

=====

<콤보박스><리스트박스>

lstc.Items.Add(treev.SelectedNode.Text); -> 리스트에 선택노드 추가

<리스트뷰>

ListViewItem.ListViewSubItem sub1 = new ListViewItem.ListViewSubItem();

sub1.Text = txta.Text;

ListViewItem lvitem = new ListViewItem();

lvitem.Text = txtb.Text;

lvitem.SubItems.Add(sub1);

lstInfo.Items.Add(lvitem);

삭제 :

if (lstInfo.SelectedItems.Count != 0; { lstInfo.SelectedItems[0].Remove(); }

출력 :

```
if (lstInfo.SelectedItems.Count != 0) {  
    txtStuNum.Text = lstInfo.SelectedItems[0].Text; //상위  
    txtName.Text = lstInfo.SelectedItems[0].SubItems[1].Text; //하위  
}
```

=====

<트리뷰> -> 키값과 value값 저장해야함

tvCourse.ExpandAll();

추가 :

```
if (txtAddNode.Text != "") {  
    tvCourse.SelectedNode.Nodes.Add(txtAddNode.Text, txtAddNode.Text);  
}
```

삭제 :

```
if (tvCourse.SelectedNode.Nodes.Count == 0){  
    tvCourse.Nodes.Remove(tvCourse.SelectedNode);  
}
```

찾기 :

```
TreeNode t = tvCourse.Nodes.Find(txtFindNode.Text, true)[0];  
tvCourse.SelectedNode = t;
```

```
if (tvCourse.SelectedNode != null) {  
    tvCourse.ExpandAll();  
    tvCourse.Select();  
}
```

=====

//파일 쓸때는 바이트 형식(GetBytes)으로 바꾸고, 읽을 때는 문자열 형식(GetString)으로 바귀줌

```
//using System.IO 추가해야함
```

<파일 읽기>

```
if( openFileDialog1.ShowDialog() == DialogResult.OK){
byte[] b = new Byte[500]
FileStream fs = new FileStream(openFileDialog1.FileName, FileMode.Open, FileAccess.Read);
fs.Seek(0, SeekOrigin.Begin);
fs.Read(b,0,500);
txta.Text = Encoding.Default.GetString(b, 0, b.Length);
fs.close();
}
```

<파일 쓰기>

```
if( saveFileDialog1.ShowDialog() == DialogResult.OK){
byte[] b = new Byte[500]
FileStream fs = new FileStream(saveFileDialog1.FileName, FileMode.Create, FileAccess.ReadWrite);
b = Encoding.Default.GetBytes(txta.Text);

fs.Seek(0, SeekOrigin.Begin);
fs.Write(b,0,b.Length);
fs.close();
```

=====

마우스 클릭 이벤트는 마우스 기기의 버튼누르는것  
마우스 엔터 이벤트가 마우스 해당 버튼 안으로 들어갈때 발생

=====

<파일객체쓰기>

```
Using System.IO;
Using System.Runtime.Serialization;
Using System.Runtime.Serialization.Formatters.Binary;

Student s = new Student(생성자);
FileStream fs = new FileStream("파일명",FileMode.Create);

BinaryFormatter bf = new BinaryFormatter();
bf.Serialize(fs,s);

fs.Flush();
fs.Close();
```

<파일객체읽기>

```
FileStream fs = new FileStream("파일명",FileMode.Open);

BinaryFormatter bf = new BinaryFormatter();

Student s = (Student)bf.Deserialize(fs);
Txta.Text = s.No;
Txta.Text = s.Name;
Txta.Text = s.tel;
```

```
fs.Close();
```

---

```
# Cp_09
```

```
namespace FormEventProject
{
    public partial class EventForm : Form
    {
        public EventForm()
        {
            InitializeComponent();
        }

        private void EventForm_Load(object sender, EventArgs e)
        {
            //form의 Load 이벤트는 form이 열리기 전 실행되고, 실행 맞춘 후 form 이 로드된다.
            //form의 size 프로퍼티에 있는 width를 조정. this 는 현재 form 을 가리킨다.
            String formInfo = "EventForm의 넓이 : " + this.Width +
                "높이 : " + this.Height;
            MessageBox.Show(formInfo);
        }

        private void EventForm_MouseMove(object sender, MouseEventArgs e)
        {
            <마우스이벤트>
            //form에서 마우스가 움직일때 조정하도록 form에서 마우스무브 이벤트 적용함
            //MouseEventArgs에 마우스 이벤트에 해당하는 기능들을 가지고 있다.
            lblOut.Text = "마우스 좌표 : x : " + e.X + "y : " + e.Y;

            Point mousep = PointToClient(MousePosition); //현재 마우스 위치 얻을 수 있음
            mousep.X + mousep.Y;
        }

        private void EventForm_MouseClick(object sender, MouseEventArgs e)
        {
            //click 이벤트와는 다르게 마우스 클릭은 어디를 누르는것이 중요한게 아니라
            //마우스 기기에서 어떤 버튼이 눌렸는지를 출력해준다.
            MessageBox.Show(e.Button.ToString());
        }

        private void EventForm_KeyPress(object sender, KeyPressEventArgs e)
```

```

{
    //KeyPress 는 KeyDown 과 비슷하지만 키가 눌린 값에 더 초점을 두고,
    //KeyDown은 눌린 상황에 대한 것에 초점을 둔다.
    //따라서 shift는 동작을 하지 않아서 이것은 KeyDown에서 다룬다.

    MessageBox.Show(e.KeyChar.ToString());
    //keychar은 눌린 키 값을 가지고 있고,
    //이것을 문자열로 바꿔서 메시지 박스에 출력해 줘야 한다.

}

private void EventForm_KeyDown(object sender, KeyEventArgs e)
{
    //KeyDown 에서는 출력되지 않는 키값을 다룬다.
    //Shift , Menu, Control 키는 KeyDown 이벤트에서 발생하는 것이다.
    MessageBox.Show(e.KeyCode.ToString()); //keycode는 정수형이기 때문에
}
}
}

```

---

<좌표이동><좌표수정>

namespace LocationAndFontApp

```

{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void btnup_Click(object sender, EventArgs e)
        {
            lblHorse.Font = new Font("궁서", lblHorse.Font.Size + 1); //폰트 사이즈 조절
            //폰트는 객체이기 때문에 객체를 만들어 넣어줘야 한다.
        }

        private void btndown_Click(object sender, EventArgs e)
        {
            lblHorse.Font = new Font("궁서", lblHorse.Font.Size - 1);
        }

        private void btnOrigin_Click(object sender, EventArgs e)
        {
            //location은 point형 객체이기 때문에 객체 생성
            //해당 컨트롤에 Location 속성을 보면 됨
        }
    }
}

```

```

        lblHorse.Location = new Point(139, 94); //만드는것에 따라 다를 수 있음
    }

    private void btnU_Click(object sender, EventArgs e)
    {
        //위로 가는 것 : y좌표만 1증가시킴
        lblHorse.Location = new Point(lblHorse.Location.X, lblHorse.Location.Y - 1);
    }

    private void btnL_Click(object sender, EventArgs e)
    {
        lblHorse.Location = new Point(lblHorse.Location.X-1, lblHorse.Location.Y);
    }

    private void btnLU_Click(object sender, EventArgs e)
    {
        lblHorse.Location = new Point(lblHorse.Location.X - 1, lblHorse.Location.Y);
        lblHorse.Location = new Point(lblHorse.Location.X, lblHorse.Location.Y - 1);
    }

    private void btnD_Click(object sender, EventArgs e)
    {
        lblHorse.Location = new Point(lblHorse.Location.X, lblHorse.Location.Y+1);
    }

    private void btnR_Click(object sender, EventArgs e)
    {
        lblHorse.Location = new Point(lblHorse.Location.X + 1, lblHorse.Location.Y);
    }

    private void btnRU_Click(object sender, EventArgs e)
    {
        lblHorse.Location = new Point(lblHorse.Location.X + 1, lblHorse.Location.Y - 1); //->
        //lblHorse.Location = new Point(lblHorse.Location.X + 1, lblHorse.Location.Y);
        //lblHorse.Location = new Point(lblHorse.Location.X, lblHorse.Location.Y - 1);
    }

    private void btnLD_Click(object sender, EventArgs e)
    {
        lblHorse.Location = new Point(lblHorse.Location.X - 1, lblHorse.Location.Y);
        lblHorse.Location = new Point(lblHorse.Location.X, lblHorse.Location.Y + 1);
    }

    private void btnRD_Click(object sender, EventArgs e)
    {
        lblHorse.Location = new Point(lblHorse.Location.X + 1, lblHorse.Location.Y);
        lblHorse.Location = new Point(lblHorse.Location.X, lblHorse.Location.Y + 1);
    }

```

이것도 됨

}  
  }  
    }

---

#cp\_10

<폼에값전달>

```
namespace WindowsForms
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void btncreateForm_Click(object sender, EventArgs e)
        {
            SubForm sf = new SubForm();
            sf.Text = txtTitle.Text;
            sf.Location = new Point(int.Parse(txtX.Text), int.Parse(txtY.Text));
            //location은 포인터 형으로 사용해야 한다.
            sf.Size = new Size(int.Parse(txtWidth.Text), int.Parse(txtHeight.Text));
            sf.Opacity = double.Parse(txtOpacity.Text); //폼의 투명도를 나타냄

            if(rdoNone.Checked)
            {
                sf.FormBorderStyle = FormBorderStyle.None; //조절가능
            }
            else if(rdoSingle.Checked)
            {
                sf.FormBorderStyle = FormBorderStyle.FixedSingle; //폼 태두리를 조절 못하게 함
            }
            else
            {
                sf.FormBorderStyle = FormBorderStyle.FixedToolWindow; //도구상자모양으로 땀,
크기조절 x
            }

            sf.Show(); //메인폼으로 돌아올 수 있음
            //sf.ShowDialog(); // 메인폼으로 다시 돌아오지 못함
        }
    }
}
```

---

```
namespace WindowsFormsApp
{
    public partial class InputForm : Form
```

```

{
    public InputForm()
    {
        InitializeComponent();
    }

    private void btnSend_Click(object sender, EventArgs e)
    {
        String name, gender, favorite="";
        int age, year, month, day;

        name = txtname.Text;
        age = int.Parse(txtage.Text);
        year = int.Parse(txtyear.Text);
        month = int.Parse(txtMonth.Text);
        day = int.Parse(txtDay.Text);

        if (rdoMan.Checked)
        {
            gender = "Man";
        }
        else
        {
            gender = "Woman";
        }

        if (chkfishing.Checked)
        {
            favorite += "fishing" + Environment.NewLine;
        }
        if (chkGaming.Checked)
        {
            favorite += "Gaming" + Environment.NewLine;
        }
        if (chkSinging.Checked)
        {
            favorite += "Singing" + Environment.NewLine;
        }

        //서브폼에서 생성자 생성 후 값 전달

        // 1. 객체를 기반한 전달
        Info info = new Info(name, age, gender, year, month, day, favorite);
        ReceivedForm rf = new ReceivedForm(info);

        // 2. 값을 기반한 전달
        //ReceivedForm rf = new ReceivedForm(name, age, gender, year, month, day, favorite);
    }
}

```



```

        rf.Show();
    }
}

```

```

public partial class ReceivedForm : Form
{
    public ReceivedForm()
    {
        InitializeComponent();
    }
    public ReceivedForm(String name, int age, String gender, int year, int month, int day,
String favorite )
    {
        InitializeComponent();//이걸 해줘야 텍스트들이 생성되는 것임. 따라서 꼭 해줘야 함
        txtname.Text = name;
        txtage.Text = age.ToString();
        txtGender.Text = gender;
        txtFavorite.Text = favorite;
        txtyear.Text = year + " ";
        txtMonth.Text = month + " ";
        txtDay.Text = day + " ";
    }

    public ReceivedForm(Info info)//생성자 매개변수를 객체로 주는 방법 -> 프로퍼티 이용
    {
        InitializeComponent();//이걸 해줘야 텍스트들이 생성되는 것임. 따라서 꼭 해줘야 함
        txtname.Text = info.Name;
        txtage.Text = info.Age.ToString();
        txtGender.Text = info.Gender;
        txtFavorite.Text = info.Favorite;
        txtyear.Text = info.Year + " ";
        txtMonth.Text = info.Month + " ";
        txtDay.Text = info.Day + " ";
    }

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

```



---

---

#cp\_11

<메시지박스>

```
public partial class UserDialogMain : Form
{
    private string text;
    public UserDialogMain()
    {
        InitializeComponent();
    }

    private void UserDialogMain_Load(object sender, EventArgs e)
    {
        text = txtMain.Text;
    }
    private void btnUser1_Click(object sender, EventArgs e)
    {
        UserDialog1Form uDialog1 = new UserDialog1Form();

        //값을 전달 방법 : 1. 생성자에 전달 2. 프로퍼티로 전달
        uDialog1.Parameter = txtParameter.Text; //프로퍼티 이용
        uDialog1.Show(); //모달리스(창 포커스 이동 가능) 방식으로 띄움

    }
    private void btnUser2_Click(object sender, EventArgs e)
    {
        UserDialog2Form uDialog2 = new UserDialog2Form();
        uDialog2.ShowDialog();
        //ShowDialog 로 해줘야 값을 제대로 가져올 수 있다.
        txtParameter.Text = uDialog2.Parameter; //프로퍼티 이용

    }

    private void btnBasic_Click(object sender, EventArgs e)
    {
        MessageBox.Show("정보를 확인시켜주기 위한 대화상자", "기본 대화상자",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    private void btnOkCancel_Click(object sender, EventArgs e)
    {
        MessageBox.Show("사용자로부터 Ok Cancel 을 확인하기 위한 대화상자 ", "Ok Cancel
대화상자", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);
    }
}
```

```

private void btnYesNo_Click(object sender, EventArgs e)
{
    MessageBox.Show("사용자로부터 Yes No Cancel을 확인하기 위한 대화상자 ", "Yes No Cancel
대화상자", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
}

private void btnAbort_Click(object sender, EventArgs e)
{
    MessageBox.Show("사용자로부터 Abort, Rerty, Ignore 를 결정하기 위한 대화상자 ",
"Abort, Rerty, Ignore 대화상자", MessageBoxButtons.AbortRetryIgnore, MessageBoxIcon.Error);
}
//MessageBoxDefaultButton.Button2

```

<공용대화상자><

```

private void btnColorDialog_Click(object sender, EventArgs e)
{
    //백컬러를 설정
    colorDialog1.ShowDialog();
    txtMain.BackColor = colorDialog1.Color;
}

```

```

private void btnFontDialog_Click(object sender, EventArgs e)
{
    fontDialog1.ShowColor = true;//color 보려면 바꿔줘야함
    fontDialog1.ShowDialog();
    txtMain.Font = fontDialog1.Font;
    txtMain.ForeColor = fontDialog1.Color;
}

```

```

private void btnOpenFileDialog_Click(object sender, EventArgs e)
{
    openFileDialog1.ShowDialog();
    txtMain.Text = openFileDialog1.FileName;//파일의 경로를 복사
}

```

```

private void btnFolderBrowser_Click(object sender, EventArgs e)
{
    folderBrowserDialog1.ShowDialog();
    txtMain.Text = folderBrowserDialog1.SelectedPath;//폴더의 경로를 복사
}

```

```

private void btnPrintDialog_Click(object sender, EventArgs e)
{
    //using System.Drawing.Printing 를 네임스페이스에 추가해야함
    PrinterSettings printer = new PrinterSettings();//기본 프린터 설정
    PrintDocument pd = new PrintDocument();//출력물 설정 -> 인쇄될 내용을 담은 객체

    printDialog1.PrinterSettings = printer;
    printDialog1.Document = pd;
}

```

```

        DialogResult dr = printDialog1.ShowDialog();
        txtMain.Text = printer.PrinterName;
    }
}

```

<툴바><메뉴바>

```

public partial class UserDialog2Form : Form
{
    private string parameter;

    public string Parameter//프로퍼티
    {
        get { return parameter; }
        set { this.parameter = value; }
    }

    public UserDialog2Form()
    {
        InitializeComponent();
    }

    private void btnAccept_Click(object sender, EventArgs e)
    {
        parameter = txtParameter.Text;
        Close();
    }

    private void btnReject_Click(object sender, EventArgs e)
    {
        parameter = "";
        Close();
    }
}

```

---

#cp\_12

<문자정렬><클립보드>

```

private void mnuLeft_Click(object sender, EventArgs e)//지금은 이벤트연결 안해놔서 실행
안됨
{
    mnuLeft.Checked = true;
    mnuCenter.Checked = false;
    mnuRight.Checked = false;
    txtEdit.TextAlign = HorizontalAlignment.Left;
}

```

```
}
```

```
private void mnuCenter_Click(object sender, EventArgs e)
{
    mnuLeft.Checked = false;
    mnuCenter.Checked = true;
    mnuRight.Checked = false;
    txtEdit.TextAlign = HorizontalAlignment.Center;
}
```

```
private void mnuRight_Click(object sender, EventArgs e)
{
    mnuLeft.Checked = false;
    mnuCenter.Checked = false;
    mnuRight.Checked = true;
    txtEdit.TextAlign = HorizontalAlignment.Right;
}
```

`private void mnuAlign_Click(object sender, EventArgs e)` //정렬 어떤것을 선택하든지 이 이벤트 발생하도록 함

```
    //텍스트박스의 정렬을 바꿔주는 함수
    {
        //
        ToolStripMenuItem item = (ToolStripMenuItem)sender;
        //sender는 전달받은 객체로, item 변수에다가 형변환을 해서 넣어준다.

        foreach(ToolStripMenuItem it in item.Owner.Items)
        {
            if(it == item)
            {
                it.Checked = true; //클릭된 메뉴만 checked를 true로 바꿔준다
            }
            else
            {
                it.Checked = false;
            }
        }
    }
}
```

```
private void mnuCopy_Click(object sender, EventArgs e)
{
    Clipboard.SetText(txtEdit.Text); //ctrl + c 와 같은 기능을 함
}
```

```
private void mnuPaste_Click(object sender, EventArgs e)
{
    txtEdit.Text = Clipboard.GetText(); //ctrl + v 와 같은 기능을 함
}
```

```
private void MainForm_MouseMove(object sender, MouseEventArgs e)//상태바의 기능은  
프로그램의 상세 내용 표시이다.
```

```
{  
    //statusStrip1.Items[0].Text = e.X + ", " + e.Y;//이것도됨  
    toolLabel.Text = e.X + ", " + e.Y;  
}
```

```
private void toolLabel_Click(object sender, EventArgs e)  
{  
    //StatusStrip s = (StatusStrip)sender;//상태바에 여러개 있을 경우  
    //MessageBox.Show(e.ClickedItem.Text, "StatusStrip_Click");  
  
    MessageBox.Show(e.ToString(), "StatusStrip_Click");  
}
```

---

#cp\_13

<리스트뷰>

```
namespace ListViewControlApp
```

```
{
```

```
    public partial class ListViewMainForm : Form
```

```
    {
```

```
        public ListViewMainForm()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
private void btnAdd_Click(object sender, EventArgs e)
```

```
{
```

```
    //리스트 뷰의 아이템은 상위과 하위의 구조를 가진다.
```

```
    //아이템에 처음 들어간 값을 기준으로, 그 아이템의 하위로 값들이 저장된다.
```

```
    // 따라서 subitem을 이용해서 값을 추가해야 한다.
```

```
    ListViewItem.ListViewSubItem sub1 = new ListViewItem.ListViewSubItem();
```

```
    ListViewItem.ListViewSubItem sub2 = new ListViewItem.ListViewSubItem();
```

```
    ListViewItem.ListViewSubItem sub3 = new ListViewItem.ListViewSubItem();
```

```
    ListViewItem.ListViewSubItem sub4 = new ListViewItem.ListViewSubItem();
```

```
    sub1.Text = txtName.Text;
```

```
    sub2.Text = txtAddress.Text;
```

```
    sub3.Text = txtTel.Text;
```

```
    sub4.Text = txtDept.Text;
```

```
    ListViewItem lvItem = new ListViewItem();//리스트뷰 객체생성
```

```
    lvItem.Text = txtStuNum.Text;//리스트뷰의 아이템
```

```
    lvItem.SubItems.Add(sub1);//리스트뷰의 아이템에 서브아이템 추가
```

```
    lvItem.SubItems.Add(sub2);
```

```
    lvItem.SubItems.Add(sub3);
```

```

        lvlItem.SubItems.Add(sub4);

        lstInfo.Items.Add(lvlItem); //내가 지정한 리스트뷰의 아이템에 객체를 넣어줌
    }

    private void btnRemove_Click(object sender, EventArgs e) //선택항목을 제거
    {
        if (lstInfo.SelectedItems.Count != 0) //리스트뷰의 선택된 아이템이 있다면 실행
        {
            //선택된 아이템 항목 중 첫번째만 지운다.
            lstInfo.SelectedItems[0].Remove(); //하나의 아이템 항목에는 4개의 서브아이템이
            //따라서 SelectedItems 은 여러개 선택가능
        }
    }

    private void btnPrint_Click(object sender, EventArgs e) //선택항목을 텍스트 박스에 출력
    {
        if (lstInfo.SelectedItems.Count != 0) //리스트뷰의 선택된 아이템이 있다면 실행
        {
            txtStuNum.Text = lstInfo.SelectedItems[0].Text; //상위
            /*txtStuNum.Text = lstInfo.SelectedItems[0].SubItems[0].Text; //이것도 가능*/
            txtName.Text = lstInfo.SelectedItems[0].SubItems[1].Text; //하위
            txtAddress.Text = lstInfo.SelectedItems[0].SubItems[2].Text;
            txtTel.Text = lstInfo.SelectedItems[0].SubItems[3].Text;
            txtDept.Text = lstInfo.SelectedItems[0].SubItems[4].Text;
        }
    }

    private void dmView_SelectedIndexChanged(object sender, EventArgs e)
    {
        //Domain은 클릭 이벤트가 아닌 체인지 이벤트가 발생한다.
        if (dmView.Text == "Details")
        {
            lstInfo.View = View.Details; //리스트뷰의 속성인 뷰를 바꿈
        } else if (dmView.Text == "Largeicon")
        {
            lstInfo.View = View.LargeIcon; //큰 아이콘
        }
        else
        {
            lstInfo.View = View.SmallIcon; //작은 아이콘
        }
    }
}

```

---



<트리뷰>

```
public partial class TreeViewMainForm : Form
{
    public TreeViewMainForm()
    {
        InitializeComponent();
    }

    private void btnAdd_Click(object sender, EventArgs e)
    {
        lstCourse.Items.Add( tvCourse.SelectedNode.Text );//선택된 노드를 list로 추가
    }

    private void TreeViewMainForm_Load(object sender, EventArgs e)
    {
        tvCourse.ExpandAll(); //로드될때 트리가 펼쳐져있게 만들기 위해서
    }

    private void btnNodeRemove_Click(object sender, EventArgs e)
    {
        if ( tvCourse.SelectedNode.Nodes.Count == 0 )
        {
            //하위노드가 없을 경우에만 삭제할수 있도록
            <트리뷰삭제>
            tvCourse.Nodes.Remove( tvCourse.SelectedNode );//tvCourse.SelectedNode 는 객체이다.
            //tvCourse.Nodes.RemoveAt( tvCourse.SelectedNode.Index );//RemoveAt 은 선택된 노드의
            인덱스를 줘야한다.
        }
        else
        {
            MessageBox.Show( "자식 노드가 있는경우 삭제 불가" );
        }
    }

    private void btnListRemove_Click(object sender, EventArgs e)
    {
        if( lstCourse.SelectedIndex != 1 )//삭제할 아이템이 있을경우 삭제
        {
            lstCourse.Items.Remove( lstCourse.SelectedItem );
            //lstCourse.Items.RemoveAt( lstCourse.SelectedIndex );
        }
        else
        {
            MessageBox.Show( "삭제할 아이템이 없거나 선택이 안됐습니다. " );
        }
    }
}
```

```

private void btnAddNode_Click(object sender, EventArgs e)
{
    if (txtAddNode.Text != "")//텍스트가 비어있지 않다면
    {<트리뷰추가>
        //선택된 treenode의 노드들에다가 입력한 노드를 추가한다.
        tvCourse.SelectedNode.Nodes.Add(txtAddNode.Text, txtAddNode.Text);//node는 key값과
text값을 넣어주면 됨
        // tvCourse.SelectedNode.Nodes.Add(New TreeNode(txta.Text, 1, 1));
        // TreeNode(노드이름, 노드선택x일 때 이미지, 노드선택시 이미지);
    }
    else//텍스트가 비어있다면
    {
        MessageBox.Show("텍스트가 비어있습니다. 추가할 수 없습니다. ");
    }
}

private void btnNodeFind_Click(object sender, EventArgs e)
{<트리뷰찾기>
    //Find 는 해당 노드의 key값을 찾는다.
    TreeNode t = tvCourse.Nodes.Find(txtFindNode.Text, true)[0];//노드들에서 해당 key값을
찾는다.
    tvCourse.SelectedNode = t;

    if(tvCourse.SelectedNode != null)//선택된게 있다면
    {
        tvCourse.ExpandAll();//tree를 확장시키고
        tvCourse.Select();//tree를 선택하면 됨
    }
}
}

```

---

<타이머>

<트랙바><업다운컨트롤>

namespace AdControlWindowApp

```

{
    public partial class adControlForm : Form
    {
        public adControlForm()
        {
            InitializeComponent();
        }

        private void nUpDown_ValueChanged(object sender, EventArgs e)
        {//업다운컨트롤은 int 형으로 바꿔줘야 한다.
            //서로의 값이 바뀌면 둘다 바뀌도록
            tvValue.Value = (int)nUpDown.Value;
            //progress는 진행사항을 알려줌
            progress.Maximum = (int)nUpDown.Value;
            //트랙바와 업다운 컨트롤의 값으로 맥시멈 설정하면 유동적이게 됨
        }
    }
}

```

```

private void tvValue_Scroll(object sender, EventArgs e)
{
    nUpDown.Value = tvValue.Value; //스크롤 옮기면 updown 컨트롤도 같이 바뀌도록 설정
    progress.Maximum = tvValue.Value; //프로그래스 컨트롤의 Maximum 값을 변경된 값으로 설정
} //TickFrequency로 눈금조절 가능
//TickStyle 로 눈금 모양 변경
//마우스 휠로도 동작 가능

private void btnStart_Click(object sender, EventArgs e)
{
    timer1.Start(); //timer1 이라는 이벤트를 시작 -> 컴포넌트 형식으로, 사용자에게 안보임
}

private void timer1_Tick(object sender, EventArgs e) //timer1 이벤트가 발생했을때 호출되는
메소드
{
    //Interval 프로퍼티로 1000 이 1초
    //주기적으로 발생시키기 위해서는 Enable프로퍼티를 true로 해야함
    if(progress.Value < progress.Maximum) //맥시멈보다 작을때까지만 반복
    { //프로그래스 컨트롤을 이용해서 value 프로퍼티를 변경
        progress.Value += 1;
    }
}
}
}

```

---

#cp\_14

<파일입출력>

```

private void btnWrite_Click(object sender, EventArgs e)
{
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    { <파일출력>
        byte[] writeData = new byte[500];
        FileStream write = new FileStream(saveFileDialog1.FileName, FileMode.Create,
        FileAccess.Write);

        writeData = Encoding.Default.GetBytes(txtResult.Text);
        write.Seek(0, SeekOrigin.Begin); //현재 스트림의 위치를 시작점으로 옮김
        write.Write(writeData, 0, writeData.Length); //bData = txtResult의 내용을 바이트로
바꾼 정보가 저장되어있음
        write.Close();
    }
    else
    {
        MessageBox.Show("파일쓰기에 오류가 있습니다.");
    }
}

```

```

private void btnRead_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {<파일입력>
        byte[] readData = new byte[500];
        FileStream read = new FileStream(openFileDialog1.FileName, FileMode.Open,
FileAccess.Read);
        //FileMode.OpenorCreate 가능
        read.Seek(0, SeekOrigin.Begin);//현재 스트림의 위치를 시작점으로 옮김
        read.Read(readData, 0, readData.Length);//bData = txtResult의 내용을 바이트로 바꾼
정보가 저장되어있음
        txtResult.Text = Encoding.Default.GetString(readData,0,readData.Length);

        read.Close();
    }
    else
    {
        MessageBox.Show("파일오픈에 오류가 있습니다.");
    }
}
}

```