Explorer xuất hiện



Bước 2: Configure (Cấu hình) và Design (Thiết kế) MyReport.rpt

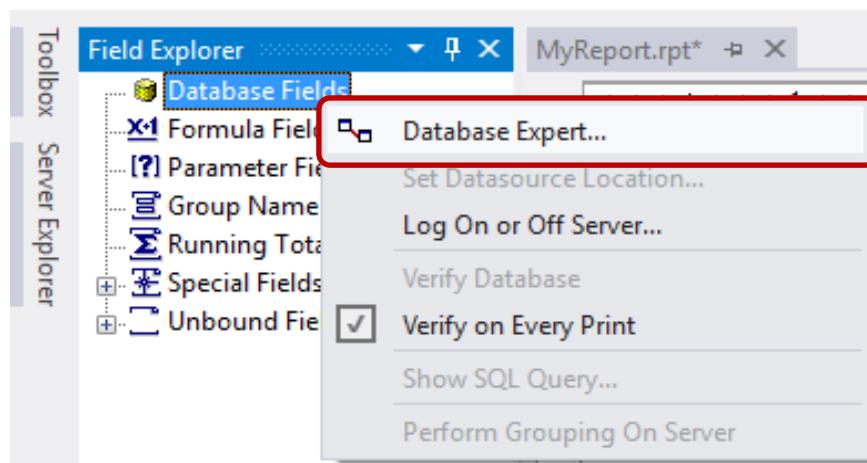
A screenshot of the Microsoft Visual Studio Report Designer for 'MyReport.rpt'. The interface shows five sections stacked vertically: Section1 (Report Header), Section2 (Page Header), Section3 (Details), Section4 (Report Footer), and Section5 (Page Footer). The 'Details' section is currently active. Along the left side, there's a toolbar with icons for 'Toolbox', 'Server Explorer', 'Field Explorer', and 'Data Sources'. At the bottom, there are tabs for 'Main Report' (selected) and 'Main Report Preview'.

Thêm dòng tiêu đề “DANH SÁCH SINH VIÊN” vào Report

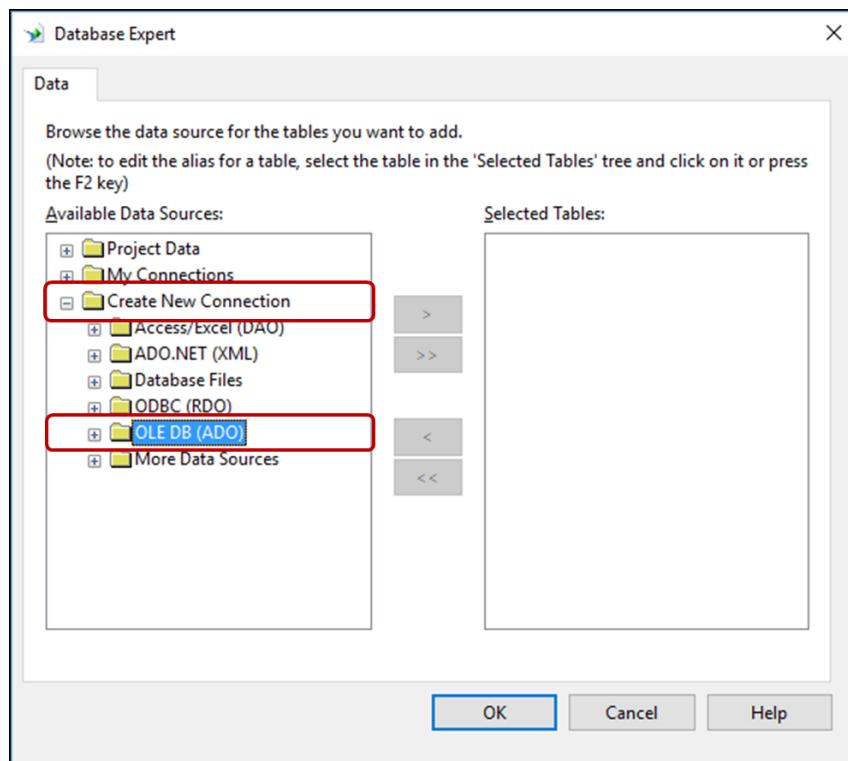


Tiến hành kết nối cơ sở dữ liệu QLSINHVIEN (bảng SINHVIEN)

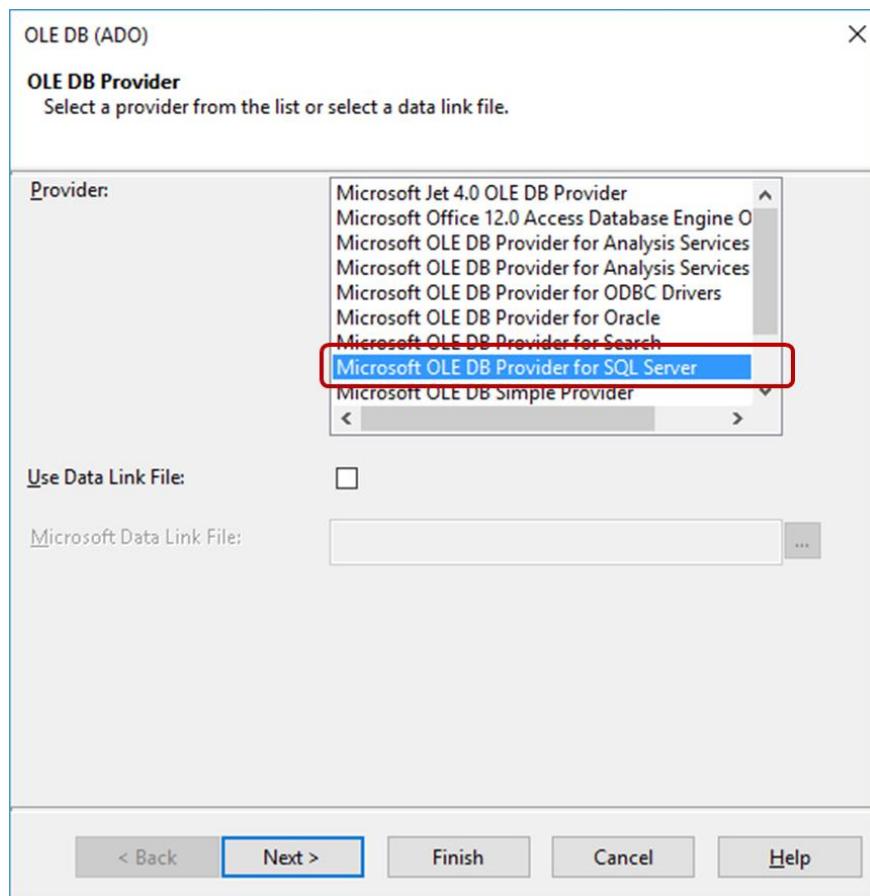
Right click Database Fields (trong Field Explorer) → chọn Database Expert...



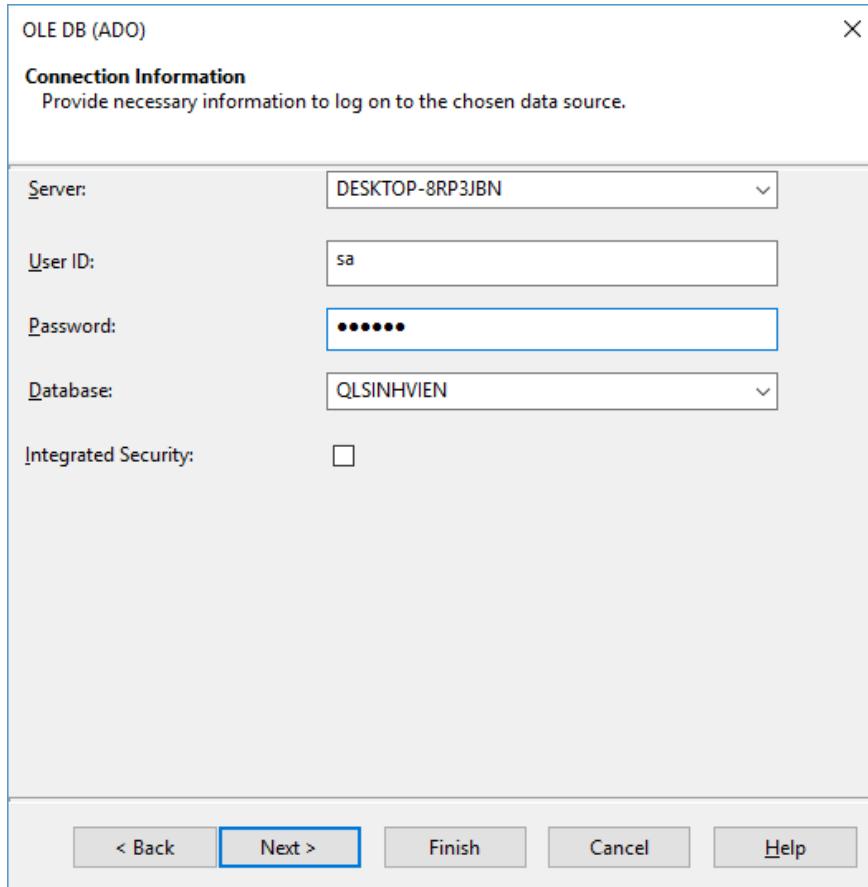
Chọn **Create New Connection** → **OLEDB (ADO)** (Nếu muốn tạo một Connection mới)



Chọn **Microsoft OLE DB Provider for SQL Server**

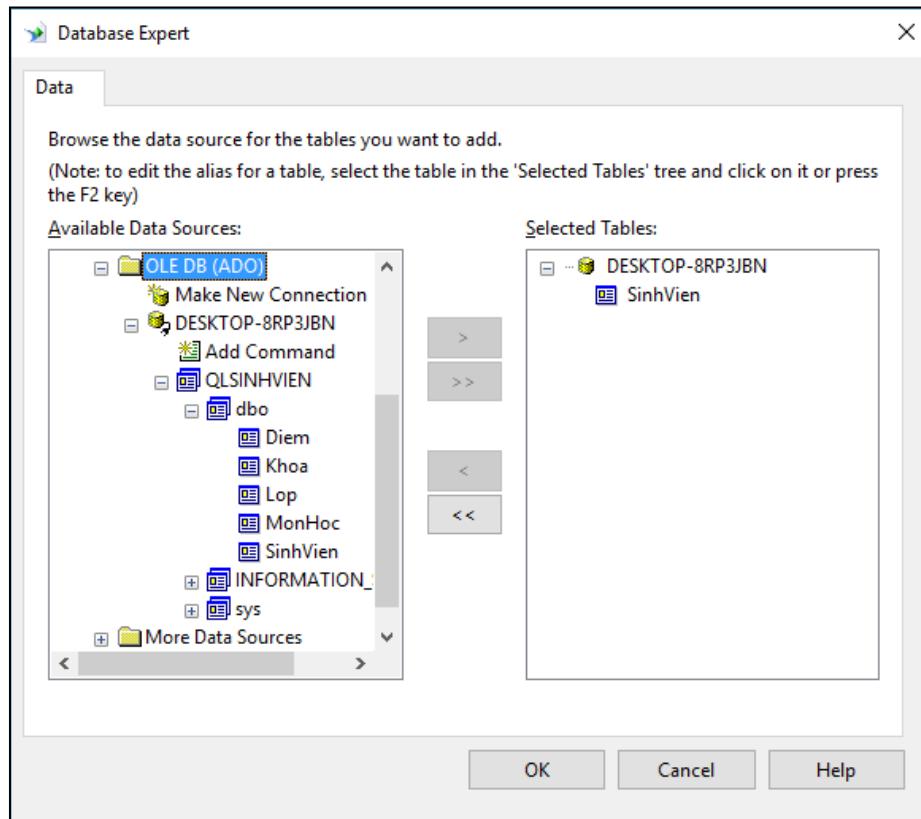


Nhập thông tin kết nối



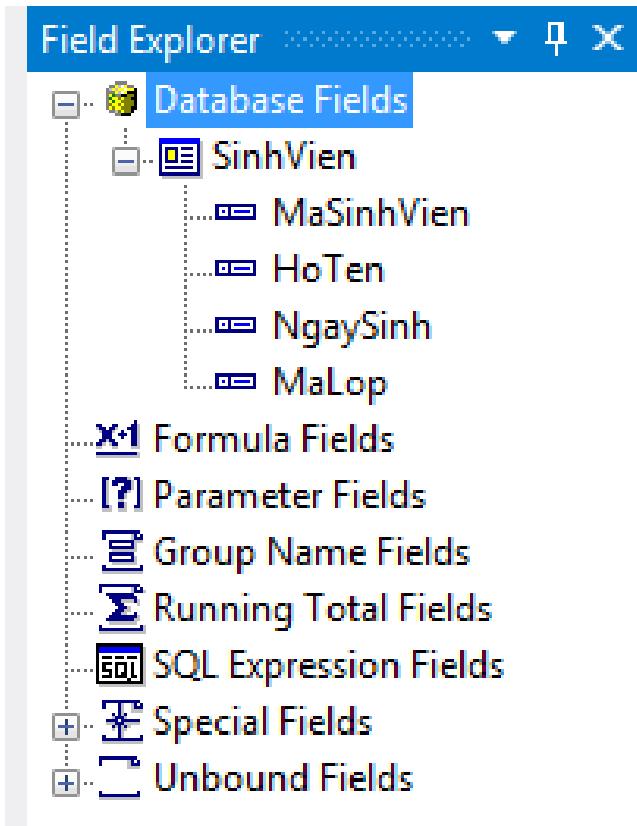
→ FINISH

Chọn Table (hoặc Add command)

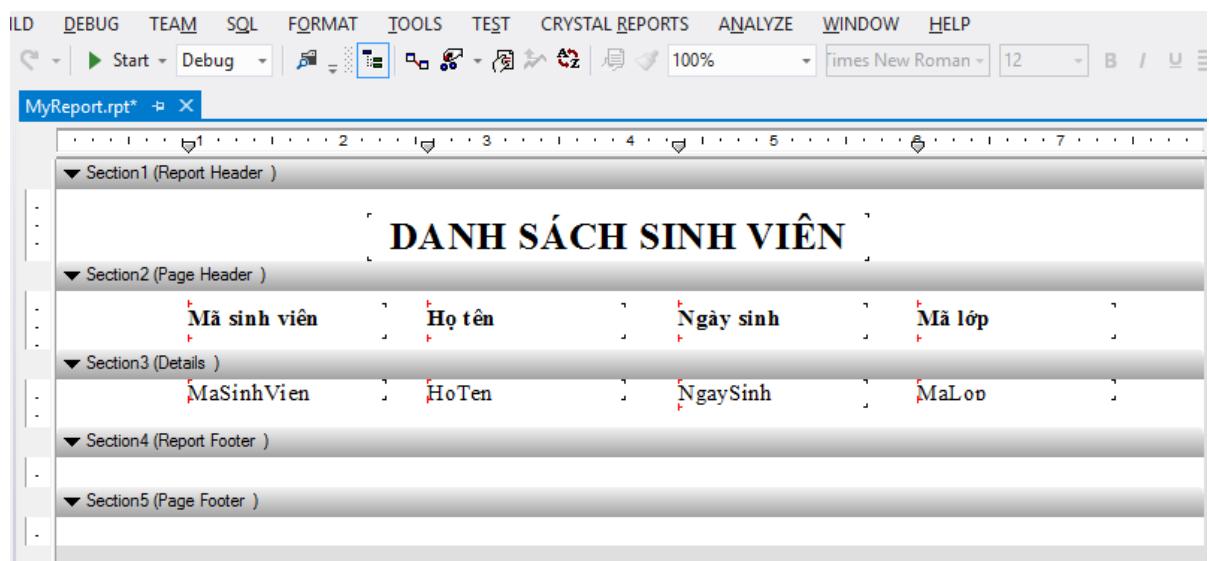


→ OK

Khi đó trong **Database Fields** xuất hiện

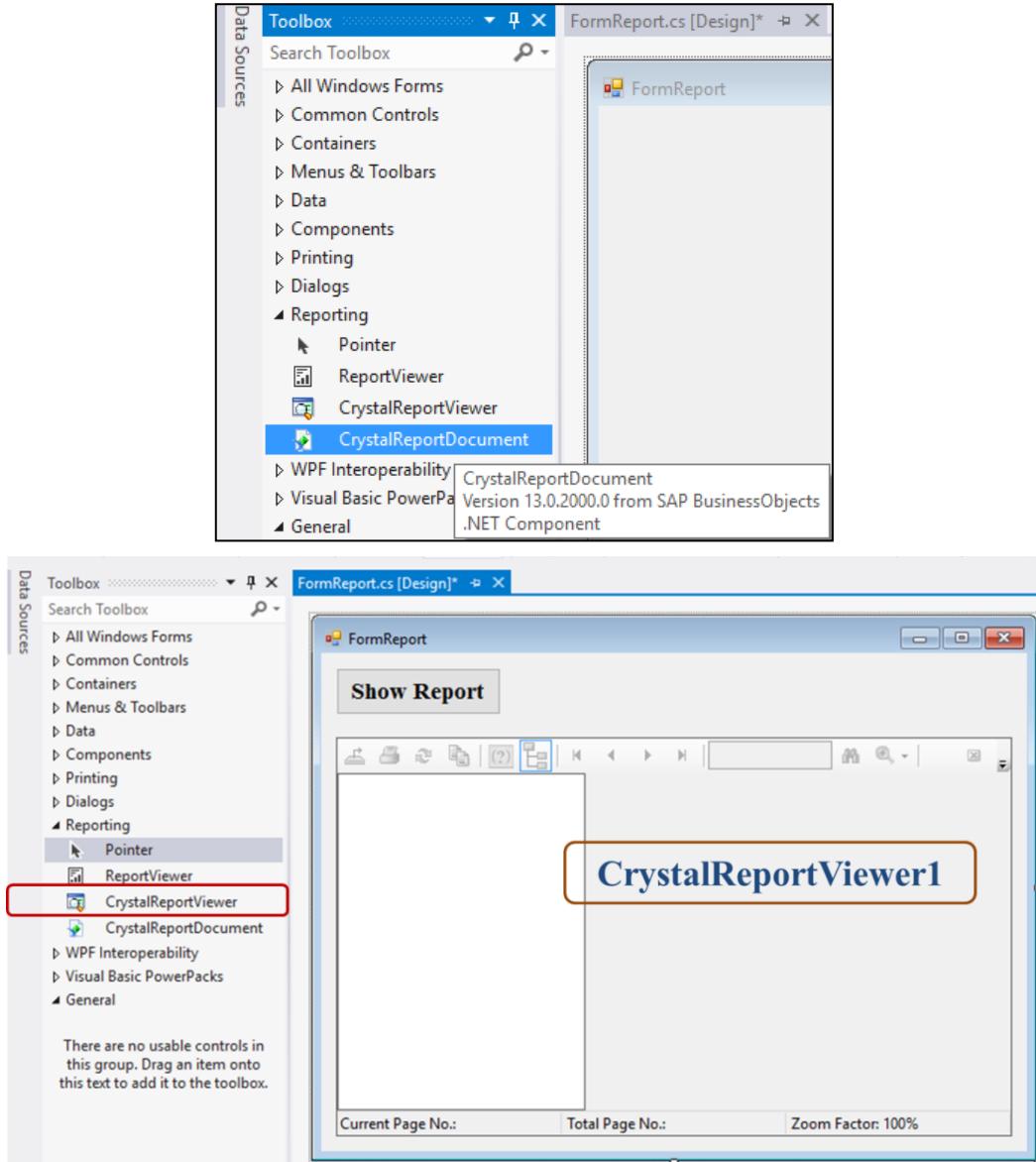


Kéo các **Fields** vào Report và sắp xếp như hình sau:



Bước 3: Thiết kế Form chính hiển thị Report

Sử dụng control **CrystalReportViewer**



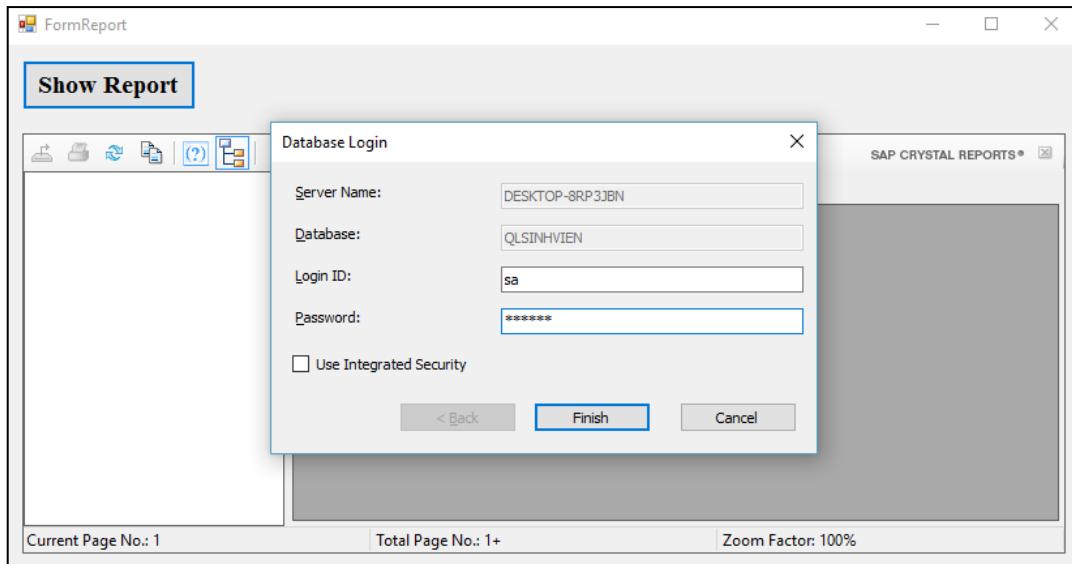
❖ Một số thuộc tính của CrystalReportViewer:

- ReportSource: nguồn Report đã thiết kế
- DisplayStatusBar: (true|false) hiển thị thanh trạng thái
- DisplayToolbar: (true|false) hiển thị thanh công cụ

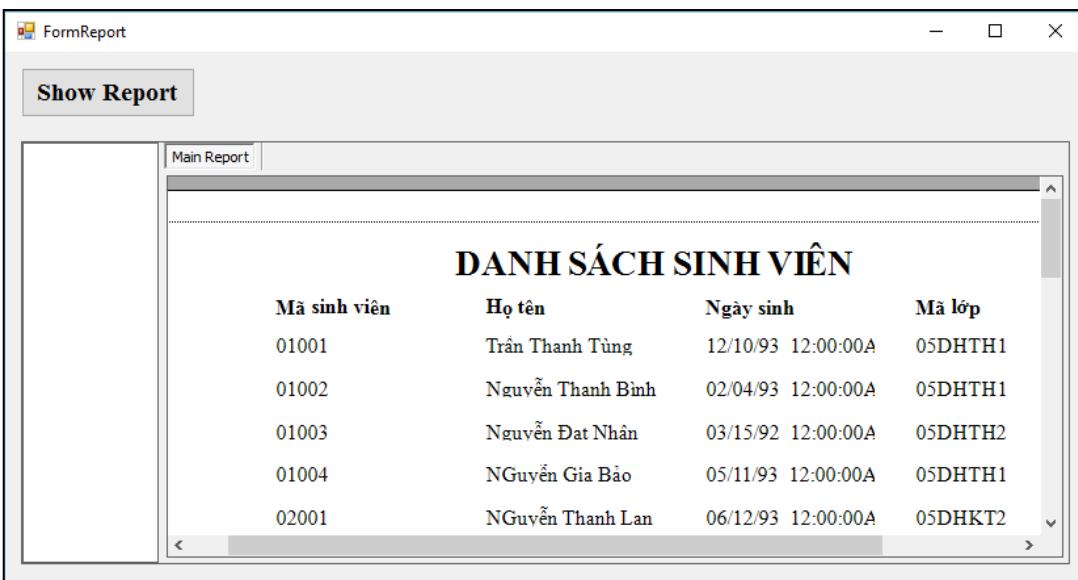
Viết phương thức xử lý sự kiện button Show Report

```
private void frmReportHS_Load(object sender, EventArgs e)
{
    MyReport rpt = new MyReport();
    crystalReportViewer1.ReportSource = rpt;
    crystalReportViewer1.DisplayStatusBar = false;
    crystalReportViewer1.DisplayToolbar = true;
    crystalReportViewer1.Refresh();
}
```

Dịch và chạy chương trình, **nhập Password** kết nối:



Kết quả như sau:

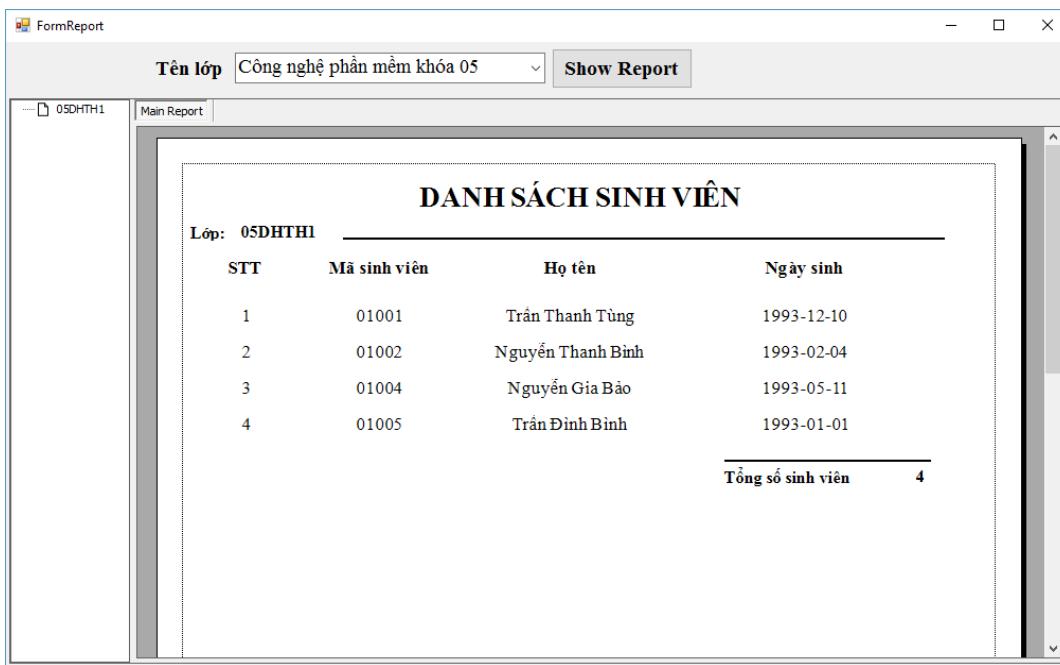


Lưu ý: Khi chạy chương trình để không phải nhập password khi kết nối thêm dòng code sau vào button Show Button

```
private void btn_showreport_Click(object sender, EventArgs e)
{
    MyReport rpt = new MyReport();
    crystalReportViewer1.ReportSource = rpt;
    rpt.SetDatabaseLogon("sa", "sa2012", "DESKTOP-8RP3JBN", "QLSINHVIEN");
    crystalReportViewer1.Refresh();
    crystalReportViewer1.DisplayToolbar = false;
    crystalReportViewer1.DisplayStatusBar = false;
}
```

2. Bài tập tại lớp

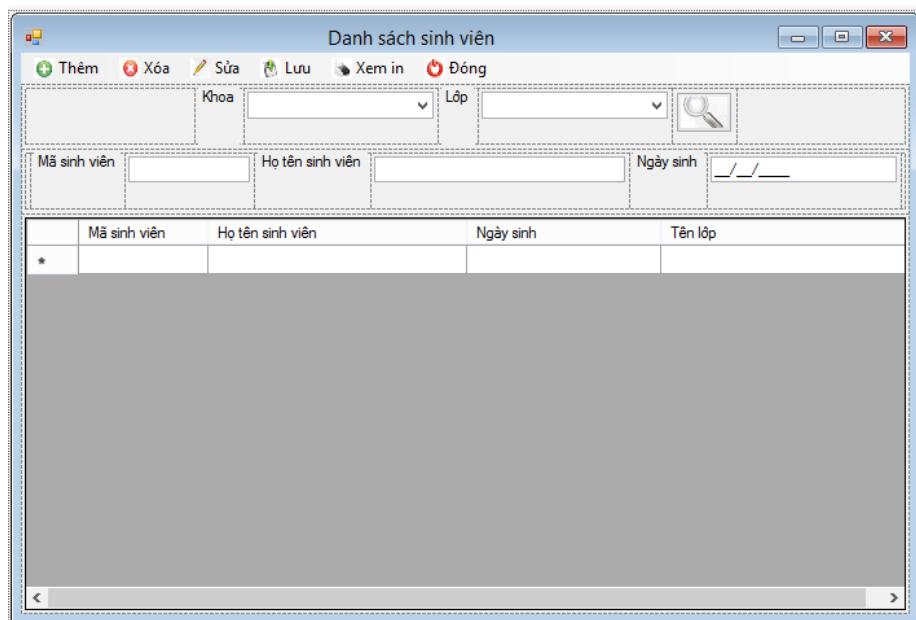
Bài 1: Thiết kế form thực hiện in danh sách theo mẫu sau:



Yêu cầu:

- Thực hiện đổ dữ liệu cho combobox tên lớp
- Thiết kế report theo mẫu
- Chọn một lớp trong combobox tên lớp thực hiện in danh sách sinh viên cho lớp đó.

Bài 2: Mở form Quản lý sinh viên (bài tập tại lớp tuần 10):



Yêu cầu: Viết lệnh cho button Xem in thực hiện in danh sách sinh viên theo mẫu report bài 1.

Trường ĐH CNTP TP.HCM
 Khoa: CNTT
 Bộ môn: CNPM
 Môn: Lập trình .NET

BÀI 13. LẬP TRÌNH THEO DỰ ÁN



A. MỤC TIÊU:

- Tổng hợp kiến thức xây dựng phần mềm ứng dụng

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

Thiết kế form Phiếu Nhập hàng như sau:

Mã phiếu nhập	Mã sản phẩm	Đơn giá	Số lượng
PN10112015001	SP001	5400000	1
PN10112015001	SP002	7600000	2
PN10112015002	SP001	5400000	2
PN10112015002	SP002	7600000	2
PN10112015003	SP001	5400000	1

Yêu cầu

- Formload:

Groupbox Phiếu nhập:

- + Textbox Mã phiếu nhập để trống, textbox ngày nhập là ngày hiện tại
- + Con trỏ nhập liệu tại textbox Mã phiếu nhập.
- + Combobox Nhà cung cấp: Chứa tất cả Mã nhà cung cấp của bảng NCC

- + Textbox thành tiền: Bị vô hiệu hóa, hiển thị số 0.

Groupbox Chi tiết phiếu nhập

- + Combobox mã phiếu nhập: Hiển thị tất cả Mã phiếu nhập trong bảng Phieunhap.
- + Combobox mã sản phẩm: Hiển thị tất cả Mã sản phẩm trong bảng Sanpham.
- + Textbox đơn giá: bị vô hiệu hóa.
- + Số lượng: Hiển thị số 0.
- + Datagridview: Hiển thị thông tin trong bảng chitietphieunhap.
- Thực hiện các chức năng còn lại tương ứng.
- Chọn button “Tạo phiếu nhập”:
 - + Lưu thông tin trong groupbox Phiếu nhập vào table PhieuNhap lưu ý kiểm tra các dữ liệu đầu vào.
 - + Load lại Combobox Mã phiếu nhập trong groupbox Chi tiết phiếu nhập.
 - + Lưu ý: Không cho thao tác trong textbox thành tiền, lúc này đang là số 0.
 - Chọn “combobox Mã phiếu nhập” trong groupbox “Chi tiết Phiếu nhập”
 - + Thông tin về phiếu nhập vừa chọn sẽ được hiển thị lên groupbox “Phiếu nhập”.
 - + Thông tin về Chi tiết phiếu nhập tương ứng hiển thị lên datagridview ChiTietPhieuNhap.
 - Chọn “Combobox Mã sản phẩm”:
 - + Tự động hiển thị đơn giá tương ứng với mã sản phẩm vừa chọn.
 - Gõ số lượng vào “textbox số lượng”:
 - + Tự động cập nhật thành tiền trên “textbox thành tiền” ứng với Phiếu nhập đang chọn.
 - Chọn “Thêm sản phẩm”
 - + Kiểm tra Mã sản phẩm trong combobox “Mã sản phẩm” đã có trong “ChiTietPhieuNhap” đang chọn hay chưa. Nếu chưa có thì lưu thông tin trong groupbox “Chi tiết phiếu nhập” vào bảng Chitietphieunhap. Ngược lại cập nhật số lượng.
 - + Cập nhật Thành tiền vào trong bảng Phieunhap.
 - Khi chọn datagridview Chi tiết phiếu nhập:
 - + Hiển thị thông tin vừa chọn lên các control trong groupbox Chi tiết phiếu nhập.

- + DataGridView Chi tiết phiếu nhập cho hiển thị ContextMenuStrip có các chức năng Sửa, Xóa Chi tiết phiếu nhập.

Trường ĐH CNTP TP.HCM
 Khoa: CNTT
 Bộ môn: CNPM
 Môn: Lập trình .NET

BÀI 14. LẬP TRÌNH THEO DỰ ÁN



A. MỤC TIÊU:

- Tổng hợp kiến thức xây dựng phần mềm ứng dụng.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Tạo cơ sở dữ liệu gồm 3 table: COSO(macoso,tencoso)
 DONVI(madonvi,tendonvi,macoso)
 GV(magv, hoten, sdt, ghichu, madonvi).
 Thí sinh tự chọn loại dữ liệu thích hợp.
2. Tạo Diagram, nhập mẫu cơ sở dữ liệu test: có 2 cơ sở, mỗi cơ sở có 2 đơn vị, mỗi đơn vị có 2 giáo viên với giá trị cột ghichu là NULL.
1. Tạo form kết nối cơ sở dữ liệu như sau:

2. Khi người dùng kích chuột vào nút thoát chương trình hiện ra cửa sổ hỏi người dùng có chắc chắn muốn thoát không
 - Nếu người dùng chọn Yes chương trình sẽ thoát khỏi chương trình
 - Nếu người dùng chọn No chương trình sẽ trở lại chương trình
3. Khi textbox tên chương trình hoặc tên cơ sở dữ liệu để trống chương trình sẽ hiển thị báo cho người dùng biết, người dùng chọn Ok chương trình sẽ quay lại textbox đó.
4. a. Nếu chương trình kết nối không thành công cơ sở dữ liệu thì thông báo cho người dùng biết biết “kết nối đến cơ sở dữ liệu thất bại”.

b. Nếu kết nối thành công thông báo cho người dùng biết kết nối thành công và hiển thị form như câu 3.

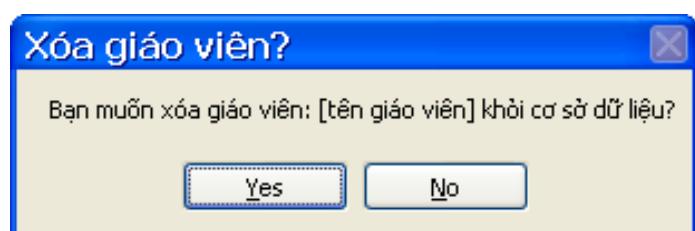
1. Tạo form theo cấu trúc:

2. a. Khi người dùng chọn một cơ sở thì chương trình hiển thị danh sách đơn vị đào tạo thuộc cơ sở đó (sắp xếp theo tên đơn vị), chưa đơn vị nào được chọn từ ComboBox này, và không có GV nào hiển thị trong danh
- b. Người dùng chọn một đơn vị để xem danh sách giáo viên (thuộc đơn vị hiện tại) hiển thị trong danh sách phía dưới (thông tin hiển thị chỉ chứa tên gv, được đánh số thứ tự).

1. Người dùng có thể bấm phải chuột trên một GV từ danh sách để hiển thị popup menu chứa 2 menu (xem hình). Nếu người dùng bấm phải chuột trên danh sách khi chưa chọn GV nào thì hai menu này sẽ bị mờ, không cho sử dụng.

2. a. Khi người dùng chọn menu [Xóa GV đang chọn], chương trình sẽ xuất hiện hộp thoại xác nhận.

b. Nếu người dùng đồng ý, chương trình sẽ xóa GV này ra khỏi database và cập nhật danh sách Trên màn hình).



c. Nếu người dùng không đồng ý trở lại chương trình.

3. a. Tạo form theo cấu trúc.

The screenshot shows a Windows application window titled "Form4". Inside, there is a sub-form with the title "Thông tin chi tiết của giáo viên". This sub-form contains five text input fields with labels: "Họ và tên giáo viên", "SĐT", "Ghi chú", "Đơn vị đào tạo", and "Cá số". Each label is followed by a text input box. At the bottom right of the sub-form is a button labeled "Quay về".

- b. Khi người dùng chọn menu [Hiển thị thông tin chi tiết GV] hoặc double click vào một GV trong danh sách, chương trình sẽ hiển thị form trên với thông tin giáo viên đang chọn.
- c. Người dùng nhấn nút quay lại để thoát khỏi form.

Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: CNPM Môn: Lập trình .NET	BÀI 15. LẬP TRÌNH THEO DỰ ÁN	
--	---	--

A. MỤC TIÊU:

- Tổng hợp kiến thức xây dựng phần mềm ứng dụng.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

Sau khi phân tích hệ thống quản lý thư viện người ta đã thiết kế các bảng cơ sở dữ liệu quản lý thư viện được tóm lược như sau:

DOCGIA

Field Name	Description
Madg	Mã độc giả
Hoten	Họ và tên độc giả
Gioitinh	Gới tính
Ngaysinh	Ngày sinh
Diachi	Địa chỉ
Coquan	Cơ quan

NHANVIEN

Field Name	Description
Manv	Mã nhân viên
Hoten	Họ và tên nhân viên
Gioitinh	Gới tính

LOAISACH

Field Name	Description
Maloai	Mã loại
Tenloai	Tên loại
Vitri	Vị trí

SACH

Field Name	Description
Masach	Mã phòng
Tensach	Tên sách
Sotrang	Số trang
Tacgia	Tác giả
Loaisach	Loại sách
Nhaxb	Nhà xuất bản
Soluong	Số lượng

MUONTRA

Field Name	Description
Manv	Mã nhân viên
Madg	Mã độc giả
Masach	Mã sách
Ngaymuon	Ngày mượn
Ngaytra	Ngày trả
Ngayhentra	Ngày hẹn trả

Hãy thực hiện các yêu cầu sau:

* *Tạo thư mục mang tên **Họ và tên** của thí sinh lên màn hình DESKTOP. Lưu bài thi vào thư mục này*

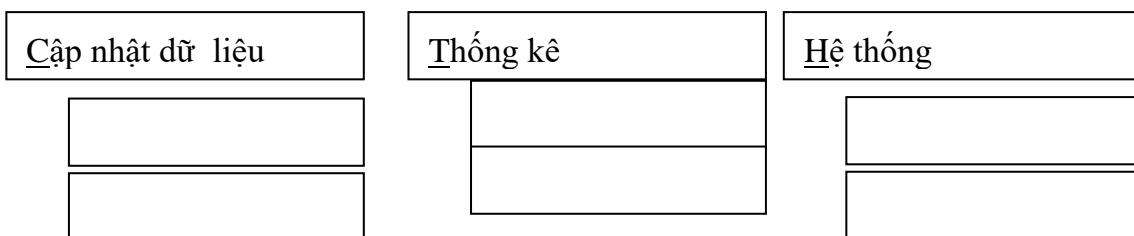
Module 1: Tạo một cơ sở dữ liệu với tên **qlthu_vien.mdf** chứa các bảng dữ liệu trên trong SQL Server. Tạo các bảng dữ liệu với kiểu dữ liệu, độ rộng,, khóa chính, khóa phụ, (các ràng buộc toàn vẹn dữ liệu đầy đủ) của các trường sao cho phù hợp của bài toán. Nhập dữ liệu tối thiểu mỗi bảng 5 bảng ghi.

Module 2: **Tạo các form cập nhật dữ liệu cho các bảng** (*Cập nhật độc giả, cập nhật sách, cập nhật bảng mượn trả*), trên form có lối hiển thị dữ liệu, bao gồm các nút thêm, sửa, lưu, xóa, in dữ liệu (trừ bảng mượn trả) thoát.

Module 3 Tạo report thống kê độc giả chưa trả sách ở thời điểm hiện tại. Tạo form thống kê độc giả chưa trả sách để sử dụng report vừa tạo.

Module 4: Tạo report thống kê các độc giả đã mượn sách theo loại sách. Tạo form sử dụng report vừa tạo.

Module 5: Tạo menu theo mẫu dưới đây rồi gắn tất cả các form đã tạo ở các câu trên vào chức năng trên menu.



.....Hết.....