COMPUTING SUBJECT:	**Socket programming**
TYPE:	Assignment
IDENTIFICATION:	UDPNumberSender
COPYRIGHT:	*Michael Claudius*
LEVEL:	Intermediate
TIME CONSUMPTION:	1-2 hours
EXTENT:	50 lines
OBJECTIVE:	UDP-sockets reversing responsibility of server/client
PRECONDITIONS:	Computer Networks Ch. 2.7, UDPEchoServer
COMMANDS:	

IDENTIFICATION: UDPNumberSender

The Mission
We are going to explore the UDP socket programming creating a server sending out data on the net and a client receiving these data.

Useful C# links

- https://msdn.microsoft.com/en-us/library/system.net.sockets.udpclient(v=vs.110).aspx
- https://msdn.microsoft.com/en-us/library/08h8s12k(v=vs.110).aspx
- https://msdn.microsoft.com/en-us/library/cdas754k(v=vs.110).aspx
- http://stackoverflow.com/questions/20038943/simple-udp-example-to-send-and-receive-data-from-same-socket

Advanced:
- http://www.codeproject.com/Articles/552497/Scalable-UDP-Client-Server

Right now you have a solution for UDP echo-programming: UDPEchoServer & UDEchoClient
These programs you will use as a starting point.

Assignment 1: Application class: UDPNumberSender
Create a new project UDPNumberSender with an application class UDPNumberSender with the usual main method just like UDPEchoClient, but in an infinite loop it is just generating and sending out data; in this case one by one the numbers: 0, 1, 2, .. Thus the usual main method has the following responsibility:

main()
After the socket initialization state an infinite while-loop.
Inside the while-loop
Print out the number
Send a sentence like: `"The number is: " + number`

Tip: Similar to UDPEchoClient, but *does not* receive data.
Compile and run!

Assignment 2: Application class: UDPNumberReceiver
Create a new project UDPNumberReceiver with an application class UDPNumberReceiver with the usual main method just like UDPEchoServer, but in an infinite loop it is just receiving and printing out the received data. Thus the usual main method has the following responsibility:

main()
After the socket initialization state an infinite while-loop.
Inside the while-loop
Receive data
Print out data.

Tip: Similar to UDPEchoServer, but *does not* send any data.

***Remark**: At the moment only one client can receive data; in principle.*
Now we shall see how to broadcast the data to all clients/receivers listening on the port.

Assignment 3: Application class: UDPNumberSenderBroad
In UDPNumberSender create a new application class UDPNumberSenderBroad with the usual main method just like UDPCNumberSender, but now the socket is set up to broadcast the data. Thus the main method has the following responsibility:

main()
Create a UdpClient udpSender, not on a specific port how ☺ ??
Enable broadcast on the the udpSender
Set up an IPEndpoint for broadcast
Inside the while-loop
Use the IPEndpoint to send data

Compile and run! Sender and receiver.

Assignment 4: Application class: UDPNumberReceiver
In UDPNumberReceiver try to change the socket setup so the IPEndPoint will accept data on the port from any IP-address.
Tip: IPAddress.Any