# 5일차 프로젝트 (기업솔루션프로젝트)

## 실습 : 윈도우 기본 컨트롤2

### 예제1 - 트레이 메시지 디자인

트레이 메시지는 지금까지 알아본 Timer, TextBox, Button 등의 윈도우 기본 컨트롤을 이용하여 메신저나 바이러스 백신 등의 애플리케이션에서 알림메시지 창으로 자주 사용되는 기능을 구현한 예제

프로젝트 이름 : **TrayMsg**

#### [1-1] 트레이 메시지(Form1)

**(1) 트래이 메시지(Form1) 디자인에 사용된 컨트롤의 주요 속성값**

| 폼 컨트롤 | 속성 | 값 |
|---|---|---|
| Form1 | Name | Form1 |
| | Text | 트레이 메시지 |
| | FormBorderStyle | FixedSingle |
| | MaximizeBox | False |
| TextBox1 | Name | txtMsg |
| Button1 | Name | btnMsg |
| | Text | 보이기 |

**(2) 트레이 메시지(Form1.cs) 코드 구현**

❶ **btnMsg Click() 이벤트 핸들러** : [보이기] 버튼을 더블클릭하여 생성한 프로시저로, Form2를 호출하여 트레이 메시지를 보이게 하는 작업 수행

#### [1-2] 트레이 메시지(Form2)

**(1) 트레이 메시지(Form2) 디자인에 사용된 컨트롤의 주요 속성값**

| 폼 컨트롤 | 속성 | 값 |
|---|---|---|
| Form2 | Name | Form2 |
| | Text | 메시지 창 |
| | FormBorderStyle | Non |
| | ShowIcon | False |
| | ShowInTaskbar | False |
| | Size | 170, 120 |
| | TopMost | True |
| Panel1 | Name | plBack |
| | BackColor | LightBlue |
| | BorderStyle | FixedSingle |
| | Dock | Fill |
| LinkLable1 | Name | lblResult |
| | Text | 결과 |
| | TextAlign | MiddleCenter |

### (2) 트레이 메시지(Form2.cs) 코드 구현

❶ using 키워드를 이용하여 Timer를 사용하기 위한 네임스페이스를 추가
using System.Timers; //Timer 클래스사용

❷ TimerEvent 개체를 클래스 내부 상단에 추가
private static System.Timers.Timer TimerEvent; //Timer 개체 생성

❸ Form2의 생성자 메서드인 Form2() 메서드 안에 폼의 위치와 사이즈를 설정하기 위한 코드 추가
메시지 창의 위치와 크기 설정은 폼이 그려지기 전에 설정하기 위해 InitializeComponent() 메서드 호출보다 앞서 이루어져야 한다.

❹ Form1에서 입력받은 문자열 값을 lblResult 컨트롤에 반영하기 위한 set 접근자를 클래스 상단에 추가

❺ 안전한 이벤트 처리를 위한 델리게이트 선언은 클래스 내부 상단에 추가
private delegate void OnDelegateHeight(int Flag); //델리게이트 선언
private OnDelegateHeight OnHeight = null; //델리게이트 개체 생성

❻ **Form2 Load() 이벤트 핸들러** : 폼을 더블클릭하여 생성한 프로시저로, 폼이 로드될 때 발생하는 이벤트를 수행하는 작업 수행

❼ **MsgView() 메서드** : Delegate에 의해 처리되는 메서드로, Flag 인자값에 따라 폼을 올리거나 내리거나 또는 종료하는 작업을 수행

❽ **OnPopUp() 메서드** : 이벤트가 구독될 때 사용되는 이벤트 핸들러로, 폼을 올리는 작업을 수행

❾ **OnPopOut() 메서드** : 폼을 서서히 내리는 작업을 수행하며, 내리는 작업이 끝나면 폼을 종료하는 작업을 수행

❿ **lblResult_LinkClicked() 이벤트 핸들러** : lblResult 컨트롤을 더블클릭하여 생성한 프로시저로, 폼과 이벤트를 종료하는 작업 수행

## [12-3] 트레이 메시지 예제 실행

텍스트 상자에 내용을 입력하고, [보이기] 버튼을 클릭하면 입력된 문자열이 트레이 영역에서 서서히 표시되었다가 사라진다.

### [Source]

```
namespace TrayMsg
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void btnMsg_Click(object sender, EventArgs e)
        {
            _____________________; //Form2의 frm2 객체 생성
            frm2.MsgText = this.txtMsg.Text;
            frm2.ShowDialog();
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Timers;
using System.Windows.Forms;

//Timer 클래스사용 : using System.Timers;

namespace TrayMsg
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();

            int x = Screen.PrimaryScreen.WorkingArea.Width - this.Width - 20; // 스크린의 가로위치
            int y = Screen.PrimaryScreen.WorkingArea.Height - this.Height; // 스크린의 세로위치

            DesktopLocation = new Point(x, y); //폼의 위치 설정
            this.Size = new Size(170, 0); //폼의 크기 설정
        }

        private static System.Timers.Timer TimerEvent; //Timer 개체 생성

        public string MsgText
        {
            set { this.lblResult.Text = value; }
        }

        private _____ void OnDelegateHeight(int Flag); //델리게이트 선언
        private _____ OnHeight = null; //델리게이트 개체 생성

        private void Form2_Load(object sender, EventArgs e)
        {
            OnHeight = new OnDelegateHeight(MsgView);
            this.Size = new System.Drawing.Size(170, 0);
            this.Location =
                new System.Drawing.Point(Screen.PrimaryScreen.WorkingArea.Width
                - this.Width - 20, Screen.PrimaryScreen.WorkingArea.Height - this.Height);
            TimerEvent = new System.Timers.Timer(2);
            TimerEvent.Elapsed += new ElapsedEventHandler(OnPopUp);
            TimerEvent.Start();
        }
```

```
        private void MsgView(int Flag)
        {
            if (Flag == 0)
            {
                Height++;
                Top--;
            }
            else if (Flag == 1)
            {
                Height--;
                Top++;
            }
            else if (Flag == 2)
            {
                this.Close();
            }
        }

        private void OnPopUp(object sender, ElapsedEventArgs e)
        {
            if (Height < 120)
            {
                Invoke(OnHeight, 0);
            }
            else
            {
                TimerEvent.Stop();
                TimerEvent.Elapsed -= new ElapsedEventHandler(OnPopUp);

                TimerEvent.Elapsed += new ElapsedEventHandler(OnPopOut);
                TimerEvent.Interval = 3000;
                TimerEvent.Start();
            }
            Application.DoEvents();
        }

        private void OnPopOut(object sender, ElapsedEventArgs e)
        {
            while (Height > 2)
            {
                Invoke(OnHeight, 1);
            }
            TimerEvent.Stop();
            Application.DoEvents();
            Invoke(OnHeight, 2);
        }
        private void llblResult_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
        {
            this.Close();
            TimerEvent.Stop();
        }
    }
}
```

## 예제2 - 프로세스 보기

윈도우 운영체제의 작업관리자의 일부 기능으로 ListView, StatusStrip, Button 컨트롤과 Thread를 이용하여 프로세스 보기 및 예제를 구현한다.

프로젝트 이름 : **ProcessView**

### [2-1] 프로세스 디자인에 사용된 컨트롤의 주요 속성값

| 폼 컨트롤 | 속성 | 값 |
|---|---|---|
| Form1 | Name | Form1 |
| | Text | 프로세스 보기 |
| | FormBorderStyle | FixedSingle |
| | MaximizeBox | False |
| | MinimizeBox | False |
| ListView1 | Name | lvView |
| | Columns | [설정] |
| | FullRowSelect | True |
| | GrideLines | True |
| | View | Details |
| Button1 | Name | btnKill |
| | Text | 프로세스 끝내기 |
| StatusStrip1 | Name | ssBar |

### [2-2] lvView 컨트롤의 Columns 속성값을 설정

❶ lvView 컨트롤을 선택하고 속성 목록 창에서 [Columns] 항목의 ...(컬렉션) 버튼을 클릭

❷ [ColumnHeader 컬렉션 편집기] 대화상자가 나타나면 [멤버] 항목 아래에서 [추가] 버튼을 클릭하여 필요한 4개 컬럼을 위한 멤버 항목을 멤버를 추가하고 멤버의 속성값 설정

| 폼 컨트롤 | 속성 | 값 |
|---|---|---|
| ColumnHeader1 | Name | chName |
| | Text | 프로세스 이름 |
| | Width | 108 |
| ColumnHeader2 | Name | chPid |
| | Text | PID |
| | TextAlign | Center |
| | Width | 60 |
| ColumnHeader3 | Name | chTime |
| | Text | Time |
| | TextAlign | Center |
| | Width | 90 |
| ColumnHeader4 | Name | chMemory |
| | Text | 메모리 사용 |
| | TextAlign | Right |
| | Width | 100 |

❸ 상태 바를 설정하기 위해 ssBar 컨트롤을 선택하고 속성 창에서 [Items] 항목의 ...(컬렉션) 버튼을 클릭하고 [항목 컬렉션 편집기] 대화상자를 연다.

❹ [항목 컬렉션 편집기] 대화상자의 왼쪽 위에 목록 상자에서 StatLabel을 선택하고 [추가] 버튼을 클릭하여 필요한 3개 항목을 추가

| 폼 컨트롤 | 속성 | 값 |
|---|---|---|
| ToolStripStatusLabel1 | Name | tsslProcess |
| | Text | 프로세스 : 0개 |
| ToolStripStatusLabel2 | Name | tsslCpu |
| | Text | CPU 사용 : 0% |
| ToolStripStatusLabel3 | Name | tsslMem |
| | Text | 실제 메모리 : 0% |

### [2-3] 프로세스 코드 구현하기

❶ using 키워드를 이용하여 필요한 네임스페이스를 추가

```
using System.Diagnostics; //Process 클래스 사용
using System.Collections; //ArrayList 사용
using System.Threading; //스레드 사용
```

❷ **ProcessUpdate 메서드** : ProcessThread 보조 스레드에 대입되어 실행되며, 프로세스 정보를 감시하여 프로세스가 생성되거나 삭제되어 변경되면 업데이트하는 작업 수행

❸ **ProcessView() 메서드** : 실행 중인 프로세스 정보를 가져와 lvView 컨트롤에 나타내는 작업 수행

❹ **NumFormat() 메서드** : Long 타입의 숫자를 산술식에 의해 계산한 후 string 타입으로 변환하여 반환

❺ **getCPU Info() 메서드** : 보조 스레드에서 실행되는 대리자로 while 문을 이용하여 1초마다 CPU와 메모리 사용량을 체크하여 OnTotal 델리게이트를 호출하는 작업 수행

❻ **TotalView() 메서드** : OnTotal 델리게이트에 의해 실행되며, CPU와 메모리 사용량을 확인하여 ToolStripStatusLabel 컨트롤에 나타내는 작업 수행

❼ **btnKill Click 이벤트 핸들러** : [프로세스 끝내기] 버튼을 더블클릭하여 생성한 프로시저로, ProcessKill() 메서드를 호출하여 선택된 프로세스를 종료하는 작업 수행

❽ **Form1 FormClosing 이벤트 핸들러** : 폼을 선택하고 이벤트 목록 창에서 [FormClosing] 항목을 더블클릭하여 생성한 프로시저로, 폼을 종료할 때 발생하는 이벤트를 제어하는 작업 수행

### [2-4] 프로세스 보기 예제 실행

프로세스 목록을 확인하고 임의의 프로세스를 선택한 뒤 [프로세스 끝내기] 버튼을 클릭하여 선택한 프로세스가 종료되는 것을 확인해 본다.

**[Source]**

```
using System;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Diagnostics; //Process 클래스 사용
using System.Collections; //ArrayList 사용
using System.Threading; //스레드 사용
```

```
namespace ProcessView
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private Thread ProcessThread; // 프로세스 나타내기 스레드
        Thread checkThread = null; //실시간 시스템 정보 체크

        private delegate void ProcessUpdateDelegate(); // 델리게이트 생성
        private ProcessUpdateDelegate UpProc = null;

        private delegate void TotalUpdateDelegate(int m, int n); // 델리게이트 생성
        private TotalUpdateDelegate OnTotal = null;

        private PerformanceCounter oCPU =
            new PerformanceCounter("Processor", "% Processor Time", "_Total");
                                                    //시스템 CPU 성능 카운터
        private PerformanceCounter oMem =
            new PerformanceCounter("Memory", "% Committed Bytes In Use");
                                                  //시스템 Mem 성능 카운터
        private PerformanceCounter pCPU =
            new PerformanceCounter();

        bool bExit = false;
        int cp = 0;

        private void Form1_Load(object sender, EventArgs e)
        {
            ProcessView(); //프로세스 출력

            UpProc = new ProcessUpdateDelegate(ProcessView); //델리게이트에 구동 메서드 입력
            OnTotal = new TotalUpdateDelegate(TotalView);

            ProcessThread = new Thread(ProcessUpdate); //스레드 대리자에 구동 메서드 입력
            ProcessThread.Start(); //스레드 시작

            checkThread = new Thread(getCPU_Info);
            checkThread.Start();
        }
```

```
private void ProcessUpdate()
{
    try
    {
        while (true)
        {
            var oldlist = new ArrayList();

            foreach (var oldproc in Process.GetProcesses())
            {
                oldlist.Add(oldproc.Id.ToString());
            }
            Thread.Sleep(1000);

            var newproc = Process.GetProcesses();

            if (oldlist.Count != newproc.Length)
            {
                Invoke(UpProc);
                continue;
            }

            int i = 0;

            foreach (var rewproc in Process.GetProcesses())
            {
                if (oldlist[i++].ToString() != rewproc.Id.ToString())
                {
                    Invoke(UpProc);
                    break;
                }
            }
        }
    }
    catch { }
}

private void ProcessView()
{
    try
    {
        this.lvView.Items.Clear();
        cp = 0;
        foreach (var proc in Process.GetProcesses())
        {
```

```
                string[] str;
                try
                {
                    str = proc.TotalProcessorTime.ToString().Split(new Char[] { '.' });
                }
                catch { str = new string[] { "" }; }

                var strArray = new string[] { proc.ProcessName.ToString(), proc.Id.ToString(),
                    str[0], NumFormat(proc.WorkingSet64) };

                var lvt = new ListViewItem(strArray);
                this.lvView.Items.Add(lvt);
                cp++;
            }
        }
        catch { }
        this.tsslProcess.Text = "프로세스 : " + cp.ToString() + "개";
    }

    private void TotalView(int m, int n)
    {
        this.tsslCpu.Text = "CPU 사용: " + m.ToString() + " %";
        this.tsslMem.Text = "실제 메모리 : " + n.ToString() + " %";
    }

    private string NumFormat(long MemNum)
    {
        MemNum = MemNum / 1024;
        return String.Format("{0:N}", MemNum) + " KB";
    }

    private void getCPU_Info()
    {
        while (!bExit)
        {
            int iCPU = (int)oCPU.NextValue();
            int iMem = (int)oMem.NextValue();
            Invoke(OnTotal, iCPU, iMem);
            Thread.Sleep(1000);
        }
    }
```

```
        private void btnKill_Click(object sender, EventArgs e)
        {
            ProccessKill();
        }

        private void ProccessKill()
        {
            try
            {
                int PID = Convert.ToInt32(this.lvView.SelectedItems[0].SubItems[1].Text);
                Process tProcess = Process.GetProcessById(PID);
                if (!(tProcess == null))
                {
                    var dlr = MessageBox.Show(this.lvView.SelectedItems[0].SubItems[0].Text +
                        " 프로세스를 끝내시겠습니까?", "알림", MessageBoxButtons.YesNo,
                        MessageBoxIcon.Warning);
                    if (dlr == DialogResult.Yes)
                    {
                        tProcess.Kill();
                        ProcessView();
                    }
                }
                else
                {
                    MessageBox.Show(this.lvView.SelectedItems[0].SubItems[0].Text +
                        "프로세스는 존재하지 않습니다", "알림", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
                    ProcessView();
                }
            }
            catch
            {
                return;
            }
        }

        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            if (!(ProcessThread == null))
                ProcessThread.Abort(); //스레드 종료
            if (!(checkThread == null))
                checkThread.Abort(); //스레드 종료
        }
    }
}
```