**ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN TIN HỌC ỨNG DỤNG**

🙢★🙠

****

**THỰC HÀNH**

**LẬP TRÌNH ỨNG DỤNG MẠNG**

**TN414**

**CÁN BỘ BIÊN SOẠN: TS. PHẠM TRƯƠNG HỒNG NGÂN**

🙣🕮🙡

**03/2021**

**NỘI DUNG THỰC HÀNH**

[BÀI THỰC HÀNH THỨ 1 1](#_gjdgxs)

[BÀI THỰC HÀNH THỨ 2 9](#_30j0zll)

[BÀI THỰC HÀNH THỨ 3 15](#_1fob9te)

[BÀI THỰC HÀNH THỨ 4 22](#_3znysh7)

[BÀI THỰC HÀNH THỨ 5 28](#_2et92p0)

[BÀI THỰC HÀNH THỨ 6 29](#_tyjcwt)

1. 1. **Các bài tập cơ bản C#**
      1. **Viết chương trình nhập vào 1 ngày. Hiển thị Thứ và Tháng của ngày đó bằng tiếng việt.**

ví dụ: nhap ngay 28/05/2013, hiển thị: thu ba, 28, thang nam, 2013.

Hướng dẫn:

- Sinh viên sử dụng lớp DateTime để khởi tạo đối tượng ngày và giờ.

- Dùng mảng để lưu thứ trong tuần.

- Dùng switch case để lưu tháng.

* + 1. **Viết chương trình nhân 2 ma trận.**

Hướng dẫn:

- Đọc giá trị 2 ma trận từ bàn phím.

- Hiển thị 2 ma trận lên màn hình.

- Hiển thị ma trận kết quả phép nhân 2 ma trận đã nhập lên màn hình.

* + 1. **Viết chương trình nhập vào danh sách sinh viên học môn Lập trình ứng dụng mạng, sắp xếp danh sách tăng dần theo ngày sinh sinh viên, hiển thị kết quả sắp xếp ra màn hình.**

Hướng dẫn:

- Tạo lớp SinhVien, với các thuộc tính sau:

+ mssv: kiểu chuỗi là Mã sinh viên

+ ho: kiểu chuỗi là Họ

+ ten: kiểu chuỗi là Tên

+ ngaysinh: kiểu ngày là Ngày sinh

+ malop: kiểu chuỗi là Mã Lớp

- Tạo lớp List SinhVien với các thuộc tính và phương thức sau:

\* Thuộc tính:

+ max: kiểu số nguyên là số lương sinh viên có trong danh sách

+ list: kiểu mảng sinh viên là mảng lưu danh sách các sinh viên

\* Phương thức:

+ readList() //đọc danh sách sinh viên.

+ sort() //sắp xếp danh sách tăng theo ngày sinh

+ printList() // hiển thị danh sách ra màn hình

- Tạo mảng lưu trữ độ dài 40 phần tử với kiểu phần tử là SinhVien.

* 1. **Lập trình winform, thao tác với các control**
     1. **Sinh viên thiết kế form với các yêu cầu sau:**

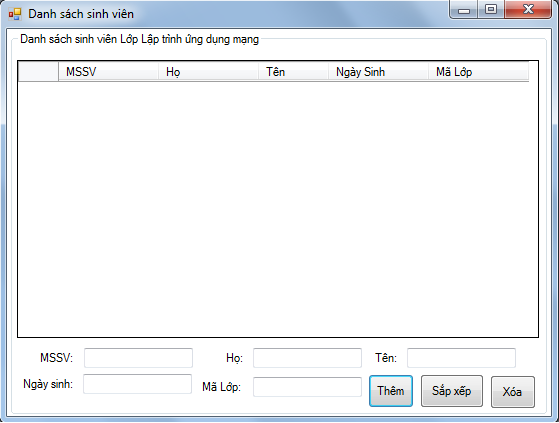
**Yêu cầu:**

- **Khi người dùng click vào nút Thêm thì thêm dòng dữ liệu sinh viên (đã nhập vào các ô textbox) vào datagrid.**

**- Khi người dùng click vào nút Sắp xếp thì thực hiện sắp xếp danh sách sinh viên theo ngày sinh.**

**- Khi click vào nút xóa thì xóa dòng dữ liệu sinh viên đang được chọn khỏi datagrid.**

**- Khi click 1 dòng dữ liệu trên datagrid thì hiển thị thông tin tương ứng vào các textbox.**



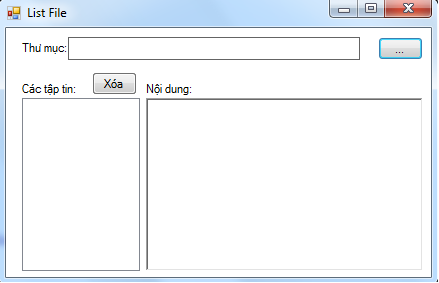
* + 1. **Sinh viên thiết kế form với các yêu cầu sau:**

**Yêu cầu:**

**- Nút ...: cho phép mở hộp thoại Open để người dùng chọn các tập tin (chỉ hiện các tập tin .txt). Các tập tin sau khi được chọn sẽ được đưa vào Danh sách tập tin, mỗi phần tử trong danh sách là tên tập tin.**

**- Nút Xóa: xóa một phần tử đang chọn trong Danh sách tập tin.**

**- Khi nhấn vào tập tin nào trong Danh sách tập tin thì nội dung của tập tin đó được hiển thị tương ứng trong phần Nội dung tập tin.**



* 1. **Bài tập đồ họa với window form**
     1. **Sinh viên thiết kế form theo yêu cầu sau:**

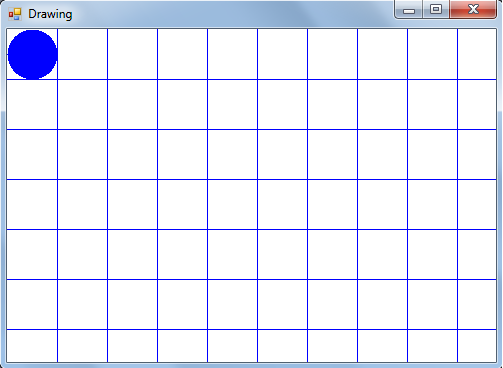
**Khi form hiển thị thì vẽ 1 lưới, bên trong có một hình trong. Hình tròn sẽ di chuyển khi người dùng bấm vào các phím tướng ứng:**

**- D di chuyển sang phải 1 ô.**

**- A di chuyển sang trái 1 ô.**

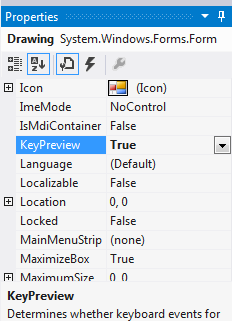
**- W di chuyển lên 1 ô.**

**- S di chuyển xuống 1 ô.**



Hướng dẫn:

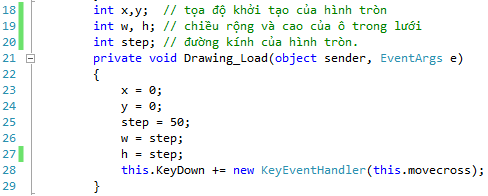
Sinh viên tạo 1 form mới, thiết lập thuộc tính KeyPreview là true.



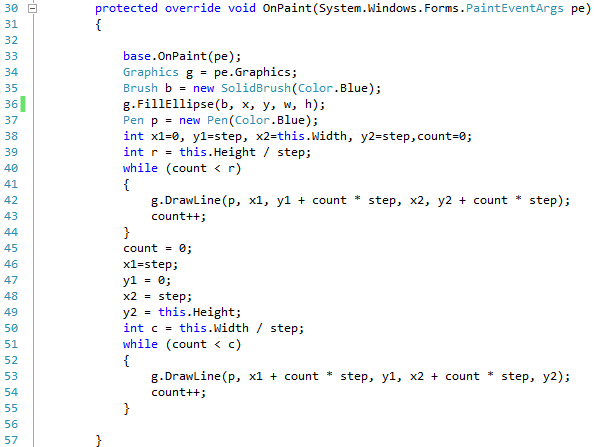
Mở phần code và thực hiện như sau:

- Khai báo các biến và khởi tạo trong phương thức load của form:

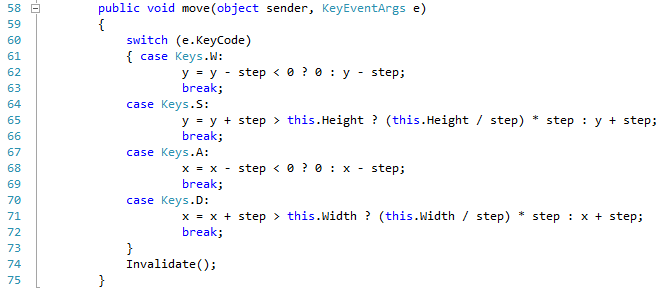
Chú ý: Dòng 28 bắt sự kiện bấm phím trên form với phương thức movecross



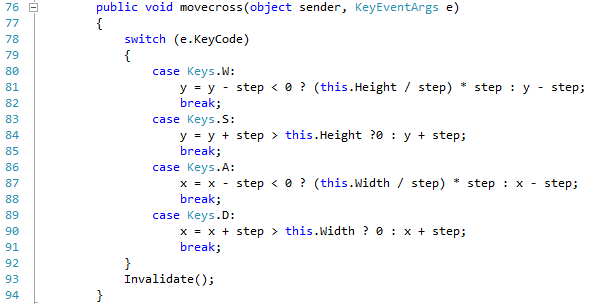
- Override phương thức OnPaint của form để vẽ các đường ngang dọc tạo thành lưới và hình vòng tròn. Sinh viên xem them tài liệu để phần vẽ trong winform.



- Khai báo phương thức move, xử lý khi bấm các phím di chuyển, khi hình tròn di chuyển đến cạnh bên phải, nếu tiếp tục bấm phím D thì hình tròn đứng yên tại chỗ, tương tự cho các hướng còn lại.

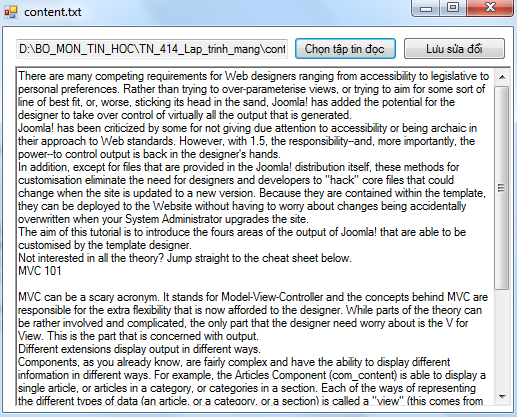


- Khai báo phương thức movecross, xử lý khi bấm các phím di chuyển, khi hình tròn di chuyển đến cạnh bên phải, nếu tiếp tục bấm phím D thì hình tròn sẽ xuất hiện ở cạnh bên trái của cửa sổ, tương tự cho các hướng còn lại.



* + 1. **Sinh viên viết thêm hàm xử lý khi click chuột trái lên 1 ô trong form thì hình tròn sẽ hiển thị tại ô đó.**
  1. **Đọc/ghi tập tin**

Sinh viên thiết kế form với giao diện như sau:



Với các control như sau:

- txtpath là textbox để hiển thị đường dẫn tập tin.

- btndoc là button chọn tập tin để đọc

- btnluu là button lưu nội dung tập tin đã sửa đổi.

- rtb là richtextbox hiển thị nội dung tập tin

Yêu cầu:

- Khi nhấn vào nút “chọn tập tin đọc” thì mở cửa sổ chọn tập tin và chỉ cho phép chọn tập tin txt.

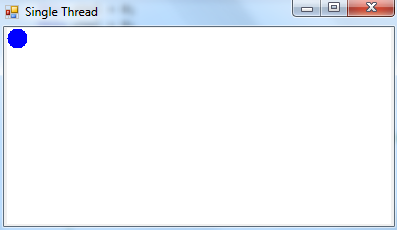
- Sau khi chọn tập tin thì đọc và hiển thị nội dung tập tin đó trong rtb.

- Người dùng có thể thay đổi nội dung tập tin trong rtb.

- Khi nhấn vào nút “Lưu sửa đổi” thì lưu nội dung trong rtb vào tập tin.

* 1. **Viết chương trình sử dụng SingleThread để tạo ra một quả banh chạy trên màn hình.**

Sinh viên thiết kế form với giao diện như sau:

****

Với các control như sau:

- panel là panel hiển thị hình tròn.

Yêu cầu:

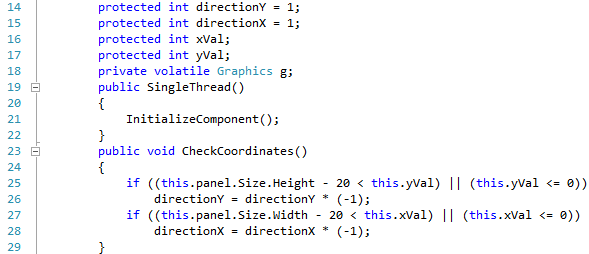
Khi form load thì quả banh xuất hiện góc trên bên trái và di chuyển xuống phía dưới 1 góc 450, va chạm với thành panel thì tiếp tục di chuyển.

Hướng dẫn:

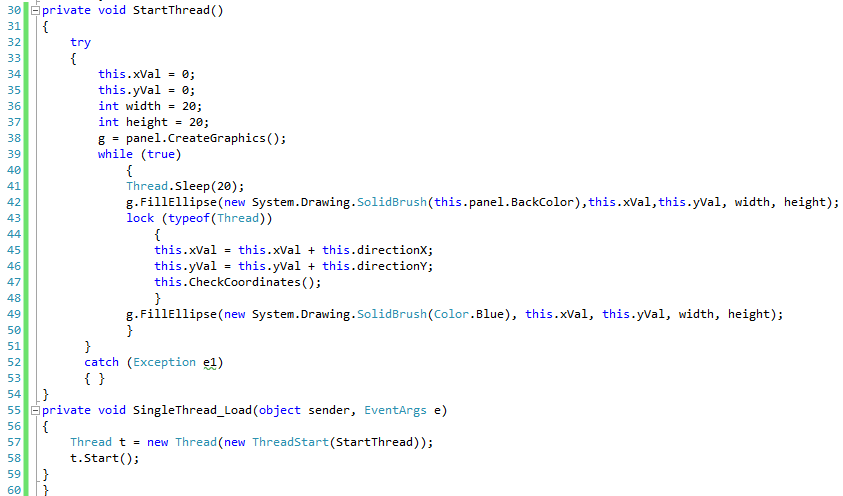
- Khi form load thì tạo một thread mới sử dụng đối tượng đồ họa của panel để vẽ quả bóng.

- Để quả bóng di chuyển thì thay đổi vị trí vẽ hình, chú ý sử dụng sleep để thấy sự di chuyển.

Code xử lý như sau:



Hàm CheckCoordinates: kiểm tra tọa độ hiện tại của quả bóng, khi va chạm thì đổi hướng di chuyển quả bóng.



Hàm StartThread: vẽ quả banh, sử dụng dòng lặp while để di chuyển quả banh.

Hàm SingleThread: khởi tạo Thread vẽ quả banh.

* 1. **Viết chương trình sử dụng MultiThread để tạo ra nhiều quả banh chạy trên màn hình.**

Sinh viên thiết kế form như bài 5

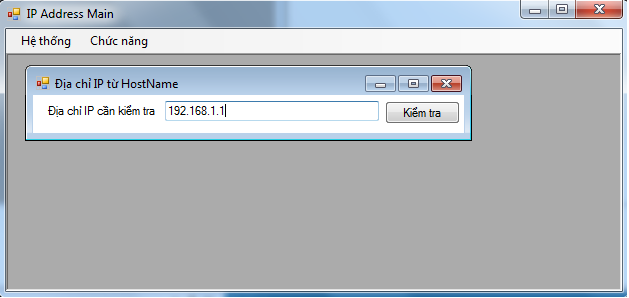
Yêu cầu:

- Khi form load thì các quả banh xuất hiện góc trên bên trái và di chuyển xuống phía dưới 1 góc 450, va chạm với thành form thì tiếp tục di chuyển.

- Mỗi quả banh xuất hiện cách nhau 2 giây.

* 1. **Thực hiện các ví dụ Socket trong bài giảng lý thuyết: ví dụ Client/Server Socket, Network Stream, UDP Socket.**
  2. **Sử dụng IPAddress, IPEndPoint, viết các chương trình:**
     1. **Kiểm tra một tên miền có địa chỉ IP là phiên bản 4 hay phiên bản 6.**

Sinh viên thiết kế form kiểm tra địa chỉ IP như sau:

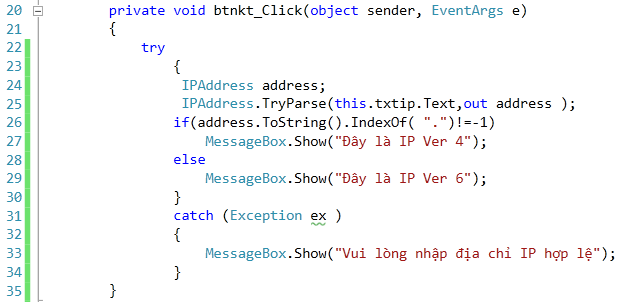


Với các control như sau:

- txtip là textbox để nhập địa chỉ IP.

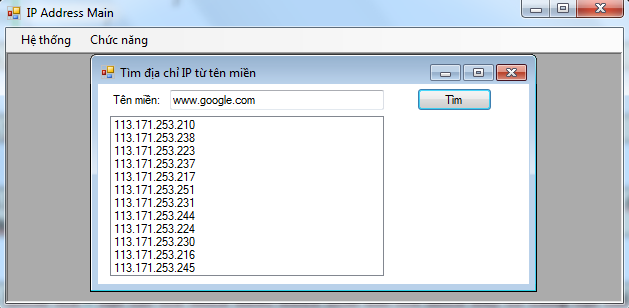
- btnkt là nút kiểm tra

Code xử lý khi bấm nút Kiểm tra như sau:



* + 1. **Tìm tất cả địa chỉ IP của tên miền cho trước.**

Sinh viên thiết kế form tìm địa chỉ IP từ tên miền như sau:

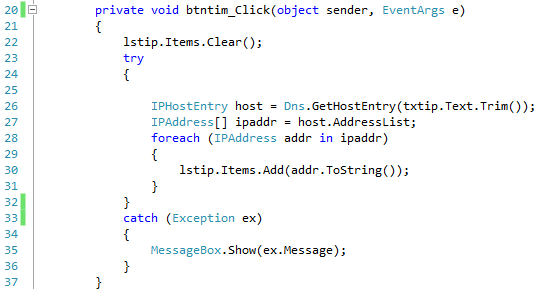


Với các control như sau:

- txtip là textbox để nhập tên miền.

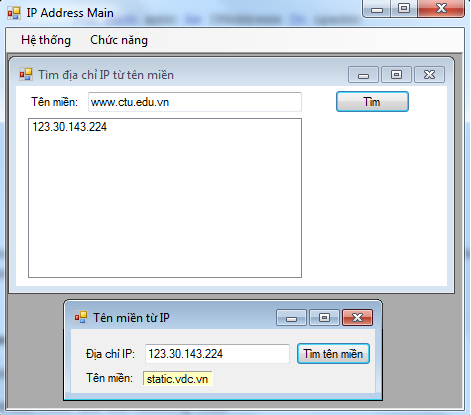
- btntim là nút Tìm

Code xử lý khi bấm nút Tìm như sau:



* + 1. **Tìm tên miền tương ứng (nếu có) của 1 địa chỉ IP.**

Sinh viên thiết kế form tìm Tên miền từ địa chỉ IP như sau:



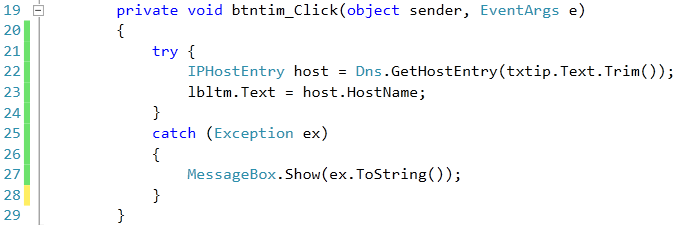
Với các control như sau:

- txtip là textbox để nhập địa chỉ IP.

- lbltm là label hiển thị tên miền tìm được.

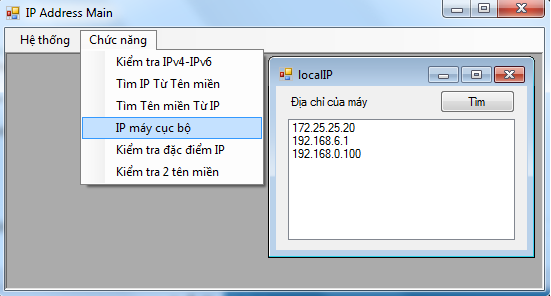
- btntim là nút Tìm tên miền

Code xử lý khi bấm nút Tìm tên miền như sau:



* + 1. **Viết chương trình cho biết địa chỉ IP của máy cục bộ**

Sinh viên thiết kế form tìm Tên miền từ địa chỉ IP như sau:

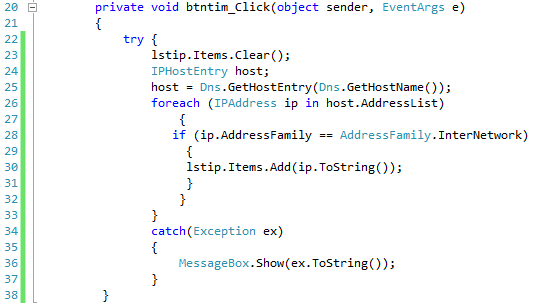


Với các control như sau:

- lstip là ListBox hiển thị các IP tìm được.

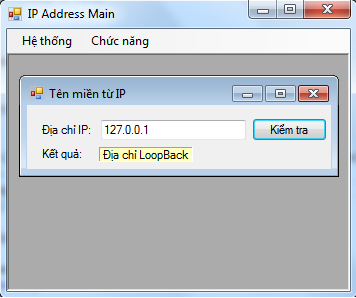
- btntim là nút Tìm

Code xử lý khi bấm nút Tìm như sau:



* + 1. **Kiểm tra đặc điểm của một địa chỉ IP là dạng địa chỉ nào: địa chỉ cục bộ, địa chỉ loopback, địa chỉ multicast.**

Sinh viên thiết kế form kiểm tra địa chỉ IP như sau:



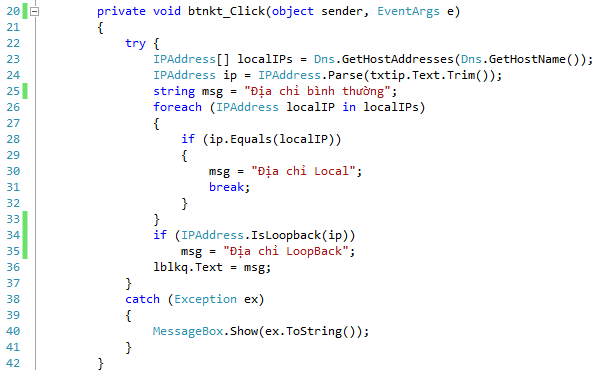
Với các control như sau:

- txtip là TextBox nhập IP cần kiểm tra.

- btntkt là nút Kiểm tra.

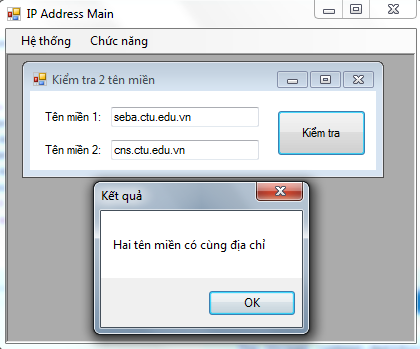
- lblkq là Label hiển thị kết quả.

Code xử lý khi bấm nút Kiểm tra như sau:



* + 1. **Kiểm tra 2 địa chỉ tên miền khác nhau xem hai tên miền này có cùng địa chỉ IP không ?**

Sinh viên thiết kế form kiểm tra địa chỉ IP như sau:



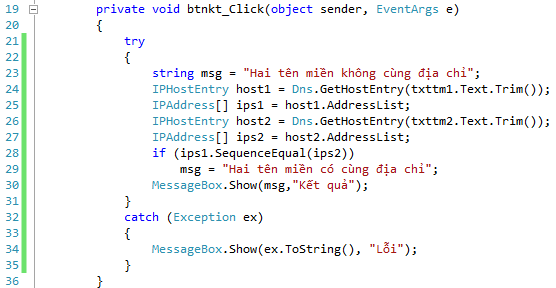
Với các control như sau:

- txttm1 là TextBox nhập tên miền thứ nhất cần kiểm tra.

- txttm2 là TextBox nhập tên miền thứ hai cần kiểm tra.

- btntkt là nút Kiểm tra.

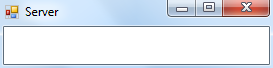
Code xử lý khi bấm nút Kiểm tra như sau:



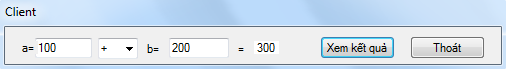
1. 1. **Viết chương trình giao tiếp giữa client và server sử dụng giao thức TCP**

Sinh viên thiết kế các form như sau:

Server: là 1 form trống

****

Client:

****

Với các control sau:

- txta là TextBox số thứ nhất

- txtb là Text Box số thứ hai

- cbo là ComboBox các phép toán +,-,\*,/

- txtkq là TextBox hiển thị kết quả trả về.

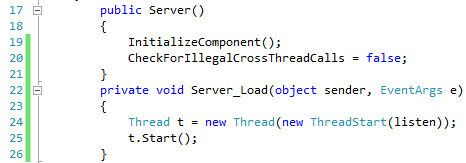
- btnxem là Button Xem kết quả

- btnexitlà Button Thoát

**Cơ chế:** Khi người dùng nhấn vào Xem kết quả, Client truyền hai số nguyên và phép toán (cộng, trừ, nhân, chia) lên Server, Server sau khi nhận được thì thực hiện phép toán giữa hai số nguyên và trả kết quả về cho Client, Client nhận lại kết quả và xuất ra màn hình.

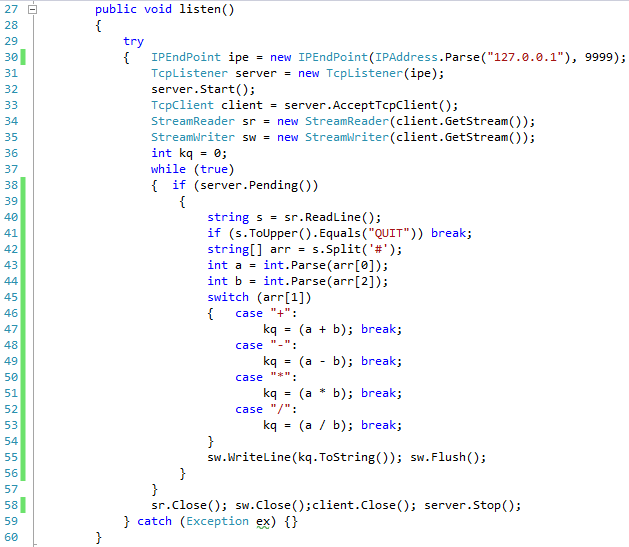
Code xử lý như sau:

Server:

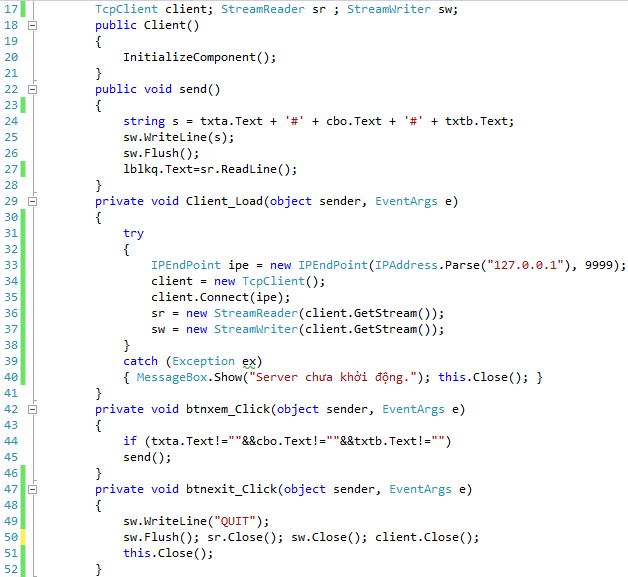
****

**Chú ý:**

***CheckForIllegalCrossThreadCalls = false;***thiết lập chế độ kiểm tra CrossThread là false.

****

Client:

****

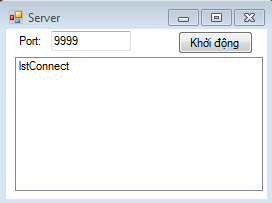
* 1. **Viết chương trình giao tiếp giữa client và server dùng giao thức TCP, (cài đặt bằng Socket và MultiThread cho server)**

Cơ chế hoạt động như bài 1, sử dụng socket tạo kết nối và MultiThread để server phục vụ nhiều client cùng lúc.

* 1. **Viết chương trình gửi/nhận tập tin có dung lượng lớn giữa 2 máy.**

Sinh viên thiết kế form như sau:

Server:

****

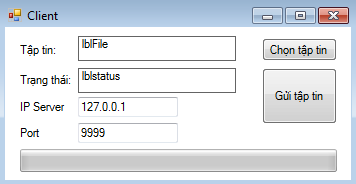
với các controller:

- txtPort là TextBox nhập Cổng làm việc của Server.

- lstConnect là ListBox hiển thị trạng thái của Server.

- btnStart là Button Khởi động.

Client:

****

với các controller:

- lblFile là Label hiển thị tên tập tin sẽ gửi đến Server.

- lblState là Label hiển thị trạng thái của Client.

- txtIP là TextBox nhập địa chỉ IP của Server.

- txtPort là TextBox nhập Cổng của Server.

- progressBar1 là ProgressBar hiển thị tiến trình tải tập tin.

- fdlg là OpenFileDialog để chọn tập tin sẽ gửi.

Cơ chế:

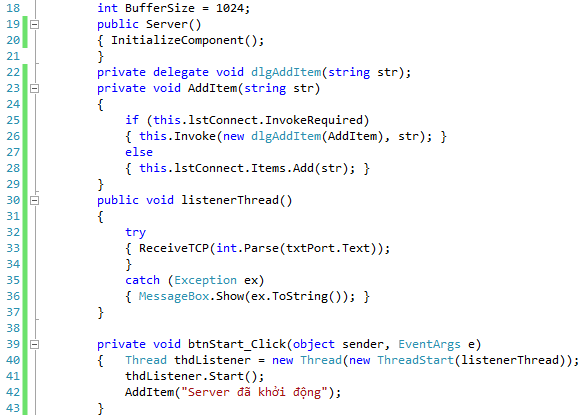
- Server: Người dùng mở chương trình, thiết lập thông số, và bấm nút Khởi động để Server bắt đầu làm việc, server nhận tập tin và lưu lại thành C:\test.abc, không yêu cầu kiểm tra sự tồn tại của tập tin .

- Client: Người dùng mở chương trình, thiết lập thông số, chọn tập tin cần gửi đến server và bấm nút Gửi tập tin để bắt đầu quá trình. Quá trình gửi có thông báo tỉ lệ hoàn thành.

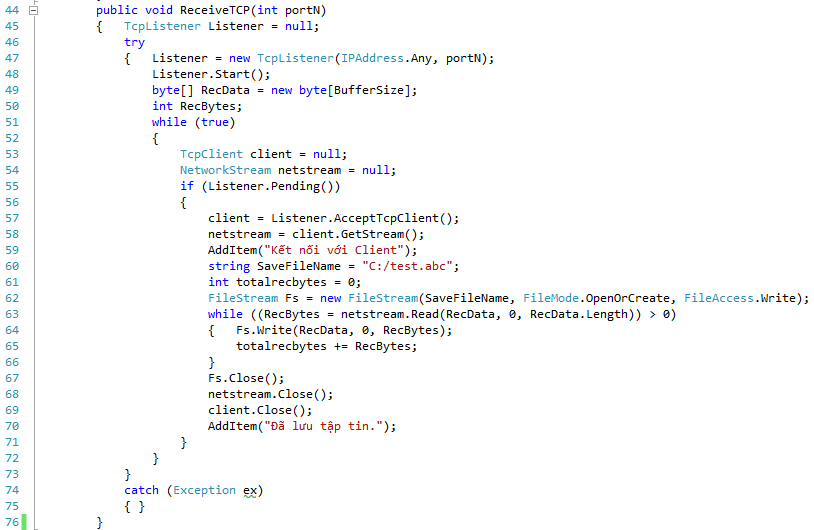
Code xử lý như sau:

Server:

Các nút lệnh

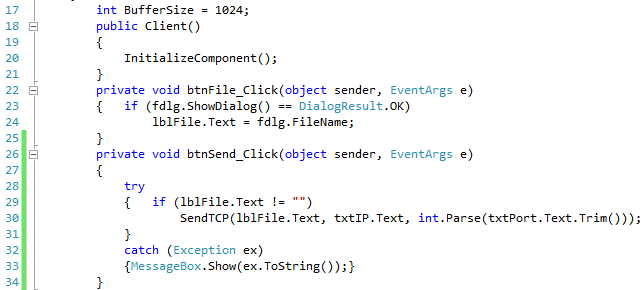
****

Phần xử lý chính của chương trình

****

Client:

Các nút lệnh

****

Phần xử lý chính của chương trình

****

* 1. **Viết chương trình gửi/nhận tập tin có dung lượng lớn giữa 2 máy, Server có thể phục vụ nhiều Client cùng lúc.**

Sinh viên thiết kế các form như bài 3.

**Yêu cầu:**

- Server có thể đáp ứng nhiều kết nối cùng lúc từ Client, do đó cần sử dụng MultiThread.

- Server lưu tập tin vào ổ C theo tên tập tin Client đã gửi.

Hướng dẫn: trước khi truyền dữ liệu tập tin, Client phải gửi tên tập tin đến Server trước.

* 1. **Viết chương trình mô phỏng giao thức FTP như sau:**

- Chương trình client: thực hiện chức năng gởi yêu cầu tập tin tới Server sau đó đợi đáp trả từ server dữ liệu đến tập tin và hiển thị thông tin dữ liệu tập tin lên màn hình và lưu thành tập tin phía máy client nếu tìm thấy. Nếu có lỗi thì hiển thị thông báo.

- Chương trình server: có nhiệm vụ lắng nghe kết nối từ client, tìm và mở tập tin liên quan theo yêu cầu của client gửi về cho client sau đó đóng Thread lại, yêu cầu sử dụng MultiThread để phục vụ cùng lúc nhiều Client.

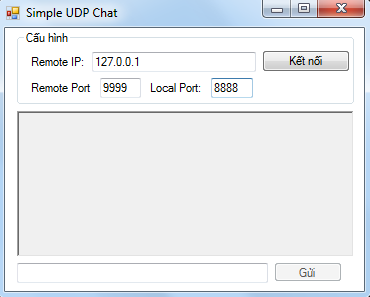
* 1. **Viết Game Caro trực tuyến trên mạng (Yêu cầu viết chỉ cho 2 người chơi với nhau) Game gồm 2 người chơi, một trắng và một đen. Mỗi người chơi ở hai máy khác nhau.**

Luật chơi:

- Mỗi người chơi lần lược chọn con cờ đặt lên bàn cờ của mình và bàn cờ cũng của người chơi thứ 2 cũng được cập nhật theo.

- Người chơi thắng nếu anh ta có 5 quân cờ thẳng hàng ngang, dọc hoặc cắt chéo.

1. 1. **Viết chương trình Text Chat giữa 2 máy sử dụng UDP.**
      1. **Sinh viên thiết kế Form theo yêu cầu sau, sử dụng Socket để gửi nhận dữ liệu**



Với các control như sau:

- txtIPR là TextBox nhập IP của máy sẽ gửi tin nhắn đi.

- txtPortR là TextBox nhập Port của máy sẽ gửi tin nhắn đi.

- txtPortL là TextBox nhập Port của máy cục bộ nhận tin nhắn gửi đến.

- btntKN là nút Kết nối theo các cấu hình đã thiết lập.

- rtxMsg là RichTextBox hiển thị tin nhắn gửi đi và gửi đến.

- txtMsg là TextBox nhập nội dung tin nhắn gửi đi.

- btnGui là nút thực hiện việc gửi tin nhắn.

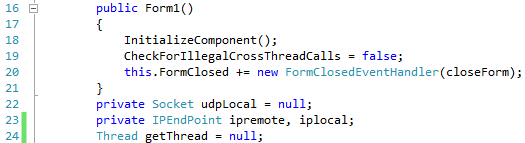
**Chú ý:**

- khi load form thì btnGui.Enable=false.

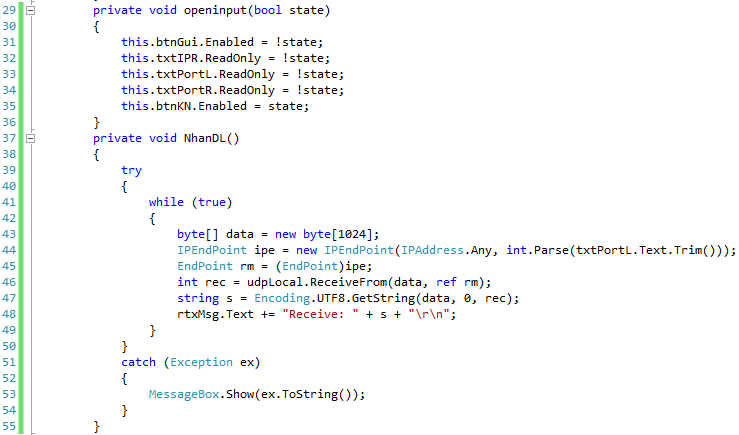
- Khi click nút btnKN thì các TextBox cấu hình sẽ có ReadOnly=true và btnKN.Enable=false; btnGui.Enable=true.

Code xử lý như sau:

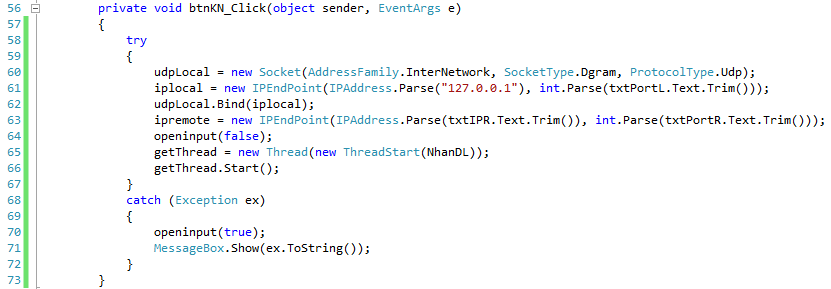
Phần Khởi tạo Form và các thuộc tính sử dụng



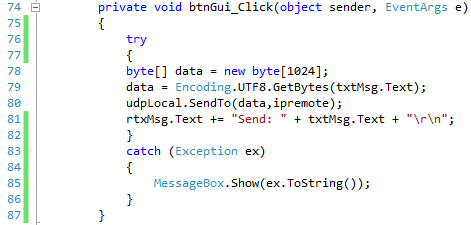
Phần xử lý các input và phương thức nhận dữ liệu



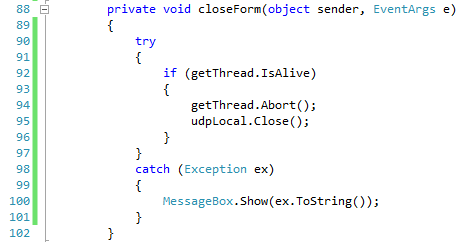
Code xử lý nút Kết nối



Code xử lý nút Gửi



Code xử lý sự kiện đóng form



* + 1. **Yêu chức năng và thiết kế Form như câu a, Sinh viên dụng UDPClient để gửi và nhận dữ liệu.**
  1. **Viết chương trình gửi nhận tập tin văn bản (txt) giữa 2 máy, sử dụng UDP.**

Sinh viên tự thiết kế form.

Cơ chế:

- Chương trình cho phép vừa gửi và vừa nhận.

- Khi gửi tập tin, đầu tiên sẽ gửi tên tập tin trước, sau đó là gửi dữ liệu tập tin.

* 1. **Viết chương trình gửi/nhận tập tin có dung lượng lớn giữa 2 máy, sử dụng UDP.**

Sinh viên thiết kế các form như bài 3 buổi 3.

Cơ chế hoạt động như bài 3 buổi 3.

Yêu cầu: sử dụng UDP thay cho TCP.

Chú ý: Sau khi Server hoàn thành việc nhận tập tin gửi từ Client, Sinh viên mở tập tin xem và ghi nhận kết quả.

* 1. **Viết chương trình Text Chat giữa 2 máy, cho phép gửi tập tin trong quá trình chat.**

Sinh viên tự thiết kế form (tham khảo form bài 1).

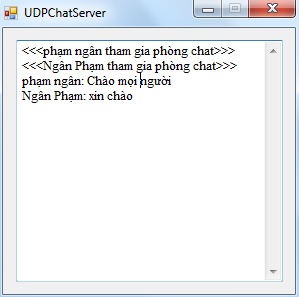
Cơ chế: chương trình cho phép chat giữa 2 máy và gửi tập tin trong quá trình chat.

Hướng dẫn: code textchat tham khảo bài 1. Code gửi tập tin tham khảo bài 3 buổi 3.

* 1. **Viết chương trình Chat Room có quản lý User, sử dụng UDP.**

**Sinh viên thiết kế các form như sau:**

Server:

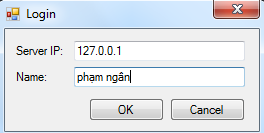


Với các control:

- txtLog là TextBox hiển thị các kết nối và các thông điệp gửi nhận giữa các Client.

Client:

Form đăng nhập



Với các control:

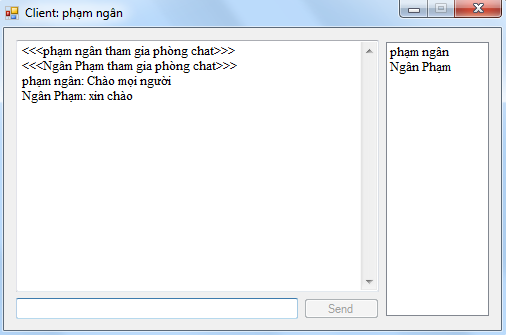
- txtServerIP là TextBox nhập địa chỉ IP Server.

- txtName là tên hiển thị khi chat.

- btnOK là Button OK.

- btnCancel là button Cancel.

From chính



Với các control:

- txtChatBox là TextBox hiển thị nội dung chat.

- txtMessage là TextBox nhập nội dung chat.

- lstChatters là ListBox hiển thị danh sách các thành viên phòng chat.

- btnSend là nút Send.

Cơ chế hoạt động:

Server:

- Khi form load thì sẵn sàng nhận các kết nối đến server.

- Khi có client kết nối thì thêm client vào danh sách các client.

- Khi có tin nhắn gửi đến thì gửi cho tất cả các client có trong danh sách (dạng Broadcast).

- Các sự kiện client kết nối, gửi tin nhắn, thoát khỏi chương trinhg được hiển thị trên khung Log.

Client:

\* Đăng nhập:

- Người dùng nhập địa chỉ IP của server, tên đăng nhập, sau đó nhấn OK.

- Client kết nối đến server, server cập nhật danh sách Client.

\* Xử lý chính:

- Người dùng nhập tin nhắn, sau đó nhấn Send, tin nhắn sẽ gửi đến server. Sau đó tin nhắn sẽ được Server broadcast đến các Client có trong danh sách.

- Client nhận hiển thị tin nhắn.

* 1. **Viết chương trình Chat có quản lý User.**

Thiết kế form như bài số 5.

Cơ chế:

Server: chức năng như bài 5.

Client: ngoài việc chat trong room, chương trình còn cho phép chat trực tiếp 2 người với nhau. Khi người dùng muốn chat trực tiếp với người dùng khác thì click vào tên người dùng trên danh sách, chương trình sẽ mở cửa sổ chat trực tiếp giữa 2 người. Có thể sử dụng code bài 4.

1. 1. **Viết chương trình MultiCast Chat (trong giáo trình lý thuyết).**
   2. **Viết chương trình BroadCast như sau:**

Chức năng chương trình:

- Server gửi dữ liệu đến địa chỉ MultiCast để nhiều client có thể nhận được.

- Các Client hiển thị các thông tin nhận được từ server.

Cơ chế:

- Server đọc từng dòng dữ liệu từ tập tin cho trước.

- Với mỗi dòng dữ liệu, gửi nó đến địa chỉ multicast là 230.0.0.1, mỗi lần gửi Server tạm dừng vài giây (là bội số của 5).

- Tất cả các client tham gia vào MultiCast ở trên sẽ nhận được dữ liệu từ Server gửi đến.

- Client hiển thị nó lên màn hình của mình, trong chương trình được thiết kế chỉ nhận 5 dòng.

* 1. **MultiCast như sau:**

Chức năng chương trình:

- Tham gia vào group của MultiCast.

- Gửi dữ liệu đến địa chỉ MultiCast.

- Nhận dữ liệu từ MultiCast.

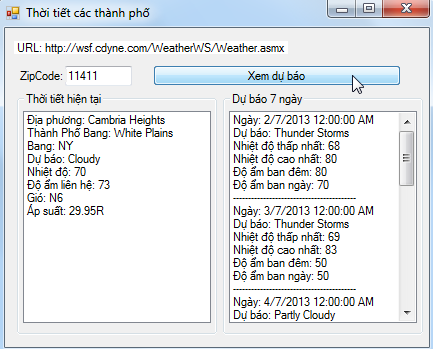
- Hiển thị lên màn hình.

- Chỉ cần các Client tham gia vào group của địa chỉ MultiCast này thì khi có dữ liệu được gửi đến Server thì tất cả Client đều nhận được.

* 1. **Sử dụng ICMP và Raw Socket cài đặt chương trình Ping**
  2. **Sử dụng ICMP và Raw Socket cài đặt chương trình Trace Route**
  3. **Viết chương trình gửi mail sử dụng SMTP**
  4. **Viết chương trình nhận mail sử dụng POP3**

1. 1. **Cài đặt mail client sử dụng POP3**
   2. **Viết chương trình HTTP Server**
   3. **Viết chương trình HTTP Client**
   4. **Viết chương trình FTP Client**
   5. **Viết chương trình lấy thông tin mail Server**
   6. **Viết chương trình lấy thông tin thời tiết một thành phố của Mỹ từ webservice thông qua zipcode.**

Sinh viên thiết kế from:



Cơ chế:

- Sử dụng dịch vụ của dự báo thời tiết Mỹ theo link sau:

http:// www.premis.cz/PremisWS/WeatherWS.asmx

- Sinh viên nhập vào ZipCode một thành phố của Mỹ, sau đó chọn Xem dự báo thì sẽ trả về thông tin thời tiết hiện tại ở ListBox bên trái, thông tin dự báo trong 7 ngày hiển thị ở bên phải.

Hướng dẫn:

- Sinh viên vào http://www.premis.cz/PremisWS/WeatherWS.asmx để xem thông tin về dich vụ.

- Sử dụng hàm GetCityWeatherByZIP để lấy thông tin thời tiết hiện tại.

- Sử dụng lớp WeatherReturn để nhận dữ liệu thời tiết hiện tại trả về.

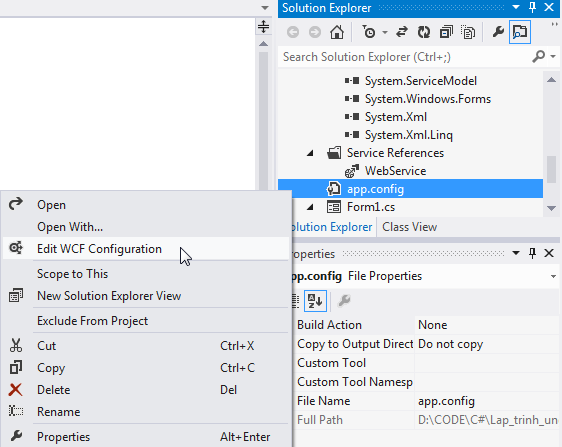
- Sử dụng hàm GetCityForecastByZIP để lấy thông tin dự báo thời tiết.

- Sử dụng lớp ForecastReturn để nhận dữ liệu dự báo trả về.

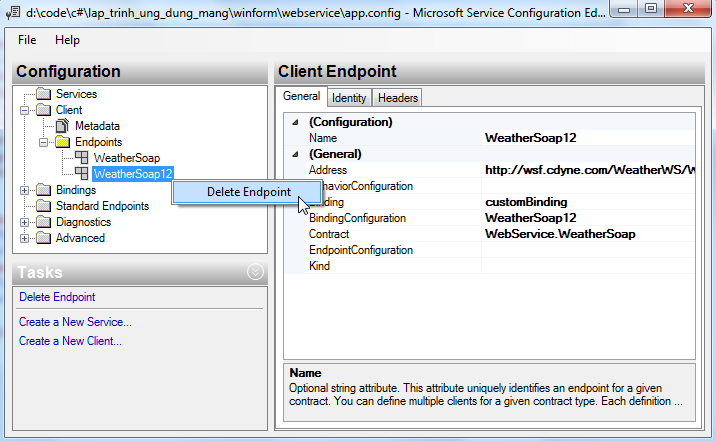
- Thời tiết được dự báo trong 7 ngày.

Chú ý: Nếu gặp lỗi“An endpoint configuration section for contract could not be loaded because more than one endpoint configuration for that contract was found. Please indicate the preferred endpoint configuration section by name.”,

Sinh viên click chuột phải lên tập tin app.config và chọn WCF Configuration như hình.



Mở rông thẻ Enpoints, click chuột phải lên EndPoint cần xóa và chọn Delete Enpoint.



Sinh viên vào xem ZipCode các thành phố Mỹ tại : http://www.zip-area.com

**CHÚ Ý**

* Test telnet trên window 7, nhớ cài đặt telnet client.
* Test POP3 với gmail, nhớ enable POP3 cho tài khoản gmail sử dụng để test.