

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN
---oOo---



BÀI TẬP MÔN LẬP TRÌNH WINDOWS

(WINDOWS PROGRAMMING WITH C#)

HỆ: ĐẠI HỌC

THÀNH PHỐ HỒ CHÍ MINH NĂM 2020

Mục Lục

BÀI TẬP TUẦN 01 MÔN LẬP TRÌNH WINDOWS (C#)	1
Chương I : Giới thiệu ngôn ngữ lập trình C#	1
I. Làm quen với Visual Studio 2010/2012/2013	1
II. Kiểu dữ liệu, giá trị mặc định và một số ký tự đặc biệt	2
III. Bài tập lập trình căn bản với C#.....	3
Bài 1. Viết chương trình xuất ra màn hình dòng chữ “Hello World!”.....	3
Bài 2. Viết chương trình nhập vào tên của mình và xuất ra màn hình “Hello + Tên”.....	4
Bài 3. Khi chạy chương trình xuất ra yêu cầu nhập vào hai số từ bàn phím xuất ra kết quả tổng của hai số vừa nhập.	4
Bài 4. Viết một chương trình giải phương trình bậc nhất.....	4
Bài 5. Viết chương trình giải phương trình bậc 2	4
Bài 6. Viết chương trình tính chu vi và diện tích của các hình tam giác, hình chữ nhật, hình vuông, hình tròn. Chương trình cho phép nhập vào 1 trong 4 hình trên và xuất kết quả ra màn hình.....	5
Bài 7. Viết chương trình tạo một menu như sau:.....	5
Bài 8. Dùng Console Application thực hiện các bài tập	6
Bài 9. Viết chương trình in ra tổng của 10 số chẵn đầu tiên (sử dụng vòng lặp for hoặc while).....	8
Bài 10. Viết chương trình in ra những số lẻ từ 1 đến 99.	8
Bài 11. Viết chương trình xuất ra tổng các số là bội số của 7 (từ 1 đến 100).....	8
Bài 12. Viết chương trình in ra tổng $1+2+3+\dots+n$ với n được nhập từ tham số command line.....	8
Bài 13. Viết chương trình in ra tổng $1+3+5+\dots+n$ nếu n là số chẵn, $2+4+6+\dots+n$ nếu n là số lẻ. Giá trị n được nhập vào từ tham số command line	8
Bài 14. Viết chương trình in ra giá trị lớn nhất và nhỏ nhất trong một dãy các giá trị user đã nhập vào từ tham số command line.....	8
Bài 15. Viết chương trình giải phương trình bậc 1 với hệ số a, b được nhập vào bởi user từ tham số command line.	8
Bài 16. Viết chương trình đọc một giá trị nguyên từ bàn phím và in ra số đó là số chẵn, lẻ hoặc zero	9
Bài 17. Viết chương trình in ra bội số của 3 từ 300 đến 3.....	9
Bài 18. Viết chương trình kiểm tra số nhập vào có phải là số nguyên tố hay không.....	9
Bài 19. Viết chương trình tìm USCLN của 2 số nhập vào	9
Bài 20. Viết chương trình tính tổng N số nguyên.	9
Bài 21. Tính tổng các số nguyên tố nhỏ hơn N	9
Bài 22. Tính tổng N số nguyên tố đầu tiên.....	9
Bài 23. Viết hàm tính giai thừa của một số nguyên n bằng phương pháp đệ qui.	9
Bài 24. Viết hàm	9
Bài 25. Viết chương trình theo dạng thực đơn đơn giản, cho phép người dùng thực hiện các thao tác sau:.....	9
IV. 4. Bài tập phần mảng.....	10

Bài 1.	Mảng	10
Bài 2.	Xử lý mảng.....	10
Bài 3.	Nhập số phần tử và các phần tử nguyên dương của mảng a.	10
Bài 4.	Xử lý mảng.....	10
Bài 5.	Xử lý mảng.....	10
Bài 6.	Xử lý mảng.....	10
Bài 7.	Viết chương trình nhập vào ma trận vuông A[N][N].	11
Bài 8.	Hãy nhập số nguyên dương cho một mảng hai chiều m dòng và n cột. Cho biết mảng đó chứa bao nhiêu số có: ba chữ số, hai chữ số và một chữ số.	11
V.	Bài tập lập trình hướng đối tượng với C#	12
Bài 1.	Viết chương trình tính diện tích, chu vi hình chữ nhật.....	12
Bài 2.	Viết chương trình OOP quản lý sinh viên đơn giản: Nhập, xuất thông tin, tính điểm TB.....	12
Bài 3.	Tạo một project có 2 class như sau:	13
Bài 4.	Tạo một project có 3 class như sau:	14
Bài 5.	(Inheritance and Interfaces) Tạo ứng dụng để tính toán thu nhập cho các cầu thủ bóng chày. Ứng dụng gồm các lớp sau:	15
Bài 6.	Viết chương trình xây dựng đối tượng Person gồm có các thuộc tính sau:	16
Bài 7.	Viết chương trình xây dựng đối tượng Shape gồm có các thuộc tính sau:	16
Bài 8.	(Exception) Viết chương trình xây dựng đối tượng Sinh viên gồm có các thuộc tính sau:.....	16
Bài 9.	Ngân hàng ABC muốn lưu trữ thông tin của mỗi tài khoản như sau:.....	17
Bài 10.	Một phân số gồm có 2 thuộc tính là tử số (numerator) và mẫu số (denominator). Hãy thiết kế class Phân số (Rational) để thực hiện các chức năng sau:	17
Bài 11.	Viết lớp HangThucPham mô tả một hàng hóa là hàng thực phẩm trong kho của một siêu thị, có các thuộc tính: mã hàng (<i>không cho phép sửa, không được để rỗng</i>), tên hàng (<i>không được để rỗng</i>), đơn giá (>0), ngày sản xuất và ngày hết hạn (<i>ngày không được để rỗng, ngày hết hạn phải sau ngày sản xuất</i>).	17
Bài 12.	18
Bài 13.	Công ty du lịch V quản lý thông tin là các chuyến xe. Thông tin của 2 loại chuyến xe:	18
Bài 14.	Thư viện X quản lý danh sách các loại sách . Thông tin về các loại sách:	18
Bài 15.	Xây dựng chương trình quản lý danh sách các giao dịch. Hệ thống quản lý 2 loại giao dịch:	19
Bài 16.	Xây dựng chương trình quản lý danh sách các giao dịch nhà đất. Thông tin bao gồm:	19
Bài 17.	Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng. Thông tin bao gồm các loại khách hàng :	19
Chương II :	Events and Delegate	21
Bài 1.	Cho đoạn chương trình sau:	21
Bài 2.	Tạo delegate sử dụng Delegate và phương thức anonymous.....	22
Bài 3.	Tạo 1 delegate để thực hiện toán tử 2 ngôi (+, -, *, /, %).	22
Bài 4.	Thực hiện kỹ thuật multicasting delegate viết hàm cho phép gán nhiều phương thức vào cùng một delegate (các phương thức này có cùng kiểu trả về và tham số với delegate).....	22

Bài 5.	Xây dựng hàm Sort tổng quát cho cho mảng đối tượng có kiểu bất kỳ.	23
Bài 6.	Quá trình xử lý sự kiện trong .NET Framework được mô tả:.....	24
Bài 7.	Xây dựng 1 lớp thực hiện yêu cầu: “cứ mỗi giây sẽ phát sinh 1 sự kiện”	25
Chương III :	Lập trình giao diện Windows	28
Bài 1.	Tạo Project dạng Windows Forms	28
II. Bài tập sử dụng các basic controls	31	
Bài 1.	Thực hiện Form sau:.....	31
Bài 2.	Tạo một chương trình dùng để qui đổi tiền từ VND sang USD và ngược lại, từ VND sang EUR và ngược lại. 34	
Bài 3.	Thêm vào project một windows form, hiện thực bài toán giải phương trình bậc 1:	35
Bài 4.	Viết chương trình đọc năm Âm lịch khi có năm Dương lịch tương ứng, biết:.....	36
Bài 5.	Viết chương trình nhập vào số nguyên N, kiểm tra N có phải là số nguyên tố hay không? (xem giao diện)36	
Bài 6.	Viết chương trình Nhập vào 2 số a, b. Xuất ra USCLN va BSCLN của 2 số a và b (tự vẽ giao diện).....	37
Bài 7.	Xử lý control động, các dãy ghế khi run time được tạo ra trên form (thiết kế không có)	37
Bài 8.	Tạo giao diện như hình: nhập vào 2 số và 4 Radio, click vào Radio Cộng sẽ thực hiện phép Cộng 2 số và xuất kết quả vào Label Kết quả. Thực hiện tương tự cho các Radio còn lại. (kiểm tra dữ liệu nhập phải là số).....	38
Bài 9.	Viết chương trình dùng để định dạng LABEL TEXT theo các Checkbox và Radio button tương ứng (Font Style, Color).....	38
Bài 10.	Viết chương trình dùng để định dạng TEXTBOX theo các font chữ tương ứng với từng Radio button...39	
Bài 11.	Viết chương trình Quản lý thu tiền cho quán Café Sinh Viên như sau:	39
Bài 12.	Viết chương trình kiểm tra sức khỏe của 1 người bằng cách nhập vào các thông số: họ tên (không được rỗng), chiều cao (cm), cân nặng (kg) dữ liệu này phải là số. Tính cân số cân chuẩn của người đó theo công thức sau (tùy theo Nam hay Nữ):	40
Bài 13.	Viết chương trình thực hiện như sau (xem hình bên dưới).....	41
Bài 14.	Viết chương trình thêm các phần tử vào listbox (listbox được chọn nhiều phần tử) các số tự nhiên N được nhập từ textbox	41
Bài 15.	Viết chương trình nhập danh sách các Sinh Viên theo yêu cầu sau: (xem hình bên dưới).	42
Bài 16.	Viết chương trình từ điển Anh – Việt và Việt – Anh (dùng TabControl).....	43
Bài 17.	Viết chương trình cho phép Sinh Viên đăng ký học các môn học trong học kỳ:	43
Bài 18.	Thực hiện Form xử lý chuỗi có giao diện như sau:	44
Bài 19.	Thiết kế giao diện để thực hiện nhập vào 4 số a, b, c, d nguyên dương khác nhau. Hãy in ra màn hình hai số không phải là số lớn nhất và số nhỏ nhất. Yêu cầu không sử dụng mảng dữ liệu trong chương trình.	45
Bài 20.	Thiết kế giao diện để thực hiện nhập vào 3 số nguyên dương a, b, c (khác nhau). Hãy in ra màn hình 3 số trên theo thứ tự tăng dần (chương trình chỉ được dùng thêm tối đa hai biến có kiểu là số nguyên, không sử dụng biến kiểu mảng).....	45
Bài 21.	Thiết kế giao diện để viết chương trình nhập vào một số nguyên dương n và thực hiện các công việc sau: 46	
Bài 22.	Viết chương trình nhập vào ngày, tháng, năm:	46

Bài 23.	Viết chương trình theo dạng thực đơn đơn giản, cho phép người dùng thực hiện các thao tác sau:.....	46
Bài 24.	Xây dựng chương trình hiển thị lịch tờ như sau:	46
Bài 25.	Xây dựng chương trình máy tính cá nhân.....	47
Bài 26.	Viết chương trình thực hiện việc thực hành đánh máy chữ, giao diện như hình bên.	48
III.	Bài tập sử dụng các GUI advanced controls	49
Bài 1.	Sử dụng ListView, PictureBox thiết kế chương trình xử lý hình ảnh đơn giản tương tự như sau:.....	49
Bài 2.	Quản lý thông tin tài khoản cần lưu trữ các thông tin sau: số tài khoản, tên khách hàng, địa chỉ khách hàng và số tiền trong tài khoản. Thiết kế chương trình quản lý thông tin tài khoản cho một ngân hàng tương tự như sau: 50	
Bài 3.	Viết chương trình quản lý cho thuê CD, thông tin của khách thuê bao gồm mã KH, họ tên, số lượng CD thuê và tình trạng cho thuê.	51
Bài 4.	Viết chương trình quản lý danh sách sinh viên ở dạng cây như hình minh họa.....	51
Bài 5.	Thiết kế ứng dụng dạng cây như sau:	52
Bài 6.	Hãy tạo một project xử lý động control. project giúp cho rạp chiếu phim quản lý việc bán vé của mình. Rạp có 6 hàng ghế, mỗi hàng có 3 ghế được tạo ra khi chạy chương trình, các ghế được đánh số từ 1 đến 15 và được phân thành 3 lô:.....	53
Bài 7.	Hãy tạo một project có giao diện như sau:.....	54
Bài 8.	Viết chương trình quản lý thông tin thiệp chúc mừng nhân dịp các ngày lễ của một cửa hàng bán thiệp, thông tin của thiệp bao gồm mã thiệp (mã số này là duy nhất), mô tả, nhà sản xuất, loại thiệp (chúc xuân, giáng sinh, sinh nhật), năm sản xuất và số lượng hiện có.....	55
Bài 9.	Viết chương trình quản lý thông tin truyện tranh của một cửa hàng bán sách, thông tin của một cuốn truyện tranh bao gồm mã truyện (mã số này là duy nhất), tên truyện, nhà xuất bản, họ tên tác giả, năm xuất bản và số lượng hiện có.....	56
Bài 10.	Viết chương trình quản lý thông tin về lời các bài nhạc của một website nghe nhạc trực tuyến, thông tin của lời một bài nhạc bao gồm mã số (mã số này là duy nhất), tên bài, nội dung, tác giả, ca sĩ thể hiện và dòng nhạc (nhạc trẻ, nhạc trữ tình ...).	57
Bài 11.	Viết chương trình quản lý sách chuyên ngành CNTT.....	58
Bài 12.	Viết chương trình quản lý giá ngoại tệ.....	59
Bài 13.	Viết chương trình quản lý nhân sự.	60
Chương IV :	Tập hợp, chuỗi, Lambda Expressions và Extension Methods	63
I.	Bài tập phần Collection.....	65
Bài 1.	Viết một chương trình tạo ra một lớp tên là LopHoc, trong đó có khai báo bộ chỉ mục chỉ đến tên của từng học viên trong lớp. Cho phép một lớp có tối đa 30 học viên. Tạo chương trình minh họa cho phép người dùng nhập vào tên của từng học viên. Xuất kết quả danh sách học viên của lớp thông qua bộ chỉ mục.....	65
Bài 2.	Viết chương trình sử dụng ArrayList để tạo một mảng.	65
Bài 3.	Viết chương trình tạo ra đối tượng Queue tên là myQueue. Khởi tạo myQueue có 5 giá trị ngẫu nhiên. Hãy thực hiện các bước sau, mỗi bước thực hiện phải xuất tình trạng của myQueue:	65
Bài 4.	Viết chương trình tạo đối tượng Stack tên là myStack. Khởi tạo myStack có 5 giá trị ngẫu nhiên. Hãy thực hiện các bước sau, mỗi bước thực hiện phải xuất tình trạng của myStack:.....	67

Bài 5.	Viết chương trình sử dụng kiểu dữ liệu từ điển để quản lý thông tin của một lớp học. Trong đó khóa là chuỗi mã số học viên còn giá trị là tên của học viên. Viết chương trình minh họa cho phép nhập vào 10 học viên, và cho phép người dùng tìm kiếm tên của học viên thông qua mã số học viên.....	67
Bài 6.	Công ty TrueLove:	67
Bài 7.	Phòng học được quản lý trong một trường đại học gồm: phòng học lý thuyết, phòng máy tính và phòng thí nghiệm	67
Bài 8.	Quản lý khách hàng xếp hàng mua vé tại nhà ga. Thông tin lưu trữ cho khách hàng gồm: số CMND khác hàng (String), Tên khách hàng, Ga đến, giá tiền (double).....	68
Bài 9.	Viết phương thức Generic cho phép in ra mảng các phần tử, phương thức này cho phép in phần tử mảng của nhiều kiểu dữ liệu khác nhau.	69
Bài 10.	Một quân bài trong bộ bài gồm hai thuộc tính loại bài (cơ, rô, chuồn, bích) và thứ tự quân bài(2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A). Dùng Generic.....	69
Bài 11.	Viết chương trình cho phép người dùng nhập vào một chuỗi. Sau đó đếm số ký tự xuất hiện của từng ký tự trong chuỗi.....	70
Bài 12.	Viết chương trình tìm một chuỗi con trong một chuỗi cho trước. Chương trình cho phép người dùng nhập vào một chuỗi, và chuỗi con cần tìm. Kết quả là chuỗi con có tìm thấy hay không, nếu tìm thấy thì hãy đưa ra vị trí đầu tiên tìm thấy.....	70
Bài 13.	Viết chương trình tìm số lần xuất hiện một chuỗi con trong một chuỗi cho trước. Chương trình cho phép người dùng nhập vào một chuỗi và chuỗi con cần đếm. Kết quả hiển thị chuỗi, chuỗi con và các vị trí mà chuỗi con xuất hiện trong chuỗi.	70
Bài 14.	Viết chương trình cho phép người dùng nhập vào một chuỗi, rồi thực hiện việc đảo các ký tự trong chuỗi theo thứ tự ngược lại.....	71
Bài 15.	Viết chương trình cắt các từ có nghĩa trong câu.....	71
Bài 16.	Hãy nhập một chuỗi kí tự. In ra màn hình đảo ngược của chuỗi đó.	71
Bài 17.	Nhập chuỗi. Thống kê số các chữ số '0', số chữ số '1', ..., số chữ số '9' trong chuỗi.	71
Bài 18.	In ra vị trí kí tự trắng đầu tiên từ bên trái (phải) một chuỗi kí tự.....	71
Bài 19.	Nhập chuỗi. In ra tất cả các vị trí của chữ 'a' trong chuỗi và tổng số lần xuất hiện của nó.	71
Bài 20.	Nhập chuỗi. Tính số từ có trong chuỗi. In mỗi từ trên một dòng.....	71
Bài 21.	Nhập chuỗi họ tên, in ra họ, tên dưới dạng viết hoa.....	71
Bài 22.	Thay kí tự x trong chuỗi s bởi kí tự y (s, x, y được đọc vào từ bàn phím).....	71
Bài 23.	Viết chương trình xóa chuỗi con trong một chuỗi.....	71
Bài 24.	Viết chương trình trích chuỗi con trong một chuỗi.....	71
Bài 25.	Viết chương trình in các từ của chuỗi trên mỗi dòng	72
Bài 26.	Viết chương trình tìm vị trí một chuỗi con trong một chuỗi đã cho.....	72
Bài 27.	Viết chương trình ghép hai chuỗi thành một chuỗi.....	72
Bài 28.	Viết chương trình loại bỏ các khoảng trắng thừa trong một chuỗi.....	72
II. 3	Bài tập phần Lambda Expressions và Extension Methods	72
Bài 2.	Tạo Windows Application project.....	78
Chương V :	Xử lý lưu trữ dữ liệu với LINQ và 3-Tier Applications.....	80

I.	LINQ to Object	80
Bài 1.	Thực hiện chương trình quản lý thông tin của công ty tương tự như sau:	80
Bài 2.	Thực hiện chương trình quản lý thông tin của <i>lớp học và sinh viên</i> tương tự như sau:	81
Bài 3.	Thực hiện chương trình quản lý danh bạ.....	81
Bài 4.	Thiết kế form cho phép người sử dụng nhập vào danh sách họ tên (lưu trong List<String> và hiển thị trên ListBox) 82	
Bài 5.	Thực hiện các ví dụ và bài tập: https://code.msdn.microsoft.com/101-LINQ-Samples-3fb9811b	82
II.	LINQ to SQL	82
Bài 1.	(Computer Manager)	82
Bài 2.	(Ebook Manager).....	84
Bài 3.	(Warehouse Manager).....	87
Bài 4.	(Quản lý câu hỏi trắc nghiệm).....	90
III.	Xây dựng ứng dụng 3-Tier	95
Bài 1.	Tạo cơ sở dữ liệu bằng Access quản lý thông tin bác sĩ điều trị trong một bệnh viện, cấu trúc các bảng: 95	
Bài 2.	Thực hiện quản lý thông tin đội bóng và các thành viên trong các đội này. Chương trình cho phép hiển thị thông tin các đội bóng trên ComboBox, khi sự kiện chọn trên ComboBox này, thông tin các cầu thủ sẽ hiện trên DataGridView	96
Bài 3.	Tạo cơ sở dữ liệu bằng SQL Server quản lý sản phẩm với cấu trúc các bảng như hình sau:	97
Bài 4.	Cho 1 phần CSDL quản lý sinh viên ở phần đăng ký môn học có lược đồ quan hệ được cho như hình vẽ sau: 99	

Phân bổ thời gian thực hành (10 tuần, 3 tiết/tuần)

- Tuần 01: *Chương 1: Giới thiệu ngôn ngữ lập trình C# (bài?? Bat buoc phai nop)*
- Tuần 02: *Chương 2: Events & Delegate*
- Tuần 03-04-05: *Chương 3: Lập trình giao diện Windows*
Kiểm tra giữa kỳ (thực hành)
- Tuần 06-07: *Chương 4: Tập hợp, chuỗi, Lambda Expressions và Extension Methods*
- Tuần 08-09-10: *Chương 5: Xử lý lưu trữ dữ liệu với LINQ và 3-Tier Applications*
Ôn tập, bài tập tổng hợp và kiểm tra cuối kỳ (thực hành)

BÀI TẬP TUẦN 01 MÔN LẬP TRÌNH WINDOWS (C#)

Chương I : Giới thiệu ngôn ngữ lập trình C#

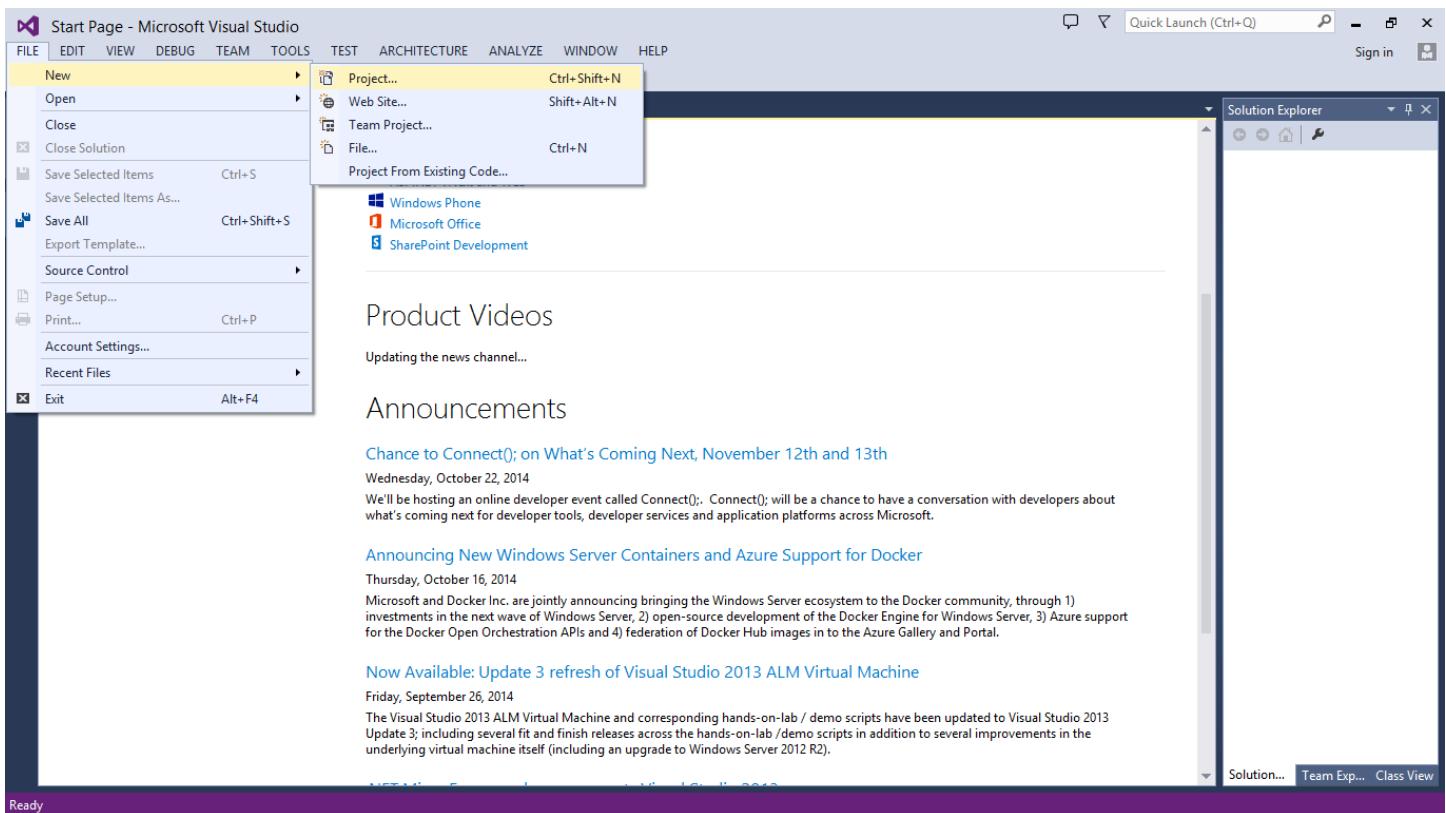
Mục tiêu:

- Sử dụng thành thạo các thao tác tạo Solution, Project trong Visual Studio 2010/2012/2013
- Viết được chương trình ứng dụng ở dạng Console Application
- Hiểu rõ và áp dụng vào chương trình: Kiểu dữ liệu, biến - hằng, biểu thức, các câu lệnh, toán tử trong ngôn ngữ lập trình C#.
- Hiểu rõ và áp dụng vào chương trình: Các khái niệm hướng đối tượng (object, class, inheritance, encapsulation, polymorphism, ...) trong ngôn ngữ lập trình C#.
- Viết được chương trình ứng dụng các khái niệm HDT ở dạng Console Application

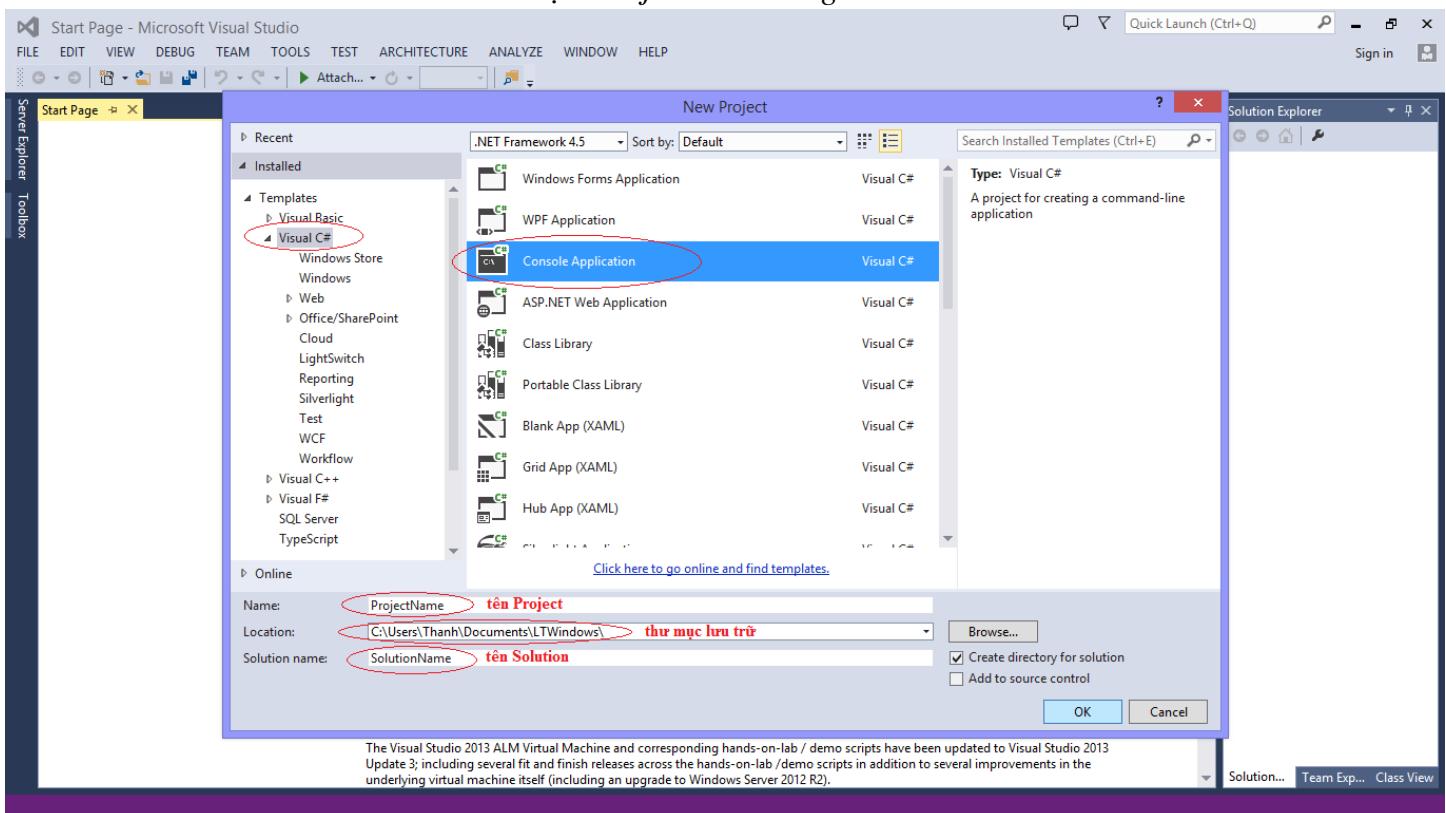
Yêu cầu:

- Tất cả các bài tập lưu trong thư mục: T:\MaSV_HoTen\Tuan01\
- Tạo Solution Tuan01 trong thư mục T:\MaSV_HoTen\Tuan01\ trong Visual Studio 2010/2012/2013. Mỗi bài tập là một Project riêng biệt.
- Cuối mỗi buổi thực hành, SV phải nén (.rar hoặc .zip) thư mục làm bài và nộp lại bài tập đã thực hiện trong buổi đó.
- Bài tập bắt buộc của Tuần XX, GV sẽ công bố cuối giờ. Bài tập bắt buộc sẽ gửi vào email giáo viên quy định ví dụ thangvnguyen2001@gmail.com với Subject và tập tin nén kèm theo: **XY_ZK_LTWINDOWS_TUANXX_HOTENSINHVIEN** (trong đó XY là lớp ZK là nhóm XX sẽ là tuần)

I. Làm quen với Visual Studio 2010/2012/2013



Hình 1. Tạo Project mới trong Visual Studio



Hình 2. Đặt tên cho Project, Solution và chọn thư mục lưu trữ

II. Kiểu dữ liệu, giá trị mặc định và một số ký tự đặc biệt

Kiểu C#	Số byte	Kiểu .NET	Mô tả
byte	1	Byte	Số nguyên dương không dấu từ 0-255
char	2	Char	Ký tự Unicode
bool	1	Boolean	Giá trị logic true/ false
sbyte	1	Sbyte	Số nguyên có dấu (-128 đến 127)
short	2	Int16	Số nguyên có dấu giá trị từ -32768 đến 32767.
ushort	2	UInt16	Số nguyên không dấu 0 – 65.535
int	4	Int32	Số nguyên có dấu -2.147.483.647 và 2.147.483.647
uint	4	UInt32	Số nguyên không dấu 0 – 4.294.967.295
float	4	Single	Kiểu dấu chấm động, giá trị xấp xỉ từ 3,4E-38 đến 3,4E+38, với 7 chữ số có nghĩa..
double	8	Double	Kiểu dấu chấm động có độ chính xác gấp đôi, giá trị xấp xỉ từ 1,7E-308 đến 1,7E+308, với 15,16 chữ số có nghĩa.
decimal	8	Decimal	Có độ chính xác đến 28 con số và giá trị thập phân, được dùng trong tính toán tài chính, kiểu này đòi hỏi phải có hậu tố “m” hay “M” theo sau giá trị.

Hình 3. Kiểu dữ liệu

Ký tự	Ý nghĩa
\'	Dấu nháy đơn
\\"	Dấu nháy kép
\\"\\	Dấu chéo
\0	Ký tự null
\a	Alert
\b	Backspace
\f	Sang trang form feed
\n	Dòng mới
\r	Đầu dòng
\t	Tab ngang
\v	Tab dọc

Kiểu dữ liệu	Giá trị mặc định
int, long, byte,...	0
bool	false
char	'\0' (null)
enum	0
reference	null

Hình 4. Một số ký tự đặc biệt và giá trị mặc định cho kiểu dữ liệu

III. Bài tập lập trình căn bản với C#

Bài 1. Viết chương trình xuất ra màn hình dòng chữ “Hello World!”.

Hướng dẫn:

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

Bài 2. Viết chương trình nhập vào tên của mình và xuất ra màn hình “Hello + Tên”.

Hướng dẫn:

- Khai báo biến chuỗi:
`string ten;`
- Nhập chuỗi từ bàn phím:
`Ten = Console.ReadLine();`
Nếu là số: `so = Console.Read();`
- Xuất ra màn hình:
`Console.WriteLine("Chuỗi" + biến);` Hoặc `Console.WriteLine("Hello {0}!", biến);`

Bài 3. Khi chạy chương trình xuất ra yêu cầu nhập vào hai số từ bàn phím xuất ra kết quả tổng của hai số vừa nhập.

Xuất ra màn hình yêu cầu có dạng:

Nhap So thu nhat A: ??

Nhap So thu hai B: ??

Tong cua ?? Va ?? La: ??

Hướng dẫn:

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

Bài 4. Viết một chương trình giải phương trình bậc nhất.

Hướng dẫn:

```
int a;
float b;
a= Convert.ToInt32(Console.ReadLine());
b = float.Parse(Console.ReadLine());
```

Bài 5. Viết chương trình giải phương trình bậc 2.

Hướng dẫn:

- Lưu ý $ax^2+bx+c=0$ phương trình có 5 trường hợp: vô nghiệm, vô số nghiệm và có một nghiệm, có 2 nghiệm phân biệt, có 1 nghiệm kép.

- *Lớp Math*

`Math.Pow(x,y)` lấy x^y

`Math.Sqrt(x)` lấy căn bậc 2 của x

`Math.Round(x,2)` làm tròn x tới 2 đơn vị

`Math.PI` lấy số PI

`Math.Sin(a)` tính Sin, với a là radian

`Math.Cos(a)` tính Cos, với a là radian

`Math.Tan(a)` tính Tan, với a là radian

Bài 6. Viết chương trình tính chu vi và diện tích của các hình tam giác, hình chữ nhật, hình vuông, hình tròn. Chương trình cho phép nhập vào 1 trong 4 hình trên và xuất kết quả ra màn hình.

Hướng dẫn:

- Tạo menu Chọn chức năng

1. Tính tam giác

2. Tính

`switch (chucnang)`

{

case 1:

```
Xử lý 1();
break;
```

case 2:

```
Xử lý 1();
break;
```

default:

...

}

- Trong trường hợp hình là tam giác, nhập cạnh a > 0, b > 0, c > 0 . Nếu a, b, c tạo thành tam giác thì hãy tính và xuất chu vi, diện tích hình tam giác. Ngược lại thông báo “Không tạo thành tam giác”

$$cv = a + b + c, \quad p = cv/2 \text{ và } dt = \sqrt{p * (p - a) * (p - b) * (p - c)}$$

Bài 7. Viết chương trình tạo một menu như sau:

1. Nhập hai số nguyên a, b
2. Xuất tổng a + b
3. Xuất hiệu a – b
4. Xuất tích a * b
5. Xuất thương a/b
6. Kết thúc

Khi người sử dụng nhập một số nguyên, thực hiện công việc tương ứng.

Bài 8. Dùng Console Application thực hiện các bài tập

a.

```
*          * * * * * * *
* *        * * * * * *
* * *      * * * * *
* * * *    * * * *
* * * * *   * *
* * * * * * *
```

b.

```
$      $$$$$$$
$ $    $$$$$$
$ $$   $$$$ $
$ $$ $  $$$ $
$ $$ $$ $ $ $
$ $$ $$ $ $ $ $ $ $
```

c.

```
* 
* * *
* * * * *
* * * * * * *
* * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * * *
```

d.

```
1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
6 7 8 9 0 1 0 9 8 7 6
7 8 9 0 1 2 3 2 1 0 9 8 7
8 9 0 1 2 3 4 5 4 3 2 1 0 9 8
```

9 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 9
0 1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 0

e.

1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
6 7 8 9 0 1 0 9 8 7 6
7 8 9 0 1 2 3 2 1 0 9 8 7
8 9 0 1 2 3 4 5 4 3 2 1 0 9 8
9 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 9
0 1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 0

f.

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

g.

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

* *
*

h.
*
* * *
* * * *
* * * * *
* * * * * *
* * * * * *
* * * * *
* * * *
* *
*

Hướng dẫn:

- Sử dụng cấu trúc lặp
- Lưu ý các đường chéo

Bài 9. Viết chương trình in ra tổng của 10 số chẵn đầu tiên (sử dụng vòng lặp for hoặc while)

Bài 10. Viết chương trình in ra những số lẻ từ 1 đến 99.

Bài 11. Viết chương trình xuất ra tổng các số là bội số của 7 (từ 1 đến 100)

Bài 12. Viết chương trình in ra tổng $1+2+3+\dots+n$ với n được nhập từ tham số command line

Bài 13. Viết chương trình in ra tổng $1+3+5+\dots+n$ nếu n là số chẵn, $2+4+6+\dots+n$ nếu n là số lẻ. Giá trị n được nhập vào từ tham số command line

Bài 14. Viết chương trình in ra giá trị lớn nhất và nhỏ nhất trong một dãy các giá trị user đã nhập vào từ tham số command line.

Bài 15. Viết chương trình giải phương trình bậc 1 với hệ số a, b được nhập vào bởi user từ tham số command line.

Bài 16. Viết chương trình đọc một giá trị nguyên từ bàn phím và in ra số đó là số chẵn, lẻ hoặc zero

Bài 17. Viết chương trình in ra bộ số của 3 từ 300 đến 3.

Bài 18. Viết chương trình kiểm tra số nhập vào có phải là số nguyên tố hay không

Bài 19. Viết chương trình tìm USCLN của 2 số nhập vào.

Bài 20. Viết chương trình tính tổng N số nguyên.

Bài 21. Tính tổng các số nguyên tố nhỏ hơn N.

Bài 22. Tính tổng N số nguyên tố đầu tiên.

Bài 23. Viết hàm tính giai thừa của một số nguyên n bằng phương pháp đệ qui.

(a) Viết hàm tính x^n , x kiểu số thực, n là số nguyên dương.

(b) Sử dụng các hàm nói trên và các hàm cần thiết khác để tính tổng sau bằng phương pháp đệ qui.

$$S=x - x^3/3! + x^5/5! - \dots + (-1)^{n+1} * x^{2n-1}/(2n-1)!$$

Với x,n nhập từ bàn phím, x là số thực, n số nguyên dương.

Bài 24. Viết hàm

(a) Viết hàm tính x^n .

(b) Viết hàm tính $n!$ theo phương pháp đệ qui.

(c) Dùng hàm trên để tính: $S = x/1! + x^3/3! + x^5/5! + \dots$ Với x, n được nhập vào từ bàn phím, x là số thực, n là số nguyên dương.

Bài 25. Viết chương trình theo dạng thực đơn đơn giản, cho phép người dùng thực hiện các thao tác sau:

(a) Nhập vào năm dương lịch, đổi sang năm âm lịch.

(b) Nhập vào năm âm lịch, đổi sang năm dương lịch.

(c) Nhập vào ngày tháng năm, đổi ra ngày trong năm.

(d) Nhập vào ngày trong năm, đổi ra ngày thứ mấy của tháng nào.

IV. 4. Bài tập phần mảng

Bài 1. Mảng

Viết chương trình cho phép tạo giá trị ngẫu nhiên cho một mảng hai chiều $a(mxn)$ có giá trị từ $10 \rightarrow 100$. Sau đó thực hiện các công việc sau:

- + Đếm trong mảng có bao nhiêu giá trị chẵn.
- + Đếm trong mảng có bao nhiêu giá trị lẻ.
- + Tính tổng các giá trị trong mảng.
- + Sắp xếp mảng giảm dần theo chiều từ trái qua phải và trừ trên xuống.

Bài 2. Xử lý mảng

Viết chương trình nhập vào mảng nguyên dương $A[N]$, sau đó hãy sắp xếp các phần tử có giá trị lẻ ở đầu mảng và theo thứ tự tăng dần, các phần tử chẵn ở cuối mảng và theo thứ tự giảm dần.

Bài 3. Nhập số phần tử và các phần tử nguyên dương của mảng a.

- (a) In các số nguyên tố có trong mảng a.
- (b) Sắp xếp các số chẵn trong mảng theo thứ tự tăng dần.

Bài 4. Xử lý mảng

Viết chương trình nhập vào một mảng a, có n phần tử. Ta định nghĩa một mảng con tăng dần trong a là một dãy các phần tử liên tiếp gần nhau và có thứ tự tăng dần trong a.

- (a) Xác định số mảng con tăng có trong a.
- (b) In ra mảng con tăng dài nhất trong a.

Bài 5. Xử lý mảng

Hãy nhập mảng 1 chiều có n phần tử là những số nguyên dương. Hãy cho biết mảng đó chứa bao nhiêu số cùng thỏa mãn hai điều kiện: có 3 chữ số và các chữ số đều được sắp tăng. Ví dụ: 122, 457, 889,...

Bài 26. Viết chương trình nhập vào mảng một chiều có n số nguyên dương. Hãy cho biết số nào trong mảng có giá trị gần với trung bình cộng của toàn mảng.

Bài 6. Xử lý mảng

Viết chương trình tạo ngẫu nhiên hai ma trận vuông $a, b(n \times n)$ theo kiểu cấp phát động.

- (a) In hai ma trận a, b đã được tạo.
- (b) In ra ma trận tổng.
- (c) In ra ma trận tích.

Lưu ý: Phép nhân ma trận

Phép nhân hai ma trận chỉ thực hiện được khi số cột của ma trận bên trái bằng số dòng của ma trận bên phải. Nếu ma trận A có kích thước $m \times n$ và ma trận B có kích thước $n \times p$, thì ma trận tích AB có kích thước $m \times p$ xác định bởi:

$$(AB)_{i,j} = a_{i,1}b_{1,j} + a_{i,2}b_{2,j} + \dots + a_{i,n}b_{n,j}$$

với mọi cặp (i,j) .

Chẳng hạn:

$$\begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix} \times \begin{bmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} (1 \times 3 + 0 \times 2 + 2 \times 1) & (1 \times 1 + 0 \times 1 + 2 \times 0) \\ (-1 \times 3 + 3 \times 2 + 1 \times 1) & (-1 \times 1 + 3 \times 1 + 1 \times 0) \end{bmatrix}$$
$$= \begin{bmatrix} 5 & 1 \\ 4 & 2 \end{bmatrix}$$

Phép nhân ma trận có các tính chất sau:

$$(AB)C = A(BC) \text{ với mọi ma trận cấp } k \times m \text{ } A, \text{ ma trận } m \times n \text{ } B \text{ và ma trận } n \times p \text{ } C \text{ ("kết hợp").}$$

$$(A + B)C = AC + BC \text{ với mọi ma trận cấp } m \times n \text{ các ma trận } A \text{ và } B \text{ và ma trận cấp } n \times k \text{ } C \text{ ("phân phối bên phải").}$$

$$C(A + B) = CA + CB \text{ ("phân phối bên trái").}$$

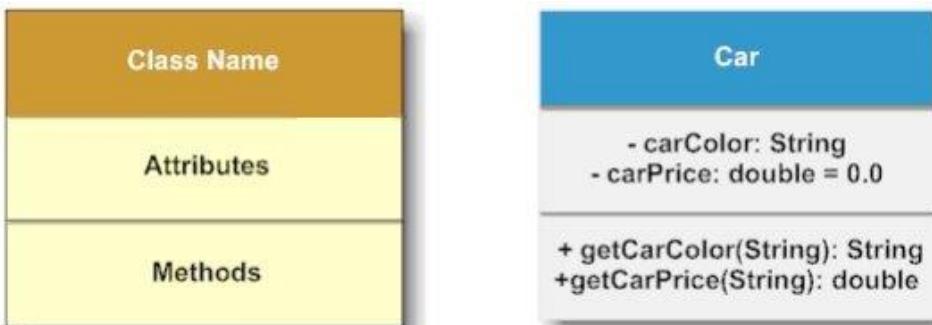
Bài 7. Viết chương trình nhập vào ma trận vuông $A[N][N]$.

- (a) Hãy đếm số phần tử có giá trị lớn nhất (Max) và nhỏ nhất (Min) trong ma trận A và đồng thời hãy xuất ra màn hình các vị trí (i,j) với $A(i,j) = \text{Max}$, hoặc $A(i,j) = \text{Min}$.
- (b) Trong ma trận A hãy thay thế giá trị của các phần tử có giá trị bằng Max hoặc bằng Min bằng trung bình cộng của Max và Min

Bài 8. Hãy nhập số nguyên dương cho một mảng hai chiều m dòng và n cột. Cho biết mảng đó chứa bao nhiêu số có: ba chữ số, hai chữ số và một chữ số.

V. Bài tập lập trình hướng đối tượng với C#

Định nghĩa 1 lớp trong UML2



Cách truy xuất (visibility) gồm public, private và protected

public	+
private	-
protected	#
package	~

Bài 1. Viết chương trình tính diện tích, chu vi hình chữ nhật.

- Hãy viết lớp **HinhChuNhat** gồm có:
 - Attributes : chiều dài, chiều rộng.
 - Phương thức thiết lập (set), và lấy (get) thông tin chiều dài, chiều rộng.
 - Phương thức tính diện tích, chu vi.
 - Phương thức `toString` gồm các thông tin dài, rộng, diện tích, chu vi.
- Xây dựng lớp chứa hàm `main` cho phần kiểm nghiệm. Dài, rộng có thể nhập từ bàn phím.

Bài 2. Viết chương trình OOP quản lý sinh viên đơn giản: Nhập, xuất thông tin, tính điểm TB.

- Viết lớp Sinh viên như sau:

Attributes (private):

- Mã sinh viên là số nguyên.
- Họ tên: chuỗi ký tự.
- Điểm LT, điểm TH : float

Constructor:

- Constructor mặc định (để khởi tạo đối tượng với các thông tin kiểu số là 0, kiểu chuỗi là chuỗi rỗng).
- Constructor thứ hai nhận đầy đủ thông tin để khởi tạo giá trị cho tất cả các biến instance.

Methods:

- Các getter và setter cho mỗi thuộc tính.
- Tính điểm trung bình.
- Phương thức `toString` để diễn tả đối tượng ở dạng chuỗi.

- Xây dựng class chứa hàm main: tạo 3 đối tượng sinh viên sv1, sv2, sv3, trong đó:
 - sv1 chứa thông tin của chính mình (tạo bằng constructor đủ thông số, thông tin biết rồi khỏi nhập từ bàn phím).
 - sv2 là thông tin người bạn thân nhất của em (tạo bằng constructor đủ thông số, thông tin biết rồi khỏi nhập từ bàn phím).
 - sv3 tạo bằng constructor mặc định. Nhập các thông tin cho sv3 từ bàn phím rồi sau đó dùng các setter để gán vào cho các thuộc tính tương ứng.
 - In bảng danh sách sinh viên gồm 4 cột là MSSV, họ tên, điểm LT”, điểm TH, điểm TB (bảng có 3 dòng cho 3 sinh viên).

Bài 3. Tạo một project có 2 class như sau:

BankAccount:

- Có các field:
 - private int accNumber;
 - private String firstName;
 - private String lastName;
 - private int balance;
- Có một constructor với 4 đối số (int accNo, String fName, String lName, int bal) để khởi tạo giá trị cho các field tương ứng khi tạo đối tượng
- Có các phương thức:
 - public boolean withdraw(int amount): cho phép rút số tiền amount từ tài khoản.
 - Số tiền cần rút phải hợp lệ ($0 < amount < balance$)
 - Hàm trả về true nếu thành công, ngược lại trả về false
 - public boolean deposit(int amount): cho phép gửi tiền vào tài khoản
 - Số tiền gửi vào phải hợp lệ ($0 < amount$)
 - Hàm trả về true nếu thành công, ngược lại trả về false
 - public String toString()


```

{
    return "Account Num: " + accNumber + "\nName: " + firstName + " " + lastName + "\nBalance: " + balance;
}
```

Class BankTest: Chứa hàm main thực hiện các công việc sau:

- Khai báo một biến kiểu BankAccount ba
- Tạo menu với các chức năng như sau:
 1. Create new Account
 - Khai báo biến: accNum, fName, lName, Bal
 - Nhập giá trị cho các biến từ bàn phím
 - Gọi new để tạo một account mới (nhớ truyền giá trị)

2. Withdraw

- Nhập số tiền muốn rút
- Gọi phương thức Withdraw sau đó thông báo kết quả dựa vào giá trị trả về

3. Deposit

- Các bước tương tự Withdraw

4. Print account Information

5. Exit

Khi người sử dụng chọn một công việc thì thực hiện chức năng tương ứng

Bài 4. Tạo một project có 3 class như sau:

CDinfo:

- Có các field:
 - private int CDno;
 - private String title;
 - private String author;
 - private int price;
- Có một constructor với 4 đối số (int cdNum, String cdTitle, String cdAuthor, int price) để khởi tạo giá trị cho các field tương ứng khi tạo đối tượng
- Có phương thức:
- **public String toString():** trả về một chuỗi chứa tất cả thông tin của CD

CDList: (dùng indexer)

- Khai báo và khởi tạo một mảng **list** kiểu CDinfo
- Khai báo int count=0 (lưu số CD đã có trong mảng)
- Viết constructor **CDList(int n):** dùng khởi tạo mảng list với n phần tử ($n > 0$)
- Viết phương thức : **int AddCD(CDinfo newCD):** thêm newCD vào mảng list tại vị trí **count**. (list[count]=newCD) Sau khi thêm tăng count lên .
Lưu ý: Chỉ thêm nếu mảng list còn chỗ (count<list.length), hàm trả về 0 nếu thêm thành công, -1 nếu không còn chỗ để thêm
- Viết phương thức **CD searchCD(String CDtitle):** tìm một CD trong **list** theo tựa CD (title). Phương thức trả về CD tìm thấy trong mảng **list** hoặc null nếu không tìm thấy.
- Viết phương thức **printAllCD()** in thông tin tất cả CD có trong mảng **list**

Class CDTTest: Chứa hàm main thực hiện các công việc sau:

- Khai báo và khởi tạo **CDCollection** kiểu CDList n phần tử (n nhập từ bàn phím)
- Tạo menu với các chức năng như sau:
 1. Create new CD
 - Khai báo các biến để nhập: thông tin của một CD
 - Khai báo và khởi tạo một CD mới
 - Thêm cd vừa tạo vào CDCollection (**CDCollection.AddCD()**)
 - Thông báo kết quả của việc thêm CD
 2. Search CD by Title

- Nhập tên CD muốn tìm (nhập vào một biến CDTITLE kiểu String)
 - Gọi phương thức searchCD() của **CDCollection** để tìm, nếu kết quả trả về khác null thì in thông tin của CD vừa tìm thấy, ngược lại thông báo “Không tìm thấy”
3. Display all CD
 - In tất cả các CD lên màn hình
 4. Exit

Bài 5. (Inheritance and Interfaces) Tạo ứng dụng để tính toán thu nhập cho các cầu thủ bóng chày. Ứng dụng gồm các lớp sau:

1. Player.java
2. Tax.java
3. PlayerIncome.java
4. GradeBonus.java
5. TournamentIncome.java
6. PlayerTest.java

Player:

- Abstract class
- Thuộc tính: Tên cầu thủ
- Constructor: gán giá trị cho Tên cầu thủ
- Abstract Method: **displayDetail:** Hiển thị tên cầu thủ

Tax:

- Interface
- Abstract Method: **calculateTax()** tính thuế thu nhập của cầu thủ

PlayerIncome:

- Thừa kế lớp **Player** và thực thi (implement) lớp **Tax**
- Chứa Constant TAX_PERCENT
- Thuộc tính: Thu nhập của cầu thủ (income)
- Constructor: có hai tham số chứa tên và thu nhập của cầu thủ.
 - Gán giá trị cho thuộc tính thu nhập
 - Gọi Constructor của lớp Player để gán giá trị cho thuộc tính tên cầu thủ
- Viết code(implement) cho phương thức **calculateTax** { return TAX_PERCENT* income; }
- Viết code(implement) cho phương thức **displayDetail:** hiện thị tên ,thu nhập và thuế thu nhập của cầu thủ xuất (name + “, income=” + income + “tax=” + calculateTax());

GradeBonus:

- Interface
- Abstract method: **calculateGradeBonus:** tính tiền thưởng cho cầu thủ

TournamentIncome:

- Thừa kế lớp **Player** và thực thi Interface **GradeBonus**
- Thuộc tính: grade (hạng cầu thủ), rate(mức đáng giá của cầu thủ)
- Constant: GRADE_BONUS_PERCENT: tỉ lệ thưởng cho cầu thủ (giá trị=0.3)
- Constructor: có ba tham số: chứa tên, hạng và đánh giá của cầu thủ:

- Gán giá trị cho thuộc tính hạng, đánh giá
- Gọi Constructor của lớp Player để gán giá trị cho thuộc tính tên cầu thủ
- Viết code(implement) cho phương thức
 - **calculateGradeBonus:**
if (grade>=2 && rate>=2) return GREAT_BONUS_PERCENT * grade;
return 0; // code tự định
 - **displayDetail:**
xuất (name + “,Bonus=” + calculateGreatBonus());

PlayerTest:

```
{ main()
{ PlayerIncom p1= new PlayerIncome( "Albert", 10000);
TournamentIncome p2= new TournamentIncome ("Alberto",3,2);
p1.displayDetail();
p2.displayDetail();
}
```

Bài 6. Viết chương trình xây dựng đối tượng Person gồm có các thuộc tính sau:

- + ID int,
- + FullName String,
- + Add String.

Sau đó xây dựng class cho phép người dùng nhập vào 10 Person. Xuất danh sách trên ra màn hình theo thứ tự tăng dần của ID.

Bài 7. Viết chương trình xây dựng đối tượng Shape gồm có các thuộc tính sau:

- + Mảng điểm chứa danh sách các Điểm.
- + Tên hình, String

Xây dựng phương thức tính chu vi của hình Shape trên, đồng thời nhập vào 4 hình Shape xuất ra danh sách các hình trên theo thứ tự tăng dần của thuộc tính Chuvi.

Bài 8.(Exception) Viết chương trình xây dựng đối tượng Sinh viên gồm có các thuộc tính sau:

- Mã sinh viên là số nguyên (>0),
- Họ tên: chuỗi ký tự (không để trống),
- Địa chỉ: chuỗi ký tự,
- Số điện thoại: là chuỗi bao gồm 7 chữ số,
- Các thuộc tính khai báo private, định nghĩa properties cho từng thuộc tính, override phương thức equal và toString của lớp Object.
- Viết các constructor để khởi tạo đối tượng.
- Xây dựng lớp cho phép nhập vào N sinh viên. Xuất danh sách trên ra màn hình theo thứ tự tăng dần của mã sinh viên.

Bài 9. Ngân hàng ABC muốn lưu trữ thông tin của mỗi tài khoản như sau:

Mỗi tài khoản chứa các thông tin:

- Số tài khoản (Kiểu long),
- Tên tài khoản (kiểu chuỗi),
- Số tiền trong tài khoản (kiểu double)

(a). Thiết kế lớp Account để lưu trữ các thông tin, lớp bao gồm các phương thức sau:

- Constructor: Có 2 constructor (mặc định và đầy đủ tham số)
- Các phương thức get, set cho từng thuộc tính
- Phương thức toString để trả về chuỗi chứa toàn bộ thông tin tài khoản, yêu cầu định dạng tiền tệ.

(b). Thêm các thông tin sau vào lớp Account

- Hằng số lãi suất có giá trị khởi tạo 0.035
- Constructor có 2 đối số: số tài khoản, tên tài khoản. Constructor này sẽ khởi tạo số tiền mặc định là 50.
- Phương thức nạp tiền vào tài khoản: Lấy số tiền hiện tại trong tài khoản + số tiền nạp vào
- Phương thức rút tiền: Lấy số tiền hiện tại trong tài khoản - (số tiền muốn rút + phí rút tiền)
- Phương thức đáo hạn: Mỗi lần đến kỳ đáo hạn thì số tiền trong tài khoản = số tiền trong tài khoản + số tiền trong tài khoản * LAISUAT
- Phương thức chuyển khoản từ tài khoản này sang tài khoản khác

Chú ý: Mỗi thao tác phải kiểm tra số tiền nạp, rút, chuyển có hợp lệ hay không? (VD: tiền nhập vào <0, tiền rút nhiều hơn tiền trong tài khoản thì thông báo không hợp lệ và yêu cầu nhập lại)

Bài 10. Một phân số gồm có 2 thuộc tính là tử số (numerator) và mẫu số (denominator).

Hãy thiết kế class Phân số (Rational) để thực hiện các chức năng sau:

- Tối giản phân số (reduce)
- Nghịch đảo phân số (reciprocal)
- Cộng 2 phân số (add)
- Trừ 2 phân số (subtract)
- Nhân 2 phân số (multiply)
- Chia 2 phân số (divide)
- So sánh 2 phân số, sử dụng sai số 0.0001(tolerance) để so sánh.

Bài 11. Viết lớp HangThucPham mô tả một hàng hóa là hàng thực phẩm trong kho của một siêu thị, có các thuộc tính: mã hàng (*không cho phép sửa, không được để rỗng*), tên hàng (*không được để rỗng*), đơn giá (>0), ngày sản xuất và ngày hết hạn (*ngày không được để rỗng, ngày hết hạn phải sau ngày sản xuất*).

Ràng buộc chặt chẽ các ràng buộc trên các trường dữ liệu. Nếu dữ liệu không hợp lệ thì gán giá trị mặc định cho phép tương ứng của trường đó.

- Tạo 1 constructor có đầy đủ tham số, 1 constructor có tham số là mã hàng.
- Viết các phương thức setters/getters.
- Viết phương thức kiểm tra một hàng thực phẩm đã hết hạn chưa?

- Phương thức `toString`, trả về chuỗi chứa thông tin của hàng thực phẩm. Trong đó: Định dạng đơn giá có phân cách hàng nghìn. Định dạng kiểu ngày là `dd/MM/yyyy`.
Viết lớp cho phần kiểm nghiệm.

Bài 12.

- a. Viết chương trình xây dựng đối tượng Sinh viên gồm có các thuộc tính sau:
 - o Mã sinh viên là số nguyên,
 - o Họ tên: chuỗi ký tự,
 - o Địa chỉ: chuỗi ký tự,
 - o Số điện thoại: là số bao gồm 7 chữ số
- Các thuộc tính khai báo private, định nghĩa các phương thức get/set cho từng thuộc tính.
- Viết các constructor để khởi tạo đối tượng (constructor mặc định, constructor có tham số).
- Override phương thức `toString` của lớp Object.
- b. Xây dựng lớp cho phép nhập vào N sinh viên (*dùng mảng*). Xuất danh sách trên ra màn hình theo thứ tự tăng dần của mã sinh viên.

Bài 13. Công ty du lịch V quản lý thông tin là các chuyến xe. Thông tin của 2 loại chuyến xe:

- Chuyến xe nội thành: Mã số chuyến, Họ tên tài xế, số xe, số tuyến, số km đi được, doanh thu.
- Chuyến xe ngoại thành: Mã số chuyến, Họ tên tài xế, số xe, nơi đến, số ngày đi được, doanh thu.

Thực hiện các yêu cầu sau:

- Xây dựng các lớp với chức năng thừa kế.
- Viết chương trình quản lý các chuyến xe theo dạng cây thừa kế với các phương thức sau:
 - Nhập, xuất danh sách các chuyến xe (danh sách có thể dùng cấu trúc mảng).
 - Tính tổng doanh thu cho từng loại xe.

Bài 14. Thư viện X quản lý danh sách các loại sách . Thông tin về các loại sách:

- Sách giáo khoa: Mã_sách, ngày nhập (ngày, tháng, năm), đơn giá, số lượng, nhà xuất bản, tình trạng (mới, cũ).

Nếu tình trạng sách là mới thì: $\text{thành tiền} = \text{số lượng} * \text{đơn giá}$.

Nếu tình trạng sách là cũ thì: $\text{thành tiền} = \text{số lượng} * \text{đơn giá} * 50\%$

- Sách tham khảo: Mã sách, ngày nhập (ngày, tháng, năm), đơn giá, số lượng, nhà xuất bản, thuế.
 $\text{Thành tiền} = \text{số lượng} * \text{đơn giá} + \text{thuế}$

Thực hiện các yêu cầu sau:

- Xây dựng các lớp với chức năng thừa kế.
- Nhập xuất danh sách các loại sách.
- Tính tổng thành tiền cho từng loại.
- Tính trung bình cộng đơn giá của các sách tham khảo.

Xuất ra các sách giáo khoa của nhà xuất bản X.

Bài 15. Xây dựng chương trình quản lý danh sách các giao dịch. Hệ thống quản lý 2 loại giao dịch:

- Giao dịch vàng: Mã giao dịch, ngày giao dịch (ngày, tháng, năm), đơn giá, số lượng, loại vàng. Thành tiền được tính như sau:
 thành tiền = số lượng * đơn giá.
- Giao dịch tiền tệ: Mã giao dịch, ngày giao dịch (ngày, tháng, năm), Đơn giá, số lượng, tỉ giá, loại tiền tệ có 3 loại: tiền Việt Nam, tiền USD, tiền Euro. Thành tiền được tính như sau:
 - Nếu là tiền USD hoặc Euro thì: thành tiền = số lượng * đơn giá * tỉ giá
 - Nếu là tiền VN thì: thành tiền = số lượng * đơn giá

Thực hiện các yêu cầu sau:

- Xây dựng các lớp với chức năng thừa kế.
- Nhập xuất danh sách các giao dịch.
- Tính tổng số lượng cho từng loại.
- Tính trung bình thành tiền của giao dịch tiền tệ.
- Xuất ra các giao dịch có đơn giá > 1 tỷ.

Bài 16. Xây dựng chương trình quản lý danh sách các giao dịch nhà đất. Thông tin bao gồm:

- Giao dịch đất: Mã giao dịch, ngày giao dịch (ngày, tháng, năm), đơn giá, loại đất (loại A, B, C), diện tích.
 - Nếu là loại B, C thì: thành tiền = diện tích * đơn giá.
 - Nếu là loại A thì: thành tiền = diện tích * đơn giá * 1.5
- Giao dịch nhà: Mã giao dịch, ngày giao dịch (ngày, tháng, năm), đơn giá, loại nhà (cao cấp, thường), địa chỉ, diện tích.
 - Nếu là loại nhà cao cấp thì: thành tiền = diện tích * đơn giá.
 - Nếu là loại thường thì: thành tiền = diện tích * đơn giá * 90%

Thực hiện các yêu cầu sau:

- Xây dựng các lớp với chức năng thừa kế.
- Nhập xuất danh sách các giao dịch.
- Tính tổng số lượng cho từng loại.
- Tính trung bình thành tiền của giao dịch đất.
- Xuất ra các giao dịch của tháng 9 năm 2013.

Bài 17. Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng. Thông tin bao gồm các loại khách hàng :

- Khách hàng Việt Nam: mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), đối tượng khách hàng (sinh hoạt, kinh doanh, sản xuất): số lượng (số KW tiêu thụ), đơn giá, định mức. Thành tiền được tính như sau:
 - Nếu số lượng \leq định mức thì: thành tiền = số lượng * đơn giá.
 - Ngược lại thì: thành tiền = số lượng * đơn giá * định mức + số lượng KW vượt định mức * Đơn giá * 2.5.

- Khách hàng nước ngoài: mã khách hàng, họ tên, ngày ra hóa đơn (ngày, tháng, năm), quốc tịch, số lượng, đơn giá. Thành tiền được tính = số lượng * đơn giá.

Thực hiện các yêu cầu sau:

- Xây dựng các lớp với chức năng thừa kế.
- Nhập xuất danh sách các hóa đơn khách hàng.
- Tính tổng số lượng cho từng loại khách hàng.
- Tính trung bình thành tiền của khách hàng người nước ngoài.
- Xuất ra các hóa đơn trong tháng 09 năm 2013 (của cả 2 loại khách hàng).

BÀI TẬP TUẦN 02 MÔN LẬP TRÌNH WINDOWS (C#)

Chương II : Events and Delegate

Mục tiêu:

- Hiểu rõ và áp dụng vào chương trình các delegates và events

Yêu cầu:

- Tất cả các bài tập lưu trong thư mục: T:\MaSV_HoTen\Tuan01\
- Tạo Solution Tuan01 trong thư mục T:\MaSV_HoTen\Tuan01\ trong Visual Studio 2010/2012/2013. Mỗi bài tập là một Project riêng biệt.
- Cuối mỗi buổi thực hành, SV phải nén (.rar hoặc .zip) thư mục làm bài và nộp lại bài tập đã thực hiện trong buổi đó.
- Bài tập bắt buộc của Tuần XX, GV sẽ công bố cuối giờ. Bài tập bắt buộc sẽ gửi vào email giáo viên quy định ví dụ thangvnguyen2001@gmail.com với Subject và tập tin nén kèm theo: **XY_ZK_LTWINDOWS_TUANXX_HOTENSINHVIEN** (trong đó XY là lớp ZK là nhóm XX sẽ là tuần)

Bài 1. Cho đoạn chương trình sau:

```
using System;  
public class Application {
```

```
    public delegate double NumericFunction(double d);  
    static double factor = 4.0;
```

```
    public static NumericFunction MakeMultiplier(double factor){  
        return delegate(double input){return input * factor;};  
    }
```

```
    public static void Main(){  
        NumericFunction f = MakeMultiplier(3.0);  
        double input = 5.0;
```

```
        Console.WriteLine("factor = {0}", factor);  
        Console.WriteLine("input = {0}", input);  
        Console.WriteLine("f is a generated function which multiplies its input with factor");  
        Console.WriteLine("f(input) = input * factor = {0}", f(input));  
    }  
}
```

Khi chạy chương trình, kết quả:

```
factor = 4  
input = 5  
f is a generated function which multiplies its input with factor  
f(input) = input * factor = 15
```

Bài 2. Tạo delegate sử dụng Delegate và phương thức anonymous

```
public delegate bool IntPredicate(int x);  
public delegate void IntAction(int x);
```

Bài 3. Tạo 1 delegate để thực hiện toán tử 2 ngôi (+, -, *, /, %).

- int Add(int a, int b) {}
- int Subtract(int a, int b) {}
- int Devide(int a, int b) {}
- int Mod(int a, int b) {}
- int Multiply(int a, int b) {}

Bài 4. Thực hiện kỹ thuật multicasting delegate viết hàm cho phép gán nhiều phương thức vào cùng một delegate (các phương thức này có cùng kiểu trả về và tham số với delegate).

- public void DisplayMsg(string msg){...}
- public void ShowHello(string name){...}

Viết hàm gọi delegate này, tất cả các phương thức đã được gán vào sẽ thi hành cùng lúc.

Hướng dẫn:

- Một multicast delegate chỉ chấp nhận phương thức có kiểu trả về là void.
- Để thực hiện Multicasting, tạo các thể hiện (instance) của một multicast delegate, gắn với các phương thức tương ứng. Sau đó dùng toán tử “+” để gom các delegate này vào thành 1 delegate duy nhất.

```

using System;

namespace DelegateDemo
{
    public delegate void MyEventHandler(string msg);

    public class Demo
    {
        public static void Main()
        {
            Demo d = new Demo();
            MyEventHandler handler1 = new MyEventHandler(d.DisplayMsg);
            MyEventHandler handler2 = new MyEventHandler(d.DisplayMsg);
            MyEventHandler handler = handler1 + handler2;
            handler("Test");
            Console.ReadLine();
        }

        public void DisplayMsg(string msg)
        {
            Console.WriteLine(msg);
        }

        public void ShowHello(string name)
        {
            Console.WriteLine("Hello " + name);
        }
    }
}

```

Hoặc dùng:

```

Demo d = new Demo();
MyEventHandler handler = new MyEventHandler(d.DisplayMsg);
handler += d.ShowHello;
handler("Test");

```

Bài 5. Xây dựng hàm Sort tổng quát cho cho mảng đối tượng có kiểu bất kỳ.

Xét cả trường hợp nếu đối tượng là kiểu số như int, long, float và trường hợp đối tượng phức tạp khác (user defined).

Hướng dẫn:

- Cho phép đối tượng tự quy định thứ tự của chúng
- Sử dụng delegate để truyền phương thức so sánh này vào hàm Sort

void Sort(object[] list, CompareObj cmp).

Trong đó Delegate này sẽ tham chiếu tới hàm Compare của lớp MyClass. Chính lớp MyClass sẽ quy định thứ tự của các đối tượng.

- Mô tả delegate CompareObj cho hàm Sort:
public delegate bool CompareObj(object o1, object o2); Delegate này sẽ trả về true: nếu o1 “trước” o2 false: ngược lại, trong đó o1, o2 là 2 đối tượng cần so sánh.
- Định nghĩa hàm Sort tổng quát cho các lớp
public static void Sort(object[] objs, CompareObj cmp)
{

```

for(int i=0; i < objs.Length-1; i++)
    for(int j=objs.Length-1; j>i; j--)
        if( cmp( objs[j], objs[j-1] ) )
        {
            Swap( objs[j], objs[j-1] );
        }
    }
}

```

Delegate sẽ trỏ tới hàm Compare riêng của lớp tương ứng

- Các lớp hỗ trợ cần Sort phải:
 - Cung cấp hàm Compare riêng
 - Signature phải thoả delegate CompareObj

class Person

{

```

    private string name;
    private int weight;
    private int yearOfBirth;
    public static bool CompareName(object p1, object p2)
    {
        if(string.CompareName(((Person)p1).name, ((Person)p2).name)<0)
            return true;
        return false;
    }
}

```

- Lớp thử nghiệm

public delegate bool CompareObj(object o1,object o2); ...

```

Person[ ] persons = new Person[4];
persons[0] = new Person("Thanh Hòa", 2, 2004);
persons[1] = new Person("Hồng Nguyên", 65, 2000);
persons[2] = new Person("Hoàng Ngân", 47, 1999);
persons[3] = new Person("Thảo Trần", 65, 1996);

```

CompareObj cmp = new CompareObj(Person.CompareName);

Tên_Lớp.Sort(persons, cmp);

Bài 6. Quá trình xử lý sự kiện trong .NET Framework được mô tả:

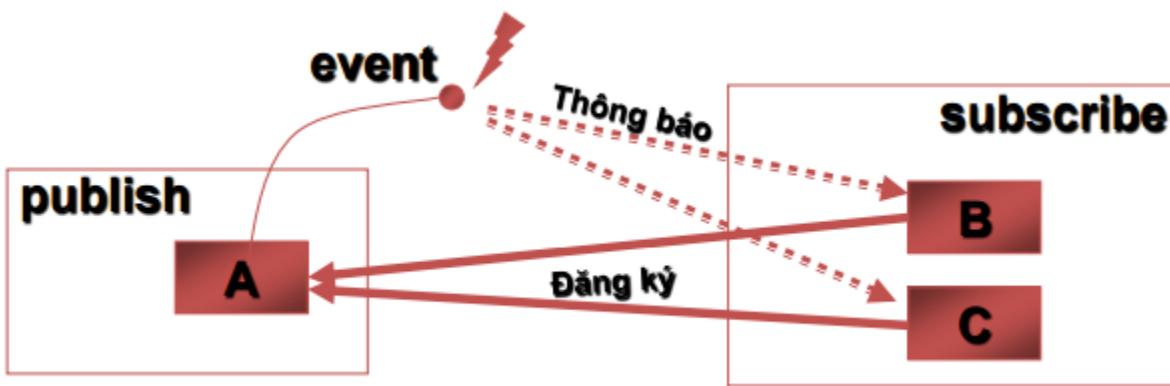
- Trả về giá trị void
- Tham số 1: nguồn phát sinh sự kiện, đây chính là đối tượng publisher
- Tham số 2: là đối tượng thuộc lớp dẫn xuất từ EventArgs

Khai báo delegate xử lý sự kiện

public delegate void HandlerName(object obj, EventArgs arg);

Khai báo event

```
public event HandlerName OnEventName;
```



Hình 1. Cơ chế thông điệp giữa các lớp hay các đối tượng (Event)

Viết một chương trình đơn giản minh họa quản lý tài khoản ATM: khi rút tiền hoặc chuyển tiền thì hệ thống sẽ gửi tự động tin nhắn đến chủ tài khoản.

Lưu ý Khi rút tiền hoặc chuyển tiền xong: phát sinh sự kiện “đã rút tiền” hoặc “đã chuyển tiền”

Bài 7. Xây dựng 1 lớp thực hiện yêu cầu: “cứ mỗi giây sẽ phát sinh 1 sự kiện”

Chương trình cho phép 2 lớp khác đăng ký xử lý sự kiện này, mỗi lớp có cách xử lý riêng:

- Lớp A: hiển thị thời gian theo “mô phỏng đồng hồ analog”
- Lớp B: hiển thị thời gian theo “mô phỏng đồng hồ digital”

Hướng dẫn:

- Tạo một lớp Clock gồm các thông tin:
 - Khai báo event: OnSecondChange
 - Phương thức Run: cứ 1s thì phát sinh sự kiện OnSecondChange
- Tạo 2 lớp: AnalogClock và DigitalClock nhận xử lý sự kiện OnSecondChange của lớp Clock

Khai báo delegate xử lý event

```
delegate void SecondChangeHandler(object clock, EventArgs info);
```

Khai báo event có hàm xử lý mô tả trên

event SecondChangeHandler OnSecondChange; Trong đó từ khóa event: thể hiện cơ chế publishing & subscribing. SecondChangeHandler là kiểu delegate và OnSecondChange là tên của event.

```
public class Clock
{
    public delegate void SecondChangeHandler(object clock, EventArgs info);
    public event SecondChangeHandler OnSecondChange;
    public void Run()
    {
        while (true)
    }
}
```

```

        Thread.Sleep(1000);
        if (OnSecondChange != null)
            OnSecondChange(this, new EventArgs());
    }
}

```

Lớp DigitalClock

Định nghĩa trình xử lý sự kiện của Clock

Đúng mô tả delegate hàm xử lý của lớp Clock

Thực hiện một số thao tác riêng của DigitalClock

Đăng ký xử lý với trình xử lý sự kiện trên khi có sự kiện OnSecondChange của Clock

Chức năng đăng ký với lớp Clock là có xử lý khi sự kiện OnSecondChange của Clock phát sinh

Ủy thác cho lớp Clock sẽ gọi trình xử lý định nghĩa bên trên của DigitalClock

```

public class DigitalClock
{
    //Đăng ký xử lý sự kiện. Trong đó đối tượng theClock sẽ phát sinh sự kiện.
    public void Subscribe(Clock theClock)
    {
        theClock.OnSecondChange += new Clock.SecondChangeHandler>Show);
    }
    public void Show(object obj, EventArgs args)
    {
        DateTime date = DateTime.Now;
        Console.WriteLine("Digital Clock: {0}:{1}:{2}", date.Hour, date.Minute, date.Second);
    }
}

```

Lớp AnalogClock

Định nghĩa trình xử lý sự kiện của Clock

Đúng mô tả delegate hàm xử lý của lớp Clock

Thực hiện một số thao tác riêng của AnalogClock

Đăng ký xử lý với trình xử lý sự kiện trên khi có sự kiện OnSecondChange của Clock

Chức năng đăng ký với lớp Clock là có xử lý khi sự kiện OnSecondChange của Clock phát sinh

Ủy thác cho lớp Clock sẽ gọi trình xử lý định nghĩa bên trên của AnalogClock

```

public class AnalogClock
{

```

```

public void Subscribe(Clock theClock)
{
    theClock.OnSecondChange += new Clock.SecondChangeHandler>Show);
}
public void Show(object obj, EventArgs args)
{
    DateTime date = DateTime.Now;
    Console.WriteLine("Analog Clock: {0}:{1}:{2}", date.Hour,date.Minute,date.Second);
}

```

Minh họa cơ chế event

```

public class Tester
{
    public static void Main()
    {
        Clock myClock = new Clock();
        AnalogClock c1 = new AnalogClock();
        DigitalClock c2 = new DigitalClock();
        c1.Subscribe(myClock);
        c2.Subscribe(myClock);
        myClock.Run();
    }
}

```

Hoặc hàm main xử lý?

```

public static void Main( )
{
    Clock myClock = new Clock();
    AnalogClock c1 = new AnalogClock();
    DigitalClock c2 = new DigitalClock();
    myClock.OnSecondChange += new Clock.SecondChangeHandler>c1.Show);
    myClock.OnSecondChange += new Clock.SecondChangeHandler>c2.Show);
    myClock.Run();
}

```

BÀI TẬP TUẦN 03, 04, 05 MÔN LẬP TRÌNH WINDOWS (C#)

Chương III : Lập trình giao diện Windows

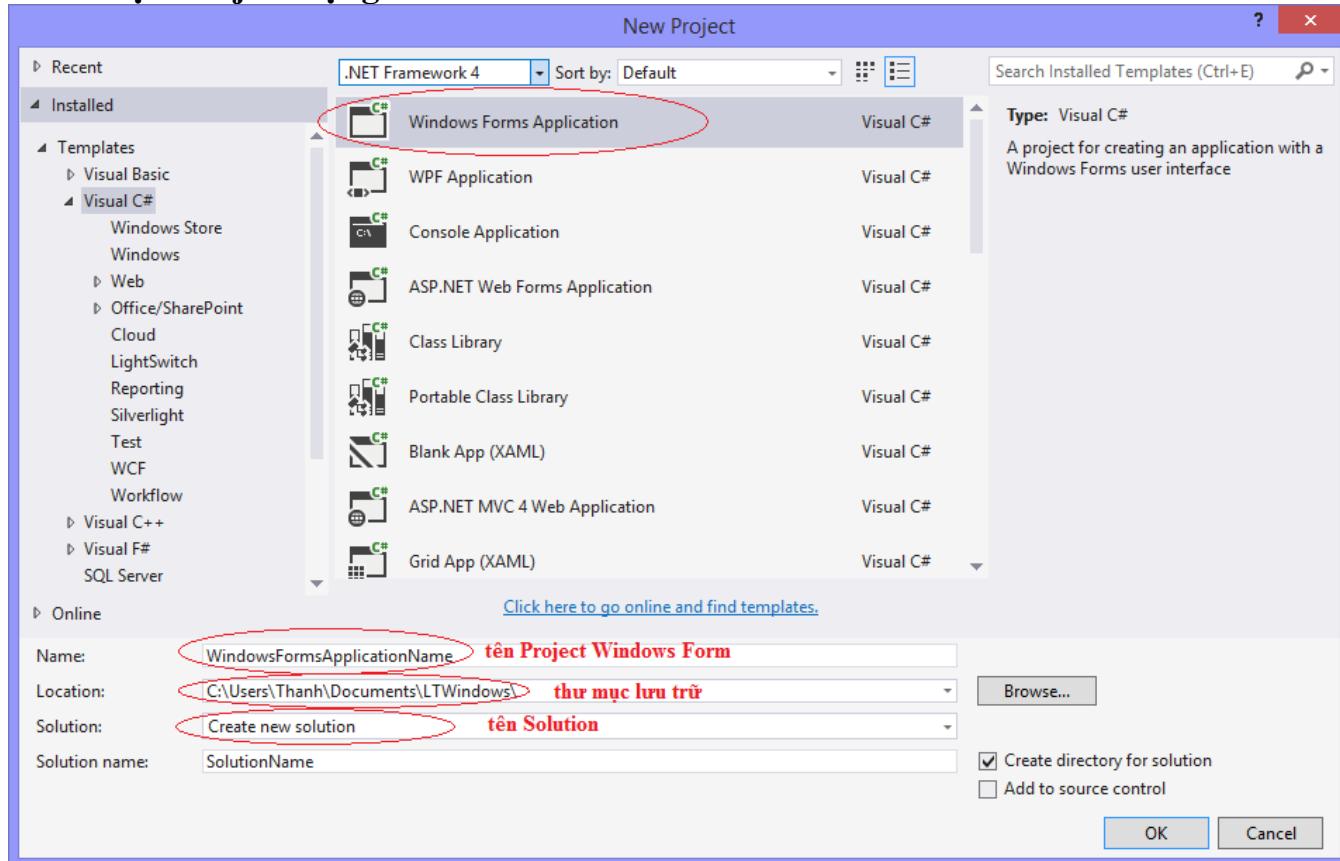
Mục tiêu:

- Hiểu và áp dụng được các *GUI controls* (*basic controls, advanced controls*) vào chương trình cụ thể.
- Viết được chương trình ứng dụng ở dạng *Windows Form Application*

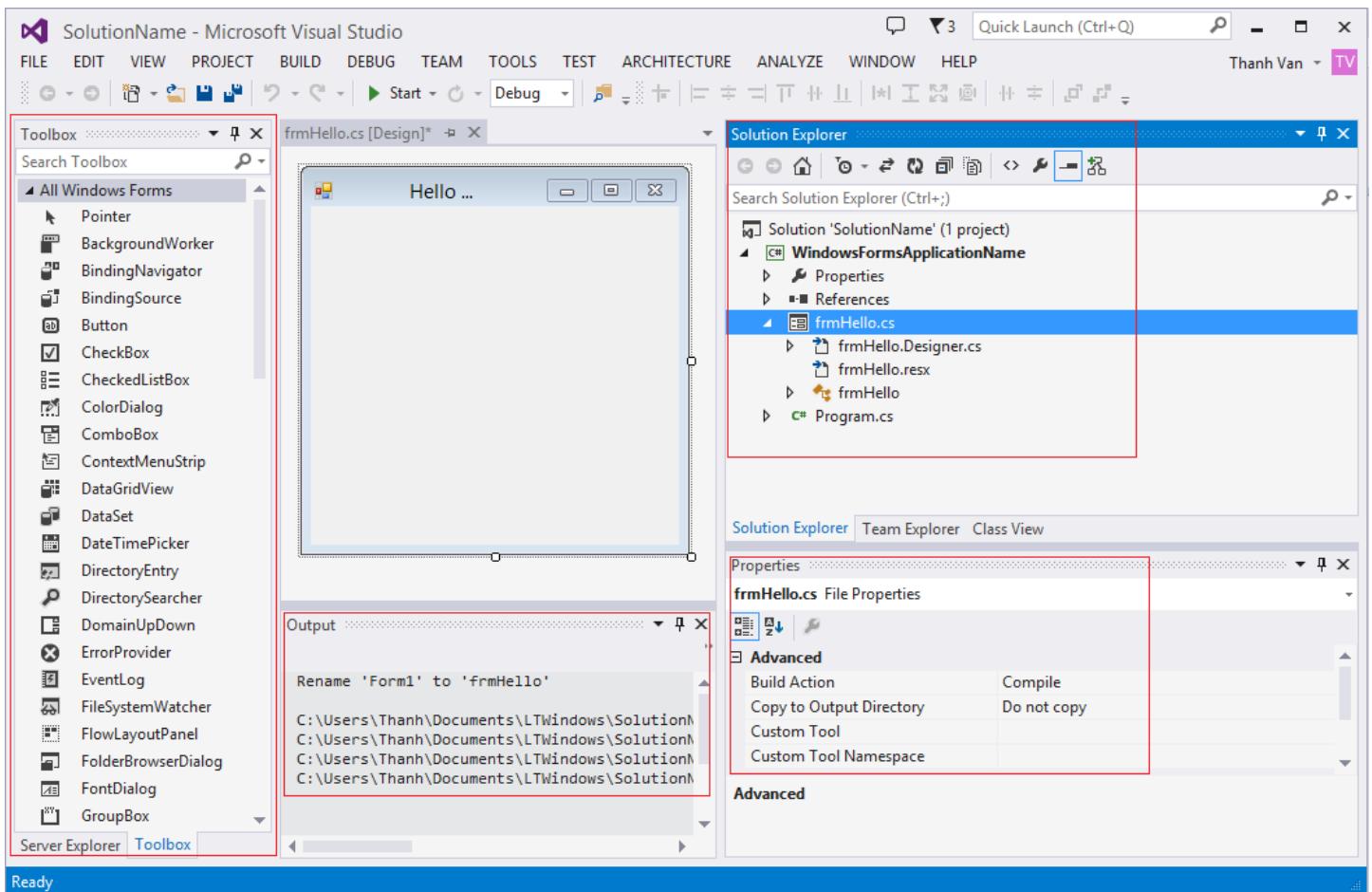
Yêu cầu:

- Tất cả các bài tập lưu trong thư mục: *T:\MaSV_HoTen\Tuan01*
- Tạo Solution *Tuan01* trong thư mục *T:\MaSV_HoTen\Tuan01* trong Visual Studio 2010/2012/2013. Mỗi bài tập là một Project riêng biệt.
- Cuối mỗi buổi thực hành, SV phải nén (.rar hoặc .zip) thư mục làm bài và nộp lại bài tập đã thực hiện trong buổi đó.
- Bài tập bắt buộc của Tuần XX, GV sẽ công bố cuối giờ. Bài tập bắt buộc sẽ gửi vào email giáo viên quy định ví dụ thangvnguyen2001@gmail.com với Subject và tập tin nén kèm theo: **XY_ZK_LTWINDOWS_TUANXX_HOTENSINHVIEN** (trong đó XY là lớp ZK là nhóm XX sẽ là tuần)

Bài 1. Tạo Project dạng Windows Forms



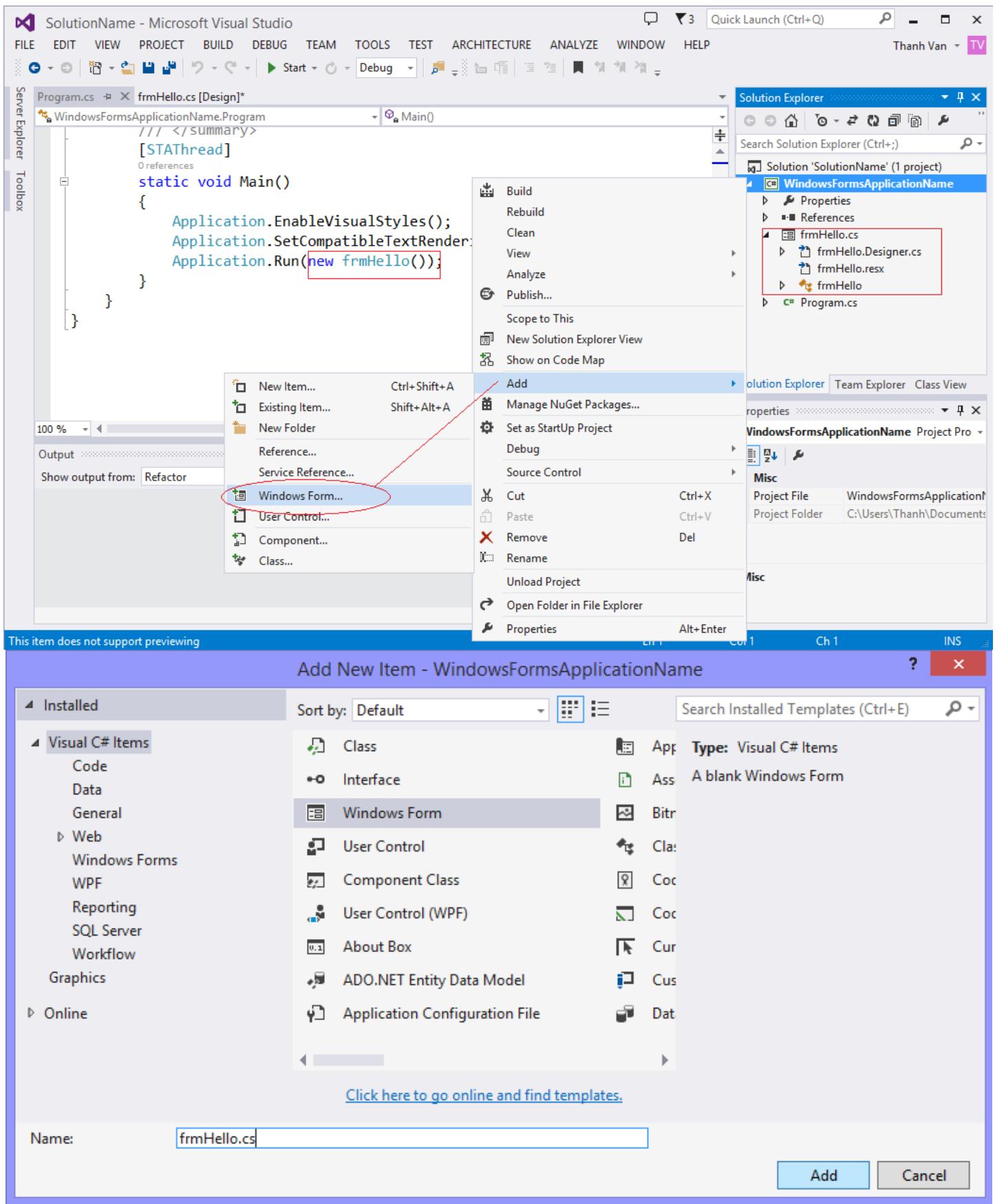
Hình 5. Tạo Project Windows Forms Application mới trong Visual Studio



Hình 6. Giao diện của Windows Forms Application

Một số lưu ý khi tạo Project Windows Forms:

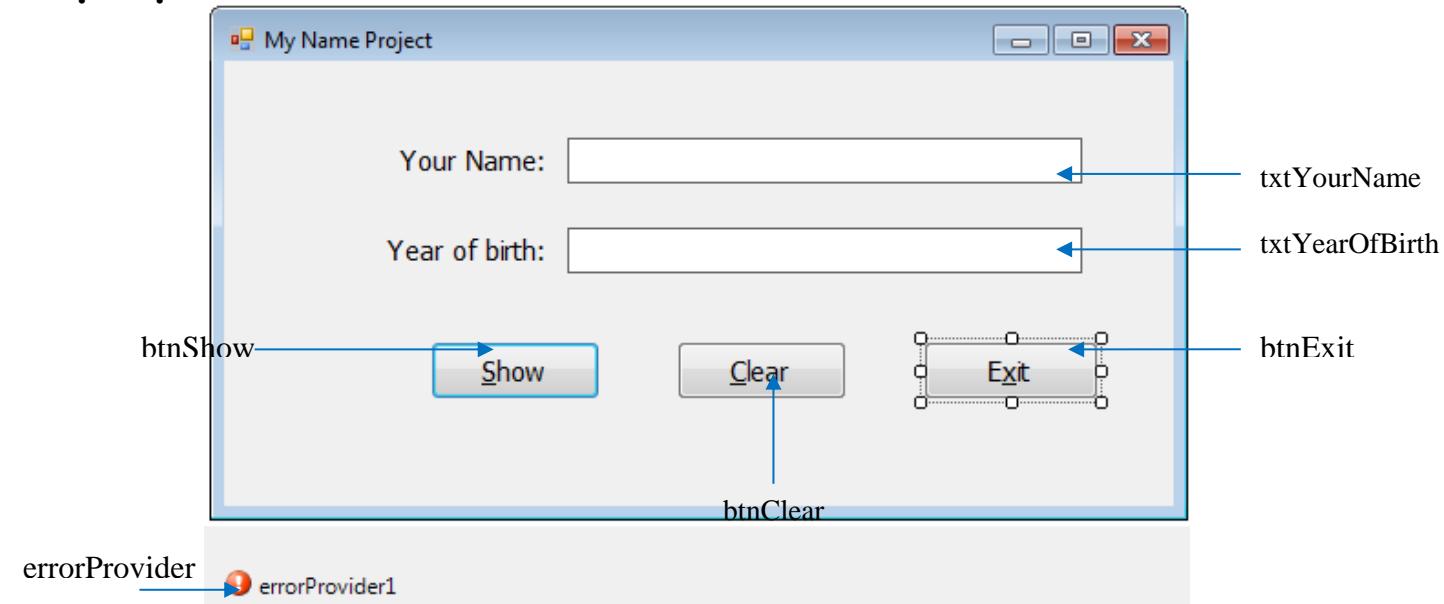
- Nhấn chuột phải trên tên Project (trong Solution Explorer) → Add → Windows Form.
- Đặt tên cho Form mới (Không nên để tên mặc định Form1, Form2, ...)



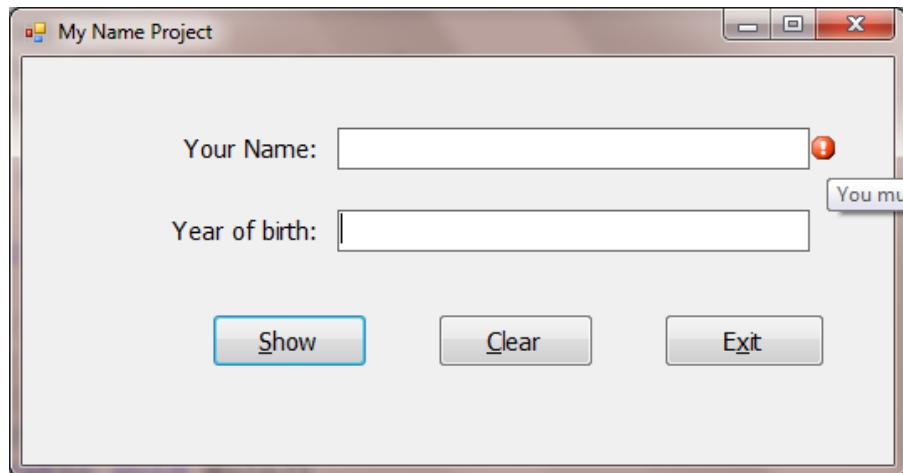
Hình 7. Tạo Form mới trong Project Windows Forms

II. Bài tập sử dụng các basic controls

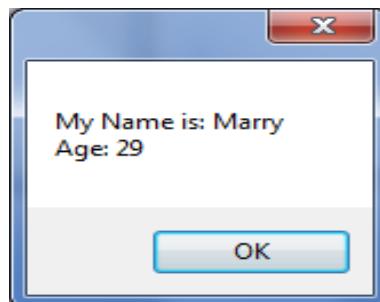
Bài 1. Thực hiện Form sau:



- Khi Form load thực hiện nhập tên, năm sinh vào TextBox YourName và Year of Birth tương ứng. Nếu YourName không nhập dữ liệu, năm sinh không phải là số thì phải thông báo lỗi (dùng ErrorProvider)



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



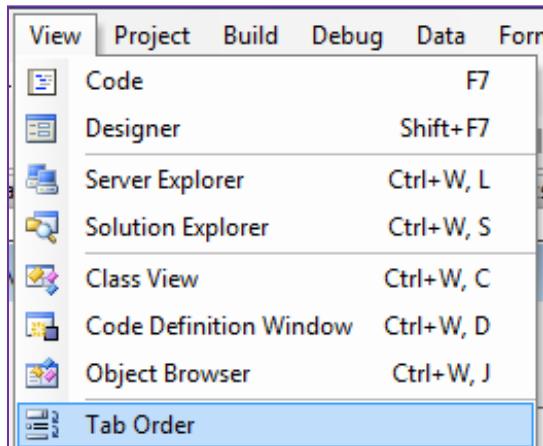
- Khi người dùng nhấn nút Clear sẽ xóa thông tin đã nhập trên các TextBox, đặt con trỏ văn bản vào Textbox YourName.
- Nút Exit xác nhận người dùng có thực sự muốn thoát khỏi chương trình không? Yes: thoát, No: không.

Hướng dẫn:

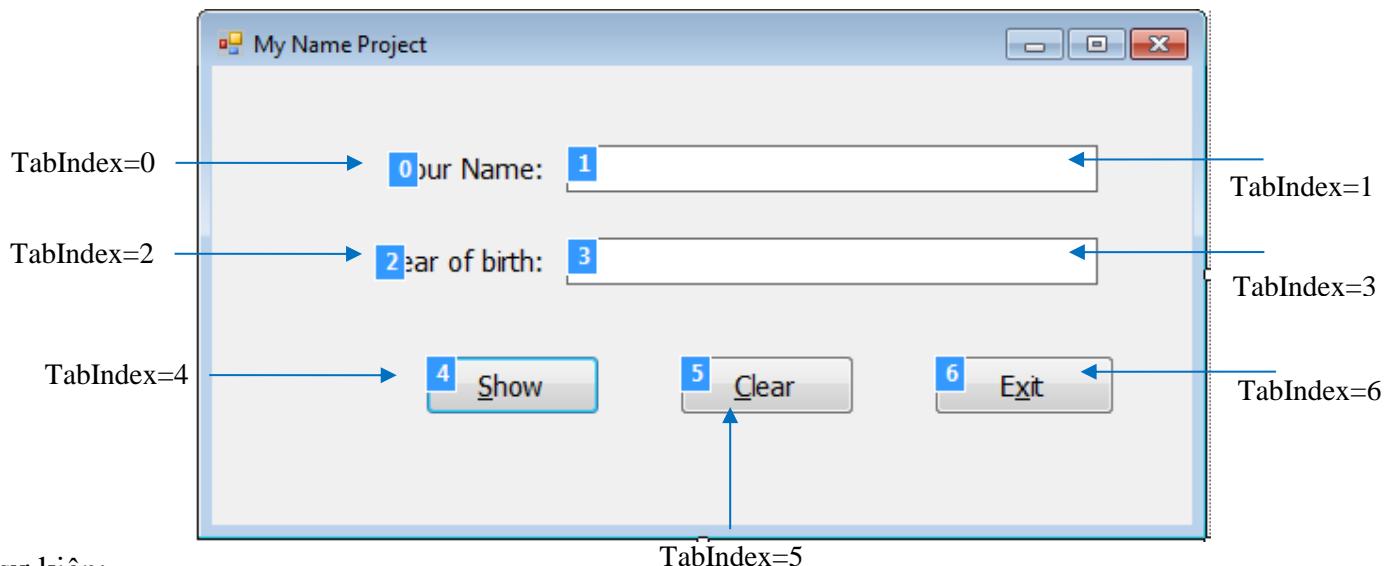
- ❖ Danh sách các thuộc tính của các Object:

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 focus)
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 tự nhận tiêu điểm focus trên Form: chọn menu View → Tab Order



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



❖ Các sự kiện:

```
private void frmBaiTap1_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult r;
    r = MessageBox.Show("Do you really want to close?", "Ex1",
                        MessageBoxButtons.YesNo,
                        MessageBoxIcon.Question,
                        MessageBoxDefaultButton.Button1);
    if (r == DialogResult.No)
        e.Cancel = true;
}
```

```
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();
}
```

```
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();
}
```

```

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); //Current Year - Enter year
    s = s + "Age: " + age.ToString();
    MessageBox.Show(s);
}

private void btnClear_Click(object sender, EventArgs e)
{
    //clear text
    txtYourName.Clear();
    txtYear.Clear();
    //TextBox Your Name reset focus
    txtYourName.Focus();
}

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

```

- ❖ Tập tin Program.cs: thay đổi Form thực thi đầu tiên là frmBaiTap1:

```

namespace MyProject
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new frmBaiTap1());
        }
    }
}

```

Bài 2. Tạo một chương trình dùng để quy đổi tiền từ VND sang USD và ngược lại, từ VND sang EUR và ngược lại.



Yêu cầu:

- Khi form load thực hiện nhập số tiền cần đổi vào Textbox, nếu nhập dữ liệu không phải là số hoặc để trống thì sẽ hiển thị lỗi vào errorProvider.
- btnVNDtoUSD: thực hiện quy đổi từ VND sang USD (với tỷ giá là: 1 USD= 17861 VND) và hiển thị vào lblKetQua, làm tròn 2 chữ số thập phân (dùng phương thức Math.Round)
- btnUSDtoVND: thực hiện quy đổi từ USD sang VND (với tỷ giá là: 1 USD= 17861 VND) và hiển thị vào lblKetQua, làm tròn 2 chữ số thập phân (dùng phương thức Math.Round)
- btnVNDtoEUR: thực hiện quy đổi từ VND sang EUR (với tỷ giá là: 1 USD= 27043 VND) và hiển thị vào lblKetQua, làm tròn 2 chữ số thập phân (dùng phương thức Math.Round)
- btnEURtoVND: thực hiện quy đổi từ EUR sang VND (với tỷ giá là: 1 USD= 27043 VND) và hiển thị vào lblKetQua, làm tròn 2 chữ số thập phân (dùng phương thức Math.Round)
- sự kiện FormClosing: xác nhận người dùng có chắc chắn thoát không? Yes: thoát, No: không.

Bài 3. Thêm vào project một windows form, hiện thực bài toán giải phương trình bậc 1:

Yêu cầu:

- Khi form load thực hiện nhập số A, B. Nếu dữ liệu nhập không hợp lệ thì thông báo lỗi (dùng errorProvider). btnTinh, btnXoa bị mờ (Enabled=false). Sau khi nhập dữ liệu hợp lệ và đầy đủ thì btnTinh có tác dụng (Enabled=true).
- btnTinh: tính nghiệm phương trình (xét tất cả các trường hợp xảy ra: PT có 1 nghiệm, vô nghiệm, vô số nghiệm) và lưu kết quả vào lblKetQua. Khi đó btnXoa có tác dụng, btnTinh bị mờ.
- btnXoa: Xóa các Textbox và label Kết Quả, đặt con trỏ vào TextBox A, btnXoa bị mờ
- btnThoat: xác nhận người dùng có chắc chắn thoát khỏi ứng dụng hay không? Yes: thoát, No: không.

Giải Phương Trình Bậc 1

PHƯƠNG TRÌNH: AX + B = 0

Nhập A:

Nhập B:

Nghiệm PT:

Bài 4. Viết chương trình đọc năm Âm lịch khi có năm Dương lịch tương ứng, biết:

Năm DL MOD 10=0_9: Canh – Tân – Nhâm – Quý – Giáp - Át – Bính – Đinh – Mậu – Kỷ

Năm DL MOD 12=0_11: Thân – Dậu – Tuất – Hợi – Tý – Sửu – Dần – Mẹo – Thìn – Ty - Ngọ - Mùi.

(Lưu ý: Kiểm tra nhập dữ liệu trong textbox Nhập Năm Dương lịch)

frmDoiNam

Nhập năm dương lịch:

Năm Âm lịch:

Bài 5. Viết chương trình nhập vào số nguyên N, kiểm tra N có phải là số nguyên tố hay không? (xem giao diện)

Tương tự bài trên: hãy thêm các button “Tính tổng N số nguyên Dương đầu tiên”, “Tính tổng N số nguyên Dương chẵn đầu tiên”, “Tính giai thừa N”, “Kiểm tra số hoàn thiện”,...

Lưu ý: số hoàn thiện là số có Tổng các ước số của nó bằng chính nó.

VD: $6 = 1 + 2 + 3$: 6 là số hoàn thiện.

Bài 6. Viết chương trình Nhập vào 2 số a, b. Xuất ra USCLN và BSCLN của 2 số a và b (tự vẽ giao diện)

Bài 7. Xử lý control động, các dãy ghép khi run time được tạo ra trên form (thiết kế không có)

Hãy 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 lô A 1000/vé

Giá vé lô B 1500/vé

Giá vé lô C 2000/vé

Trên Form trình bày một sơ đồ các chỗ ngồi để người sử dụng chọn vị trí muốn mua. Trên sơ đồ này cũng thể hiện những vị trí đã bán vé và những vị trí chưa bán vé bằng cách thể hiện màu khác nhau (ghế chưa bán vé màu trắng, ghế đã bán vé màu vàng).

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

- Nếu đây là vị trí chưa bán vé thì đổi màu của vị trí này sang màu xanh để cho biết đây là vị trí đang chọn.

- 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 Message box thông báo cho người sử dụng biết.

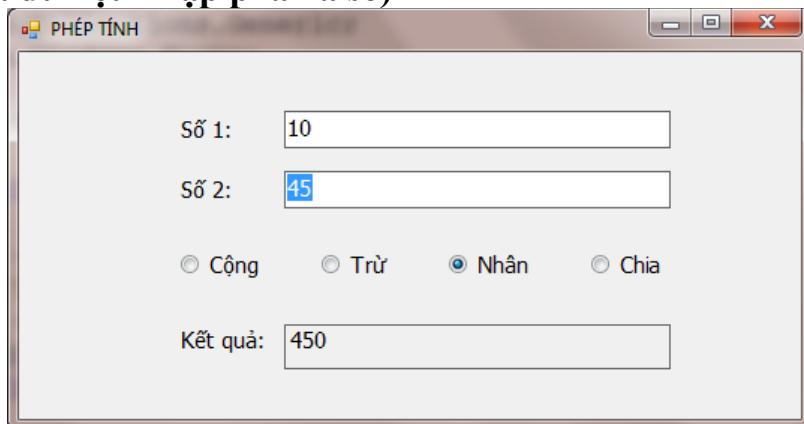
Sau khi đã chọn các vị trí người sử dụng có thể click chuột vào nút **CHỌN** hoặc **HỦY BỎ**

Nếu click vào nút **CHỌN** thì:

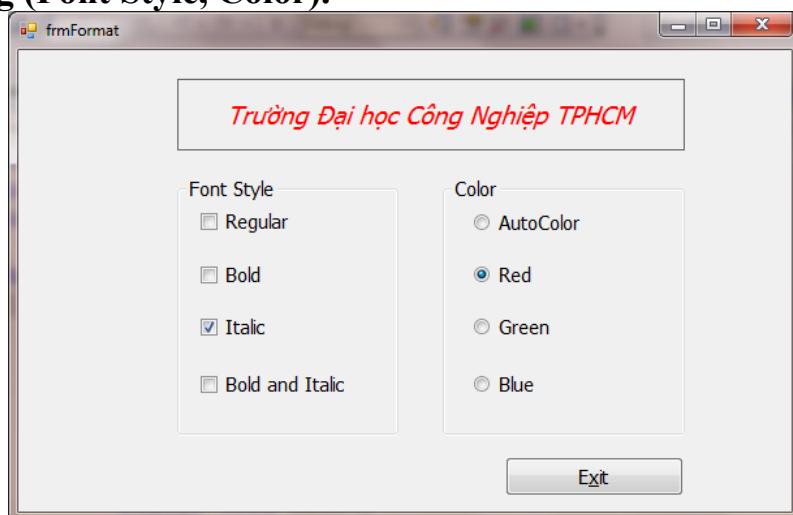
- Đổi màu các vị trí đã chọn (màu xanh) trên sơ đồ sang màu vàng (cho biết vị trí đã bán vé)
- Xuất lên một Label tổng số tiền phải trả cho số vé đã mua (phụ thuộc vào các vị trí đã chọn)
- Nếu click vào nút **HỦY BỎ** thì:
- Đổi màu các vị trí đã chọn (màu xanh) trên sơ đồ sang màu trắng trở lại
- Xuất lên label giá trị 0

Textbox, Button, Radiobutton, Checkbox.

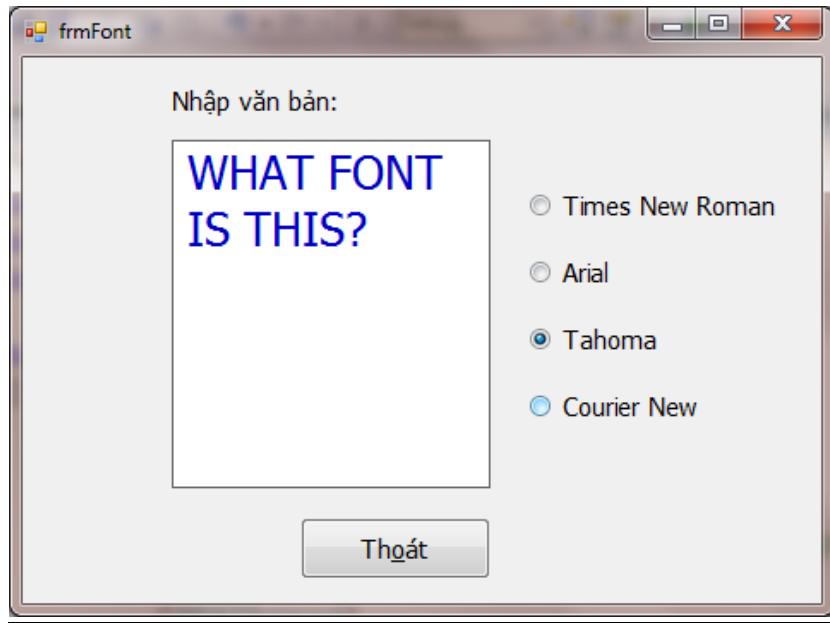
Bài 8. Tạo giao diện như hình: nhập vào 2 số và 4 Radio, click vào Radio Cộng sẽ thực hiện phép Cộng 2 số và xuất kết quả vào Label Kết quả. Thực hiện tương tự cho các Radio còn lại. (kiểm tra dữ liệu nhập phải là số)



Bài 9. Viết chương trình dùng để định dạng LABEL TEXT theo các Checkbox và Radio button tương ứng (Font Style, Color).



Bài 10. Viết chương trình dùng để định dạng TEXTBOX theo các font chữ tương ứng với từng Radio button.



Bài 11. Viết chương trình Quản lý thu tiền cho quán Café Sinh Viên như sau:

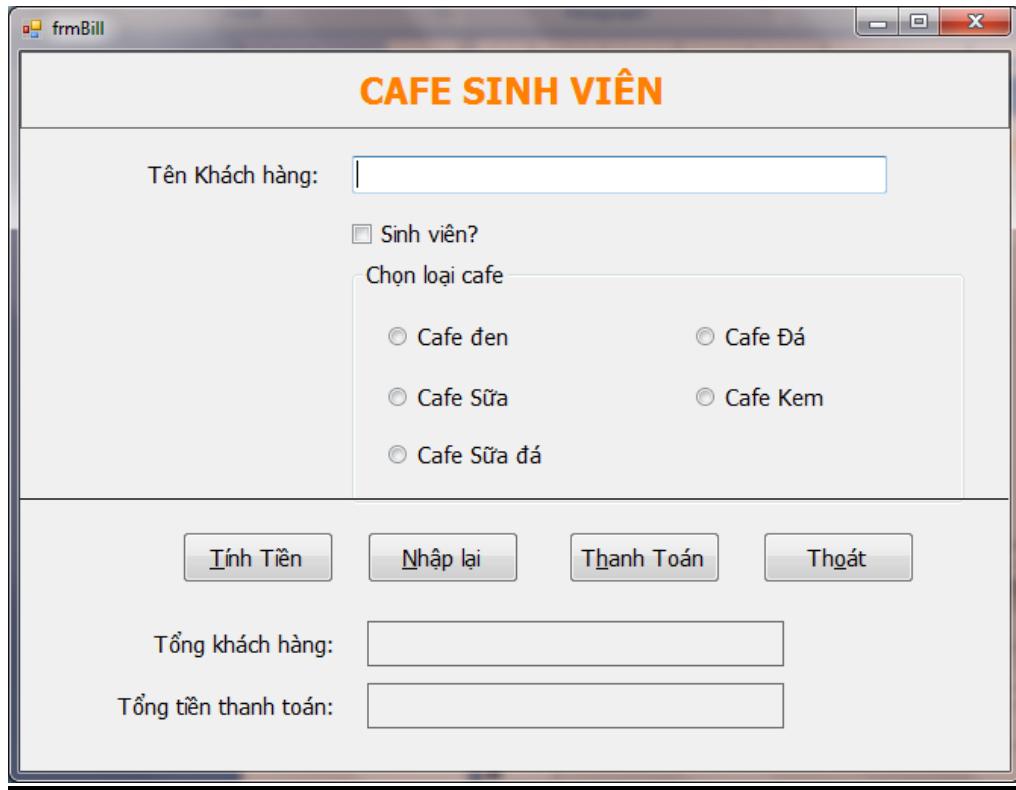
- Khi nhóm khách hàng vào quán gọi café sẽ được chọn 1 trong các loại café. Giá cho từng loại café tương ứng như sau:

Café đen:	20.000đ
Café đá:	25.000đ
Café sữa:	25.000đ
Café sữa đá:	30.000đ
Café kem:	35.000đ
- Khi khách hàng gọi tính tiền, thu ngân sẽ nhập thông tin tên khách hàng và tùy vào loại café 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%). Cuối cùng thu ngân sẽ tính tổng số người trong nhóm và thanh toán tiền.

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 café 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, đồng thời lưu lại thông tin tổng số tiền và tổng số lượt khách. btnNhapLai, btnThanhToan sáng lên sẵn sàng cho việc nhập khách hàng mới hoặc thanh toán tiền.
- btnNhapLai_Click: khởi tạo lại trạng thái ban đầu của Form, btnNhapLai bị mờ.

- **btnThanhToan_Click:** 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, 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.



Bài 12. Viết chương trình kiểm tra sức khỏe của 1 người bằng cách nhập vào các thông số: họ tên (không được rỗng), chiều cao (cm), cân nặng (kg) dữ liệu này phải là số. Tính cân số cân chuẩn của người đó theo công thức sau (tùy theo Nam hay Nữ):

Công thức tính cân nặng chuẩn:

Cân nặng lý tưởng (kg) = $T - 100 - ((T - 150)/ N)$

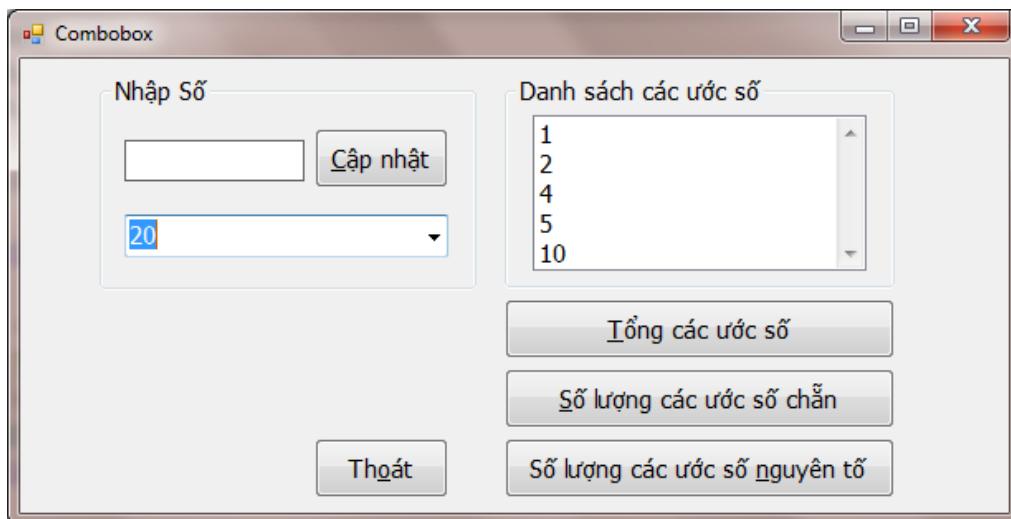
(Trong đó: T là chiều cao (cm), N = 4 với Nam và N = 2 với Nữ.)

- **btnTinh:** Thực hiện tính số cân chuẩn theo công thức, sau đó hiển thị kết quả vào label kết quả như hình.
- **btnXoa:** trả lại trạng ban đầu của Form, cho phép nhập người mới
- **btnThoat:** thoát khỏi chương trình.

4 ListBox, Combobox, CheckListBox GroupBox, Panel & TabControl.

Bài 13. Viết chương trình thực hiện như sau (xem hình bên dưới)

- **Form_Load:** nhập số vào Textbox (nhớ kiểm tra dữ liệu nhập)
- **btnCapNhat:** thêm số vừa nhập vào Combobox, đồng thời xóa textbox nhập, đặt con trỏ lại textbox nhập.
- Khi người dùng chọn 1 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 sẽ hiển thị thông tin tương ứng vào messagebox dựa vào các ước số trên listbox.
- **btnThoat:** thoát khỏi ứng dụng.



Bài 14. Viết chương trình thêm các phần tử vào listbox (listbox được chọn nhiều phần tử) các số tự nhiên N được nhập từ textbox

Viết code cho các nút 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 15. Viết chương trình nhập danh sách các Sinh Viên theo yêu cầu sau: (xem hình bên dưới).

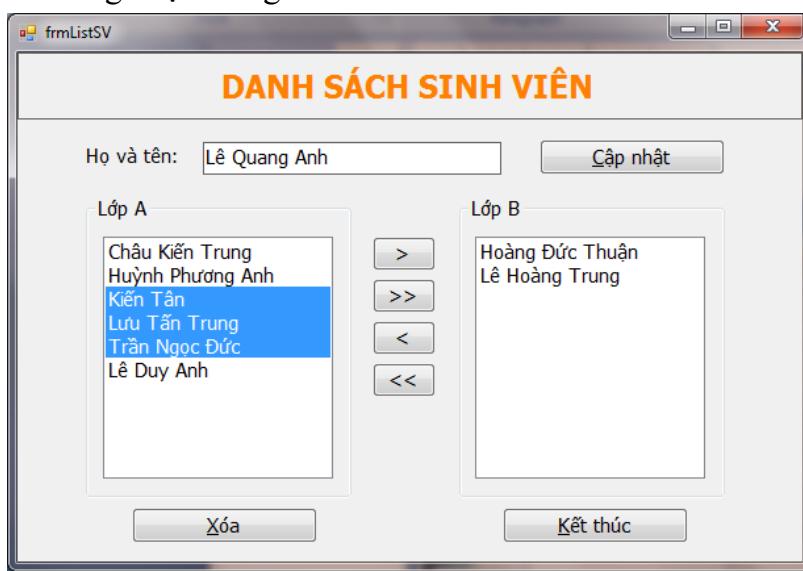
Các listbox được phép chọn nhiều mục (kết hợp giữa phím Shift, Ctrl và chuột)

Khi người dùng nhập họ và tên của Sinh viên vào Textbox, click nút Cập Nhật thì tên Sinh Viên đó sẽ được nhập vào Listbox Trái.

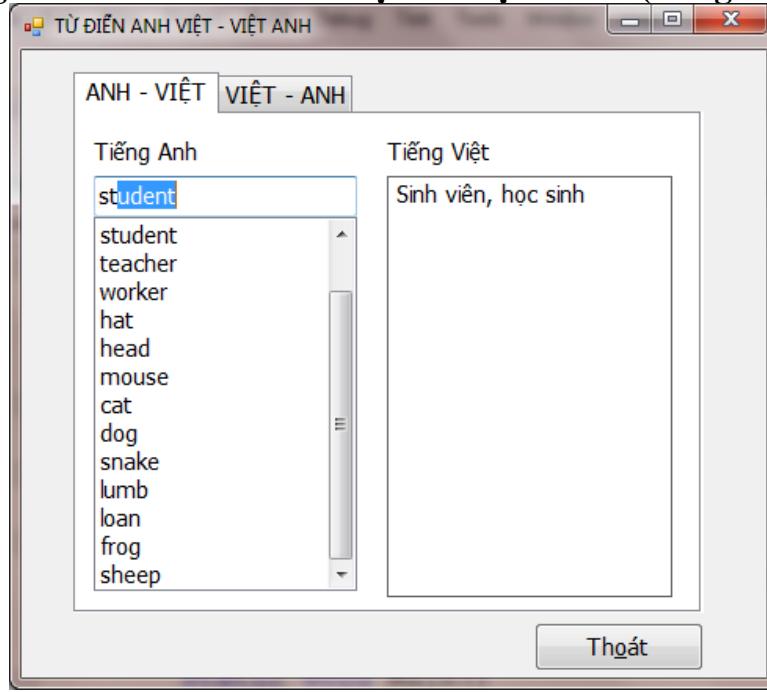
, chuyển các tên đang chọn từ Listbox Trái sang Listbox Phải và ngược lại.

, chuyển hết toàn bộ các tên từ Listbox Trái sang Listbox Phải và ngược lại.

btnXoa: cho phép xóa các tên đang chọn trong Listbox Trái.



Bài 16. Viết chương trình từ điển Anh – Việt và Việt – Anh (dùng TabControl)



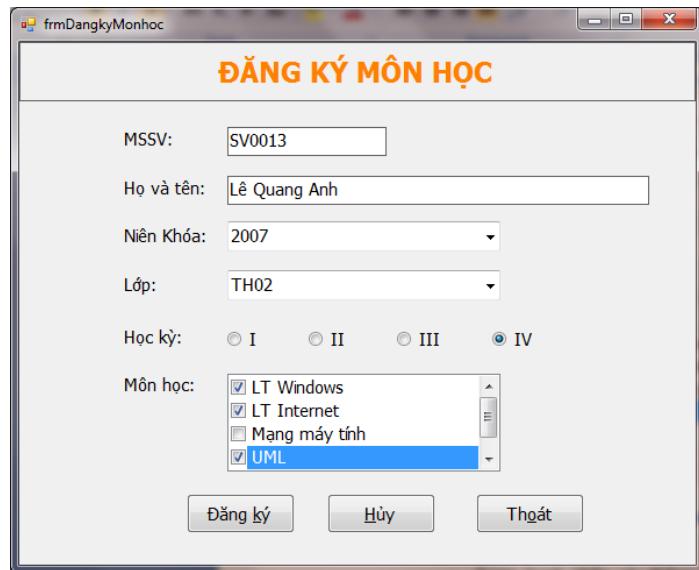
Bên trái là Combobox thể hiện dưới dạng Simple, chứa danh sách các từ cần tra cứu.

Bên phải là TextBox thể hiện dưới dạng MultiLine, ghi nghĩa của các từ được chọn bên Combobox.

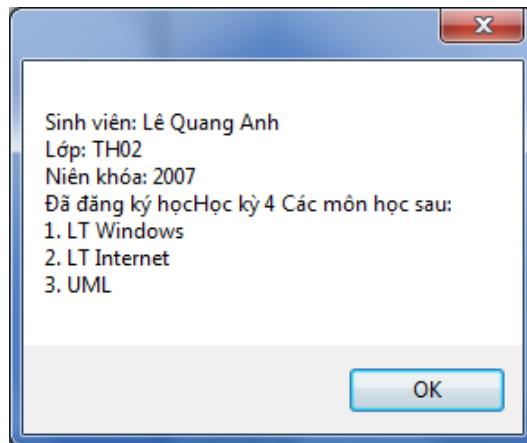
Yêu cầu:

- Khi nhập vào combobox 1 từ cần tra thì chương trình sẽ dò tìm đến chữ nào khớp với ký tự gần nhất
- Khi nhấn nút Enter hoặc Doubleclick 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 object (word) → ArrayList

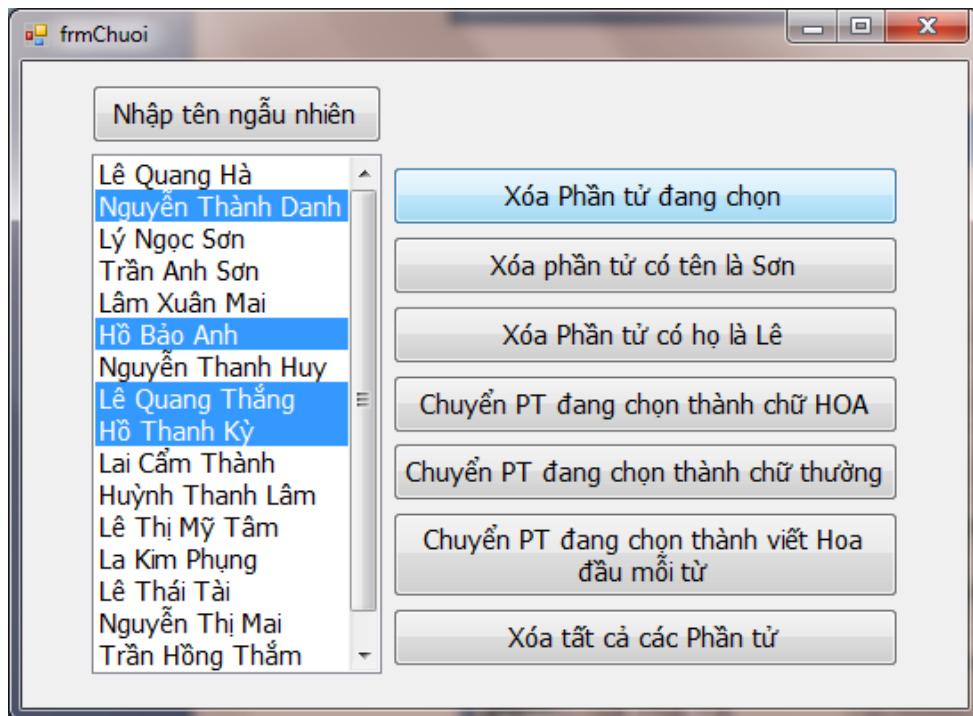
Bài 17. Viết chương trình cho phép Sinh Viên đăng ký học các môn học trong học kỳ:



- Form_load: nhập các thông tin mssv, họ tên sv, chọn niên khóa, lớp học, học kỳ đăng ký, các môn học đăng ký (kiểm tra dữ liệu nhập)
- btnDangKy: Hiển thị các thông tin mà Sinh viên đã đăng ký lên Messagebox như hình bên dưới.
- btnHuy: trả lại trạng thái ban đầu của Form
- btnThoat: thoát khỏi ứng dụng.



Bài 18. Thực hiện Form xử lý chuỗi có giao diện như sau:



- btnNgauNhien: sẽ thực hiện nhập vào listbox 50 tên dựa vào mảng chuỗi HO, TENLOT, TEN:
 - HO={“Lê”, “Nguyễn”, “Lý”, “Trần”, “Lâm”, “Hồ”, “Lai”, “Huỳnh”, “La”}
 - TENLOT={“Quang”, “Thành”, “Ngọc”, “Anh”, “Xuân”, “Bảo”, “Cẩm”, “Thị”, “Kim”, “Thái”, “Hồng”}
 - TEN={“Hà”, “Danh”, “Sơn”, “Mai”, “Thắng”, “Kỳ”, “Thành”, “Lâm”, “Tâm”, “Phụng”, “Thắm”}
 - Thực hiện các button tương ứng
 - 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.

Bài 19. Thiết kế giao diện để thực hiện nhập vào 4 số a, b, c, d nguyên dương khác nhau. Hãy in ra màn hình hai số không phải là số lớn nhất và số nhỏ nhất. Yêu cầu không sử dụng mảng dữ liệu trong chương trình.

Bài 20. Thiết kế giao diện để thực hiện nhập vào 3 số nguyên dương a, b, c (khác nhau). Hãy in ra màn hình 3 số trên thứ tự tăng dần (chương trình chỉ được dùng thêm tối đa hai biến có kiểu là số nguyên, không sử dụng biến kiểu mảng).

Kiểm tra 3 số đó có lập thành tam giác hay không? Nếu có thì xác định loại tam giác (phân biệt thành các loại: đều, vuông, vuông cân, cân, thường).

Bài 21. Thiết kế giao diện để viết chương trình nhập vào một số nguyên dương n và thực hiện các công việc sau:

- (a) Kiểm tra n có phải là số nguyên tố không?
- (b) Nếu n không phải là số nguyên tố thì xác định số nguyên tố gần n nhất và bé hơn n.

Bài 22. Viết chương trình nhập vào ngày, tháng, năm:

- (a) Kiểm tra tính hợp lệ của số liệu nhập vào. Cho biết năm này có phải là năm nhuận không?
- (b) Nhập vào ngày, tháng, năm tính khoảng cách giữa ngày tháng năm vừa nhập với ngày 1/1 của cùng năm.

Bài 23. Viết chương trình theo dạng thực đơn đơn giản, cho phép người dùng thực hiện các thao tác sau:

- (a) Nhập vào năm dương lịch, đổi sang năm âm lịch.
- (b) Nhập vào năm âm lịch, đổi sang năm dương lịch.
- (c) Nhập vào ngày tháng năm, đổi ra ngày trong năm.
- (d) Nhập vào ngày trong năm, đổi ra ngày thứ mấy của tháng nào.

Bài 24. Xây dựng chương trình hiển thị lịch tò như sau:

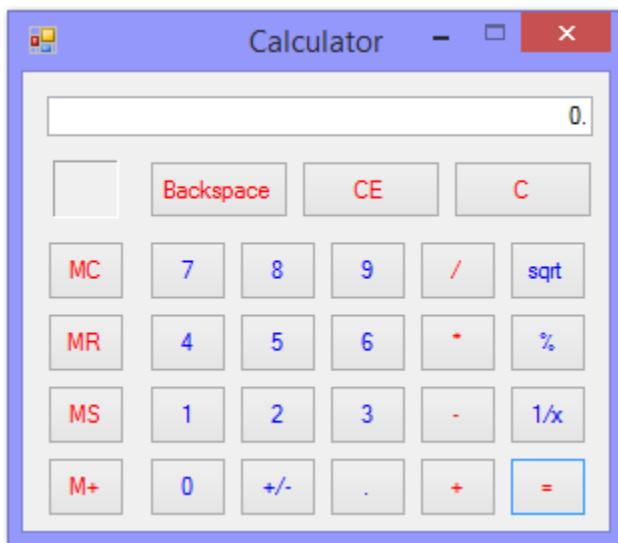


Khi mới mở Form sẽ hiển thị lịch theo ngày hiện hành.

Khi thay đổi ngày, tháng, năm trên form PHẢI kiểm tra ngày hợp lệ, sau đó click nút “Xem lịch” thì sẽ hiển thị đúng ngày tháng năm đó vào groupbox phía trên (bao gồm cả thứ trong tuần).

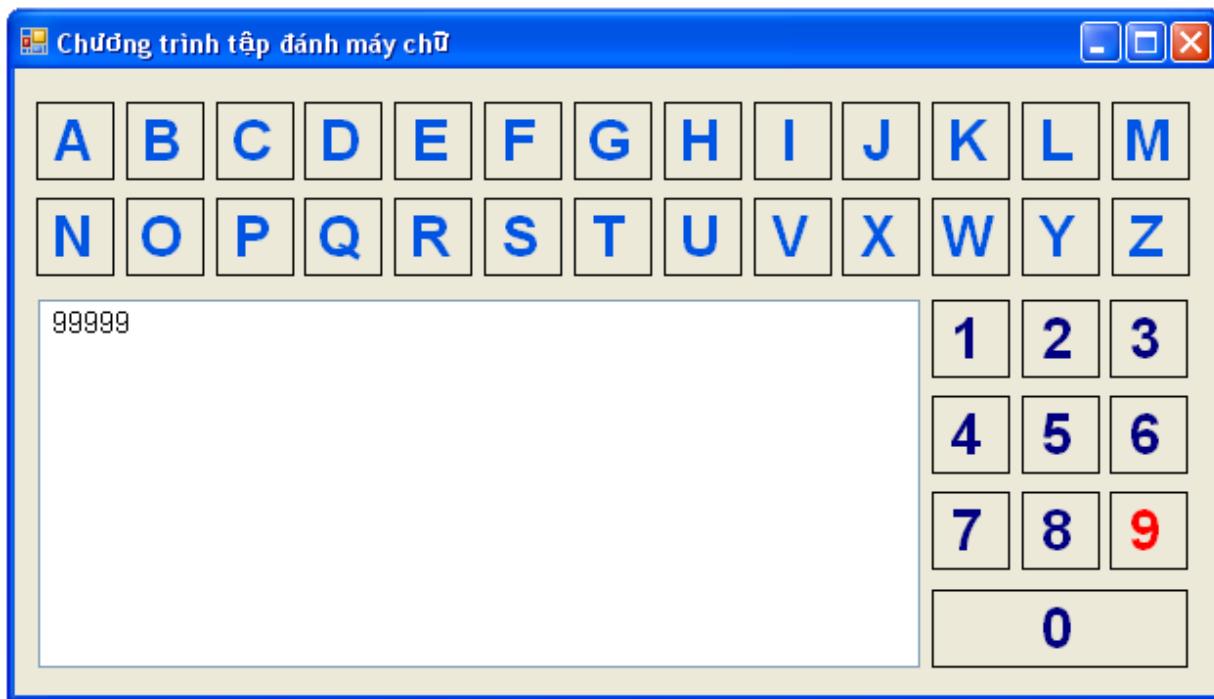
Mouse Event handling, Keyboard event handling

Bài 25. Xây dựng chương trình máy tính cá nhân.



Bài 26. Viết chương trình thực hiện việc thực hành đánh máy chữ, giao diện như hình bên.

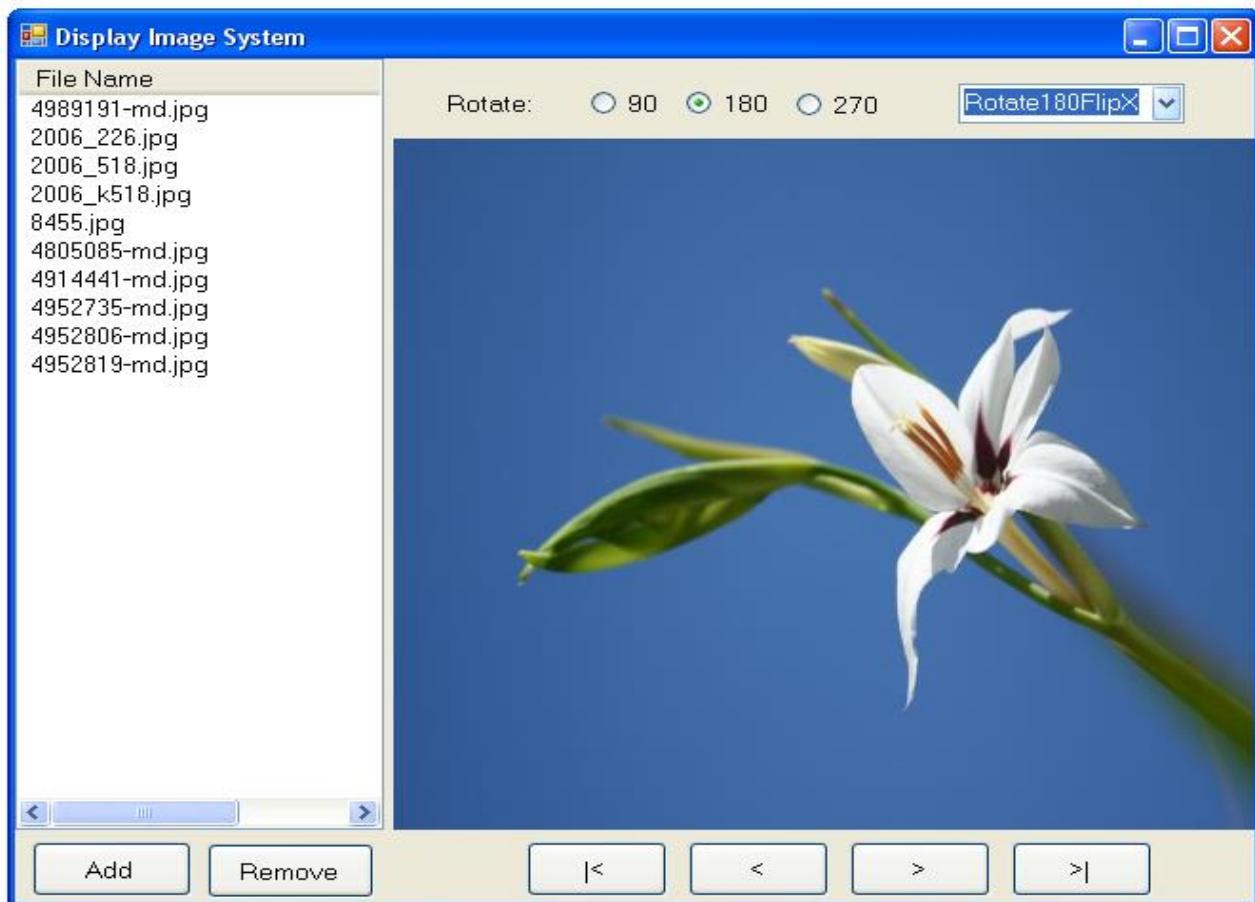
Yêu cầu: Khi nhấn 1 kí tự hoặc 1 số nào thì kí tự hoặc số đó chuyển sang màu đỏ, khi thả phím ra thì màu chữ trở lại màu đen. Đồng thời ký tự vừa được nhấn sẽ hiện bên dưới Text Box



III. Bài tập sử dụng các GUI advanced controls

Bài 1. Sử dụng ListView, PictureBox thiết kế chương trình xử lý hình ảnh đơn giản tương tự như sau:

- Khi nhấp vào nút “Add”, chương trình hỗ trợ mở hộp thoại chọn các tập tin hình ảnh từ các thư mục hình ảnh. (*.jpg, *.gif, *.bmp, *.png). Các tập tin chọn từ hộp thoại sẽ được đưa vào danh sách các tập tin hình ảnh (ListView).
- Khi nhấp nút “Remove” sẽ loại bỏ hình ảnh chọn lựa trong ListView.
- Các option rotate cho phép xoay hình ảnh.
- Các nút |<, <, >, >| cho phép di chuyển tới hình ảnh first, previous, next, last trong danh sách hình ListView.



Bài 2. Quản lý thông tin tài khoản cần lưu trữ các thông tin sau: số tài khoản, tên khách hàng, địa chỉ khách hàng và số tiền trong tài khoản. Thiết kế chương trình quản lý thông tin tài khoản cho một ngân hàng tương tự như sau:

Khi nhấn vào nút “Thêm”, caption của nút chuyển thành “Huỷ” và enable nút “Lưu”

Nút lưu có tác dụng lưu dữ liệu từ các textbox vào trong ListView.

Nút xoá có tác dụng xoá mẫu tin hiện hành trong ListView.

STT	Mã tài khoản	Tên khách hàng	Địa chỉ	Số tiền
1	0160901609	Nguyễn Văn An	12 Nguyễn Văn Bảo Gò Vấp	1000000
2	0160901610	Trần Văn Anh	123 Cách Mạng Tháng 8, Quận 10	1200000
3	0160901611	Hồ Văn Ân	86 Trần Bình Trọng, Quận 5	5200000

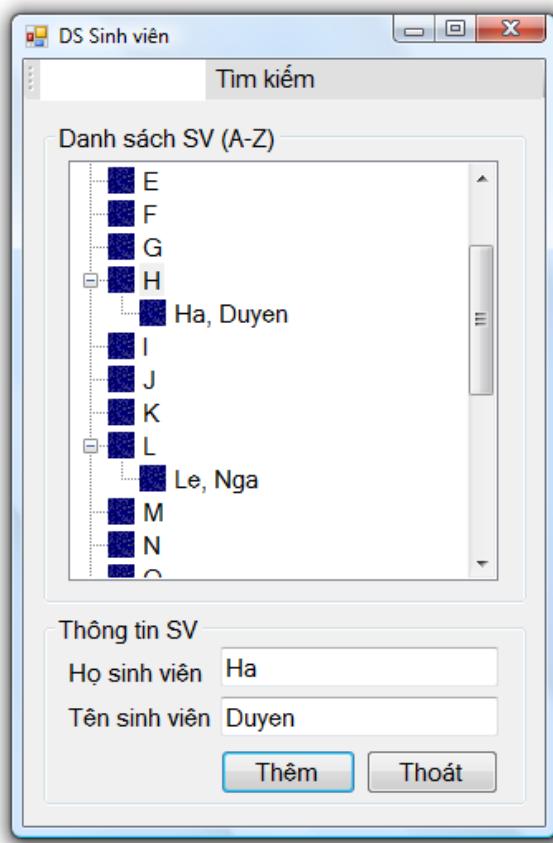
Bài 3. Viết chương trình quản lý cho thuê CD, thông tin của khách thuê bao gồm mã KH, họ tên, số lượng CD thuê và tình trạng cho thuê.

- Tạo lớp KhachThue (thuộc tính, set/get ...)
- Tạo lớp DanhSachKhachThue (dùng mảng hoặc collection ArrayList)
- Khi có một khách hàng đến thuê CD tại cửa hàng, nhập dữ liệu liên quan đến khách hàng và nhấn nút “Tính Thuê” thông tin đó được thêm vào List View bên dưới.
- Các nút <<, <, >, >> cho phép di chuyển giữa các mẫu tin liên quan đến khách thuê trong danh sách khách thuê. Khi di chuyển, các thông tin tương ứng sẽ được hiển thị lên các control ở phía trên.
- Nút “Xóa” cho phép xóa khách thuê (hiện hành) trong danh sách khách thuê.



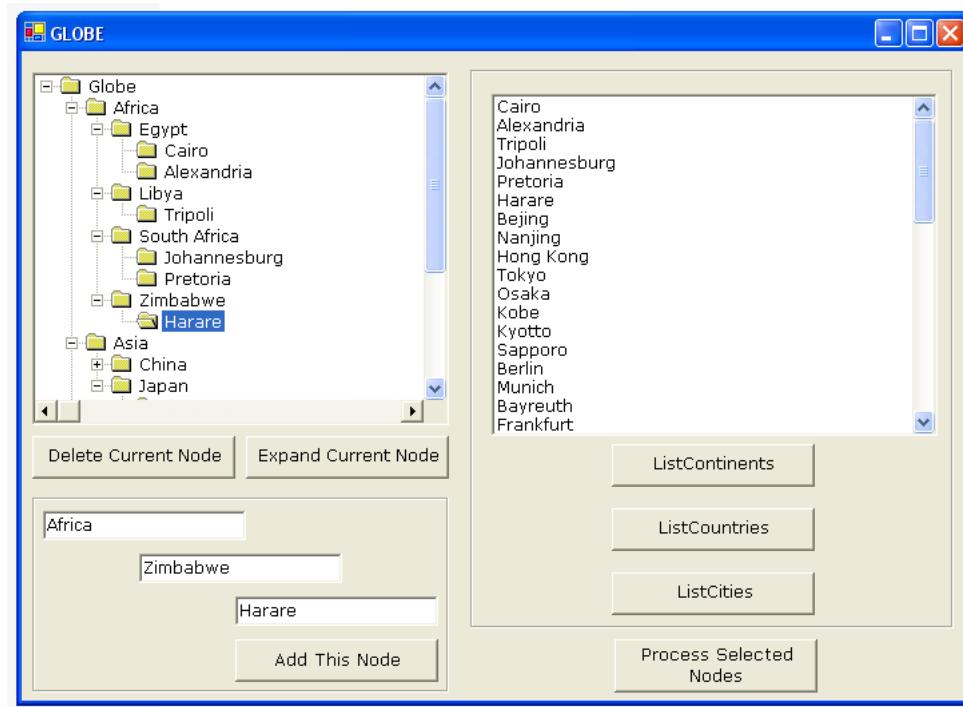
Bài 4. Viết chương trình quản lý danh sách sinh viên ở dạng cây như hình minh họa.

- Khi chương trình bắt đầu chạy, cây được phát sinh theo bảng chữ cái A-Z
- Cho phép thêm mới sinh viên, sinh viên mới thêm sẽ hiển thị trên cây tùy theo họ.
- Khi tìm kiếm, nếu sinh viên đó tồn tại, hiển thị chọn SV đó trên cây.



Bài 5. Thiết kế ứng dụng dạng cây như sau:

- Nút gốc: Globe
- Nút con cấp 1: Các châu lục: Africa, Asia, Europe, N. America, Oceania, S.America
- Nút con cấp 2: Các nước thuộc châu lục ở cấp 1.

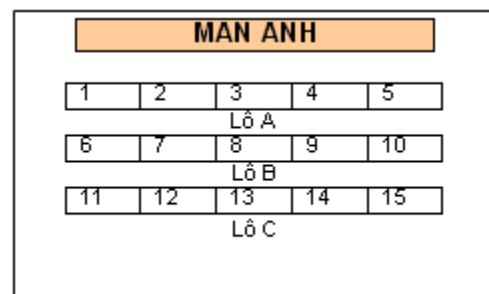


Bài 6. Hãy tạo một project xử lý động control. project giúp cho rạp chiếu phim quản lý việc bán vé của mình. Rạp có 6 hàng ghế, mỗi hàng có 3 ghế được tạo ra khi chạy chương trình, các ghế được đánh số từ 1 đến 15 và được phân thành 3 lô:

Giá vé lô A 1000/vé

Giá vé lô B 1500/vé

Giá vé lô C 2000/vé



Trên Form trình bày một sơ đồ các chỗ ngồi để người sử dụng chọn vị trí muốn mua. Trên sơ đồ này cũng thể hiện những vị trí đã bán vé và những vị trí chưa bán vé bằng cách thể hiện màu khác nhau (ghế chưa bán vé màu trắng, ghế đã bán vé màu vàng). Khi người sử dụng click chuột tại một vị trí trên sơ đồ thì:

- Nếu đây là vị trí chưa bán vé thì đổi màu của vị trí này sang màu xanh để cho biết đây là vị trí đang chọn.
- 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 Message box thông báo cho người sử dụng biết.

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

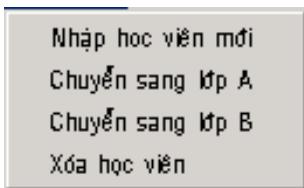
Nếu click vào nút CHỌN thì:

- Đổi màu các vị trí đã chọn (màu xanh) trên sơ đồ sang màu vàng (cho biết vị trí đã bán vé)
- Xuất lên một Label tổng số tiền phải trả cho số vé đã mua (phụ thuộc vào các vị trí đã chọn)
- Nếu click vào nút HỦY BỎ thì:
- Đổi màu các vị trí đã chọn (màu xanh) trên sơ đồ sang màu trắng trở lại
- Xuất lên label giá trị 0

Bài 7. Hãy tạo một project có giao diện như sau:

Khi người sử dụng click vào mục kết thúc thì xuất hiện một Msgbox 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 message box 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: chứa hai mục Lớp A và Lớp B
 - Khi người sử dụng Click vào nút cập nhật :
 - Nếu họ và tên khác rỗng thì:
 - Phụ thuộc vào lớp đã chọn trong cập nhật họ tên vào ListBox tương ứng chính
 - Xóa nội dung của Textbox
 - Đưa con trỏ về Textbox
 - Nếu họ và tên bằng rỗng:
 - Xuất hiện một Msgbox yêu cầu phải nhập họ tên
 - Đưa con trỏ về Textbox
 - Khi người sử dụng Click vào nút Trở về thì đóng form nhập thông tin lại và quay trở về form chính.



ComboBox
trên Form

Bài 8. Viết chương trình quản lý thông tin thiệp chúc mừng nhân dịp các ngày lễ của một cửa hàng bán thiệp, thông tin của thiệp bao gồm mã thiệp (mã số này là duy nhất), mô tả, nhà sản xuất, loại thiệp (chúc xuân, giáng sinh, sinh nhật), năm sản xuất và số lượng hiện có.

Quản lý dữ liệu

THÔNG TIN THIỆP CHÚC MỪNG

Mã thiệp	<input type="text"/>	Mô tả	<input type="text"/>	
Loại thiệp	Giáng sinh	Nhà sản xuất	<input type="text"/>	
Năm sản xuất	<input type="text"/>	Số lượng	<input type="text"/>	
Mã thiệp	Mô tả	Nhà sản xuất	Năm sản xuất	Số lượng











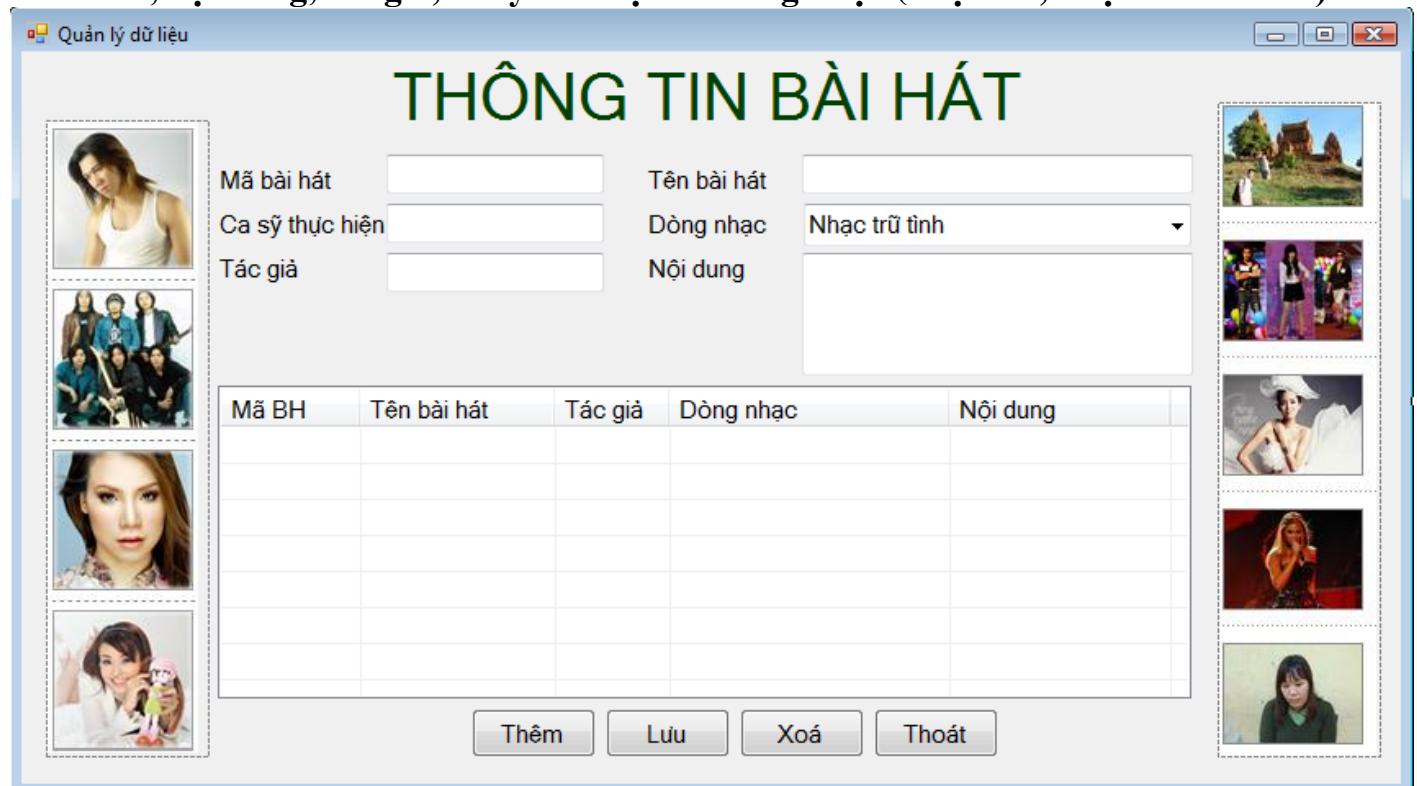

- Thiết kế giao diện, khi chạy chương trình, nút “Lưu” chưa có tác dụng.
 - Tạo lớp ThongTinThiep (khai báo các thuộc tính, set/get và các constructor)
 - Tạo lớp DanhSachThiep (dùng collection bất kỳ trừ ArrayList)
 - Khi nhấn vào nút “Thêm”, caption của nút chuyển thành “Huỷ” và cho phép nút “Lưu” hoạt động.
 - Nút “Lưu” có tác dụng lưu dữ liệu từ các Textbox vào trong DanhSachThiep và ListView.
 - Nút “Xóa” có tác dụng xoá mẫu tin hiện hành trong DanhSachThiep và ListView.
 - Nút “Thoát” có tác dụng để đóng chương trình và phải hỏi người sử dụng trước khi thật sự kết thúc chương trình.
 - Các nút <<, <, >, >> cho phép di chuyển giữa các mẫu tin liên quan đến thông tin bài hát trong danh sách các thiệp. Khi di chuyển, các thông tin tương ứng sẽ được hiển thị lên các control ở phía trên.

Bài 9. Viết chương trình quản lý thông tin truyện tranh của một cửa hàng bán sách, thông tin của một cuốn truyện tranh bao gồm mã truyện (mã số này là duy nhất), tên truyện, nhà xuất bản, họ tên tác giả, năm xuất bản và số lượng hiện có.



- Thiết kế giao diện, khi chạy chương trình, nút “Lưu” chưa có tác dụng.
- Tạo lớp TruyenTranh (khai báo các thuộc tính, set/get và các constructor)
- Tạo lớp DanhSachTruyenTranh (dùng collection bất kỳ trừ ArrayList)
- Khi nhấn vào nút “Thêm”, caption của nút chuyển thành “Huỷ” và cho phép nút “Lưu” hoạt động.
- Nút “Lưu” có tác dụng lưu dữ liệu từ các Textbox vào trong DanhSachTruyenTranh và ListView.
- Nút “Xóa” có tác dụng xoá mẫu tin hiện hành trong DanhSachTruyenTranh và ListView.
- Nút “Thoát” có tác dụng để đóng chương trình và phải hỏi người sử dụng trước khi thật sự kết thúc chương trình.
- Các nút <<, <, >, >> cho phép di chuyển giữa các mẫu tin liên quan đến truyện tranh trong danh sách truyện tranh. Khi di chuyển, các thông tin tương ứng sẽ được hiển thị lên các control ở phía trên.

Bài 10. Viết chương trình quản lý thông tin về lời các bài nhạc của một website nghe nhạc trực tuyến, thông tin của lời một bài nhạc bao gồm mã số (mã số này là duy nhất), tên bài, nội dung, tác giả, ca sỹ thể hiện và dòng nhạc (nhạc trẻ, nhạc trữ tình ...).



- Thiết kế giao diện, khi chạy chương trình, nút “Lưu” chưa có tác dụng.
- Tạo lớp ThongTinBaiHat (khai báo các thuộc tính, set/get và các constructor)
- Tạo lớp DanhSachBaiHat (dùng collection bất kỳ trừ ArrayList)
- Khi nhấn vào nút “Thêm”, caption của nút chuyển thành “Huỷ” và cho phép nút “Lưu” hoạt động.
- Nút “Lưu” có tác dụng lưu dữ liệu từ các Textbox vào trong DanhSachBaiHat và ListView.
- Nút “Xóa” có tác dụng xoá mẫu tin hiện hành trong DanhSachBaiHat và ListView.
- Nút “Thoát” có tác dụng để đóng chương trình và phải hỏi người sử dụng trước khi thật sự kết thúc chương trình.

- Các nút <<, <, >, >> cho phép di chuyển giữa các mẫu tin liên quan đến thông tin bài hát trong danh sách bài hát. Khi di chuyển, các thông tin tương ứng sẽ được hiển thị lên các control ở phía trên.

Bài 11. Viết chương trình quản lý sách chuyên ngành CNTT.



- Tạo lớp ThongTinSach (khai báo các thuộc tính, set/get và các constructor)
- Tạo lớp DanhSachSach (dùng collection bất kỳ trừ LinkedList)
- Thiết kế giao diện với những điều kiện sau:
 - Khi chạy chương trình, nút “Lưu” chưa có tác dụng.
 - Khi nhấp vào nút “Thêm”, caption của nút chuyển thành “Huỷ” và cho phép nút “Lưu” hoạt động.
 - ComboBox không cho phép ghi dữ liệu vào.
- Nút “Lưu” có tác dụng lưu dữ liệu từ các Textbox vào trong DanhSachSach và ListView.
- Nút “Xóa” có tác dụng xoá mẫu tin hiện hành trong DanhSachSach và ListView.
- Nút “Thoát” có tác dụng để đóng chương trình và phải hỏi người sử dụng trước khi thật sự kết thúc chương trình.
- Các nút <<, <, >, >> cho phép di chuyển giữa các mẫu tin liên quan đến thông tin sách trong danh sách sách. Khi di chuyển, các thông tin tương ứng sẽ được hiển thị lên các control TextBox/ComboBox ở phía trên.
- Nút “Lưu tập tin” cho phép lưu dữ liệu trên ListView vào trong tập tin (SV có thể dùng BinaryFormatter hoặc XmlSerializer), khi lưu cho phép chọn thư mục và tập tin để lưu. Bổ sung thêm chức năng khi Form chính được mở lên, nạp dữ liệu từ tập tin đã được lưu.

- Tạo Form mới (gồm TextBox và nút “Lưu”) cho phép nhập dữ liệu mới về Nhà xuất bản. Sau khi nhấn nút “Lưu” của Form này dữ liệu sẽ được cập nhật vào ComboBox Nhà xuất bản trên Form chính.

Bài 12. Viết chương trình quản lý giá ngoại tệ.

DHTH9ABTT

QUẢN LÝ TỶ GIÁ NGOẠI TỆ

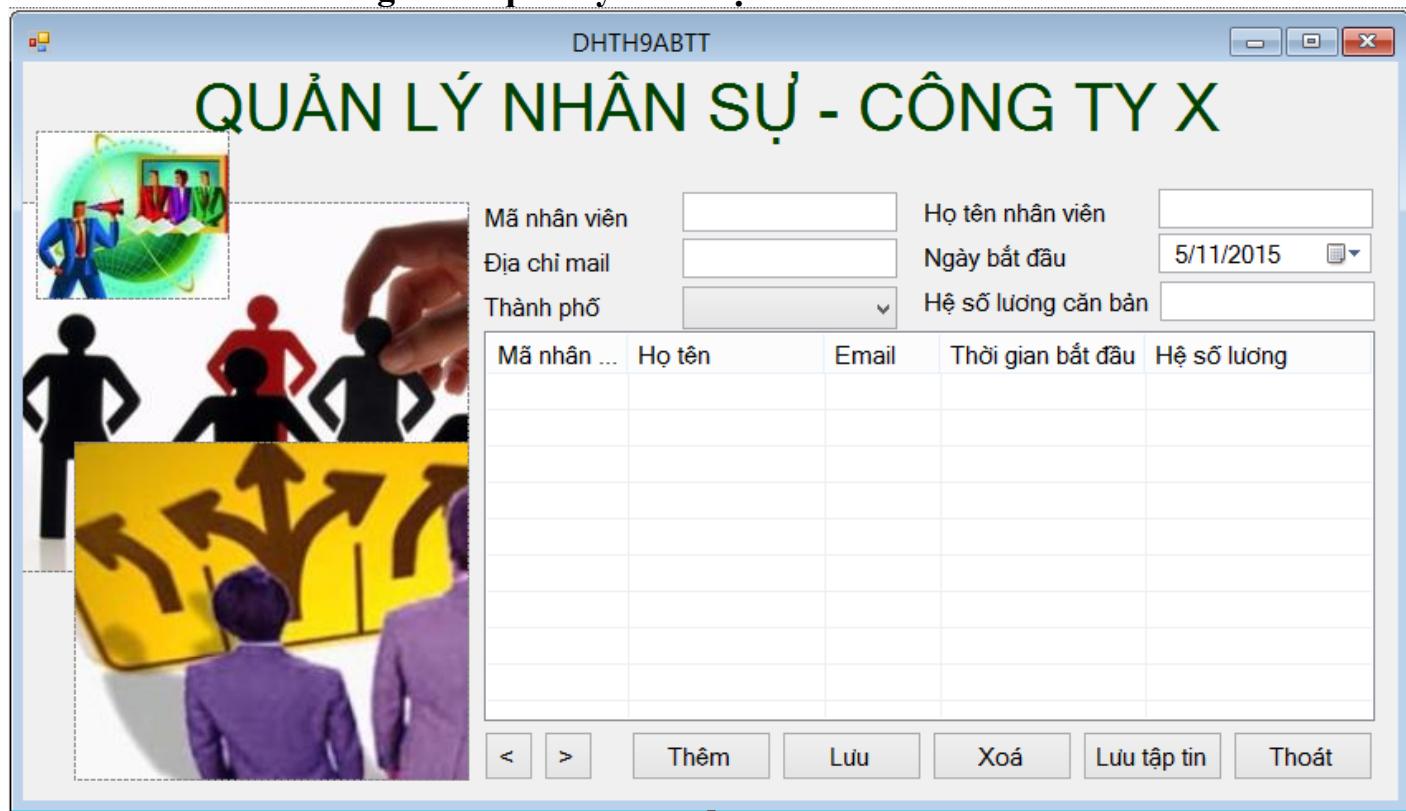
Ký hiệu	<input type="text"/>	Tên ngoại tệ	<input type="text"/>
Giá mua vào	<input type="text"/>	Ngày cập nhật	5/11/2015 <input type="button" value="▼"/>
Giá bán ra	<input type="text"/>	Tên nước	<input type="text"/>

Ký hiệu	Tên ngoại tệ	Giá mua vào	Giá bán ra	Ngày cập nhật

- Tạo lớp NgoaiTe (khai báo các thuộc tính, set/get và các constructor)
- Tạo lớp DanhSachNgoaiTe (dùng collection bất kỳ trừ LinkedList)
- Thiết kế giao diện với những điều kiện sau:
 - Khi chạy chương trình, nút “Lưu” chưa có tác dụng.
 - Khi nhấn vào nút “Thêm”, caption của nút chuyển thành “Huỷ” và cho phép nút “Lưu” hoạt động.
 - ComboBox không cho phép ghi dữ liệu vào.
- Nút “Lưu” có tác dụng lưu dữ liệu từ các Textbox vào trong DanhSachNgoaiTe và ListView.
- Nút “Xóa” có tác dụng xoá mẫu tin hiện hành trong DanhSachNgoaiTe và ListView.

- Nút “Thoát” có tác dụng để đóng chương trình và phải hỏi người sử dụng trước khi thật sự kết thúc chương trình.
- Các nút <<, <, >, >> cho phép di chuyển giữa các mẫu tin liên quan đến ngoại tệ trong danh sách ngoại tệ. Khi di chuyển, các thông tin tương ứng sẽ được hiển thị lên các control TextBox/ComboBox ở phía trên.
- Nút “Lưu tập tin” cho phép lưu dữ liệu trên ListView vào trong tập tin (SV có thể dùng BinaryFormatter hoặc XmlSerializer), khi lưu cho phép chọn thư mục và tập tin để lưu. Bổ sung thêm chức năng khi Form chính được mở lên, nạp dữ liệu từ tập tin đã được lưu.
- Tạo Form mới (gồm TextBox và nút “Lưu”) cho phép nhập dữ liệu mới về Tên nước. Sau khi nhấn nút “Lưu” của Form này dữ liệu sẽ được cập nhật vào ComboBox Tên nước trên Form chính.

Bài 13. Viết chương trình quản lý nhân sự.



- Tạo lớp NhanSu (khai báo các thuộc tính, set/get và các constructor)
- Tạo lớp DanhSachNhanSu (dùng collection bất kỳ trừ LinkedList)
- Thiết kế giao diện với những điều kiện sau:
 - Khi chạy chương trình, nút “Lưu” chưa có tác dụng.
 - Khi nhấn vào nút “Thêm”, caption của nút chuyển thành “Huỷ” và cho phép nút “Lưu” hoạt động.

- ComboBox không cho phép ghi dữ liệu vào.
 - Nút “Lưu” có tác dụng lưu dữ liệu từ các Textbox vào trong DanhSachNhanSu và ListView.
 - Nút “Xóa” có tác dụng xoá mẫu tin hiện hành trong DanhSachNhanSu và ListView.
 - Nút “Thoát” có tác dụng để đóng chương trình và phải hỏi người sử dụng trước khi thật sự kết thúc chương trình.
 - Các nút <<, <, >, >> cho phép di chuyển giữa các mẫu tin liên quan đến nhân sự trong danh sách nhân sự. Khi di chuyển, các thông tin tương ứng sẽ được hiển thị lên các control TextBox/ComboBox ở phía trên.
 - Nút “Lưu tập tin” cho phép lưu dữ liệu trên ListView vào trong tập tin (SV có thể dùng BinaryFormatter hoặc XmlSerializer), khi lưu cho phép chọn thư mục và tập tin để lưu. Bổ sung thêm chức năng khi Form chính được mở lên, nạp dữ liệu từ tập tin đã được lưu.
 - Tạo Form mới (gồm TextBox và nút “Lưu”) cho phép nhập dữ liệu mới về Thành phố. Sau khi nhấn nút “Lưu” của Form này dữ liệu sẽ được cập nhật vào ComboBox Thành phố trên Form chính.



- Tạo lớp SanPham (khai báo các thuộc tính, set/get và các constructor)
 - Tạo lớp DanhSachSanPham (dùng collection bất kỳ trừ LinkedList)
 - Thiết kế giao diện với những điều kiện sau:

- Khi chạy chương trình, nút “Lưu” chưa có tác dụng.
- Khi nhấp vào nút “Thêm”, caption của nút chuyển thành “Huỷ” và cho phép nút “Lưu” hoạt động.
- ComboBox không cho phép ghi dữ liệu vào.
- Nút “Lưu” có tác dụng lưu dữ liệu từ các Textbox vào trong DanhSachSanPham và ListView.
- Nút “Xóa” có tác dụng xoá mẫu tin hiện hành trong DanhSachSanPham và ListView.
- Nút “Thoát” có tác dụng để đóng chương trình và phải hỏi người sử dụng trước khi thoát chương trình.
- Các nút <<, <, >, >> cho phép di chuyển giữa các mẫu tin liên quan đến sản phẩm trong danh sách sản phẩm. Khi di chuyển, các thông tin tương ứng sẽ được hiển thị lên các control TextBox/ComboBox ở phía trên.
- Nút “Lưu tập tin” cho phép lưu dữ liệu trên ListView vào trong tập tin (SV có thể dùng BinaryFormatter hoặc XmlSerializer), khi lưu cho phép chọn thư mục và tập tin để lưu. **Bổ sung thêm chức năng khi Form chính được mở lên, nạp dữ liệu từ tập tin đã được lưu.**
- Tạo Form mới (gồm TextBox và nút “Lưu”) cho phép nhập dữ liệu mới về Nhà cung cấp. Sau khi nhấp nút “Lưu” của Form này dữ liệu sẽ được cập nhật vào ComboBox Nhà cung cấp trên Form chính.

BÀI TẬP TUẦN 06, 07 MÔN LẬP TRÌNH WINDOWS (C#)

Chương IV : Tập hợp, chuỗi, Lambda Expressions và Extension Methods

Mục tiêu:

- Hiểu rõ và áp dụng được cách tổ chức dữ liệu theo Collection, Strings trong C#
- Viết được chương trình ứng dụng ở dạng Windows Form Application với thao tác của Strings và Tập hợp (collection)
- Hiểu, xây dựng được biểu thức Lambda (Lambda Expression)
- Hiểu và áp dụng được các phương thức mở rộng (Extension Methods) trên các đối tượng, tập hợp.

Yêu cầu:

- Tất cả các bài tập lưu trong thư mục: T:\MaSV_HoTen\Tuan01\
- Tạo Solution Tuan01 trong thư mục T:\MaSV_HoTen\Tuan01\ trong Visual Studio 2010/2012/2013. Mỗi bài tập là một Project riêng biệt.
- Cuối mỗi buổi thực hành, SV phải nén (.rar hoặc .zip) thư mục làm bài và nộp lại bài tập đã thực hiện trong buổi đó.
- Bài tập bắt buộc của Tuần XX, GV sẽ công bố cuối giờ. Bài tập bắt buộc sẽ gửi vào email giáo viên quy định ví dụ thangvnguyen2001@gmail.com với Subject và tập tin nén kèm theo: **XY_ZK_LTWINDOWS_TUANXX_HOTENSINHVIEN** (trong đó XY là lớp ZK là nhóm XX sẽ là tuần)

Các phương thức và thuộc tính của Queue, Stack.

Phương thức- thuộc tính	Mục đích
Synchronized()	Phương thức static trả về một Queue wrapper được thread-safe.
Count	Thuộc tính trả về số thành phần trong hàng đợi
IsReadOnly	Thuộc tính xác định hàng đợi là chỉ đọc
IsSynchronized	Thuộc tính xác định hàng đợi được đồng bộ
SyncRoot	Thuộc tính trả về đối tượng có thể được sử dụng để đồng bộ truy cập Queue.
Clear()	Xóa tất cả các thành phần trong hàng đợi
Clone()	Tạo ra một bản sao
Contains()	Xác định xem một thành phần có trong mảng.
CopyTo()	Sao chép những thành phần của hàng đợi đến mảng một chiều đã tồn tại
Dequeue()	Xóa và trả về thành phần bắt đầu của hàng đợi.
Enqueue()	Thêm một thành phần vào hàng đợi.
GetEnumerator()	Trả về một enumerator cho hàng đợi.
Peek()	Trả về phần tử đầu tiên của hàng đợi và không xóa nó.
ToArray()	Sao chép những thành phần qua một mảng mới

Hình 8. Phương thức, thuộc tính của Queue

Phương thức- thuộc tính	Mục đích
Synchronized()	Phương thức static trả về một Stack wrapper được thread-safe.
Count	Thuộc tính trả về số thành phần trong ngăn xếp
IsReadOnly	Thuộc tính xác định ngăn xếp là chỉ đọc
IsSynchronized	Thuộc tính xác định ngăn xếp được đồng bộ
SyncRoot	Thuộc tính trả về đối tượng có thể được sử dụng để đồng bộ truy cập Stack.
Clear()	Xóa tất cả các thành phần trong ngăn xếp
Clone()	Tạo ra một bản sao
Contains()	Xác định xem một thành phần có trong mảng.
CopyTo()	Sao chép những thành phần của ngăn xếp đến mảng một chiều đã tồn tại
Pop()	Xóa và trả về phần tử đầu Stack
Push()	Đưa một đối tượng vào đầu ngăn xếp
GetEnumerator()	Trả về một enumerator cho ngăn xếp.
Peek()	Trả về phần tử đầu tiên của ngăn xếp và không xóa nó.
ToArray()	Sao chép những thành phần qua một mảng mới

Hình 9. Phương thức, thuộc tính của Stack

Phương thức- thuộc tính	Mục đích
Adapter()	Phương thức static tạo một wrapper ArrayList cho đối tượng IList
FixedSize()	Phương thức static nạp chồng trả về danh sách đối tượng như là một wrapper. Danh sách có kích thước cố định, các thành phần của nó có thể được sửa chữa nhưng không thể thêm hay xóa.
ReadOnly()	Phương thức static nạp chồng trả về danh sách lớp như là một wrapper, chỉ cho phép đọc.
Repeat()	Phương thức static trả về một ArrayList mà những thành phần của nó được sao chép với giá trị xác định.
Synchronized()	Phương thức static trả về danh sách wrapper được thread-safe
Capacity	Thuộc tính để get hay set số thành phần trong ArrayList.
Count	Thuộc tính nhận số thành phần hiện thời trong mảng
IsFixedSize	Thuộc tính kiểm tra xem kích thước của ArrayList có cố định hay không

Hình 10. Thuộc tính, phương thức của ArrayList (1)

IsReadOnly	Thuộc tính kiểm tra xem ArrayList có thuộc tính chỉ đọc hay không.
IsSynchronized	Thuộc tính kiểm tra xem ArrayList có thread-safe hay không
Item()	Thiết lập hay truy cập thành phần trong mảng tại vị trí xác định. Đây là bộ chỉ mục cho lớp ArrayList.
SyncRoot	Thuộc tính trả về đối tượng có thể được sử dụng để đồng bộ truy cập đến ArrayList
AddRange()	Phương thức public để thêm nhiều thành phần của một ICollection vào cuối của ArrayList
BinarySearch()	Phương thức nạp chồng public sử dụng tìm kiếm nhị phân để định vị một thành phần xác định trong ArrayList được sắp xếp.
Clear()	Xóa tất cả các thành phần từ ArrayList
Clone()	Tạo một bản copy
Contains()	Kiểm tra một thành phần xem có chứa trong mảng hay không
CopyTo()	Phương thức public nạp chồng để sao chép một ArrayList đến một mảng một chiều.
GetEnumerator()	Phương thức public nạp chồng trả về một enumerator dùng để lặp qua mảng
GetRange()	Sao chép một dãy các thành phần đến một ArrayList mới
IndexOf()	Phương thức public nạp chồng trả về chỉ mục vị trí đầu tiên xuất hiện giá trị
Insert()	Chèn một thành phần vào trong ArrayList

Hình 11. Thuộc tính, phương thức của ArrayList (2)

I. Bài tập phần Collection

Bài 1. Viết một chương trình tạo ra một lớp tên là LopHoc, trong đó có khai báo bộ chỉ mục chỉ đến tên của từng học viên trong lớp. Cho phép một lớp có tối đa 30 học viên. Tạo chương trình minh họa cho phép người dùng nhập vào tên của từng học viên. Xuất kết quả danh sách học viên của lớp thông qua bộ chỉ mục.

Bài 2. Viết chương trình sử dụng ArrayList để tạo một mảng.

Bài 3. Viết chương trình tạo ra đối tượng Queue tên là myQueue. Khởi tạo myQueue có 5 giá trị ngẫu nhiên. Hãy thực hiện các bước sau, mỗi bước thực hiện phải xuất tình trạng của myQueue:

- Lấy một giá trị ra.

- Lấy tiếp một giá trị nữa.
- Xem một giá trị ở đầu queue.
- Đưa vào queue một giá trị.

Phương thức- thuộc tính	Mục đích
Synchronized()	Phương thức static trả về một Hashtable wrapper được thread-safe.
Count	Thuộc tính trả về số thành phần trong hashtable
IsReadOnly	Thuộc tính xác định hashtable là chỉ đọc
IsSynchronized	Thuộc tính xác định hashtable được đồng bộ
SyncRoot	Thuộc tính trả về đối tượng có thể được sử dụng để đồng bộ truy cập Hastable.
Keys	Thuộc tính trả về một ICollection chứa những khóa trong hashtable.
Values	Thuộc tính trả về một ICollection chứa những giá trị trong hashtable.
Add()	Thêm một thành phần mới với khóa và giá trị xác định.
Clear()	Xóa tất cả đối tượng trong hashtable.
Item()	Chi mục cho hastable
Clone()	Tạo ra một bản sao
Contains()	Xác định xem một thành phần có trong hashtable.
ContainsKey()	Xác định xem hashtable có chứa một khóa xác định
CopyTo()	Sao chép những thành phần của hashtable đến mảng một chiều đã tồn tại
GetEnumerator()	Trả về một enumerator cho hashtable.
GetObjectData()	Thực thi ISerializable và trả về dữ liệu cần thiết để lưu trữ.
OnDeserialization	Thực thi ISerializable và phát sinh sự kiện deserialization khi hoàn thành.
Remove()	Xóa một thành phần với khóa xác định.

Hình 12. Phương thức, thuộc tính của Hashtable

Bài 4. Viết chương trình tạo đối tượng Stack tên là myStack. Khởi tạo myStack có 5 giá trị ngẫu nhiên. Hãy thực hiện các bước sau, mỗi bước thực hiện phải xuất tình trạng của myStack:

- Lấy một giá trị ra.
- Lấy tiếp một giá trị nữa.
- Xem một giá trị ở đầu stack.
- Đưa vào stack một giá trị.

Bài 5. Viết chương trình sử dụng kiểu dữ liệu từ điển để quản lý thông tin của một lớp học. Trong đó khóa là chuỗi mã số học viên còn giá trị là tên của học viên. Viết chương trình minh họa cho phép nhập vào 10 học viên, và cho phép người dùng tìm kiếm tên của học viên thông qua mã số học viên.

Bài 6. Công ty TrueLove:

- a) Công ty TrueLove cần lưu tên của các nhân viên của mình. Mỗi tháng một nhân viên sẽ được chọn ngẫu nhiên để nhận một quà tặng. Hãy dùng tập hợp Collection để viết chương trình quản lý danh sách nhân viên.
- b) Công ty TrueLove cần đặt tên cho sản phẩm mới, tên sản phẩm được chọn từ tên của nhân viên, vì vậy tên không được trùng, tên chỉ được dùng có 1 lần. Hãy dùng tập hợp Collection để viết chương trình cung cấp tên cho sản phẩm.
- c) Công ty TrueLove muốn dùng tên phổ biến nhất cho sản phẩm của họ, tên phổ biến là tên giống nhau nhiều nhất. Hãy dùng tập hợp Collection để viết chương trình cung cấp tên cho sản phẩm.
- d) Công ty TrueLove muốn cho nhân viên đi du lịch, chính sách được tạo ra là ưu tiên cho những người đăng ký trước. Hãy dùng tập hợp Collection để viết chương trình đăng ký du lịch.
- e) Công ty TrueLove muốn tạo danh sách các khách hàng theo thứ tự tăng dần theo doanh số. Hãy dùng tập hợp Collection để viết chương trình quản lý danh sách khách hàng.

Bài 7. Phòng học được quản lý trong một trường đại học gồm: phòng học lý thuyết, phòng máy tính và phòng thí nghiệm .

Mỗi phòng học đều có mã phòng, dãy nhà, diện tích, số bóng đèn.

Phòng học lý thuyết thì cần quan tâm xem có máy chiếu không.

Phòng máy tính thì cần biết là trang bị bao nhiêu máy tính.

Phòng thí nghiệm thì thêm thông tin chuyên ngành, sức chứa, có bồn rửa không (*rửa dụng cụ thí nghiệm / rửa tay*).

Ngoài ra, người quản lý cần phải xem xét phòng học có đạt chuẩn không.

Bài 8. Quản lý khách hàng xếp hàng mua vé tại nhà ga. Thông tin lưu trữ cho khách hàng gồm: số CMND khác hàng (String), Tên khách hàng, Ga đến, giá tiền (double).

Hệ thống menu gồm các mục:

- Thêm một khách hàng mới vào hàng đợi mua vé.
- Bán một vé cho khách hàng. Chỉ bán cho người đăng ký trước.
- Hiển thị danh sách khách hàng.
- Hủy một khách hàng ra khỏi danh sách. (khách hàng không mua vé nữa).
- Thống kê tình hình bán vé
- Lưu danh sách vào file
- Hiển thị danh sách các ga đang chờ mua vé.
- Hiển thị danh sách các ga đang chờ mua vé và số vé tương ứng cho ga.

Lưu ý:

- Số khách hàng trong danh sách hiện tại là số khách đang chờ, nhưng chưa có vé. Khi một khách hàng đã mua vé, thì loại khách hàng này ra khỏi danh sách chờ mua vé.
- Việc mua vé phải có thứ tự: ai vào trước thì mua vé trước (FIFO).
- Mỗi khi khách hàng mua được vé phải lưu lại khách hàng này để dùng cho việc thống kê.
- Mỗi khi thêm một khách hàng mới, nếu Số CMND khách hàng đã có thì không tạo phần tử mới mà chỉ cập nhật lại ga và giá tiền đến cho khách hàng đó.
- Mục thống kê tình hình: cho biết còn bao nhiêu khách hàng chờ nhận vé, bao nhiêu khách hàng đã nhận vé, tổng số tiền đã thu về là bao nhiêu.
- Việc lưu danh sách: chỉ lưu các khách hàng chờ mua vé. Các khách hàng đã nhận vé xem như kết sổ trong ngày không cần lưu lại.
- Khi chương trình vừa được chạy, lập tức tự động nạp toàn bộ danh sách khách hàng từ file (cách khách hàng chưa có vé).
- Khi hiển thị danh sách các ga đến đang chờ mua vé, chỉ hiển thị tên ga đó một lần. (Ví dụ: giả sử 10 khách hàng nhưng đăng ký đi đến 2 ga, thì chỉ hiển thị 2 hàng).

Viết phương thức Generic tính max, min trong một tập hợp. Sử dụng cách viết có dùng wildcard.

Bài 9. Viết phương thức Generic cho phép in ra mảng các phần tử, phương thức này cho phép in phần tử mảng của nhiều kiểu dữ liệu khác nhau.

Bài 10. Một quân bài trong bộ bài gồm hai thuộc tính loại bài (cờ, rô, chuồn, bích) và thứ tự quân bài(2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A). Dùng Generic.

Viết các lớp sau:

- Lớp mô tả quân bài
- Lớp tạo bộ bài 52 quân bài không trùng nhau với phương thức là xào bài
- Lớp kiểm tra

2. Bài tập phần chuỗi (String)

Một số lưu ý

System.String	
Phương thức/ Trường	Ý nghĩa
Empty	Trường public static thể hiện một chuỗi rỗng.
Compare()	Phương thức public static để so sánh hai chuỗi.
CompareOrdinal()	Phương thức public static để so sánh hai chuỗi không quan tâm đến thứ tự.
Concat()	Phương thức public static để tạo chuỗi mới từ một hay nhiều chuỗi.
Copy()	Phương thức public static tạo ra một chuỗi mới bằng sao từ chuỗi khác.
Equal()	Phương thức public static kiểm tra xem hai chuỗi có cùng giá trị hay không.
Format()	Phương thức public static định dạng một chuỗi dùng ký tự lệnh định dạng xác định.
Intern()	Phương thức public static trả về tham chiếu đến thể hiện của chuỗi.
IsInterned()	Phương thức public static trả về tham chiếu của chuỗi
Join()	Phương thức public static kết nối các chuỗi xác định giữa mỗi thành phần của mảng chuỗi.
Chars()	Indexer của chuỗi.
Length()	Chiều dài của chuỗi.

Hình 13. Phương thức, thuộc tính của lớp String (1)

<code>Clone()</code>	Trả về chuỗi.
<code>CompareTo()</code>	So sánh hai chuỗi.
<code>CopyTo()</code>	Sao chép một số các ký tự xác định đến một mảng ký tự Unicode.
<code>EndsWith()</code>	Chỉ ra vị trí của chuỗi xác định phù hợp với chuỗi đưa ra.
<code>Insert()</code>	Trả về chuỗi mới đã được chèn một chuỗi xác định.
<code>LastIndexOf()</code>	Chỉ ra vị trí xuất hiện cuối cùng của một chuỗi xác định trong chuỗi.
<code>PadLeft()</code>	Canh lề phải những ký tự trong chuỗi, chèn vào bên trái khoảng trắng hay các ký tự xác định.
<code>PadRight()</code>	Canh lề trái những ký tự trong chuỗi, chèn vào bên phải khoảng trắng hay các ký tự xác định.
<code>Remove()</code>	Xóa đi một số ký tự xác định.
<code>Split()</code>	Trả về chuỗi được phân định bởi những ký tự xác định trong chuỗi.
<code>StartsWith()</code>	Xem chuỗi có bắt đầu bằng một số ký tự xác định hay không.
<code>SubString()</code>	Lấy một chuỗi con.
<code>ToCharArray()</code>	Sao chép những ký tự từ một chuỗi đến mảng ký tự.
<code>ToLower()</code>	Trả về bản sao của chuỗi ở kiểu chữ thường.
<code>ToUpper()</code>	Trả về bản sao của chuỗi ở kiểu chữ hoa.
<code>Trim()</code>	Xóa bỏ tất cả sự xuất hiện của tập hợp ký tự xác định từ vị trí đầu tiên đến vị trí cuối cùng trong chuỗi.
<code>TrimEnd()</code>	Xóa như nhung ở vị trí cuối.
<code>TrimStart()</code>	Xóa như Trim nhưng ở vị trí đầu.

Hình 14. Phương thức, thuộc tính của lớp String (2)

Bài 11. Viết chương trình cho phép người dùng nhập vào một chuỗi. Sau đó đếm số ký tự xuất hiện của từng ký tự trong chuỗi.

Bài 12. Viết chương trình tìm một chuỗi con trong một chuỗi cho trước. Chương trình cho phép người dùng nhập vào một chuỗi, và chuỗi con cần tìm. Kết quả là chuỗi con có tìm thấy hay không, nếu tìm thấy thì hãy đưa ra vị trí đầu tiên tìm thấy.

Bài 13. Viết chương trình tìm số lần xuất hiện một chuỗi con trong một chuỗi cho trước. Chương trình cho phép người dùng nhập vào một chuỗi và chuỗi con cần đếm. Kết quả hiển thị chuỗi, chuỗi con và các vị trí mà chuỗi con xuất hiện trong chuỗi.

System.StringBuilder	
Phương thức	Ý nghĩa
Capacity()	Truy cập hay gán một số ký tự mà StringBuilder nắm giữ.
Chars()	Chi mục.
Length()	Thiết lập hay truy cập chiều dài của chuỗi
MaxCapacity()	Truy cập dung lượng lớn nhất của StringBuilder
Append()	Nối một kiểu đối tượng vào cuối của StringBuilder
AppendFormat()	Thay thế định dạng xác định bằng giá trị được định dạng của một đối tượng.
EnsureCapacity()	Đảm bảo rằng StringBuilder hiện thời có khả năng tối thiểu lớn như một giá trị xác định.
Insert()	Chèn một đối tượng vào một vị trí xác định
Replace()	Thay thế tất cả thể hiện của một ký tự xác định với những ký tự mới.

Hình 15. Phương thức của lớp StringBuilder

Bài 14. Viết chương trình cho phép người dùng nhập vào một chuỗi, rồi thực hiện việc đảo các ký tự trong chuỗi theo thứ tự ngược lại.

Bài 15. Viết chương trình cắt các từ có nghĩa trong câu.

Bài 16. Hãy nhập một chuỗi kí tự. In ra màn hình đảo ngược của chuỗi đó.

Bài 17. Nhập chuỗi. Thống kê số các chữ số '0', số chữ số '1', ..., số chữ số '9' trong chuỗi.

Bài 18. In ra vị trí kí tự trắng đầu tiên từ bên trái (phải) một chuỗi kí tự.

Bài 19. Nhập chuỗi. In ra tất cả các vị trí của chữ 'a' trong chuỗi và tổng số lần xuất hiện của nó.

Bài 20. Nhập chuỗi. Tính số từ có trong chuỗi. In mỗi từ trên một dòng.

Bài 21. Nhập chuỗi họ tên, in ra họ, tên dưới dạng viết hoa.

Bài 22. Thay kí tự x trong chuỗi s bởi kí tự y (s, x, y được đọc vào từ bàn phím)

Bài 23. Viết chương trình xóa chuỗi con trong một chuỗi.

Bài 24. Viết chương trình trích chuỗi con trong một chuỗi.

Bài 25. Viết chương trình in các từ của chuỗi trên mỗi dòng .

Bài 26. Viết chương trình tìm vị trí một chuỗi con trong một chuỗi đã cho.

Bài 27. Viết chương trình ghép hai chuỗi thành một chuỗi.

Bài 28. Viết chương trình loại bỏ các khoảng trắng thừa trong một chuỗi.

II. 3Bài tập phần Lambda Expressions và Extension Methods

a) Lựa chọn cú pháp truy vấn.

Cú pháp được sử dụng để viết các truy vấn LINQ:

- phương thức mở rộng (Extension method),
- biểu thức truy vấn (Query Expression) và
- kết hợp cả hai cú pháp trên (Query Dot).

b) Phương thức mở rộng (Extension method)

Mỗi phương thức mở rộng sẽ xử lý và trả về một tập kết quả **IEnumerable**, phương thức mở rộng tiếp theo tiếp tục sử dụng tập kết quả này để xử lý tiếp. Sử dụng biểu thức Lambda để tạo Extension method.

```
int[] nums = new int[] { 0, 4, 2, 6, 3, 8, 3, 1 };

var result1 = nums.Where(n => n < 5).OrderBy(n => n);

// or with line-breaks added for clarity
var result2 = nums
    .Where(n => n < 5)
    .OrderBy(n => n);
```

c) Biểu thức truy vấn (Query Expression)

Cú pháp gần giống với SQL, dễ viết và đọc. Tuy nhiên Query Expression không hỗ trợ đầy đủ các toán tử truy vấn nên có một số công việc không thể viết bằng Query Expression.

```
int[] nums = new int[] { 0, 4, 2, 6, 3, 8, 3, 1 };

var result = from n in nums
    where n < 5
    orderby n
    select n;
```

d) Biểu thức kết hợp (Query Dot)

Là sự kết hợp của hai cú pháp ở trên. Biểu thức truy vấn sẽ được bôi phương thức mở rộng ở bên ngoài.

```
int[] nums = new int[] { 0, 4, 2, 6, 3, 8, 3, 1 };

var result = (from n in nums
    where n < 5
    orderby n
```

```
select n).Distinct();
```

e) 2. Biểu thức truy vấn (Query Expression)

Trình biên dịch sẽ chuyển các biểu thức truy vấn về dạng phương thức mở rộng lúc biên dịch.

f) Tạo biến cục bộ (từ khóa let)

Sử dụng biến cục bộ để lưu trữ giá trị của một tính toán trung gian hoặc giá trị trả về của một truy vấn con.

```
var variance = from element in source
                let average = source.Average()
                select Math.Pow((element - average), 2);
```

Từ khóa let được biên dịch hoàn toàn bởi trình biên dịch, nó tạo một kiểu nặc danh để chứa cả biến element và biến average. Đoạn mã dưới đây sẽ được tạo ra sau khi trình biên dịch tạo ra khi biên dịch.

```
var variance = source.Select (
    element => new
    {
        element = element,
        average = source.Average()
    }
).Select(temp0 => Math.Pow(((double)temp0.element -
    temp0.average), 2));
```

g) Mở rộng truy vấn (từ khóa into)

Từ khóa group, join, select cho phép lưu trữ kết quả trả về vào một biến cục bộ để sử dụng ở những nơi khác của truy vấn. Dùng từ khóa into.

```
var groupings = from element in source
                 group element by element into groups
                 select new
                 {
                     Key = groups.Key,
                     Count = groups.Count()
                 };
};
```

h) So sánh phương thức mở rộng và biểu thức truy vấn

Truy vấn lấy tất cả các thông tin liên hệ ở bang “WA” và sắp xếp tăng dần theo họ, tên:

```
List<Contact> contacts = Contact.SampleData();

var q = contacts.Where(c => c.State == "WA")
    .OrderBy(c => c.LastName)
    .ThenBy(c => c.FirstName);

foreach (Contact c in q)
    Console.WriteLine("{0} {1}", c.FirstName, c.LastName);

List<Contact> contacts = Contact.SampleData();

var q = from c in contacts
    where c.State == "WA"
    orderby c.LastName, c.FirstName
    select c;

foreach (Contact c in q)
    Console.WriteLine("{0} {1}", c.FirstName, c.LastName);
```

Truy vấn lấy 5 dòng nhật ký cuộc gọi gần đây nhất:

```
List<Contact> contacts = Contact.SampleData();
List<CallLog> callLog = CallLog.SampleData();

var q = callLog.Join(contacts,
    call => call.Number,
    contact => contact.Phone,
    (call, contact) => new
    {
        contact.FirstName,
        contact.LastName,
        call.When,
        call.Duration
    })
    .OrderByDescending(call => call.When)
    .Take(5);

foreach (var call in q)
    Console.WriteLine("{0} - {1} {2} ({3}min)",
        call.When.ToString("ddMMM HH:m"),
        call.FirstName, call.LastName, call.Duration);

List<Contact> contacts = Contact.SampleData();
List<CallLog> callLog = CallLog.SampleData();

var q = (from call in callLog
    join contact in contacts on
    call.Number equals contact.Phone
    orderby call.When descending
```

```

    select new
    {
        contact.FirstName,
        contact.LastName,
        call.When,
        call.Duration
    }).Take(5);

foreach (var call in q)
    Console.WriteLine("{0} - {1} {2} ({3}min)",
        call.When.ToString("ddMMM HH:m"),
        call.FirstName, call.LastName, call.Duration);

```

i) Lọc kết quả trả về (mệnh đề where)

Sử dụng toán tử where, LINQ kiểm tra mỗi phần tử trong tập đích có phù hợp với tiêu chí được đưa ra không, nếu phù hợp thì mới trả về.

Tạo bộ lọc kết quả trả về sử dụng phương thức mở rộng và biểu thức truy vấn:

```

string[] animals = new string[] { "Koala", "Kangaroo",
    "Spider", "Wombat", "Snake", "Emu", "Shark",
    "Sting-Ray", "Jellyfish" };

var q = animals.Where(
    a => a.StartsWith("S") && a.Length > 5);

foreach (string s in q)
    Console.WriteLine(s);

string[] animals = new string[] { "Koala", "Kangaroo",
    "Spider", "Wombat", "Snake", "Emu", "Shark",
    "Sting-Ray", "Jellyfish" };

var q = from a in animals
    where a.StartsWith("S") && a.Length > 5
    select a;

foreach (string s in q)
    Console.WriteLine(s);
//Kết quả
Spider
Sting-Ray

```

j) Thay đổi kiểu trả về

- Trả về một giá trị hoặc phần tử đơn.
- Trả về một tập hợp IEnumerable với T có cùng kiểu dữ liệu với các phần tử trong danh sách gốc.
- Trả về một tập hợp IEnumerable với T là kiểu dữ liệu được định nghĩa và được khởi tạo sau lệnh select.
- Trả về một tập hợp IEnumerable với T là kiểu dữ liệu nặc danh được tạo sau lệnh select.
- Trả về một tập hợp IEnumerable> là một danh sách được gom theo nhóm.

Trả về một giá trị hoặc phần tử đơn

Toán tử truy vấn trả về dữ liệu dạng giá trị hoặc phần tử đơn từ danh sách gốc.

Kiểu trả về	Toán tử truy vấn
Số	Aggregate, Average, Max, Min, Sum, Count, LongCount
Boolean	All, Any, Contains, SequenceEqual
Type	ElementAt, ElementAtOrDefault, First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault, DefaultIfEmpty

```
int[] nums = new int[] { 5, 3, 4, 2 };
int last = nums.Last();
Console.WriteLine(last);
```

Trả về danh sách cùng kiểu với danh sách gốc (IEnumerable)

```
List<Contact> contacts = Contact.SampleData();

// có thể thay thế IEnumerable bằng từ khóa var cho ngắn gọn
IEnumerable<Contact> q = from c in contacts
                           where c.State == "WA"
                           orderby c.LastName, c.FirstName ascending
                           select c;
```

Trả về danh sách có kiểu đã được định nghĩa nhưng khác với kiểu của danh sách gốc. (IEnumerable)

```
List<Contact> contacts = Contact.SampleData();
IEnumerable<ContactName> q2 = from c in contacts
                               select new ContactName
                               {
                                   FullName = c.LastName + ", " + c.FirstName,
                                   YearsOfAge = (DateTime.Now - c.DateOfBirth).Days / 365
                               };

// ContactName class definition
public class ContactName
{
```

```

    public string FullName { get; set; }
    public int YearsOfAge { get; set; }
}

```

Trả về danh sách có kiểu dữ liệu nặc danh (IEnumerable)

```

List<Contact>; contacts = Contact.SampleData();
var q2 = from c in contacts
         select new
        {
            FullName = c.LastName + ", " + c.FirstName,
            YearsOfAge = (DateTime.Now - c.DateOfBirth).Days / 365
        };

```

Trả về danh sách được gom nhóm (IEnumerable)

Trả về một danh sách các dữ liệu được gom theo nhóm.

```

string[] partNumbers = new string[] { "SCW10", "SCW1",
    "SCW2", "SCW11", "NUT10", "NUT1", "NUT2", "NUT11" };

var q = from pn in partNumbers
        group pn by pn.Substring(0,3);

foreach (var group in q)
{
    Console.WriteLine("Group key: {0}", group.Key);
    foreach (var part in group) {
        Console.WriteLine(" - {0}", part);
    }
}

```

Bài 2. Tạo Windows Application project

- Viết class StringExtension chứa các phương thức mở rộng cho lớp String bao gồm các phương thức:
 - String **LTrim(..)** trả về chuỗi sau khi cắt bỏ các khoảng trắng đầu chuỗi
 - String **RTrim(..)** trả về chuỗi sau khi cắt bỏ các khoảng trắng cuối chuỗi
 - String **MTrim(..)** trả về chuỗi sau khi cắt bỏ các khoảng trắng thừa giữa 2 từ trong chuỗi
 - String **PerfectString(...)** trả về một chuỗi sau khi cắt bỏ tất cả các khoảng trắng thừa trong chuỗi, đổi các ký tự đầu từ sang chữ hoa và các ký tự còn lại sang chữ thường
 - String **FWord(..)** trả về từ đầu tiên của chuỗi
 - String **LWord(..)** trả về từ cuối của chuỗi
 - Int **WordCount(...)** trả về số từ có trong chuỗi
 - Boolean **NameCheck(...)**: trả về giá trị true/false cho biết một chuỗi có phải là tên hợp lệ không

- Boolean **EmailCheck(...)**: trả về giá trị true/false cho biết một chuỗi có phải là địa chỉ email hợp lệ không
 - Boolean **PhoneCheck(...)**: trả về giá trị true/false cho biết một chuỗi có phải là số điện thoại hợp lệ không
 - Boolean **DOBCheck(...)**: trả về giá trị true/false cho biết một chuỗi có phải là ngày sinh hợp lệ không
- Tạo form cho phép nhập thông tin của sinh viên bao gồm: MASV, HOTEN, NGAYSINH, PHONE, EMAIL.
 - Dùng các hàm mở rộng để kiểm tra các thông tin đã nhập
 - Biến đổi HOTEN thành perfectstring

BÀI TẬP TUẦN 08, 09, 10 MÔN LẬP TRÌNH WINDOWS (C#)

Chương V : Xử lý lưu trữ dữ liệu với LINQ và 3-Tier Applications

Mục tiêu:

- Hiểu rõ và áp dụng được cách tổ chức dữ liệu mô hình đa tầng (3-Tier application)
- Hiểu được LINQ To Object, LINQ to SQL
- Xây dựng được ứng dụng kết nối cơ sở dữ liệu SQL Server sử dụng LINQ.

Yêu cầu:

- Tất cả các bài tập lưu trong thư mục: T:\MaSV_HoTen\Tuan01\
- Tạo Solution Tuan01 trong thư mục T:\MaSV_HoTen\Tuan01\ trong Visual Studio 2010/2012/2013. Mỗi bài tập là một Project riêng biệt.
- Cuối mỗi buổi thực hành, SV phải nén (.rar hoặc .zip) thư mục làm bài và nộp lại bài tập đã thực hiện trong buổi đó.
- Bài tập bắt buộc của Tuần XX, GV sẽ công bố cuối giờ. Bài tập bắt buộc sẽ gửi vào email giáo viên quy định ví dụ thangvnguyen2001@gmail.com với Subject và tập tin nén kèm theo: **XY_ZK_LTWINDOWS_TUANXX_HOTENSINHVIEN** (trong đó XY là lớp ZK là nhóm XX sẽ là tuần)

I. LINQ to Object

Bài 1. Thực hiện chương trình quản lý thông tin của công ty tương tự như sau:

LINQ Công ty

Mã công ty	AdDV
Tên công ty	Paxeco
Địa chỉ	123 Nguyen Son - Go Vap
Số điện thoại	0133483958

Thêm

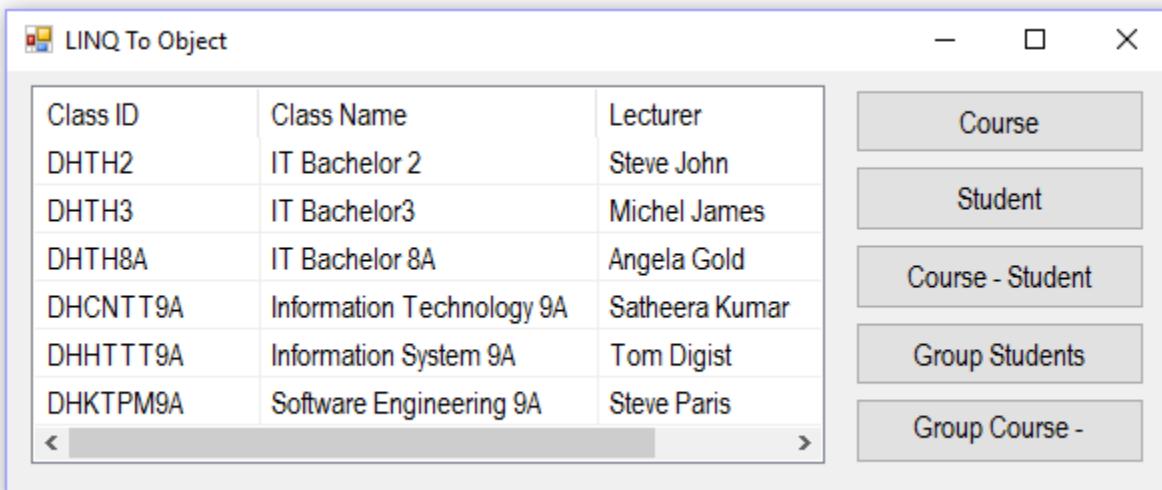
1. Hiển thị danh sách gồm tên CT và số
2. Hiển thị thông tin công ty có mã DV
3. Sắp xếp theo mã CT

TenCT	MaCT	DiaChi
Dịch vụ thương mại Thiên Ân	A1SX	123 Nguyễn Văn Lư...
Cổ phần sản xuất Ngọc Hồng	A2SX	123 Nguyen Thai S...

Exit

Thông tin 1 công ty bao gồm: mã, tên, địa chỉ, số điện thoại. Dùng biểu thức Lambda và các phương thức mở rộng để truy vấn trên tập hợp các công ty.
 Lưu ý ràng buộc dữ liệu khi nhập vào chương trình.

Bài 2. Thực hiện chương trình quản lý thông tin của lớp học và sinh viên tương tự như sau:



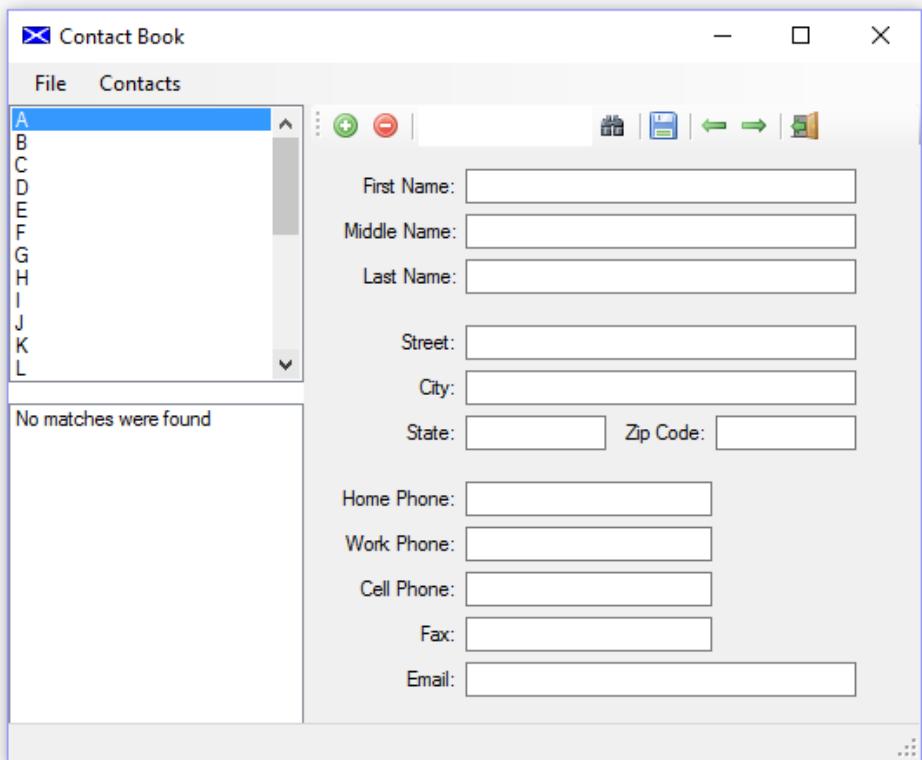
The screenshot shows a Windows application window titled "LINQ To Object". On the left is a grid table with columns: Class ID, Class Name, and Lecturer. The data is as follows:

Class ID	Class Name	Lecturer
DHTH2	IT Bachelor 2	Steve John
DHTH3	IT Bachelor3	Michel James
DHTH8A	IT Bachelor 8A	Angela Gold
DHCNTT9A	Information Technology 9A	Satheera Kumar
DHHTT9A	Information System 9A	Tom Digist
DHKTPM9A	Software Engineering 9A	Steve Paris

On the right side of the window, there is a vertical stack of five rectangular buttons labeled: Course, Student, Course - Student, Group Students, and Group Course -.

Thông tin 1 lớp bao gồm: mã, tên, tên giảng viên quản lý lớp. Thông tin về sinh viên gồm: mã sinh viên, tên SV và mã lớp của sinh viên đó. Dùng biểu thức Lambda và các phương thức mở rộng để truy vấn trên tập hợp các lớp và tập hợp các sinh viên.

Bài 3. Thực hiện chương trình quản lý danh bạ.



Bài 4. Thiết kế form cho phép người sử dụng nhập vào danh sách họ tên (lưu trong List<String> và hiển thị trên ListBox)

- Dùng Linq các câu truy vấn trên danh sách họ tên đã nhập theo các yêu cầu như sau (mỗi câu truy vấn đặt trong một phương thức, tên các phương thức phức được đưa lên một Listbox. Khi người sử dụng chọn một item trong listbox thì thực hiện phương thức tương ứng. Kết quả câu truy vấn hiển thị trong một textbox)
 - Lấy danh sách họ tên có số từ > 2
 - Lấy danh sách họ tên có tên = "Minh". (mở rộng: có tên giống tên được nhập trong một textbox)
 - Lấy danh sách họ tên có chứa ký tự 'a' trong tên (mở rộng: chứa ký tự được nhập trong một textbox)
 - Lấy danh sách họ tên có họ = "Trần". (mở rộng: có họ giống họ được nhập trong một textbox)
 - Lấy danh sách họ tên có họ kết thúc bằng một ký tự giống như ký tự nhập trong một textbox)
 - Lấy danh sách họ tên có tên kết thúc bằng một ký tự giống như ký tự nhập trong một textbox)
 - Lấy danh sách họ tên có họ kết thúc bằng ký tự 'a' trong tên (mở rộng: chứa ký tự được nhập trong một textbox)
 - Lấy danh sách họ tên có họ và tên giống nhau
 - Lấy danh sách họ tên và nhóm theo Họ
 - Lấy danh sách họ tên và nhóm theo Tên
 - Lấy danh sách họ tên và nhóm theo ký tự đầu của Tên
 - Lấy danh sách họ tên và nhóm theo số ký tự của Họ
 - *Xây dựng một phương thức duy nhất để sắp xếp danh sách theo một trong các thứ tự:
 - Họ
 - Tên
 - Tên Họ
 - Họ Tên
 - Chiều dài của họ
 - Chiều dài của tên
 - *Xây dựng một phương thức duy nhất để sắp xếp danh sách theo một trong các thứ tự:
 - Tên Họ
 - Họ Tên

Bài 5. Thực hiện các ví dụ và bài tập: <https://code.msdn.microsoft.com/101-LINQ-Samples-3fb9811b>

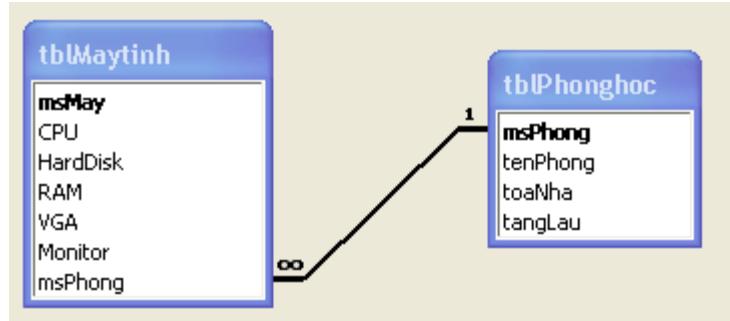
II. LINQ to SQL

Bài 1.(Computer Manager)

Tạo cơ sở dữ liệu bằng SQL Server quản lý các máy tính trong trường học với cấu trúc các bảng như hình sau

tblPhonghoc : Table		
Field Name	Data Type	Description
msPhong	Text	Mã phòng học (size=15)
tenPhong	Text	Tên phòng học (size=100)
toaNha	Text	Tòa nhà (size=20)
tangLau	Number	Tầng lầu mà phòng thuộc về

Field Name	Data Type	Description
msMay	Text	Mã số của máy tính (size=15)
CPU	Text	Đặc tả về CPU (size 100)
HardDisk	Text	Đặc tả về đĩa cứng (size 100)
RAM	Text	Đặc tả về RAM (size 100)
VGA	Text	Đặc tả về card màn hình (size 100)
Monitor	Text	Đặc tả về màn hình (size 100)
msPhong	Text	Phòng đặt máy tính (size=15)



Nhập liệu bảng tblPhonghoc ít nhất 5 mẫu tin, tblMaytinh 10 mẫu tin.

Tạo 1 project bằng ngôn ngữ lập trình C# nhằm thực hiện công việc sau:

- Khi chương trình khởi động sẽ nạp dữ liệu từ bảng tblPhonghoc lên TreeView.
- Khi chọn 1 phòng học ở treeview, danh sách các máy tính thuộc phòng đó sẽ hiển thị trong. Khi chọn 1 dòng trong DataGridView, dữ liệu chi tiết sẽ hiển thị trong các textBox bên dưới:

	msMay	CPU	HardDisk	RAM	VGA	Monitor
05	Intel Core Du...	80GB Sam s...	1024 MB Ki...	onboard	Dell	
06	Intel P4 3.2 G...	120 GB Mat...	2048 MB Ki...	nVidia	Samsung	
07	AMD K7 3200+	160GB Sea...	1024 MB Sa...	Ati	Sony LCD 18	

Thông tin chi tiết

Mã số máy: 06

CPU: Intel P4 3.2 GHz

HardDisk: 120 GB Matrox

RAM: 2048 MB Kingstons

VGA: nVidia

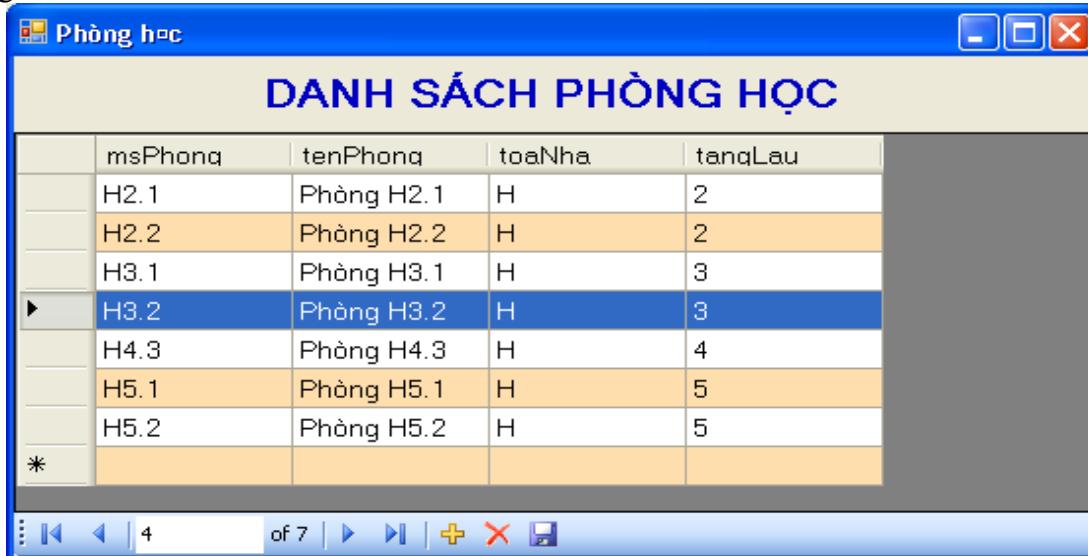
Monitor: Samsung

Phòng: H3.1

Thêm Lưu Xóa Thoát

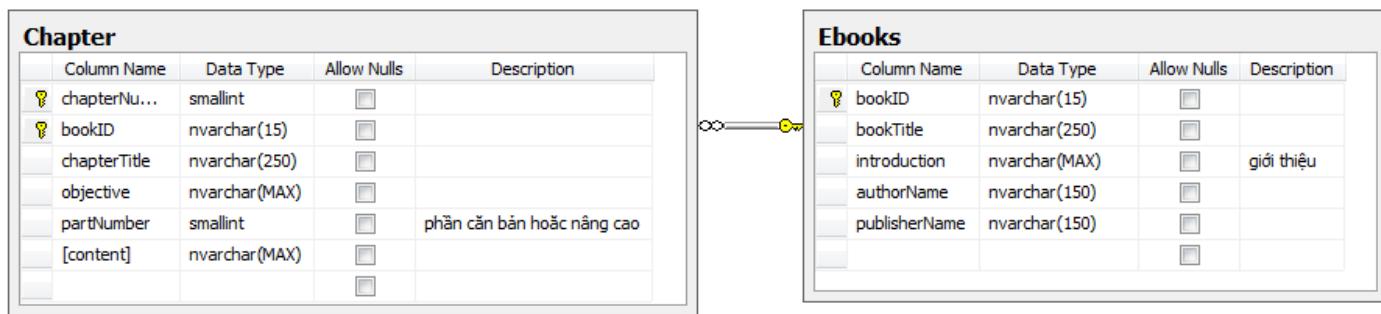
- Các TextBox: mặc định thuộc tính ReadOnly=true. Nút Lưu mặc định Enable=false.

- Khi người dùng nhấn nút Thêm, Nút này sẽ trở thành nút Hủy để hủy bỏ tác vụ nếu cần, nút Lưu sẽ được Enable lên sẵn sàng cho việc lưu mẫu tin. Các nút khác có thuộc tính enable =false. Khi người dùng nhấn nút Hủy, tình trạng trước đó phải được trả lại.
- Khi người dùng nhấn nút Lưu, sau khi kiểm tra dữ liệu hợp lệ sẽ cho phép thêm mẫu tin xuống CSDL và cập nhật lên DataGridView. Sau khi lưu xong phải khôi phục lại các trạng thái trước đó.
- Nút Xóa dùng để xóa mẫu tin đang được chọn sau khi xác minh rằng người dùng có muốn xóa hay không. Nút thoát dùng để thoát ứng dụng.
- Khi nhấn nút “Thêm phòng học” sẽ hiện ra form sau cho phép người dùng cập nhật danh sách phòng học.



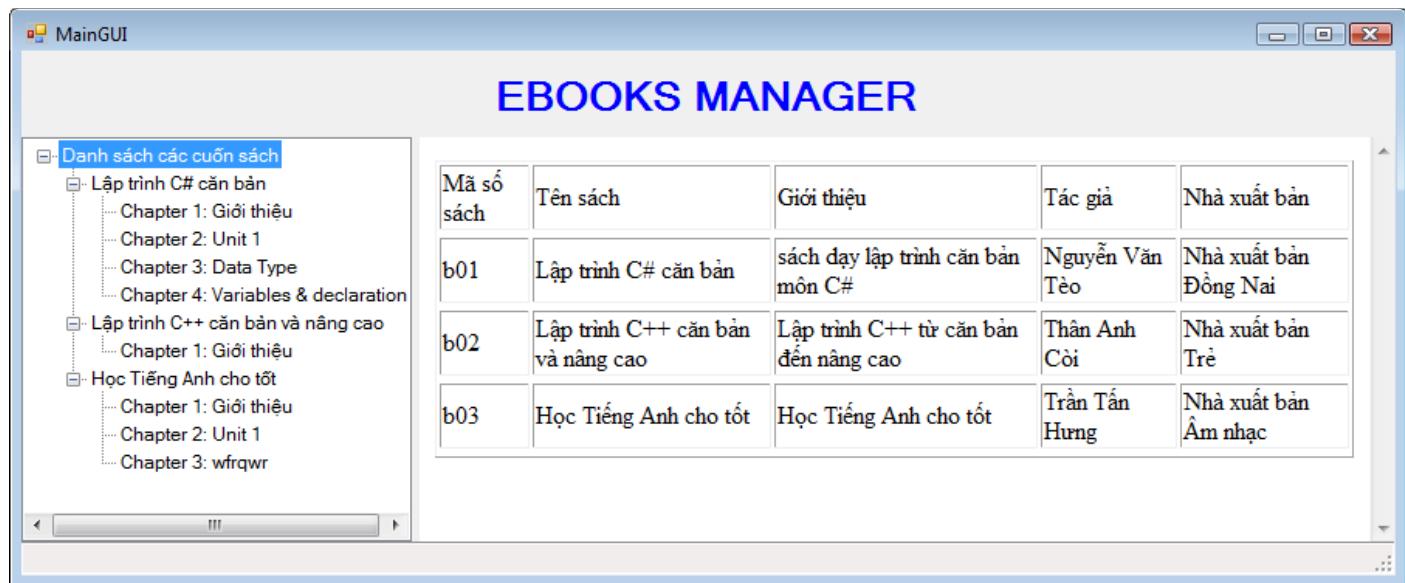
Bài 2. (Ebook Manager)

Để quản lý sách, người ta cho 1 phần lược đồ quan hệ được mô tả như hình sau:

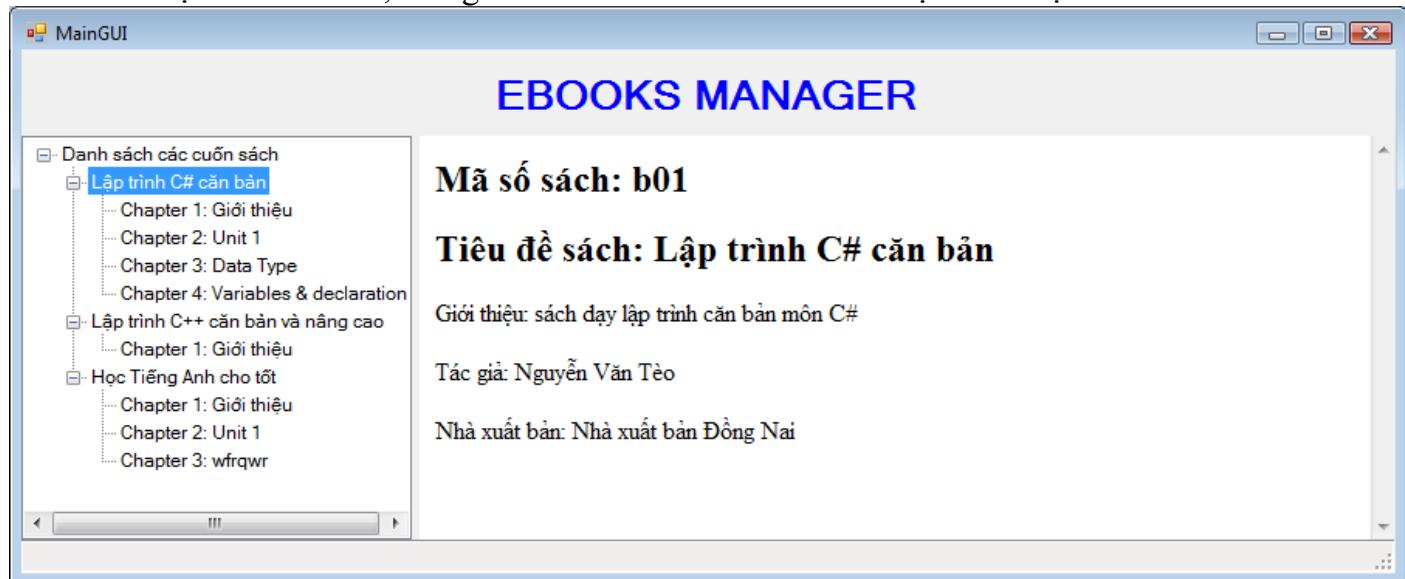


Viết chương trình thực hiện các yêu cầu sau:

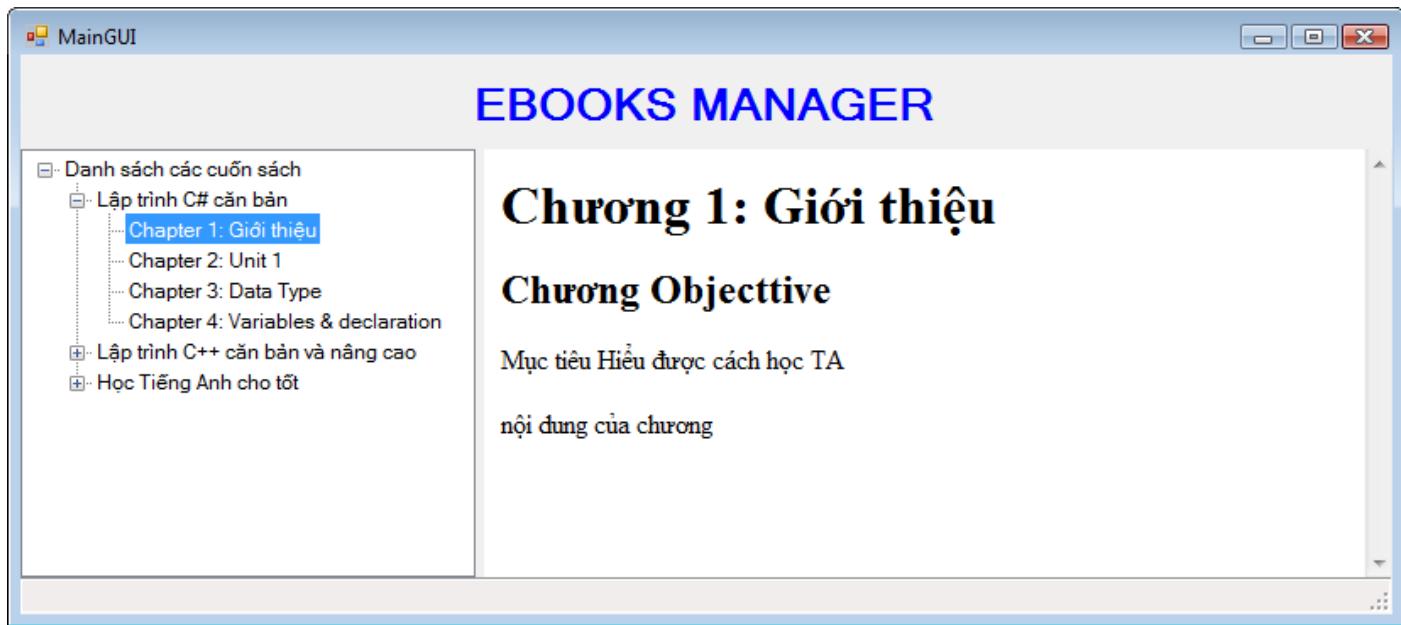
- Khi chương trình khởi động sẽ xây dựng TreeView hiển thị danh sách các cuốn sách cùng các chương thuộc về cuốn sách đó.
- Khi chọn danh sách các cuốn sách, ứng dụng sẽ hiển thị danh sách sách như hình vẽ



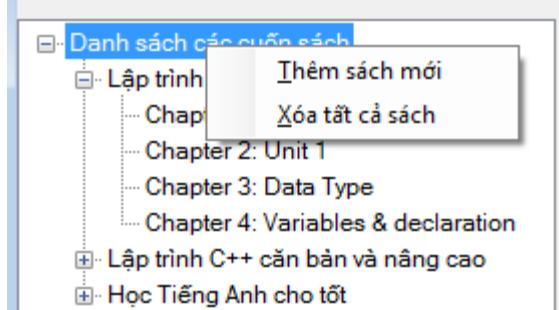
- Khi chọn 1 cuốn sách, thông tin chi tiết của cuốn sách sẽ được hiển thị



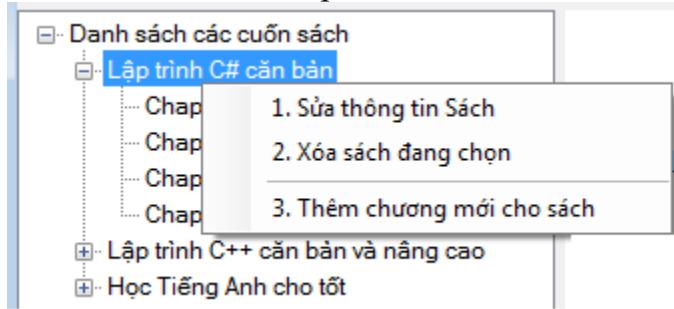
- Khi chọn 1 chương, nội dung của chương sẽ được hiển thị như hình vẽ



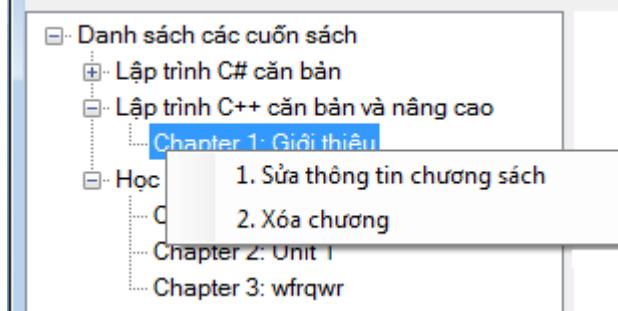
- Khi nhấn chuột phải lên danh sách các cuốn sách, context menu sẽ hiển thị như hình sau



- Khi nhấn chuột phải 1 cuốn sách được chọn, context menu sẽ hiển thị như hình sau



- Khi nhấn chuột phải 1 chương sách được chọn, context menu sẽ hiển thị như hình sau



- Form Thêm/ Sửa thông tin sách

Ebooks_WinForm

BOOK INFORMATIONS

Mã sách:

Tựa đề: Lập trình C# căn bản

Giới thiệu: sách dạy lập trình căn bản môn C#

Tác giả: Nguyễn Văn Tèo

Nhà xuất bản: Nhà xuất bản Đồng Nai

- Form Thêm/sửa chương sách

Chapters_WinForm

CHAPTER INFORMATIONS

Số thứ tự chương: Mã số sách: b02

Tên chương: Giới thiệu

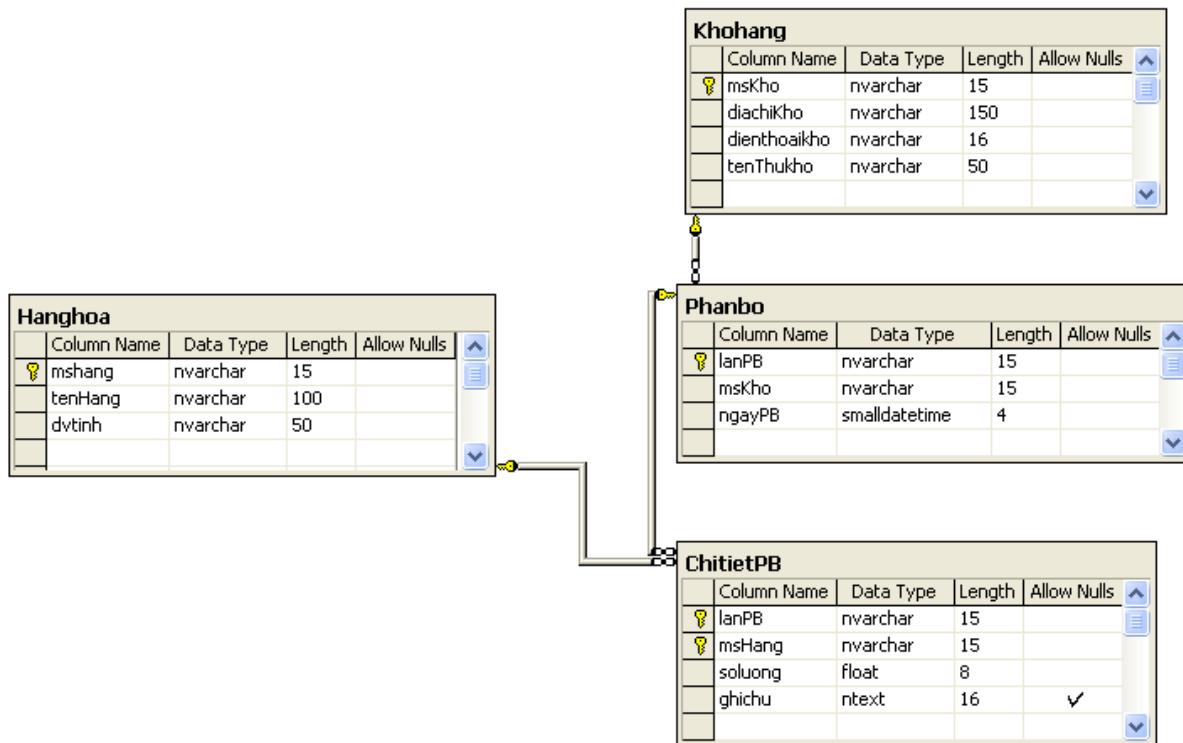
Mục tiêu chương: Hiểu được lập trình C++

Phần:

Nội dung:

Bài 3. (Warehouse Manager)

Để quản lý việc phân bổ hàng hóa về các kho trong 1 công ty, người ta tổ chức CSDL với mô hình ERD được cho như sau:



Hãy xây dựng 1 ứng dụng để quản lý việc phân bổ này.

- Chương trình có 1 form Main theo cơ chế MDI với các menustrip & toolstrip.
- Tạo 2 Form danh mục Kho hàng và danh mục hàng hóa với các chức năng thêm, xóa, sửa.
- Tạo form thực hiện chức năng phân bổ hàng hóa về các kho.
- Tạo form cho phép tìm xem 1 mặt hàng được chỉ định nằm ở những kho nào với số lượng bao nhiêu.
- Tạo form cho phép tìm xem các mặt hàng được phân bổ vào 1 kho trong 1 khoảng thời gian chỉ định.

Phân bổ hàng về kho

Phân bổ hàng hóa

PHÂN BỐ HÀNG HÓA

Lần phân bổ:	20071108_004	
Mã số kho:	KH04	Kho hàng số 4
Ngày phân bổ:	08/11/2007	

STT	Mã số hàng	Tên hàng	Số lượng	Chi chú
01	HH96	Bàn học sinh	10	
02	HH67	Ghế nhựa	100	
03	HH14	Tủ kiêm Hàn Quốc model 2007	75	Hàng dễ vỡ, để ở sân sau kho
04	HH54	Máy nước nóng	8	

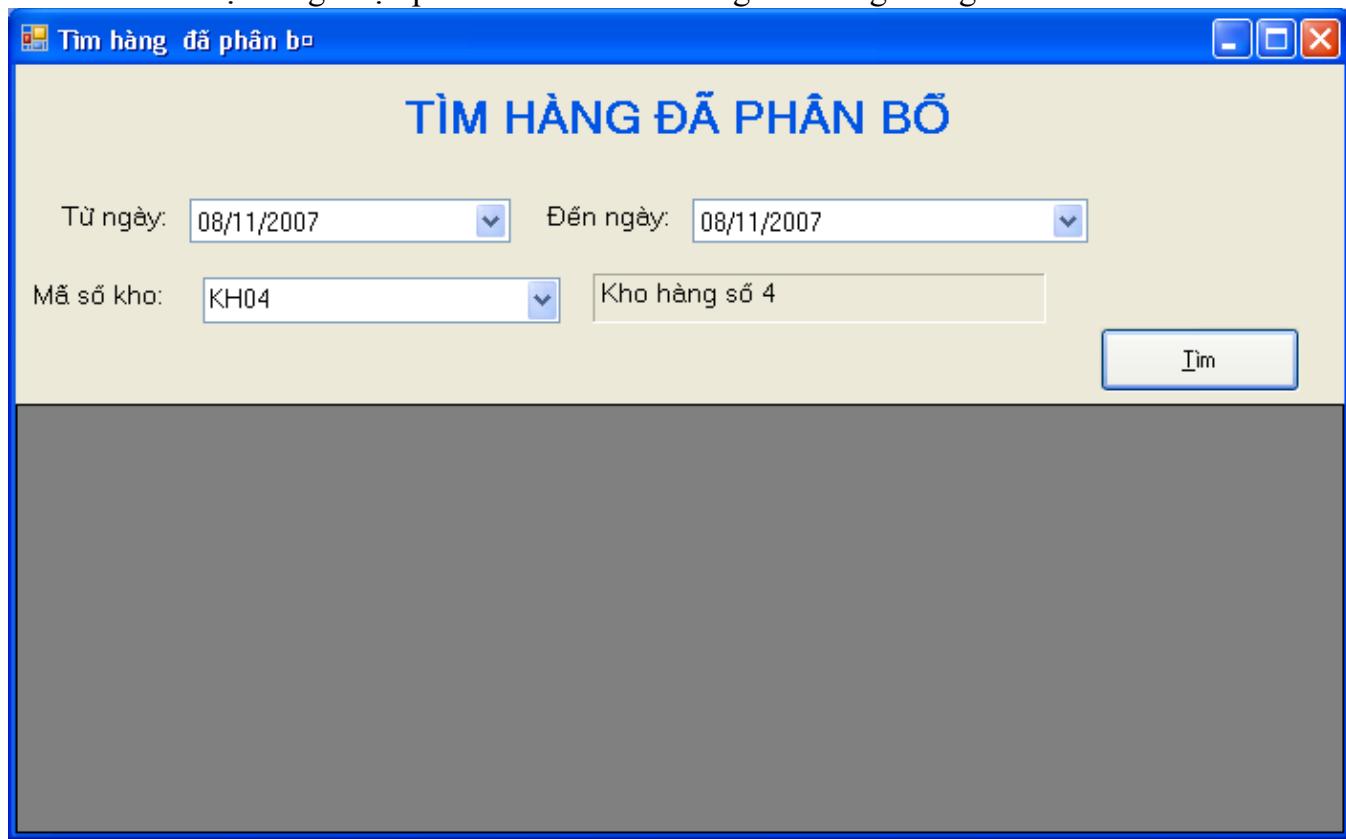
Tìm mặt hàng

Tìm hàng trong kho

TÌM HÀNG TRONG KHO

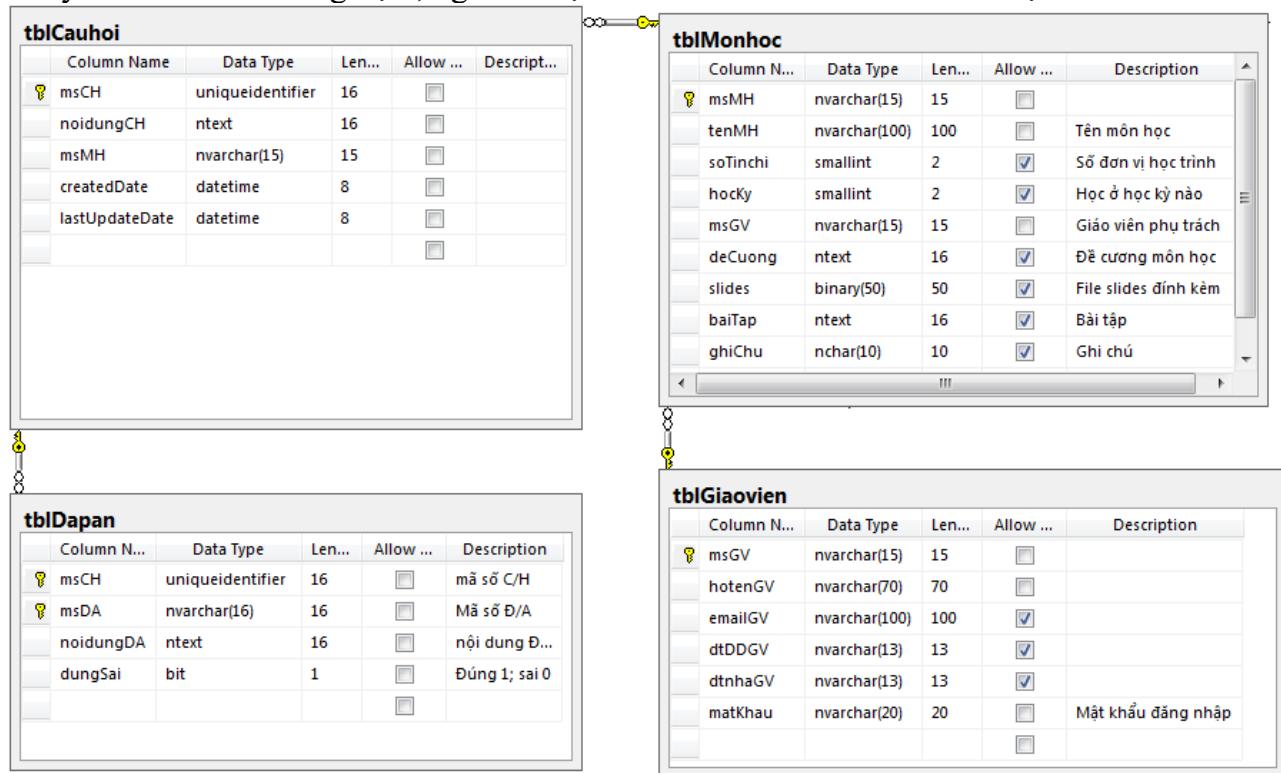
Mã số hàng:	HH67	Ghế nhựa
-------------	------	----------

Tìm xem các mặt hàng được phân bổ vào 1 kho trong 1 khoảng thời gian:



Bài 4. (Quản lý câu hỏi trắc nghiệm)

Để quản lý các câu hỏi trắc nghiệm, người ta tạo 1 CSDL với mô hình ERD được cho như sau:

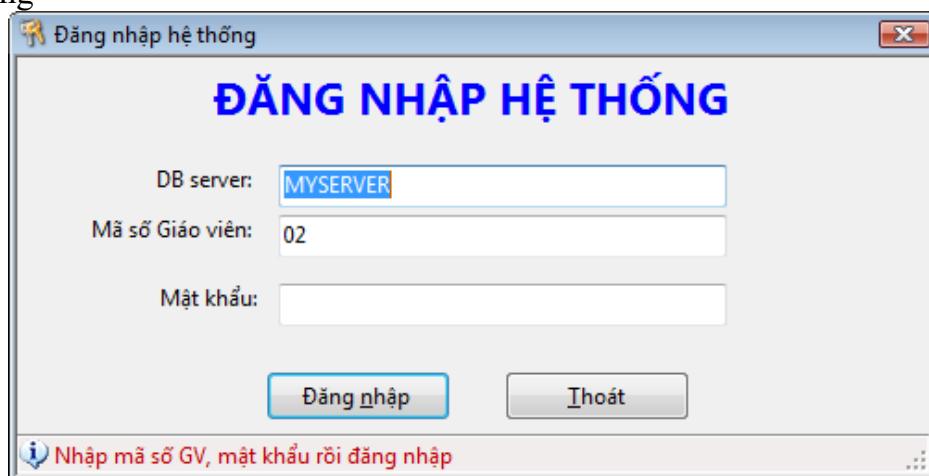


Hãy viết chương trình cho phép quản lý hệ thống này với các chức năng:

- Cho phép đăng nhập hệ thống. Nếu là Administrator thì có quyền tạo lập các mẫu tin trong các danh mục Môn học và giáo viên. Nếu là giáo viên thì chỉ được thao tác với môn học mà mình phụ trách.
- Cho phép nhập, chỉnh sửa câu trắc nghiệm.
- Cho phép ra đề thi trắc nghiệm trên giấy.

Sau đây là 1 số hình ảnh minh họa

Đăng nhập hệ thống



Danh mục môn học

msMH	tenMH	soTinchi	hocKy	msGV	deCuong	slides
MH01	Lập trình HDT 1	4	2	01		
MH02	Lập trình HDT 2	4	3	01		
ThiTNjava	THi tốt nghiệp ...		5	211370363		

Soạn thảo câu trắc nghiệm

The screenshot shows a Windows application window titled "Soạn thảo câu Trắc nghiệm". In the top-left corner, there is a dropdown menu labeled "Môn học: Tin học văn phòng". The main area contains a text editor with a toolbar above it. The text in the editor is:

```
1 Cho biết kết quả của x khi thực thi đoạn chương trình sau:  
2  
3 int x=0;  
4 for(int i=0;i<10;i++)  
5     x+=i;  
6  
7  
8  
9  
10  
11
```

Below the text editor, there is a row of buttons:

- Xem Đáp án đang chọn
- Đáp án 1
- Đáp án 2
- Xóa Đáp án đang chọn
- Thêm đáp án

At the bottom of the window, there are two buttons: "Soạn thảo" and "Preview".

Thêm đáp án cho câu trắc nghiệm

The screenshot shows a Windows application window titled "Đáp án câu TN". The main area contains a text editor with a toolbar above it. The text in the editor is:

```
1 55  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

At the bottom of the window, there is a message in red text: "Đây là câu trả lời ĐÚNG?" (This is the correct answer). There are two buttons at the bottom right: "Cập nhật" (Update) and "Trở về" (Back).

Xem trước(preview) câu trắc nghiệm đang soạn

Cho biết kết quả của x khi thực thi đoạn chương trình sau:

```
int x=0;
for(int i=0;i<10;i++)
    x+=i;
```

a)55
b)50

Đáp án đúng của câu này là: a

Danh sách các câu trắc nghiệm đã soạn

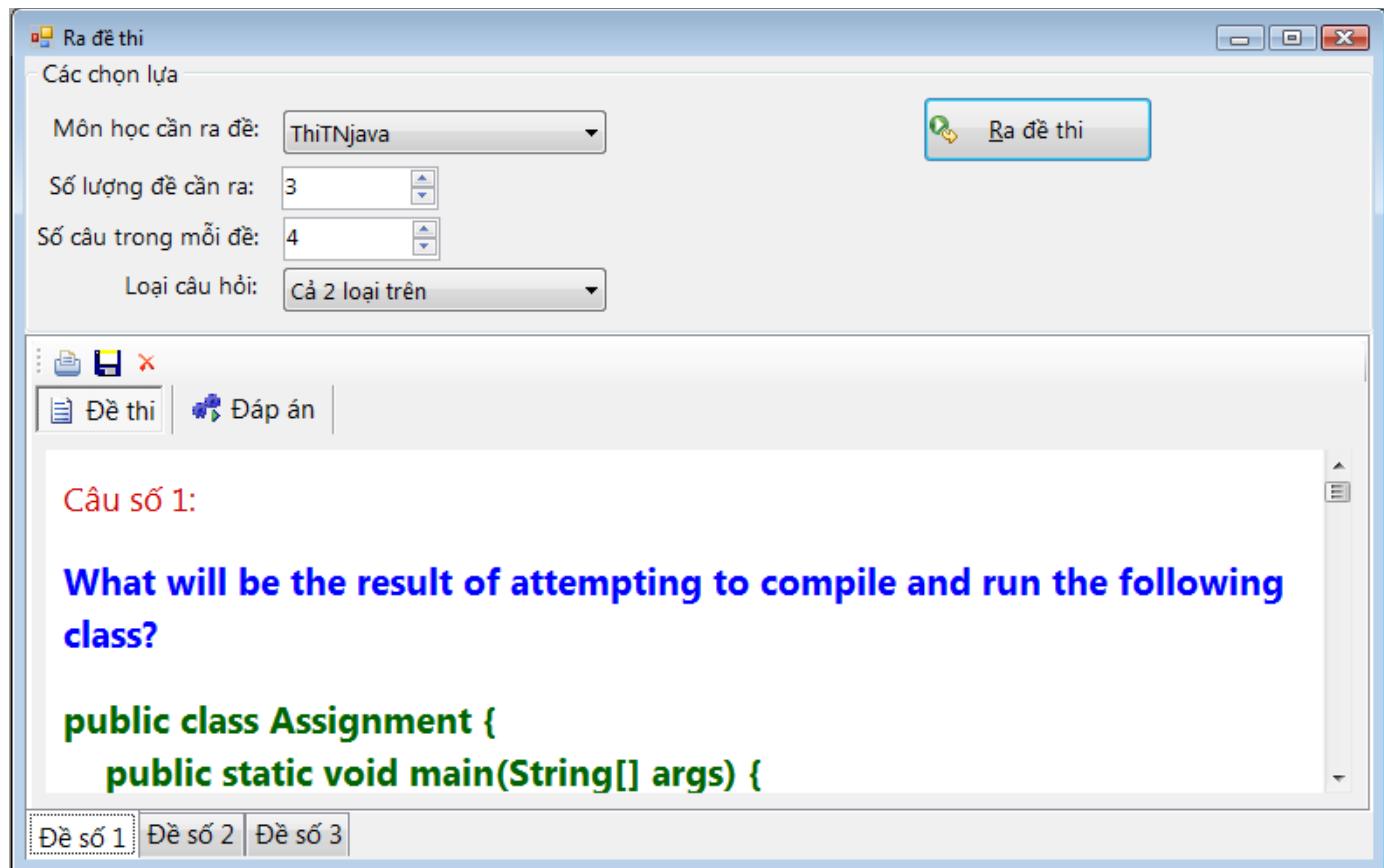
STT	Mã số câu trắc nghiệm
1	ef738dc7-25ed-4bd4-a5b7-0420
2	44813631-02ee-43f5-aad3-06a8
3	e2b09084-e65f-4895-a9d8-0a22
4	7929dd05-3f71-4345-8288-0cd3
5	7b90a76a-3aab-4a25-b9f0-0fd5
6	f3d928a2-5a94-4e3f-a7ee-12ed
7	b84bef88-7b76-40aa-bf3c-1309
8	523f9e3e-e57d-4a72-82ac-13e5
9	4e4e2b1f-ca62-45c5-b230-144f
10	249886c4-4d50-41d0-aae7-1517
11	ca651ead-6117-42e9-8d12-168e
12	14bb1832-c4e2-4e5f-a969-181a
13	2a00a8e9-da38-42d7-9da0-18b
14	ff6b791c-2de2-4c55-8bfd-1d41
15	fcffb151-f4f4-4ebb-a881-1ee37
16	8611afr9_401f_43h7_9a19_2303

What method follow checks whether the JCheckBox jchk is selected?

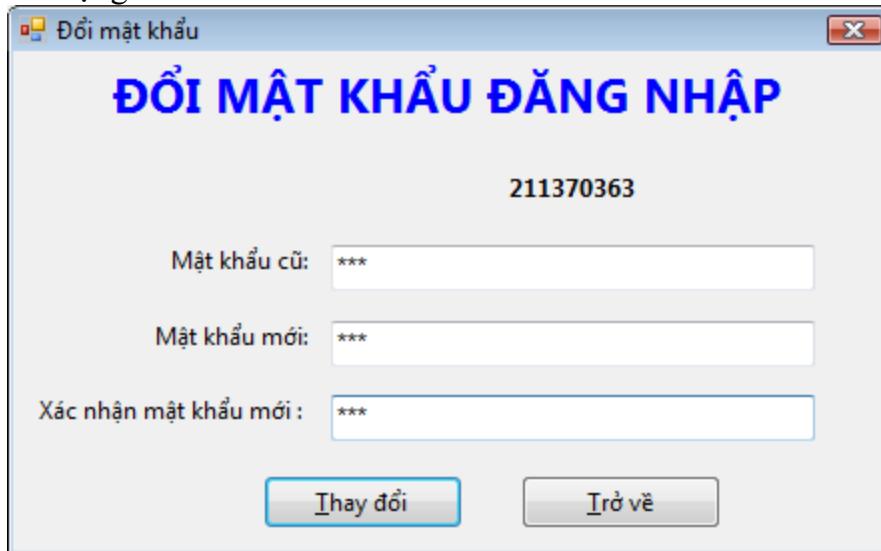
- a) jchk.select();
- b) jchk.getSelected();
- c) jchk.isSelected();
- d) jchk.selected();

Đáp án đúng của câu này là: c

Ra đề thi



Đổi mật khẩu người sử dụng



III. Xây dựng ứng dụng 3-Tier

Bài 1. Tạo cơ sở dữ liệu bằng Access

quản lý thông tin bác sỹ điều trị trong một bệnh

viên, cấu trúc các bảng:

BACSY(MABS, HOTENBS, DIACHIBS, SODIENTHOAI, CHUYENKHOA)

BENHNHAN(SOCMND, HOTENBN, DIACHIBN, MABS)

Danh sách bệnh nhân				
	DiaChiBN	TenBN	MaBS	SocMND
▶	34 Bạch Đằng, Q Bì...	Nguyễn Ngọc Hà	BS02	250358498
	34 Ngô Văn Năm Q1	Hồng Hoa	BS02	250358488
	9/12 Bình Triệu, Q T...	Nguyễn Thanh Hoà	BS02	250358411

Hình 16: Quản lý thông tin bệnh nhân của các BS

Xây dựng chương trình với một số các yêu cầu sau:

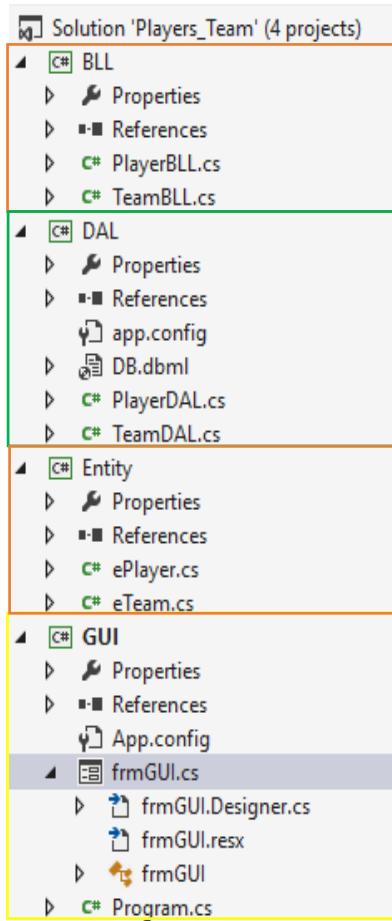
- Khi chương trình khởi động sẽ nạp dữ liệu từ bảng BACSYDIEUTRI lên TreeView.
- Khi chọn 1 bác sỹ điều trị ở TreeView, danh sách các bệnh nhân mà bác sỹ đó điều trị sẽ hiển thị trong DataGridView (hay ListView).
- Khi chọn 1 dòng trong DataGridView (hay ListView), dữ liệu chi tiết sẽ hiển thị trong các TextBox bên dưới.
- Các TextBox: mặc định thuộc tính ReadOnly là True. Nút **Lưu** mặc định Enable=False.

- Khi nhấn nút **Thêm**, Nút này sẽ trở thành nút **Hủy** để hủy bỏ tác vụ nếu cần, nút **Lưu** sẽ được Enable lên sẵn sàng cho việc lưu mẫu tin. Các nút lệnh khác có thuộc tính Enable =False. Khi nhấn nút **Hủy**, tình trạng trước đó phải được trả lại.
- Khi nhấn nút **Lưu**, sau khi kiểm tra dữ liệu hợp lệ sẽ cho phép thêm mẫu tin vào CSDL và cập nhật lên DataGridView (hay ListView). Sau khi lưu xong phải khôi phục lại các trạng thái trước đó.

Hình 17: Form thêm thông tin BS điều trị

Khi nhấn nút “**Thêm bác sỹ điều trị**” sẽ hiện ra form “Thông tin bác sỹ” (như hình trên) cho phép thêm bác sỹ điều trị mới vào hệ thống. Khi nhấn nút **Lưu**, dữ liệu được lưu vào CSDL, nhấn nút **Thoát** trở về form “Quản lý bệnh nhân” và danh sách bác sỹ điều trị phải được cập nhật trên TreeView.

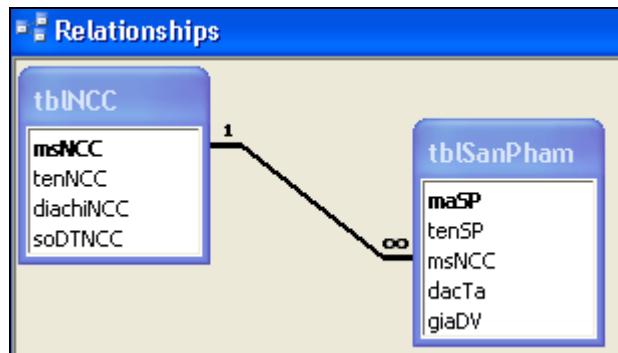
Bài 2. Thực hiện quản lý thông tin đội bóng và các thành viên trong các đội này. Chương trình cho phép hiển thị thông tin các đội bóng trên ComboBox, khi sự kiện chọn trên ComboBox này, thông tin các cầu thủ sẽ hiện trên DataGridView.



Bài 3. Tạo cơ sở dữ liệu bằng SQL Server quản lý sản phẩm với cấu trúc các bảng như hình sau:

tblNCC : Table		
Field Name	Data Type	Description
msNCC	Text	Mã số Nhà cung cấp (size=15)
tenNCC	Text	Tên nhà cung cấp (size=100)
diachiNCC	Text	Địa chỉ NCC (size=150)
soDTNCC	Text	Số ĐT NCC (size =15)

tblSanPham : Table			
	Field Name	Data Type	Description
1	maSP	Text	Mã sản phẩm (size=15)
	tenSP	Text	Tên sản phẩm (size=100)
	msNCC	Text	Nhà cung cấp (size=15)
	dacTa	Memo	Đặc tả của sản phẩm
	giaDV	Number	Giá đơn vị (size=Double)



- Nhập liệu mỗi bảng tblNCC ít nhất 5 mẫu tin, tblSanPham 10 mẫu tin.

- Tạo 1 project bằng ngôn ngữ lập trình C# nhằm thực hiện công việc sau:
- Khi chương trình khởi động sẽ nạp dữ liệu từ bảng tblNCC lên TreeView.
- Khi chọn 1 nhà cung cấp ở treeview, danh sách các sản phẩm thuộc nhà cung cấp đó sẽ hiển thị trong DataGridView.
- Khi chọn 1 dòng trong DataGridView, dữ liệu chi tiết sẽ hiển thị trong các textBox bên dưới như hình vẽ dưới.

Sinh viên hiện thực các nút chức năng của chương trình.

	maSP	tenSP	dacTa	giaDV
sp001	Dầu gội Clear	Dầu gội trị g...	25000	
sp002	Dầu gội Dou...	Dầu gội ngă...	23000	
sp003	Sữa rửa mặt...	Sữa rửa mặt...	32000	

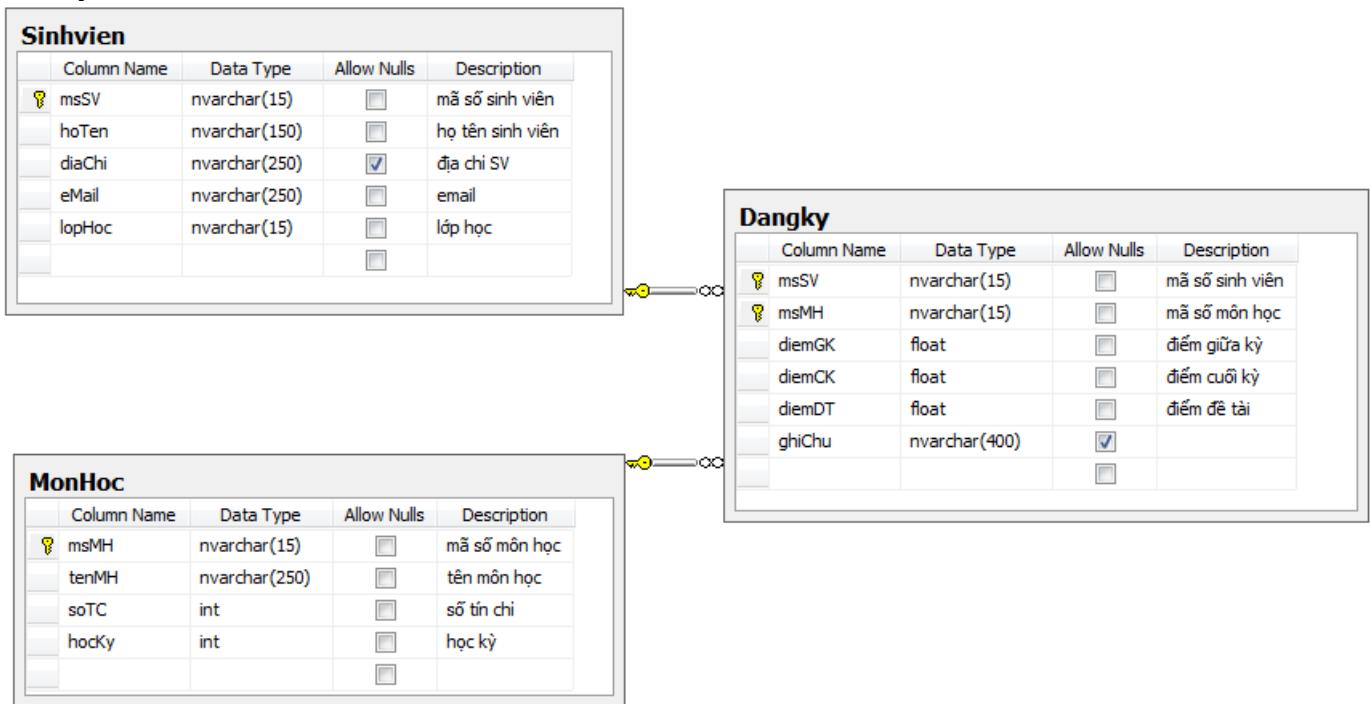
- Các TextBox: mặc định thuộc tính `ReadOnly=true`. Nút Lưu mặc định `Enable=false`.
- Khi người dùng nhấn nút **Thêm**, nút này sẽ trở thành nút **Hủy** để hủy bỏ tác vụ nếu cần, nút **Lưu** sẽ được `Enable` lên sẵn sàng cho việc lưu mẫu tin. Các nút khác có thuộc tính `enable =false`. Khi người dùng nhấn nút **Hủy**, tình trạng trước đó phải được trả lại.
- Khi người dùng nhấn nút **Lưu**, sau khi kiểm tra dữ liệu hợp lệ sẽ cho phép thêm mẫu tin xuống CSDL và cập nhật lên DataGridView. Sau khi lưu xong phải khôi phục lại các trạng thái trước đó.
- Nút **Xóa** dùng để xóa mẫu tin đang được chọn sau khi xác minh rằng người dùng có muốn xóa hay không. Nút **Thoát** dùng để thoát ứng dụng.
- Khi nhấn nút **Thêm nhà cung cấp**, hiển thị form sau cho người dùng cập nhật các mẫu tin của bảng `tblNCC`

Danh sách các nhà cung cấp

	msNCC	tenNCC	diachiNCC	soDTNCC
▶	NCC002	Unilever	1118 Pasteur Q3 Tp HCM	+84 08 657423
	NCC001	Protec&Gamble	32SA java Indonesia	+79 87 34 56 78
	NCC003	Dunhill	12CA F7 wall Bir England	
*	NCC004	Kasper	12 Minks Russia	

... ◀ ▶ | 1 of 4 | + - X

Bài 4. Cho 1 phần CSDL quản lý sinh viên ở phần đăng ký môn học có lược đồ quan hệ được cho như hình vẽ sau:



Viết chương trình bằng NNLT C# theo mô hình 3-tiers thực hiện các công việc sau:

- Cho phép cập nhật dữ liệu lên các bảng Sinh viên, Môn học
- Cho phép Sinh viên đăng ký các môn học trong 1 học kỳ

- c. Cho phép nhập điểm cho sinh viên theo môn học
- d. Cho phép in bảng điểm kết quả của sinh viên trong 1 học kỳ
- e. In danh sách sinh viên dự thi môn học.
- f. In danh sách sinh viên thi lại theo môn học.