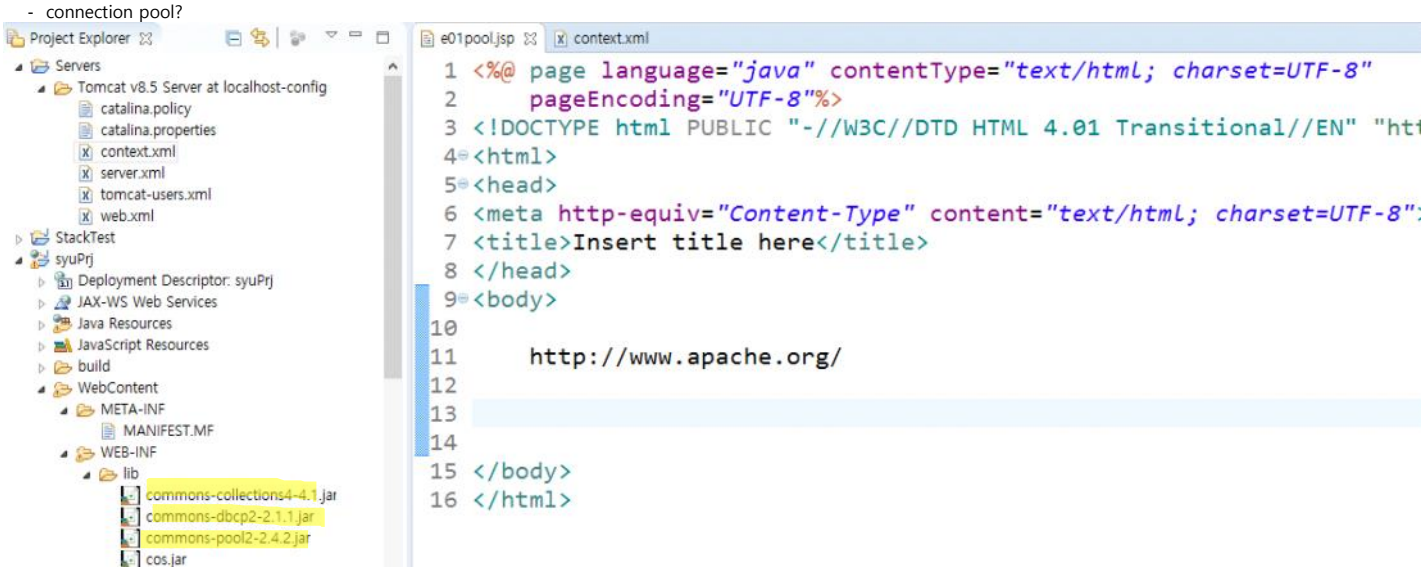
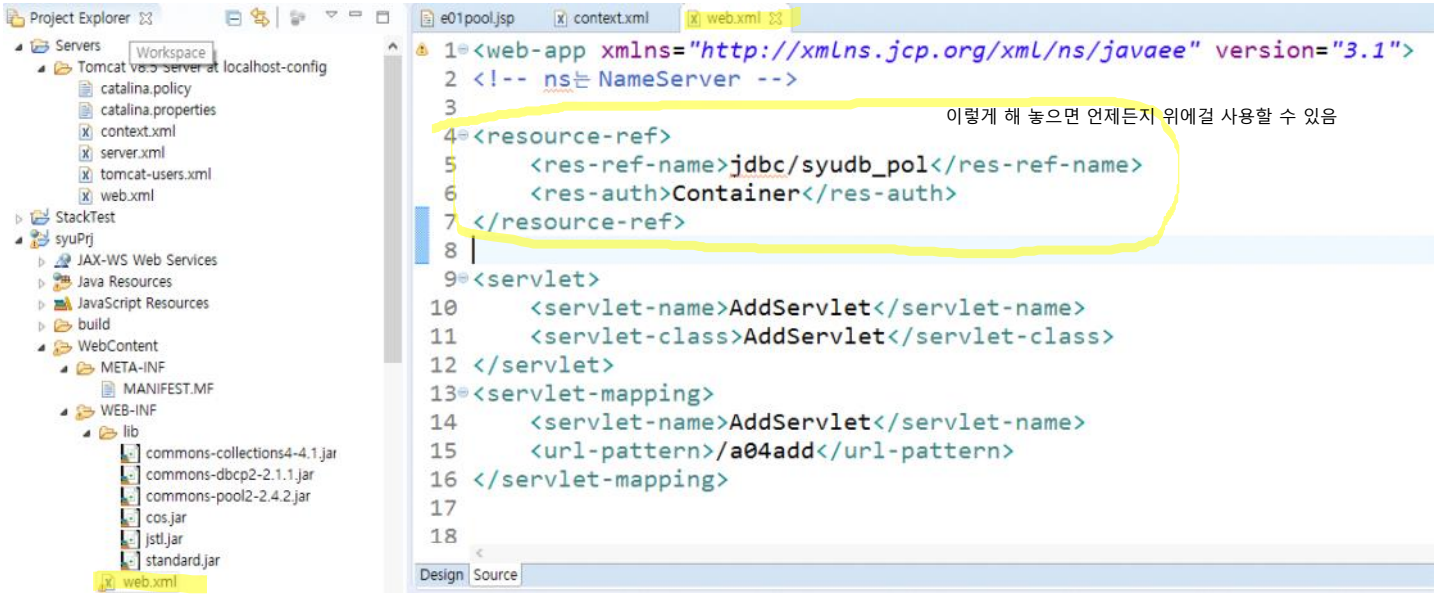
170929

2017년 9월 29일 금요일 오전 9:25

- connection pool?





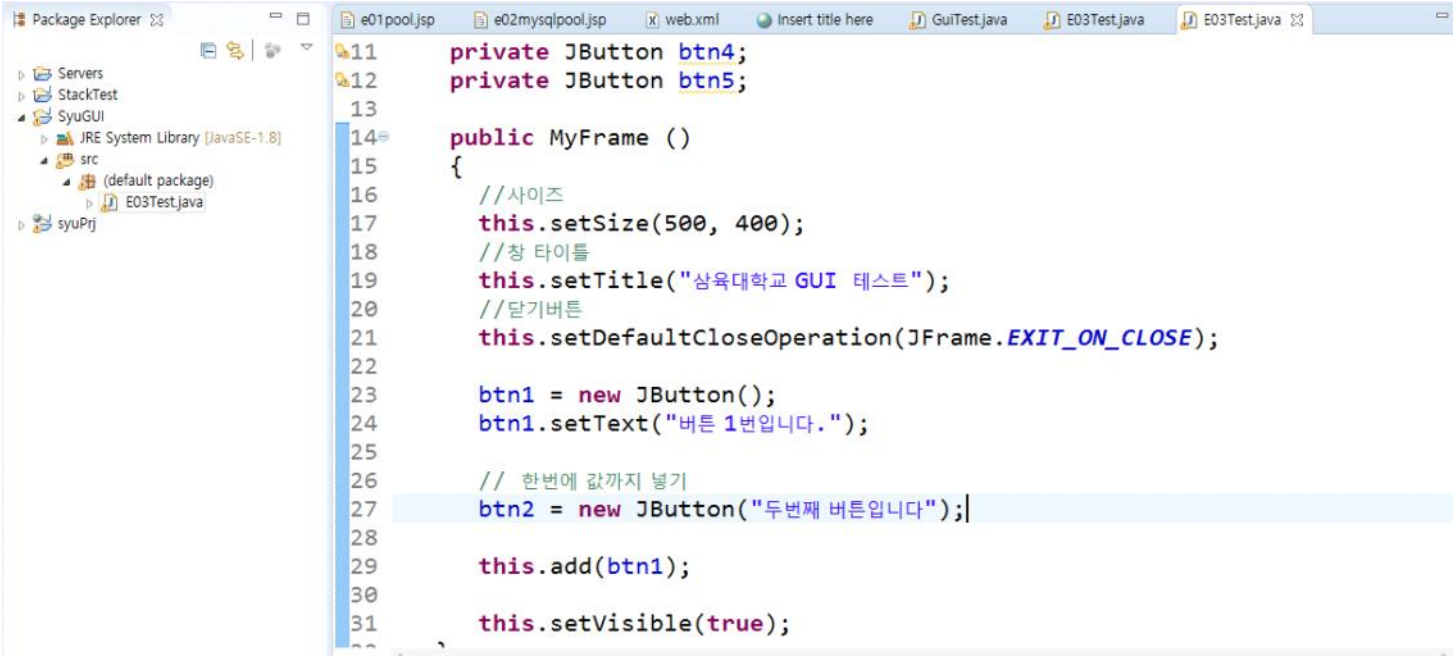
- GUI와 배치관리자

3. FlowLayout Class

3-2. FlowLayout

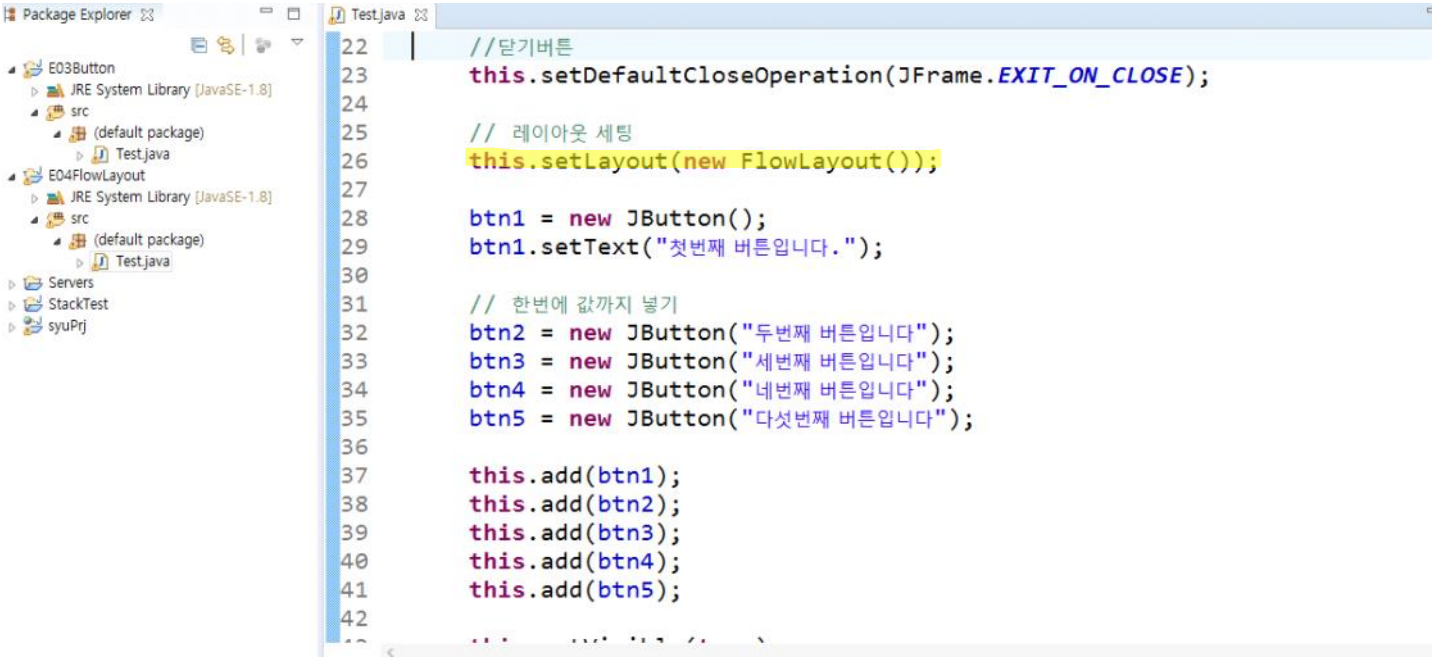
```
1  import java.awt.*;
2  import javax.swing.*;
3
4  class MyFrame extends JFrame {
5      public MyFrame() {
6          setTitle("FlowLayout");
7          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
8
9          JPanel panel;
10         panel = new JPanel();
11         panel.setLayout(new FlowLayout(FlowLayout.CENTER));
12
13         panel.add(new JButton("Button1"));
14         panel.add(new JButton("Button2"));
15         panel.add(new JButton("Button3"));
16         panel.add(new JButton("Btn4"));
17         panel.add(new JButton("Long Button5"));
18         add(panel);
19         pack();
20         setVisible(true);
21     }
22
23
24 }
25 public class FlowLayoutTest {
26     public static void main(String[] args){
27         MyFrame f = new MyFrame();
28     }
29 }
```

// frame 크기를
// component 크기에
// 맞춤

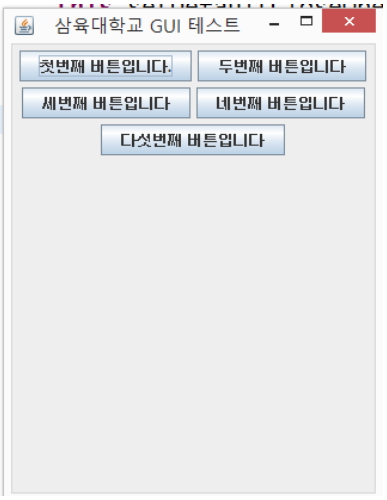


frame이라는 녀석은 기본적으로 border로 설정되어있기 때문에
무언가를(여기에선 버튼) 집어넣어라 하면 센터에다가 집어넣기때문에 이러한 결과가 나오게 됨.

하지만 레이아웃 셋팅을 다시 해주고 나면



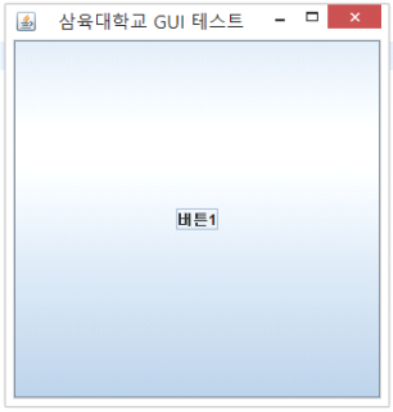
이렇게 Flow형태의 레이아웃으로 변경된 것을 볼 수 있음.



이제는 borderLayout할거임

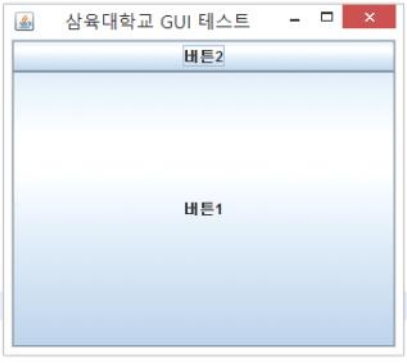
Test.java

```
18        //사이즈
19        this.setSize(500, 400);
20        //창 타이틀
21        this.setTitle("삼육대학교 GUI 테스트");
22        //닫기버튼
23        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24
25        // 레이아웃 세팅
26        this.setLayout(new BorderLayout());
27
28        btn1 = new JButton();
29        btn1.setText("버튼1");
30
31        // 한번에 값까지 넣기
32        btn2 = new JButton("버튼2");
33        btn3 = new JButton("버튼3");
34        btn4 = new JButton("버튼4");
35        btn5 = new JButton("버튼5");
36
37        this.add(btn1, BorderLayout.CENTER);
38
39        this.setVisible(true);
```



Test.java

```
21        this.setTitle("삼육대학교 GUI 테스트");
22        //닫기버튼
23        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24
25        // 레이아웃 세팅
26        this.setLayout(new BorderLayout());
27
28        btn1 = new JButton();
29        btn1.setText("버튼1");
30
31        // 한번에 값까지 넣기
32        btn2 = new JButton("버튼2");
33        btn3 = new JButton("버튼3");
34        btn4 = new JButton("버튼4");
35        btn5 = new JButton("버튼5");
36
37        this.add(btn1, BorderLayout.CENTER);
38        this.add(btn2, BorderLayout.NORTH);
39
40        this.setVisible(true);
41    }
```



```java
        // 레이아웃 세팅
        this.setLayout(new BorderLayout());

        btn1 = new JButton();
        btn1.setText("버튼1");

        // 한번에 값까지 넣기
        btn2 = new JButton("버튼2");
        btn3 = new JButton("버튼3");
        btn4 = new JButton("버튼4");
        btn5 = new JButton("버튼5");

        this.add(btn1, BorderLayout.CENTER);
        this.add(btn2, BorderLayout.NORTH);
        this.add(btn3, BorderLayout.WEST);
        this.add(btn4, BorderLayout.SOUTH);
        this.add(btn5, BorderLayout.EAST);

        this.setVisible(true);
    }
```

이번엔 Gridlayout임

```java
        // 레이아웃 세팅
        this.setLayout(new GridLayout(0,3));

        btn1 = new JButton();
        btn1.setText("버튼1");

        // 한번에 값까지 넣기
        btn2 = new JButton("버튼2");
        btn3 = new JButton("버튼3");
        btn4 = new JButton("버튼4");
        btn5 = new JButton("버튼5");

        this.add(btn1, BorderLayout.CENTER);
        this.add(btn2, BorderLayout.NORTH);
        this.add(btn3, BorderLayout.WEST);
        this.add(btn4, BorderLayout.SOUTH);
        this.add(btn5, BorderLayout.EAST);

        this.setVisible(true);
    }
```

이번엔 Calculator를 만들거임

..

이번엔 Event를 처리

event란 사용자의 반응에 대한 처리를 말한다.

```java
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;

class MyFrame extends JFrame implements ActionListener
{
    private JButton btn1;
    private JButton btn2;
    private JButton btn3;
    private JButton btn4;
```

```
16      private JButton btn2;
17      private JButton btn3;
18      private JButton btn4;
19      private JButton btn5;
20      private JButton btn6;
```

Test.java Test.java Test.java

```
110             btnPanel.add(btn0);
111             btnPanel.add(btnEqual);
112             btnPanel.add(btnCE);
113             btnPanel.add(btnDivide);
114
115         this.add(btnPanel, BorderLayout.CENTER);
116
117
118         this.setVisible(true);
119     }
120
121     // method 를 만들 예정
122     @Override
123     public void actionPerformed(ActionEvent arg0) {
124         // TODO Auto-generated method stub
125
126     }
127
128 }
129
130 public class Test {
```

Test.java

```
71          btnPanel = new JPanel();
72              // 버튼 패널의 레이아웃 : 그리드(0,4)
73              btnPanel.setLayout(new GridLayout(0,4, 5, 10));
74              // 버튼들을 순서대로 계속 붙이기
75              btn1 = new JButton("1");
76              btn2 = new JButton("2");
77              btn3 = new JButton("3");
78              btnPlus = new JButton("+");
79
80              btnPanel.add(btn1);
81              btn1.addActionListener(this); //버튼 1번이 눌러지면, 이벤트 처리를 해라
82              btnPanel.add(btn2);
83              btn2.addActionListener(this); //버튼 2번이 눌러지면, 이벤트 처리를 해라
```

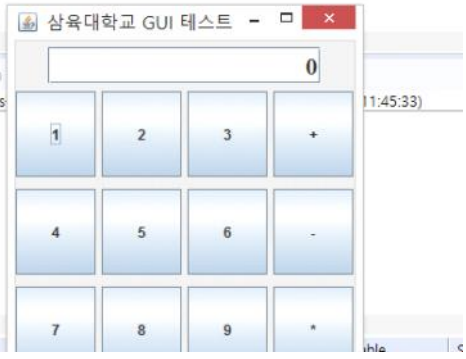
```
    // method 를 만들 예정
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub

        if(e.getSource()==btn1)
        {
            System.out.println("1 clicked");
        }

    }
```

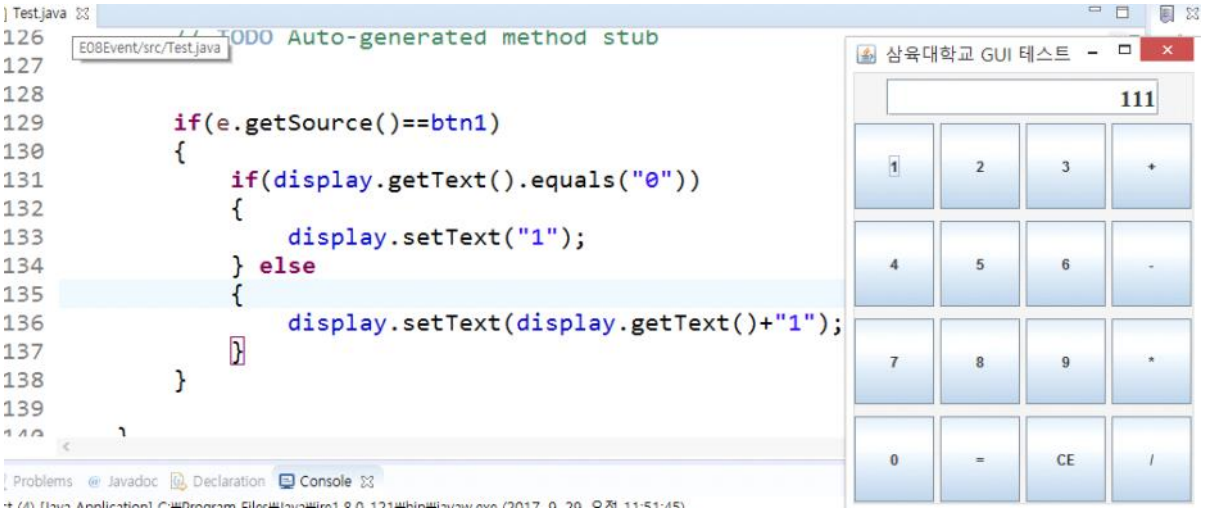
```
134
```

Problems Javadoc Declaration

Test (4) [Java Application] C:₩Program Files 11:45:33)

```
1 clicked
1 clicked
1 clicked
```





- E09Chat

```
Test.java  Test.java
 1 import java.awt.BorderLayout;
12
13 class MyFrame extends JFrame implements ActionListener
14 {
15     //변수들은 반드시 여기에 만들어야 써먹고싶을 때 써먹을 수 있음
16     private JPanel displayPanel;
17     private JPanel inputPanel;
18
19     private JTextArea display;
20     private JTextField input;
21
22     //생성자
23     public MyFrame()
24     {
25         this.setSize(500,400);
26         this.setTitle("SYU Chatting");
27         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
28
29         //프레임의 레이아웃을 센터, 동서남북으로 구분해줘 (여기선 중앙, 남쪽만 사용)
30         this.setLayout(new BorderLayout());
31
32         Font font = new Font("Serif", Font.BOLD, 20);
33
```

Test.java Test.java

```
31
32          Font font = new Font("Serif", Font.BOLD, 20);
33
34          displayPanel = new JPanel();
35          displayPanel.setLayout(new FlowLayout());
36          this.add(displayPanel, BorderLayout.CENTER);
37          display = new JTextArea(12,30);
38
39          //displayPanel.add(display)를 이렇게 바꾸면 스크롤바가 생김
40          JScrollPane scroll = new JScrollPane(display);
41          displayPanel.add(scroll);
42
43          display.setFont(font);
44
45          inputPanel = new JPanel();
46          inputPanel.setLayout(new FlowLayout());
47          this.add(inputPanel, BorderLayout.SOUTH);
48          input = new JTextField(30);
49          inputPanel.add(input);
50
51          input.setFont(font);
52          input.addActionListener(this);
53
```

Test.java Test.java

```
55          this.setVisible(true);
56      }
57
58      @Override
59      public void actionPerformed(ActionEvent e) {
60          if(e.getSource()==input)
61          {
62              //display.setText(input.getText() + "\n" + input.getText());
63              display.append(input.getText()+"\n");
64              input.selectAll();
65
66              //스크롤이 알아서 최근 내용이 있는 곳으로 내려가게 함
67              display.setCaretPosition(display.getDocument().getLength());
68          }
69      }
70
71  }
72
73
74  public class Test {
75
76      public static void main(String[] args) {
77          new MyFrame();
78      }
```

- E10HtmlReader
  - 10.20.30.40이라는 아이피주소가 있을 때

IP주소 = 네트워크 부분(network part) + 호스트 부분(host part)
통신이 되기 위해서는 네트워크 부분은 모두 같고 호스트 부분은 모두 달라야 함
호스트 부분이 전부 0 => 네트워크 자체 / 네트워크 부분이 전부 1 => 네트워크의 브로드캐스트 주소
TCP/IP 프로토콜을 만들 때 이 프로토콜을 사용하는 모든 장비들을 구분해 주기 위해 만듦

IPv4 이진수 32자리(2^32승 개, 대략 42억 9천개)
0000 0000.0000 0000.0000 0000.0000 0000~1111 1111.1111 1111.1111 1111.1111 1111(255.255.255.255)

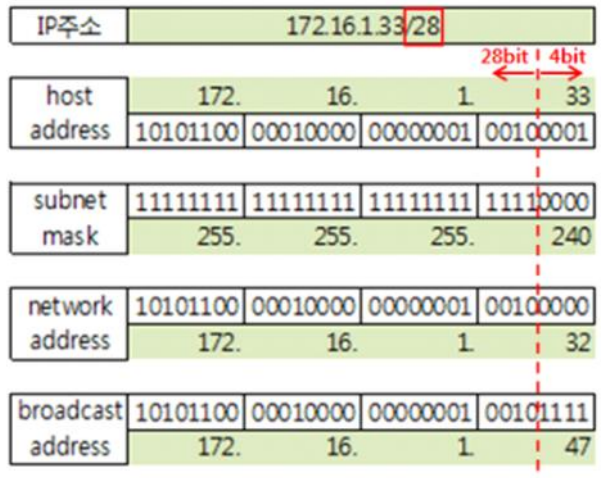
이진수 8자리(8bit) = 옥테트(octet), 총 4개 옥테트

- *IP 주소 5가지 Class(Class A/B/C/D/E)

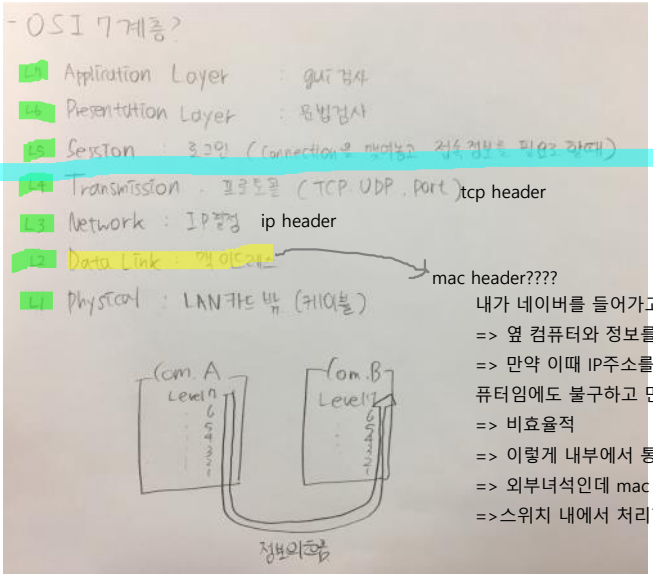
Class	Subnet Mask decimal	No. of Hosts per Network	No. of Networks	Start -End Address
A	255.0.0.0	16 Million	127	1.0.0.0 - 126.255.255.255
B	255.255.0.0	65000	16000	128.0.0.0 - 191.255.255.255
C	255.255.255.0	254	2 Million	192.0.0.0 - 223.255.255.255
D	Reserved for multicast groups			224.0.0.0 - 239.255.255.255
E	Reserved for future use, or Research and Development Purposes			240.0.0.0 - 254.255.255.254

- *서브넷 마스크(Subnet Mask)
-메인이 아닌 어떤 가공을 통한 네트워크를 만들기 위해서 씌우는 마스크
-주어진 IP주소를 네트워크 환경에 맞게 나누어 주기 위해서 씌워주는 이진수 조합
-네트워크 부분과 호스트 부분을 나타내는 역할(서브넷을 만들 때 사용하는 마스크)
-네트워크 주소의 효율적인 이용과 브로드캐스트 도메인을 줄이기 위해 사용

- *서브넷 마스크 기본 성질
-서브넷 마스크로 나누어진 서브넷(Subnet Network)끼리는 라우터를 통해서만 통신이 가능
-이진수로 썼을 때 1이 연속적으로 나와야 함



[그림2. IP주소와 서브넷팅]



TcP Header에 IP Header까지 붙어있는 녀석을 **PACKET**이라고 함

tcp header

ip header

mac header????

내가 네이버를 들어가고 다음을 들어가는 일보다 더 자주하는일
=> 옆 컴퓨터와 정보를 주고받는 일
=> 만약 이때 IP주소를 가지고 (Router)통신을 하게되면 같은 switch와 연결되어있는 컴퓨터임에도 불구하고 먼 길을 떠나게 되는것임
=> 비효율적
=> 이렇게 내부에서 통신을 할 때는 mac address만 보고 내부적으로 처리함
=> 외부녀석인데 mac address가 같은 녀석이 있으면 처리가 안됨
=>스위치 내에서 처리할 녀석은 mac주소를 떼고 보낸다 => 알아서 내부적으로 처리

- OSI 7 Layer는 사용하다보면 불합리할때가 많다 -> 4 Layer로 나눠는게 더 좋다 -> **TCP / IP의 탄생**
(!! **TCP/IP**가 TCP로만 통신하는 것은 아님)

- loop back……..
- socket……

- Test.java.. 개어렵

```
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.net.UnknownHostException;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;

class ChatServer
{
	//변수들은 반드시 여기에 만들어야 써먹고싶을 때 써먹을 수 있음
	DatagramSocket socket;
	DatagramPacket packet;
	InetAddress address = null;

	private int myPort = 10000;
	private int yourPort = 10000;

	private JPanel displayPanel;
	private JPanel inputPanel;

	private JTextArea display;
	private JTextField input;

	public ChatServer()
	{
		new MyFrame();

		try {
			address = InetAddress.getByName("172.30.116.6"); // 상대방주소
			socket = new DatagramSocket(myPort);
		} catch (Exception e) {
			// TODO Auto-generated catch block
		}
	}

	public void receive()
	{
		while(true)
		{
			try {
				byte[] buf = new byte[1024];
				packet = new DatagramPacket(buf,buf.length);
				socket.receive(packet);

				display.append(" << " + new String(buf)+"₩n");
				//스크롤이 알아서 최근 내용이 있는 곳으로 내려가게 함
				display.setCaretPosition(display.getDocument().getLength());

	} catch (Exception e) {
		// TODO: handle exception
	}
	}
```

```
    }

    class MyFrame extends JFrame implements ActionListener
    {

        //생성자
        public MyFrame()
        {
            this.setSize(500,400);
            this.setTitle("SYU Chatting");
            this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            //프레임의 레이아웃을 센터, 동서남북으로 구분해줘 (여기선 중앙, 남쪽만 사용)
            this.setLayout(new BorderLayout());

            Font font = new Font("Serif", Font.BOLD, 20);

            displayPanel = new JPanel();
            displayPanel.setLayout(new FlowLayout());
            this.add(displayPanel, BorderLayout.CENTER);
            display = new JTextArea(12,30);

            //displayPanel.add(display)를 이렇게 바꾸면 스크롤바가 생김
            JScrollPane scroll = new JScrollPane(display);
            displayPanel.add(scroll);

            display.setFont(font);

            inputPanel = new JPanel();
            inputPanel.setLayout(new FlowLayout());
            this.add(inputPanel, BorderLayout.SOUTH);
            input = new JTextField(30);
            inputPanel.add(input);

            input.setFont(font);
            input.addActionListener(this);


            this.setVisible(true);
        }

        @Override
        public void actionPerformed(ActionEvent e) {
            if(e.getSource()==input)
            {
                String msg = input.getText();
                byte[] buf = msg.getBytes();
                DatagramPacket packet;
                packet = new DatagramPacket(buf, buf.length, address, yourPort);

                try {
                    socket.send(packet);
                } catch (Exception e2) {
                    // TODO: handle exception
                }


                //display.setText(input.getText() + "₩n" + input.getText());
                display.append(" 나 >> "+msg+"₩n");
                input.selectAll();
                display.setCaretPosition(display.getDocument().getLength());

            }
        }

    }
}
```

```
public class Test {

	public static void main(String[] args) {
		ChatServer chat = new ChatServer();
		chat.receive();
	}

}
```

-