

WinForm

[0]

• 목 차

- *Chapter 01* WinForm
- *Chapter 02* 필수구조체
- *Chapter 03* Graphics 개체사용
- *Chapter 04* 텍스트출력, 폰트 사용
- *Chapter 05* 펜과 브러쉬, 패스, 클리핑, Region
- *Chapter 06* 이미지와 비트맵
- *Chapter 07* 타이머
- *Chapter 08* GDI+ 좌표 변환
- *Chapter 09* 마우스와 키보드
- *Chapter 10* 인쇄하기

[1]

 비트교육센터

• 목 차

- *Chapter 11* 버튼, 라벨, 스크롤, 컨테이너 클래스
- *Chapter 12* 텍스트 박스, 리스트 박스, 콤보박스
- *Chapter 13* 대화상자
- *Chapter 14* 드래그 앤 드롭, 클립보드

**01**  
Chapter  
WinForm

- **원폼 프로그래밍 (WinForm Programming)**

- 원도우, 버튼, 대화상자 등을 추가해서 보다 그래픽적이고 다이나믹하게 사용자의 정보를 입력받고, 표현하는 것을 윈도우즈 폼(Windows Form)
- 공용 언어 런타임의 Windows 응용 프로그래밍

- **Windows Form의 구성요소**

구성 요소	설명
폼(Form)	<ul style="list-style-type: none"> <li>- 프로그램이 표시되는 창</li> <li>- Form 클래스를 이용해서 구현</li> </ul>
컨트롤(Control)	<ul style="list-style-type: none"> <li>- 버튼, 텍스트 박스 등 폼에 추가되는 구성요소들</li> </ul>
이벤트(Event)	<ul style="list-style-type: none"> <li>- 버튼 클릭 등 컨트롤에 변화가 생길 때 발생하는 상황</li> <li>- 이벤트가 발생한 다음 이벤트에 맞는 처리를 해주는 것이 원폼의 핵심</li> </ul>

[4]

 비트교육센터

### 관련 네임스페이스

- WinForm
  - System.Windows.Forms  
=> System.Windows.Forms.dll에 구현되어 있음
- GDI+
  - System.Drawing : 이미지, 색, 브러시, 펜, 글꼴 등 기본 그래픽 기능
  - System.Drawing.Drawing2D : 고급 레스터 및 백터 그래픽 기능
  - System.Drawing.Imaging : 이미지 처리에 관련 고급 기능
  - System.Drawing.Printing : 인쇄 및 인쇄 미리보기 기능
  - System.Drawing.Text : 고급 글꼴 제어 기능
  - System.Drawing.Desing : 사용자 지정 컨트롤의 디자인 시점 지원  
=> 모두 System.Drawing.dll에 구현되어 있음

- 필요사항

- System.Windows.Forms 네임스페이스 선언
- [참조] 항목에 System.Windows.Forms.dll 파일 등록후 선언
- Form 클래스를 이용한 윈도우 화면 구성
- (static) Application.Run(...) 을 이용한 윈도우 프로그램 구동

[5]

 비트교육센터

### 예제

- 콘솔 프로그램에서 WinForm구현하기
  - using System;
  - using System.Windows.Forms;

```
class Program
{
    static void Main(string[] args)
    {
        // Application : 응용프로그램의 시작 및 종지를 위한 메서드,
        //           원도우 메시지 처리, 응용프로그램 정보를 가져오기
        //           위한 속성 관리등을 함

        // Run : Form클래스의 인스턴스를 등록하게 되면
        //           원도우 메시지의 처리 루틴 동작하게 됨
        Application.Run(new Form());
        //System.Windows.FormsApplication.Run(new
        System.Windows.Forms.Form());
    }
}
```

# **02**

Chapter

## 필수구조체

- Point

- Point() : 기본 생성자
- Point(int data) : 앞16bit : Y 뒤 16bit : X
- Point(Size data) : Size 구조체의 Width: X, Height:Y
- Point(int x, int y) : X, Y  
==> 마우스 좌표를 다룰 때 주로 사용

- Size

- Size() : 기본 생성자
- Size(Point data) : Point구조체의 X :Width, Y : Height
- Size(int x, int y) : 각각 Width, Height  
==> 윈도우 영역을 저장할 때 주로 사용

[?]

 비트교육센터

## 예제

- Size 구조체 연산

- using System;
- using System.Drawing;

```
class StructExam1
{
    static void Main(string[] args)
    {
        Size sz1 = new Size(100, 100);
        Size sz2 = new Size(50, -50);
        Size sz3 = new Size();
        sz3 = sz1 + sz2;
        System.Console.WriteLine(sz3.ToString());
        sz3 = sz2 - sz1;
        System.Console.WriteLine(sz3.ToString());
        sz3 += sz1;
        System.Console.WriteLine(sz3.ToString());
        sz3 -= sz2;
        System.Console.WriteLine(sz3.ToString());
    }
}
```

## 01 필수 구조체

### • Rectangle

- Offset 메서드
  - > rect.Offset(10, -10); >> rect.X += 10; rect.Y += -10;
- Inflate 메서드
  - > rect.Inflate(10, -10); >> rect.X -= 10; rect.Y -= -10;  
rect.Width += 2\*10; rect.Height += 2\*(-10);
- static형 Inflate메서드
  - > Rectangle.Inflate(rect, 10, -10); >> Inflate 메서드와 동일
- FromLTRB 메서드: 좌측상단과 우측하단의 좌표로 Rectangle객체를 만들어 반환
  - > rect = Rectangle.FromLTRB(x1, y1, x2, y2); >> rect = new Rectangle(x1, y1, x2, y2);
- Union메서드 :두 Rectnalge 영역의 합집합 반환
  - rect = Rectanble.Union(rect1, rect2);

[8]

비트교육센터

### 예제

```
• using System;
  using System.Drawing;
class StructExam3
{
    static void Main(string[] args)
    {
        Rectangle rect = new Rectangle();
        Rectangle rect1 = new Rectangle(10, 10, 100, 100);
        Rectangle rect2 = new Rectangle(50, 50, 100, 100);

        // 좌측상단과 우측하단의 좌표로 Rectangle 객체를 만들어 반환
        rect = Rectangle.FromLTRB(30, 30, 50, 50);
        System.Console.WriteLine("rect 값(FromLTRB) :" + rect.ToString());

        // x,y의 크기만큼 사각형의 시작 좌표를 이동
        rect.Offset(10, -10);
        System.Console.WriteLine("rect 값(Offset) :" + rect.ToString());

        // rect.X -= 50      rect.Y -= 50
        // rect.Width += 2*50  rect.Y += 2*50
        Rectangle.Inflate(rect1, 50, 50);
        System.Console.WriteLine("rect 값(Inflate) :" + rect.ToString());
```

```
// 두 사각형 영역의 합집합을 반환
rect1 = Rectangle.Union(rect1, rect2);
System.Console.WriteLine("rect 값(Union) :" + rect.ToString());

// 두 사각형 영역의 교집합을 반환
rect1 = Rectangle.Intersect(rect1, rect2);
System.Console.WriteLine("rect 값(Intersect) :" + rect.ToString());

// 매개변수가 사각형 영역에 포함되는 유무 반환
if (rect1.Contains(rect2))
    System.Console.WriteLine("rect2는 rect1에 포함되어 있습니다.");

// 매개변수가 사각형 영역과 교차하는 여부 반환
if (rect2.IntersectsWith(rect1))
    System.Console.WriteLine("rect2는 rect1과 교차합니다.");

}
}
```

- Color

- 매개변수가 없는 기본생성자만 존재함  
->Color color = new Color();
- 141개의 색상 값이 미리 정의된 속성을 갖고 있음  
-> 알파벳 순으로 AliceBlue(0) ~ YelloGreen(140) , Transparent(141) : 투명색

- 기본 색상 이외의 색 구현

- Color.FromArgb 메서드로 구현
  - > static Color FromArgb(int color)
  - > static Color FromArgb(int r, int g, int b)
  - > static Color FromArgb(int a, int r, int g, int b)
  - > static Color FromArgb(int a, Color color)

## 예제

- KnowColor 출력 예제
 

```
using System;
using System.Drawing;
using System.Windows.Forms;

class NewForm : Form
{
    string strText = null;
    public NewForm(string str)
    {
        this.Text = str;
        SetStyle(ControlStyles.SupportsTransparentBackColor, true);
    }

    public string TextLabel
    {
        set
        {
            this.strText = value;
        }
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics grfx = pea.Graphics;
        SolidBrush br = new SolidBrush(Color.Black);
        if (strText != null)
            grfx.DrawString(strText, new Font("Times New Roman", 12), br, 10, 10);
    }
}
```

```

        grfx.DrawString(this.strText, this.Font, br, 10, 7);
    }
}
class StructExam4 : Form
{
    Button btn = null;
    public StructExam4()
    {
        this.Text = "KnownColor";
        this.IsMdiContainer = true;

        Array arr = System.Enum.GetValues(typeof(KnownColor));
        NewForm[] frm = new NewForm[arr.Length];

        for (int i = 0; i < arr.Length; i++)
        {
            frm[i] = new NewForm(arr.GetValue(i).ToString());
            frm[i].TextLabel = arr.GetValue(i).ToString();
            frm[i].BackColor = Color.FromName(arr.GetValue(i).ToString());
            frm[i].SetBounds(0, 0, 200, 50);
            frm[i].MdiParent = this;
            frm[i].Show();
        }
        btn = new Button();
        btn.Text = "정렬하기";
        btn.Click += new EventHandler(this.btn_Click);
        btn.SetBounds(0, 0, 100, 50);
        this.Controls.Add(btn);
    }
    static void Main(string[] args)
    {
        Application.Run(new StructExam4());
    }
}

public void btn_Click(object Sender, EventArgs e)
{
    this.LayoutMdi(MdiLayout.TileVertical);
}
}

```

- ColorDialog를 이용한 색상 변경
  - using System;
  - using System.Drawing;
  - using System.Windows.Forms;

```

class StructExam5 : Form
{
    Button btn1 = null, btn2 = null;
    public StructExam5()

```

```

{
    btn1 = new Button();
    btn1.Text = "배경색 설정";
    btn1.SetBounds(10, 10, 100, 50);
    btn1.Click += new EventHandler(btn_Click);

    btn2 = new Button();
    btn2.Text = "전경색 설정";
    btn2.SetBounds(120, 10, 100, 50);
    btn2.Click += new EventHandler(btn_Click);

    this.Controls.Add(btn1);
    this.Controls.Add(btn2);
}

static void Main(string[] args)
{
    Application.Run(new StructExam5());
}

protected override void OnPaint(PaintEventArgs pea)
{
    Graphics grfx = pea.Graphics;
    SolidBrush br = new SolidBrush(this.ForeColor);
    Font font = new Font("돋움", 20);
    grfx.DrawString("글자색 변경", font, br, 10, 70);
}

protected void btn_Click(object sender, System.EventArgs e)
{
    ColorDialog colorDlg = new ColorDialog();
    colorDlg.AllowFullOpen = false;
    colorDlg.ShowHelp = true;

    if ((Button)sender == btn1) // 배경색 변경
    {
        colorDlg.Color = this.BackColor;
        if (colorDlg.ShowDialog() == DialogResult.OK)
            this.BackColor = colorDlg.Color;
    }
    else // 전경색 변경
    {
        colorDlg.Color = this.ForeColor;
        if (colorDlg.ShowDialog() == DialogResult.OK)
            this.ForeColor = colorDlg.Color;
    }
}
}

```

# 03

Chapter

## Graphics 개체 사용하기

## 01 Graphics 클래스

- **Graphics 클래스**

- System.Drawing 네임스페이스에 포함되어 있으며, GDI+ 그리기 화면을 캡슐화한 클래스임

- **Graphics 객체 얻는 방법**

- Control 클래스를 상속받은 클래스에서 Paint 이벤트 핸들러를 통해 얻기
- Control 클래스의 OnPaint() 메서드를 재정의해 얻기
- 컨트롤 클래스의 CreateGraphics 메서드를 사용해 얻어오기
- 몇몇 컨트롤이 제공하는 사용자 그리기(Owner draw) 기능을 이용해 얻어오기
- 비트맵과 메타파일 등에서는 static 메서드인 Graphics.FromImage(...)을 이용해 얻기
- 프린트 할 경우에는 PrintPage 이벤트 핸들러가 전달하는 PrintEventArgs 형의 객체를 통해 얻기
- Win32 코드를 사용할 경우 static 메서드인 Graphics.FromHwnd와 Graphics.FromHdc를 사용해 얻음

[11]

 비트교육센터

### Graphics 객체 얻는 방법

- Control 클래스를 상속받은 클래스에서 Paint 이벤트 핸들러를 통해 얻기
  - 무효화 영역 관련, Graphics 객체 얻는 매개변수가 들어와야 함
  - GDI+에서 가장 많이 사용하는 방법 중 하나임
  - 예제)

```
using System;
using System.Drawing;
using System.Windows.Forms;
class GDIExam1 : Form
{
    public GDIExam1()
    {
        this.Text = "Graphics 객체 얻기1";

        // paint 이벤트 핸들러 등록
        this.Paint += new PaintEventHandler(GDIExam_Paint);
    }

    static void Main(string[] args)
    {
        Application.Run(new GDIExam1());
    }

    // sender : Paint 이벤트가 적용될 객체를 가리킴
    // pea    : Paint 핸들러 내부 사용시 필요한 인자
```

```

//      2개의 속성을 가지고 있음
- Graphics (그래픽 출력 객체) : Graphics(자료형)
    Rectangle(무효한 사각형 인자) : Rectangle(자료형)

public void GDIExam_Paint(object sender, PaintEventArgs pea)
{
    Graphics grfx = pea.Graphics;
    grfx.FillRectangle(new SolidBrush(Color.Blue), this.ClientRectangle);
}
}

```

- Control 클래스의 OnPaint() 메서드를 재정의해 얻기

- 무효화 영역관련, Graphics객체 얻는 매개변수가 들어와야함
- 이벤트 핸들러 등록과정이 필요 없기 때문에 간단히Graphics 객체를 얻을 수 있다
- OnPaint()메서드는 Control 클래스에 정의되어 있음
- 예제)

```

using System;
using System.Drawing;
using System.Windows.Forms;
class GDIExam2 : Form
{
    public GDIExam2()
    {
        this.Text = "Graphics 개체 얻기2";
    }

    static void Main(string[] args)
    {
        Application.Run(new GDIExam2());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics grfx = pea.Graphics;
        grfx.FillRectangle(new SolidBrush(Color.Blue), this.ClientRectangle);
    }
}

```

- 컨트롤 클래스의 CreateGraphics 메서드를 사용해 얻어오기

- 사용이 끝나면 Dispose()메서드를 사용하여 반드시 해제해야 함
- 주로 윈폼 위에 배치된 컨트롤에서 컨트롤들의 개별 화면에 GDI+작업을 직접 처리할 때 주로 사용됨
- 예제)

```

using System;
using System.Drawing;
using System.Windows.Forms;
class GDIExam3 : Form
{
    Button btn = null;

```

```

public GDIExam3()
{
    this.Text = "Graphics 개체 얻기3";
    btn = new Button();
    btn.Text = "버튼위에 GDI+ 출력";
    btn.SetBounds(10, 10, 200, 100);
    btn.Click += new EventHandler(btn_Click);
    this.Controls.Add(btn);
}

static void Main(string[] args)
{
    Application.Run(new GDIExam3());
}

public void btn_Click(object sender, EventArgs e)
{
    Graphics grfx = btn.CreateGraphics();
    grfx.FillRectangle(new SolidBrush(Color.Blue), btn.ClientRectangle);
    grfx.Dispose();
    // Dispose()시키지 않으면 해당 개체의 Graphics 리소스가 회수되지 않기
    // 때문에 시스템에 문제를 일으킬 수 있음
    // 아래 구문이 더 깔끔함
    // using(Graphics grfx = btn.CreateGraphics())
    // {
    //     grfx.FillRectangle(new SolidBrush(Color.Blue), this.ClientRectangle);
    // }

}
}

```

- 몇몇 컨트롤이 제공하는 사용자 그리기(Owner draw) 기능을 이용해 얻어오기
  - MeasureItem or DrawItem 등의 이벤트들은 MeasureItemEventArgs or DrawItemEventArgs 형의 객체를 전달함
  - MeasureItem을 통해 개체를 얻을 수 있는 컨트롤 ( ListBox, ComboBox, CheckListBox, TabControl, MenuItem)
  - DrawItem을 통해 개체를 얻을 수 있는 컨트롤 ( ListBox, ComboBox)
  - 메뉴에 사용자가 원하는 아이콘을 넣거나 탭 컨트롤 상단에 이미지와 색상을 넣는 작업을 하고 싶다면 이를 통해 구현하게 됨
  - 예제)

```

using System;
using System.Drawing;
using System.Windows.Forms;
class GDIExam4 : Form
{
    ListBox lstbox = null;
    public GDIExam4()
    {

```

```

this.Text = "Graphics 개체 얻기4";
lstbox = new ListBox();
lstbox.SetBounds(10, 10, 200, 100);
lstbox.Items.Add("사과");
lstbox.Items.Add("포도");
lstbox.Items.Add("수박");
lstbox.DrawItem += new
System.Windows.Forms.DrawItemEventHandler(GDIExam_DrawItem);
lstbox.MeasureItem += new
System.Windows.Forms.MeasureItemEventHandler(GDIExam_MeasureItem);

this.Controls.Add(lstbox);
this.Load += new EventHandler(GDIExam_Load);
}

static void Main(string[] args)
{
    Application.Run(new GDIExam4());
}

private void GDIExam_Load(object sender, EventArgs e)
{
    lstbox.DrawMode = DrawMode.OwnerDrawVariable;
    //lstbox.DrawMode = DrawMode.OwnerDrawFixed;
}

// 먼저 호출됨
private void GDIExam_MeasureItem(object sender, MeasureItemEventArgs e)
{
    Graphics g = e.Graphics;
    Console.WriteLine("{0} : MeasureItem 이벤트 실행", e.ToString());
}

// ListBox가 다시 그려질 때마다 호출됨(데이터 갯수만큼 호출됨)
private void GDIExam_DrawItem(object sender, DrawItemEventArgs e)
{
    Graphics g = e.Graphics;

    Brush brush = Brushes.Black;

    switch (e.Index)
    {
        case 0:
            brush = Brushes.Red;
            break;
        case 1:
            brush = Brushes.Violet;
            break;
    }
}

```

```

        case 2:
            brush = Brushes.Green;
            break;
        }
        g.DrawString(lstbox.Items[e.Index].ToString(),
            e.Font, brush, e.Bounds, StringFormat.GenericDefault);

        Console.WriteLine("{0} : DrawItem 이벤트 실행", e.ToString());
    }
}

```

- 비트맵과 메타파일 등에서는 static 메서드인 Graphics.FromImage(...)을 이용해 얻기
  - 이미지에 대한 Graphics 객체를 얻을 때 사용
  - 예제)

```

using System;
using System.Drawing;
using System.Windows.Forms;
class GDIExam5 : Form
{
    Button btn = null;
    Image image = null;
    public GDIExam5()
    {
        this.Text = "Graphics 객체 얻기5";
        btn = new Button();
        btn.Text = "그림위에 글씨 쓰기";
        btn.SetBounds(10, 100, 200, 100);
        btn.Click += new EventHandler(btn_Click);
        this.Controls.Add(btn);
    }

    protected override void OnPaint(PaintEventArgs e)
    {
        Graphics grfx = e.Graphics;
        if (image != null)
            grfx.DrawImage(image, 0, 0);
    }

    static void Main(string[] args)
    {
        Application.Run(new GDIExam5());
    }

    public void btn_Click(object sender, EventArgs e)
    {
        Image imageFile = Image.FromFile("ocean.jpg");
        Graphics grfx = Graphics.FromImage(imageFile);

        Font font = new Font("돋움", 20);
    }
}

```

```

Brush brush = Brushes.Pink;

grfx.DrawString("이미지에 글자쓰기", font, brush, 10, 10);
grfx.Dispose();

imageFile.Save("ocean.gif", System.Drawing.Imaging.ImageFormat.Gif);
//System.Drawing.Imaging.ImageFormat.Gif 확장자 변환 가능

this.image = Image.FromFile("ocean.gif");
this.Invalidate(this.ClientRectangle);
}

}

```

- 프린트 할 경우에는 PrintPage 이벤트 핸들러가 전달하는 PrintPageEventArgs 형의 개체를 통해 얻기

- system.Drawing.Printing.PrintDocument 클래스의 PrintPage 이벤트 이용
- 예제)

```

using System;
using System.Drawing;
using System.Windows.Forms;
using System.Drawing.Printing;
class GDIExam6 : Form
{
    Button btn = null;
    public GDIExam()
    {
        this.Text = "Graphics 개체 얻기6";
        btn = new Button();
        btn.Text = "문서를 프린트합니다.";
        btn.SetBounds(10, 10, 200, 100);
        btn.Click += new EventHandler(btn_Click);
        this.Controls.Add(btn);
    }

    static void Main(string[] args)
    {
        Application.Run(new GDIExam());
    }

    public void btn_Click(object sender, EventArgs e)
    {
        try
        {
            PrintDocument pd = new PrintDocument();           // 1
            pd.PrintPage += new PrintPageEventHandler(this.pd_PrintPage); // 2
            pd.Print();                                // 3
        }
        catch (Exception ex)
        {

```

```

        MessageBox.Show("프린터 도중 예외 발생 :" + ex.ToString());
    }
}

private void pd_PrintPage(object sender, PrintPageEventArgs ppe)//4)
{
    string strText = DateTime.Today + " : 문서 작성자 [CJK] ";
    ppe.HasMorePages = false; //HasMorePages--> true 라면 무한반복

    Graphics g = ppe.Graphics;                                // 5)
    Pen pen = new Pen(Color.Black, 2);
    for (int i = 0; i < this.ClientSize.Width; i += 20)
        g.DrawLine(pen, i, 0, i, this.ClientSize.Height);

    for (int j = 0; j < this.ClientSize.Height; j += 20)
        g.DrawLine(pen, 0, j, this.ClientSize.Width, j);

    g.DrawString(strText, this.Font, Brushes.Black, 10,
    this.ClientSize.Height + 20);
}
}

```

- Win32 코드를 사용할 경우 static메서드인 Graphics.FromHwnd와 Graphics.FromHdc를 사용해 얻음

- 예제)

```

using System;
using System.Drawing;
using System.Windows.Forms;
class GDIEexam7 : Form
{
    Button btn1 = null;
    Button btn2 = null;
    public GDIEexam7()
    {
        this.Text = "Graphics 개체 얻기7";
        btn1 = new Button();
        btn1.Text = "Graphics.FromHwnd 이용";
        btn1.SetBounds(10, 10, 200, 100);
        btn1.Click += new EventHandler(btn_Click);

        btn2 = new Button();
        btn2.Text = "Graphics.FromHdc 이용";
        btn2.SetBounds(10, 130, 200, 100);
        btn2.Click += new EventHandler(btn_Click);

        this.Controls.Add(btn1);
        this.Controls.Add(btn2);
    }
}

```

```

static void Main(string[] args)
{
    Application.Run(new GDIExam7());
}

[System.Runtime.InteropServices.DllImportAttribute("gdi32.dll")]
private static extern bool Ellipse(
    IntPtr hdc, // handle to DC
    int nLeftRect, // x-coord of upper-left corner of rectangle
    int nTopRect, // y-coord of upper-left corner of rectangle
    int nRightRect, // x-coord of lower-right corner of rectangle
    int nBottomRect // y-coord of lower-right corner of rectangle
);

public void btn_Click(object sender, EventArgs e)
{
    if ((Button)sender == btn1)
    {
        IntPtr hwnd = new IntPtr();
        hwnd = this.Handle;

        Graphics grfx = Graphics.FromHwnd(hwnd);
        grfx.FillRectangle(Brushes.Blue, this.ClientRectangle);
        grfx.Dispose();

        hwnd = btn1.Handle;
        grfx = Graphics.FromHwnd(hwnd);
        grfx.DrawRectangle(new Pen(Color.Pink, 5), 10, 10, 180, 80);
        grfx.Dispose();
    }
    else
    {
        Graphics g = this.btn2.CreateGraphics();
        g.DrawEllipse(Pens.Blue, 10, 10, 100, 70);

        IntPtr hdc = new IntPtr();
        hdc = g.GetHdc(); //API 함수를 사용하기 위해 얻어냄
        Ellipse(hdc, 100, 10, 50, 50);

        Graphics new_g = Graphics.FromHdc(hdc);
        new_g.DrawRectangle(new Pen(Color.Blue, 5), 10, 10, 180, 80);

        g.ReleaseHdc(hdc);

        g.Dispose();
    }
}
}

```

## 01 Graphics 클래스

### • Graphics 클래스의 주요 메서드

- Clear(전체 그리기 화면을 지우고, 화면을 배경색으로 채움)
- DrawArc, DrawBezier(두개의 끝점을 잇는 Bzier 스플라인)
- DrawCurve, DrawEllipse, DrawImage, DrawLine, DrawPath, DrawPie, DrawPolygon(다각형그리기)
- DrawRectangle, DrawString, FillEllipse, FillPath, FillPolygon,
- FillRectangle, FillRegion

[12]

 비트교육센터

### 사용 예제

#### • GDI+ 그래픽 그리기

```
using System;
using System.Drawing;
using System.Windows.Forms;
class GDIExam8 : Form
{
    public GDIExam8()
    {
        this.Text = "그래픽 그리기";
        this.Size = new Size(200, 200);
    }

    static void Main(string[] args)
    {
        Application.Run(new GDIExam8());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;
        Pen pen = new Pen(Color.Black, 2);
        g.DrawLine(pen, 10, 10, 190, 190);
        g.DrawRectangle(pen, 10, 10, 100, 100);
        g.DrawEllipse(pen, 50, 50, 100, 100);
    }
}
```

```

        g.DrawArc(pen, 100, 100, 80, 80, 0, -90);
    }
}

• 다각형과 타원 그리기

using System;
using System.Drawing;
using System.Windows.Forms;
class GDIExam9 : Form
{
    Point[] point1 = new Point[5];
    Point[] point2 = new Point[3];

    public GDIExam9()
    {
        this.Text = "다각형과 타원 그리기";
        this.Size = new Size(300, 400);

        point1[0] = new Point(10, 20);
        point1[1] = new Point(20, 70);
        point1[2] = new Point(50, 100);
        point1[3] = new Point(10, 150);
        point1[4] = new Point(100, 100);

        point2[0] = new Point(100, 10);
        point2[1] = new Point(10, 100);
        point2[2] = new Point(190, 100);
    }

    static void Main(string[] args)
    {
        Application.Run(new GDIExam9());
    }

protected override void OnPaint(PaintEventArgs pea)
{
    Graphics g = pea.Graphics;
    Pen pen = new Pen(Color.Red, 2); // 빨간색 2의 두께를 갖는 선으로
    g.DrawPolygon(pen, point1); // 5개 점을 경유하는 다각형 그림

    pen = new Pen(Color.Blue, 3); // 파랑색 3의 두께를 갖는 선으로
    g.DrawPolygon(pen, point2); // 3개의 점을 경유하는 다각형

    pen = new Pen(Color.Black, 1); // 두께 1의 검은색 선으로
    for (int i = 0; i < 200; i += 20) // 9개의 타원을 그림
    {
        g.DrawEllipse(pen, 70, 130, i, i + 50);
    }
}

```

```

        }
    }

• FillXXX 계열 메서드 사용하기

using System;
using System.Drawing;
using System.Windows.Forms;
using System.Drawing.Drawing2D;
class GDIExam10 : Form
{
    Point[] point = new Point[3];
    public GDIExam10()
    {
        this.Text = "FillXXX 계열 메서드";
        point[0] = new Point(100, 10);
        point[1] = new Point(10, 100);
        point[2] = new Point(190, 100);
    }

    static void Main(string[] args)
    {
        Application.Run(new GDIExam10());
    }

protected override void OnPaint(PaintEventArgs pea)
{
    Graphics g = pea.Graphics;
    g.FillEllipse(Brushes.Brown, 10, 100, 50, 50); // 채워진 타원 그리기

    GraphicsPath graphPath = new GraphicsPath();
    graphPath.AddEllipse(0, 0, 150, 100); // 타원 등록
    g.FillPath(Brushes.Red, graphPath); // FillPath를 사용해 영역을 그림

    g.FillPie(Brushes.Yellow, 200, 10, 100, 100, 90, 180);

    g.FillPolygon(Brushes.Green, point);

    g.FillRectangle(Brushes.Magenta, 50, 150, 100, 100);

    Rectangle fillRect = new Rectangle(150, 150, 100, 100);
    Region fillRegion = new Region(fillRect);
    g.FillRegion(Brushes.Orange, fillRegion);
}
}

```

- 그라데이션 효과 주기

```

using System;
using System.Drawing;
using System.Windows.Forms;

```

```

class GDIExam11 : Form
{
    public GDIExam11()
    {
        this.Text = "그라데이션 효과주기";
    }

    static void Main(string[] args)
    {
        Application.Run(new GDIExam11());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;
        for (int i = 0; i < 256; i++)
        {
            // 색상을 바꿔가며, 좌측 상단에서 대각선 방향으로 선을 그림
            g.DrawLine(new Pen(Color.FromArgb(i, 0, 0)), 10, 10, 265 - i, 10 + i);
        }
    }
}

```

- 얼굴 그리기

```

using System;
using System.Drawing;
using System.Windows.Forms;
class GDIExam12 : Form
{
    public GDIExam12()
    {
        this.Text = "얼굴 그리기";
    }

    static void Main(string[] args)
    {
        Application.Run(new GDIExam12());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;
        g.DrawArc(Pens.Black, 70, 20, 130, 180, 180, -180); // 아랫턱 그리기

        for (int i = 0; i < 15; i++)
        {
            // 머리카락 그리기
            g.DrawArc(Pens.Brown, 50 + 2 * i, 50, 100, 160, 130, 80);
            g.DrawArc(Pens.Brown, 20 + 2 * i, 50, 190, 160, 140, 100);
        }
    }
}

```

```
        g.DrawArc(Pens.Brown, 90 + 2 * i, 50, 100, 160, 200, 260);
        g.DrawArc(Pens.Brown, 80 + 3 * i, 50, 100, 90, 130, 80);
    }

    g.DrawEllipse(Pens.Black, 90, 120, 20, 25);           // 눈
    g.DrawEllipse(Pens.Black, 155, 120, 20, 25);         // 눈
    g.FillEllipse(Brushes.Blue, 93, 130, 15, 15);       // 눈동자
    g.FillEllipse(Brushes.Blue, 158, 130, 15, 15);       // 눈동자
    g.DrawArc(Pens.Black, 110, 150, 50, 40, 0, 180);
}
```

## 01 Graphics 클래스

### • 무효화 영역과 Invalidate() 메서드

- Invalidate 메서드를 이용한 무효화 처리
- Invalidate() 메서드를 사용해 영역 갱신( 표면이 화면일 경우만 가능, 그리기 표면이 프린터나 이미지일 경우에는 Invalidate() 메서드를 이용한 무효화 메커니즘이 작동되지 않음)

[13]

 비트교육센터

### 사용 예제

- Invalidate 메서드를 이용한 무효화 처리

```
using System;
using System.Drawing;
using System.Windows.Forms;
class GDIExam13 : Form
{
    public GDIExam13()
    {
        this.Text = "Invalidate 예제";
        this.Click += new EventHandler(this.GDIExam_Click);
    }
    static void Main(string[] args)
    {
        Application.Run(new GDIExam13());
    }
    private void DrawGraphics()
    {
        // Graphics 객체 얻기
        Graphics g = this.CreateGraphics();
        // 사각형 테두리 그리기
        g.DrawRectangle(Pens.Black, 10, 10, 200, 200);
    }
}
```

```

        // 사각형 안쪽을 노란색으로 채움
        g.FillRectangle(Brushes.Yellow, 20, 20, 180, 180);
        // 문자열 출력
        g.DrawString("사라지는 색상", this.Font, Brushes.Black, 30, 100);

    }

    // 마우스 왼쪽 버튼 클릭시 호출
    private void GDIExam_Click(object sender, System.EventArgs e)
    {
        this.DrawGraphics();
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        // 자동으로 화면 갱신
        // this.DrawGraphics();
    }

}

• Invalidate()메서드를 사용해 영역 갱신하기

using System;
using System.Drawing;
using System.Windows.Forms;
class GDIExam14 : Form
{
    int signal = 0;
    string[] str = new string[3] { "빨강", "노랑", "녹색" };
    public GDIExam14()
    {
        this.Text = "신호등 예제";
        this.Size = new Size(150, 400);

        Timer time = new Timer(); // timer 객체 생성, 메서드 호출
        time.Interval = 1000;
        time.Enabled = true;
        time.Tick += new EventHandler(time_Tick);
    }

    static void Main(string[] args)
    {
        Application.Run(new GDIExam14());
    }

    // 1초에 한번씩 호출
    private void time_Tick(object sender, EventArgs ea)
    {
        Random rnd = new Random();

```

```

signal = rnd.Next(3);           // 난수 발생

Console.WriteLine(str[signal] + " 발생");

// 특정 영역 갱신
this.Invalidate(new Rectangle(10, 10, 120, 350));
//      this.Invalidate(); // 화면 전체 갱신
this.Update();
}

// 신호등 외곽선 그리기
private void DrawOutLine(Graphics g)
{
    Pen pen = new Pen(Color.White, 3);
    g.FillRectangle(Brushes.Black, 10, 10, 120, 350);
    g.DrawEllipse(pen, 20, 20, 100, 100);      //빨강 테두리 그리기
    g.DrawEllipse(pen, 20, 130, 100, 100);     // 노랑 테두리 그리기
    g.DrawEllipse(pen, 20, 240, 100, 100);     // 녹색 테두리 그리기
}

protected override void OnPaint(PaintEventArgs pea)
{
    Console.WriteLine("Invalidate 영역 = " + pea.ClipRectangle);
    Graphics g = pea.Graphics;
    DrawOutLine(g);                         // 신호등 외곽선 그리기

    switch (signal)
    {
        case 0: // red
            g.FillEllipse(Brushes.Red, 20, 20, 100, 100);
            break;
        case 1: // yellow
            g.FillEllipse(Brushes.Yellow, 20, 130, 100, 100);
            break;
        case 2: // green
            g.FillEllipse(Brushes.Green, 20, 240, 100, 100);
            break;
    }
}
}

```

# **04**

Chapter

## 텍스트출력, 폰트 사용

- **Font 클래스**

- 화면에 문자열 출력 시 Graphics 클래스의 Drawstring()메서드 사용

```
public void DrawString( string, Font, Brush, PointF)
public void DrawString( string, Font, Brush, RectangleF)
public void DrawString( string, Font, Brush, PointF, StringFormat)
public void DrawString( string, Font, Brush, RectangleF, StringFormat)
public void DrawString( string, Font, Brush, float)
public void DrawString( string, Font, Brush, float, StringFormat)
```

- 이 메서드 인자 중 Font 설정을 바꾸면, 출력되는 문자열의 글꼴 수정 가능

- **Font Family 클래스**

- 글꼴의 종류를 지정할 때 사용

[15]



### Drawstring()메서드

- string : 화면에 출력될 문자열
- Brush : 글자의 색상과 테두리 설정시 사용
- PointF, RectnagleF, float : 문자열이 출력될 좌표 지정
- StringFormat : 맞춤 및 줄 간격과 같은 텍스트 레이아웃 정보를 설정시 사용

### Font 클래스의 생성자 중 자주 사용되는 형태

- `Font( string, float)` // 글꼴 이름, 글꼴 크기
- `Font( string, float, FontStyle)` // Bold, italic, Regular, Strikeout(중간에 밑줄), UnderLine속성 설정

## 01 Font 클래스

### • GDI+에서 화면에 문자열 출력 단계

#### 1) Font 클래스 개체 생성

```
ex) Font font = new Font("Timesroman", 20);
```

#### 2) Brush 클래스 개체 생성

```
ex) SolidBrush brush = new SolidBrush(color.Blue);
```

#### 3) 출력할 좌표 및 텍스트 레이아웃 정보 설정

```
ex) RectangleF rect = new RectangleF(50, 10, 200, 30);
```

#### 4) Graphics 개체를 얻은 후 DrawString()메서드로 문자열 출력

```
ex) g.DrawString("안녕하세요", font, brush, rect);
```

[16]

 비트교육센터

### 사용 예제

#### • 글꼴 출력하기

```
using System;
using System.Drawing;
using System.Windows.Forms;
using System.Drawing.Text;
public class TextExam1 : Form
{
    public TextExam1()
    {
        this.Text = "Font 예제";
    }

    static void Main()
    {
        Application.Run(new TextExam1());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;

        // 글꼴 생성 객체
        Font font = new Font("Timesroman", 20);      // 글꼴 지정
        SolidBrush brush = new SolidBrush(Color.Blue); // 브러쉬 생성
```

```

    RectangleF rect = new RectangleF(50, 10, 200, 30); // 문자열 영역 설정

    // 화면에 문자를 출력할 때 사용하는 메서드
    g.DrawString("안녕하세요", font, brush, rect); // 출력

    font = new Font("돋움", 10); // 글꼴을 돋움10으로 변경
    g.DrawString("폰트 예제입니다.", font, brush, 50, 50); // 바뀐 글꼴로 문자열 출력

    font = new Font("궁서", 15);
    brush = new SolidBrush(Color.Red);
    PointF point = new PointF(10, 10);
    StringFormat sf = new StringFormat();
    sf.FormatFlags = StringFormatFlags.DirectionVertical; // 글꼴 출력을 세로로 지정

    g.DrawString("C# Font 다루기", font, brush, point, sf);
}
}

```

- 그림자 효과 주기

```

using System;
using System.Drawing;
using System.Windows.Forms;
using System.Drawing.Text;

public class TextExam : Form
{
    public TextExam()
    {
        this.Text = "그림자 효과 주기";
        this.Size = new Size(400, 150);
    }
    static void Main()
    {
        Application.Run(new TextExam());
    }
    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;

        // 글꼴 생성
        Font font = new Font("Timesroman", 30, FontStyle.Italic);
        string str = "빨주노초파남보";
        Color[] color = {Color.Red, Color.Orange, Color.Yellow,
                        Color.Green, Color.Blue,
                        Color.Magenta,
                        Color.Violet};

        for (int i = 0; i < 7; i++)
        {
            g.DrawString(str, font, color[i], 10 + i * 50, 100);
        }
    }
}

```

```

        // 출력 : 색상을 바꾸어 가면서 겹쳐서 출력
        g.DrawString(str, font, new SolidBrush(color[6 - i]), (20 + i), 30 + (i * 2));
        System.Console.WriteLine("{0},{1}", 20 + i, 30 + 2 * i);
    }
}
}

```

- 글꼴 목록 확인

```

using System;
using System.Drawing;
using System.Drawing.Text;
class FontFamilyExam
{
    static void Main(string[] args)
    {
        InstalledFontCollection installfont = new InstalledFontCollection();
        FontFamily [] ff = installfont.Families;
        for (int i = 0; i < ff.Length; i++)
        {
            System.Console.WriteLine("FontName-[{0}] : {1}", i, ff[i].Name);
        }
    }
}

```

- FontFamily

```

using System;
using System.Drawing;
using System.Windows.Forms;
using System.Drawing.Text;

public class TextExam3 : Form
{
    public TextExam3()
    {
        this.Text = "FontFamily 예제";
        this.AutoScrollMinSize = new Size(200, 500);
    }

    static void Main()
    {
        Application.Run(new TextExam3());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;

        int height = 20;

        // 시스템에 있는 글꼴 얻기 : 2가지 방법

```

```

FontFamily[] fontname = FontFamily.Families;
//FontFamily[] fontname = FontFamily.GetFamilies(g);

for (int i = 0; i < fontname.Length; i++)
{
    if (fontname[i].IsStyleAvailable(FontStyle.Regular))
    {
        Font font = new Font(fontname[i], 10);
        g.DrawString(fontname[i].Name + " : [ 글꼴, ABC 123 ]", font,
        Brushes.Black, 10, height);
        height += font.Height + 5;
        System.Console.WriteLine("{0}-{1}", i, fontname[i].Name);
        font.Dispose();           // 리소스 해제
    }
}
}
}

```

- 글꼴 수치 정보 보기

```

using System;
using System.Drawing;
public class TextExam4
{
    static void Main()
    {
        float ascent = 0.0f;
        float descent = 0.0f;
        float linespacing = 0.0f;
        float height = 0.0f;
        string format = "{0,-30}{1,10}{2,10},{3,10},{4,15}";

        FontFamily[] ff = FontFamily.Families;
        System.Console.WriteLine(format, "FontName", "Ascent", "Descent", "Height",
        "Linespacing");

        for (int i = 0; i < ff.Length; i++)
        {
            ascent = ff[i].GetCellAscent(FontStyle.Regular);
            descent = ff[i].GetCellDescent(FontStyle.Regular);
            linespacing = ff[i].GetLineSpacing(FontStyle.Regular);
            height = ff[i].GetEmHeight(FontStyle.Regular);
            System.Console.WriteLine(format, ff[i].Name, ascent, descent, height,
            linespacing);
        }
    }
}

```

- 글꼴 수치를 화면에 표시하기

```

using System;
using System.Drawing;

```

```

using System.Windows.Forms;
public class TextExam5 : Form
{
    public TextExam5()
    {
        this.Text = "글꼴 수치 예제";
        this.Size = new Size(360, 170);
    }

    static void Main()
    {
        Application.Run(new TextExam5());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;
        float EmSizeGraphicsUnit = 50; // 글꼴 크기

        // 글꼴의 Em 높이
        float EmSizeDesignUnit;

        // 어센트(머리선 – 기준선 간격 : 글자가 써지는 영역)
        float AscentGraphicsUnit;

        // 디센트(기준선 – 꼬리선 간격 : 글자에 따라 사용되는 영역)
        float DescentGraphicsUnit;

        // 라딩(꼬리선 – 다음 라인 : 문자 출력 공간 사이의 간격)
        float LinespacingGraphicsUnit;

        // 글자출력 기준선
        PointF baseLine = new PointF(30, 30);

        Font font = new Font("궁서체", EmSizeGraphicsUnit); // 궁서체 Font객체 생성
        FontFamily ff = new FontFamily("궁서체");

        EmSizeDesignUnit = ff.GetEmHeight(FontStyle.Regular);

        // DesignUnit 좌표 구하기(em 단위 좌표)
        float ascent = ff.GetCellAscent(FontStyle.Regular); // 궁서체 글꼴 정보를 변수에
저장
        float descent = ff.GetCellDescent(FontStyle.Regular);
        float linespacing = ff.GetLineSpacing(FontStyle.Regular);

        // Graphicsunit 좌표 단위로 변환
        AscentGraphicsUnit = ascent * (EmSizeGraphicsUnit / EmSizeDesignUnit);
        DescentGraphicsUnit = descent * (EmSizeGraphicsUnit / EmSizeDesignUnit);
        LinespacingGraphicsUnit = linespacing * (EmSizeGraphicsUnit /

```

```

EmSizeDesignUnit);

g.DrawString("글자 화면 출력", font, Brushes.Black, baseLine);

g.DrawLine(Pens.Red, baseLine, new PointF(baseLine.X + 300, baseLine.Y));

PointF p = new PointF(baseLine.X, baseLine.Y + LinespacingGraphicsUnit);
g.DrawLine(Pens.Blue, p, new PointF(p.X + 300, p.Y));
p = new PointF(baseLine.X, baseLine.Y + LinespacingGraphicsUnit -
AscentGraphicsUnit);
g.DrawLine(Pens.Magenta, p, new PointF(p.X + 300, p.Y));
p = new PointF(baseLine.X, baseLine.Y + LinespacingGraphicsUnit +
DescentGraphicsUnit);
g.DrawLine(Pens.Green, p, new PointF(p.X + 300, p.Y));

}
}

```

- 사각형 영역 안에 문자열 출력하기

```

using System;
using System.Drawing;
using System.Windows.Forms;
public class TextExam6 : Form
{
    public TextExam6()
    {
        this.Text = "문자열 출력";
        this.Size = new Size(450, 200);
    }

    static void Main()
    {
        Application.Run(new TextExam6());
    }

protected override void OnPaint(PaintEventArgs pea)
{
    Graphics g = pea.Graphics;
    string str = "한산섬 달밝은 밤에 수루에 훌로 앉아";
    str += "큰 칼 옆에 차고 깊은 시름하는 차에";
    str += "어디서 일성호가는 남의 애를 끊나니..";
    str += "-난중일기中-";

    Font font = new Font("궁서체", 15);
    Rectangle rect = new Rectangle(20, 20, 400, font.Height * 4);
    g.DrawRectangle(Pens.Red, rect);
    g.DrawString(str, font, Brushes.Black, rect);
    font.Dispose();           // 가비지 컬렉션 호출
}

```

```
}
```

```
// DrawString()메서드는 출력 문자열에 포함된 줄 바꿈 기호를 인식한다.  
// 단, StringFormatFlags.NoWrap 속성이 설정되면 줄바꿈 기호가 무시됨  
protected override void OnPaint(PaintEventArgs pea)
```

```
{
```

```
    Graphics g = pea.Graphics;  
    string str = "한산섬 달밝은 밤에 수루에 훌로 앉아₩n";  
    str += "큰 칼 옆에 차고 깊은 시름하는 차에₩n";  
    str += "어디서 일성호가는 남의 애를 끊나니₩n";  
    str += "-난중일기中-";
```

```
    Font font = new Font("궁서체", 15);
```

```
    StringFormat sf = new StringFormat();           // stringFormat객체 생성  
    sf.FormatFlags = StringFormatFlags.NoWrap;
```

```
    Rectangle rect = new Rectangle(20, 20, 400, font.Height * 4);
```

```
    g.DrawRectangle(Pens.Red, rect.Left, rect.Top, rect.Width, rect.Height);  
    g.DrawString(str, font, Brushes.Black, rect, sf);
```

```
//    g.DrawString(str, font, Brushes.Black, rect);
```

```
    sf.Dispose();
```

```
    font.Dispose();
```

```
}
```

```
// 문자열 정렬
```

```
protected override void OnPaint(PaintEventArgs pea)
```

```
{
```

```
    Graphics g = pea.Graphics;  
    string str = "한산섬 달밝은 밤에 수루에 훌로 앉아₩n";  
    str += "큰 칼 옆에 차고 깊은 시름하는 차에₩n";  
    str += "어디서 일성호가는 남의 애를 끊나니₩n";  
    str += "-난중일기中-";
```

```
    Font font = new Font("궁서체", 13);
```

```
    StringFormat sf = new StringFormat();           // stringFormat객체 생성
```

```
    sf.Alignment = StringAlignment.Center;          // 텍스트를 중앙에 출력
```

```
    sf.LineAlignment = StringAlignment.Center;      // 텍스트를 중앙에 출력
```

```
    Rectangle rect = new Rectangle(20, 20, 400, font.Height * 4);
```

```
    g.DrawRectangle(Pens.Red, rect.Left, rect.Top, rect.Width, rect.Height);  
    g.DrawString(str, font, Brushes.Black, rect, sf);
```

```
    sf.Dispose();
```

```
    font.Dispose();
```

```

    }

// 세로 문자열 출력
// - DirectionRightToLeft : 텍스트를 오른쪽에서 왼쪽으로 출력
// - DirectionVertical : 텍스트를 위에서 아래로 출력

protected override void OnPaint(PaintEventArgs pea)
{
    Graphics g = pea.Graphics;
    string str = "한산섬 달밝은 밤에 수루에 출로 앉아₩n";
    str += "큰 칼 옆에 차고 깊은 시름하는 차에₩n";
    str += "어디서 일성호가는 남의 애를 끊나니.₩n";
    str += "-난중일기中-";

    Font font = new Font("궁서체", 13);

    StringFormat sf = new StringFormat(StringFormatFlags.DirectionVertical);
        // stringFormat객체 생성
    Rectangle rect = new Rectangle(20, 20, 400, font.Height * 6);

    g.DrawRectangle(Pens.Red, rect.Left, rect.Top, rect.Width, rect.Height);
    g.DrawString(str, font, Brushes.Black, rect, sf);

    sf.Dispose();
    font.Dispose();
}

// SetTabStops( float firstTabOffset, float[] tabStops);
// 1인자 : 텍스트 줄의 시작 부분과 텁 사이의 공백(보통 0.0f 입력)
// 2인자 : 텁 정지 사이의 거리(공백 너비)배열
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;

    Font bFont = new Font("돋움", 12, FontStyle.Bold);
    Font font = new Font("돋움", 12);

    StringFormat sf = new StringFormat();
    float[] tab = { 50.0f, 50.0f, 50.0f, 50.0f };
    sf.SetTabStops(0.0f, tab);

    string str1 = "₩t국어₩t영어₩t수학₩t총점₩t평균";
    string str2 = "₩t75₩t80₩t70₩t225₩t75.0";
    string str3 = "₩t55₩t90₩t60₩t205₩t68.4";

    g.DrawString(str1, bFont, Brushes.Black, 10, 30, sf);
    g.DrawString(str2, font, Brushes.Black, 10, 50, sf);
}

```

```
g.DrawString(str3, font, Brushes.Black, 10, 70, sf);

font.Dispose();
bFont.Dispose();

}
```

# **05**

Chapter

펜과 브러쉬, 패스, 클리핑, Region

## 01 Pen, Brush

### • Pen 클래스

- 단넷이 제공하는 표준 색상을 사용해 정의

### • Brush 클래스

- Brush 클래스는 추상 클래스로 직접 인스턴스 생성 불가함

- 상속받은 아래와 같은 클래스로 인스턴스를 생성해서 사용함

선언	클래스	설명
System.Drawing	SolidBrush	- 영역을 단색으로 채움
	TextureBrush	- 영역을 이미지(비트맵, jpg, gif 등)로 채움
System.Drawing	LinearGradientBrush	- 영역을 그라데이션 효과로 채움
	PathGradientBrush	- 영역의 가장자리에서 중심 쪽으로 그라데이션 효과를 주어 채움
	HatchBrush	- 패턴을 사용해 영역을 채움

[18]

비트교육센터

### 사용예제

#### • Pen 개체로 그리기

- Pen 클래스를 이용한 직선 그리기

```
using System;
using System.Drawing;
using System.Windows.Forms;
public class PenExam1 : Form
{
    public PenExam1()
    {
        this.Text = "X자 그리기";
        this.Size = new Size(300, 300);
    }
}
```

```
static void Main()
{
    Application.Run(new PenExam1());
}
```

```
protected override void OnPaint(PaintEventArgs pea)
{
    Graphics g = pea.Graphics;
```

```
// 파랑색, 두께 10
Pen pen1 = new Pen(Color.Blue, 10);
```

```

// 빨강색, 두께 5
Pen pen2 = new Pen(Color.Red, 5);

g.DrawLine(pen1, 50, 50, 250, 250);
g.DrawLine(pen2, 250, 50, 50, 250);
}

}

- DashStyle을 이용한 파선 그리기
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class PenExam2 : Form
{
    public PenExam2()
    {
        this.Text = "파선 그리기";
        this.Size = new Size(450, 200);
    }

    static void Main()
    {
        Application.Run(new PenExam2());
    }

// 모눈 종이 그리기
void DrawGraphPaper(Graphics g)
{
    int i;

    // pens사용
    for (i = 0; i < this.Width; i += 10)
        g.DrawLine(Pens.Blue, i, 0, i, this.Height);

    // pens2사용
    for (i = 0; i < this.Width; i += 10)
        g.DrawLine(Pens.Blue, 0, i, this.Width, i);

}

protected override void OnPaint(PaintEventArgs pea)
{
    Graphics g = pea.Graphics;

    DrawGraphPaper(g);

    DashStyle[] dash = {DashStyle.Custom, DashStyle.Dash,

```

```

DashStyle.DashDot,
DashStyle.DashDotDot, DashStyle.Dot, DashStyle.Solid};

Pen pen = new Pen(Color.Black, 10);

for (int i = 0; i < dash.Length; i++)
{
    pen.DashStyle = dash[i];
    g.DrawLine(pen, 10, 15 + (20 * i), 400, 15 + (20 * i));
}
}
}
}

```

- 사용자 지정 패션 그리기

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class PenExam3 : Form
{
    public PenExam3()
    {
        this.Text = "사용자 지정 패션 그리기";
        this.Size = new Size(450, 200);
    }

    static void Main()
    {
        Application.Run(new PenExam3());
    }

protected override void OnPaint(PaintEventArgs pea)
{
    Graphics g = pea.Graphics;

    Pen pen1 = new Pen(Color.Red, 5);
    Pen pen2 = new Pen(Color.Black, 5);

    // Dot형태의 패션 그리기
    pen1.DashStyle = DashStyle.Dot;
    g.DrawLine(pen1, 10, 20, 410, 20);
    g.DrawRectangle(pen1, 10, 30, 400, 100);

    // 사용자 지정 패션 그리기
    float[] dashpattern = { 15, 10, 5, 10, 20 };
    pen2.DashPattern = dashpattern;
}

```

```

        g.DrawLine(pen2, 10, 20, 410, 20);
        g.DrawRectangle(pen2, 10, 30, 400, 100);

        pen1.Dispose();
        pen2.Dispose();
    }
}

```

- 선 끝 모양 출력하기

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class PenExam4 : Form
{
    public PenExam4()
    {
        this.Text = "선끝 모양";
        this.Size = new Size(300, 300);
    }

    static void Main()
    {
        Application.Run(new PenExam4());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;

        Pen pen = new Pen(Brushes.Black, 10);

        LineCap[] lcap = {LineCap.ArrowAnchor, LineCap.DiamondAnchor,
        LineCap.Flat,
                           LineCap.Round, LineCap.RoundAnchor,
        LineCap.Square,
                           LineCap.SquareAnchor, LineCap.Triangle};

        for (int i = 0; i < lcap.Length; i++)
        {
            pen.StartCap = lcap[i];
            pen.EndCap = lcap[i];
            g.DrawLine(pen, 30, 30 + (30 * i), 100, 30 + (30 * i));
            g.DrawString(lcap[i].ToString(), this.Font, Brushes.Blue, 150, 25 +
            (30 * i));
        }
    }
}

```

- 선의 결합

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;

public class PenExam5 : Form
{
    public PenExam5()
    {
        this.Text = "선의 결합";
    }

    static void Main()
    {
        Application.Run(new PenExam5());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;

        Pen pen = new Pen(Color.Black, 15);

        g.DrawRectangle(pen, 10, 10, 100, 100);
        pen.LineJoin = LineJoin.Bevel;
        g.DrawRectangle(pen, 10, 130, 100, 100);
        pen.LineJoin = LineJoin.Round;
        g.DrawRectangle(pen, 130, 130, 100, 100);
        pen.LineJoin = LineJoin.Miter;
        g.DrawRectangle(pen, 250, 130, 100, 100);
        pen.LineJoin = LineJoin.MiterClipped;
        g.DrawRectangle(pen, 370, 130, 100, 100);

        pen.Dispose();
    }
}
```

• Brush 개체로 그리기

- 솔리드 브러쉬 사용

```
using System;
using System.Drawing;
using System.Windows.Forms;
public class SolidBrushExam : Form
{
    public SolidBrushExam()
    {
        this.Text = "SolidBrush 사용 예";
    }
```

```

static void Main()
{
    Application.Run(new SolidBrushExam());
}

protected override void OnPaint(PaintEventArgs pea)
{
    Graphics g = pea.Graphics;
    SolidBrush brush = new SolidBrush(Color.Black);

    g.FillRectangle(brush, this.ClientRectangle);

    brush.Color = Color.Yellow;
    g.FillRectangle(brush, 100, 100, 100, 100);

    brush.Dispose();
}
}

```

- TextureBrush 사용

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class TextureBrushExam1 : Form
{
    public TextureBrushExam1()
    {
        this.Text = "TextureBrush 예제";
    }

    static void Main()
    {
        Application.Run(new TextureBrushExam1());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;

        Bitmap bmp = new Bitmap("logo.bmp");
        TextureBrush tb = new TextureBrush(bmp);

        g.FillRectangle(tb, 10, 10, 200, 100);
        g.FillEllipse(tb, 100, 150, 100, 100);

        bmp.Dispose();
        tb.Dispose();
    }
}

```

```
    }  
}
```

- TextureBrush의 WrapMode 속성 사용

Clamp : 질감과 그라데이션을 개체 가장자리에 맞춤  
Tile : 질감과 그라데이션을 바둑판 모양으로 배열  
TileFlipX : 질감과 그라데이션을 좌우로 대칭 이동한 다음 바둑판 모양으로 배열  
TileFlipXY: 질감과 그라데이션을 상하좌우로 대칭 이동한 다음 바둑판 모양으로 배열  
TileFilpY : 질감과 그라데이션을 상하로 대칭 이동한 다음 바둑판 모양으로 배열

```
using System;  
using System.Drawing;  
using System.Drawing.Drawing2D;  
using System.Windows.Forms;  
public class TextureBrushExam2 : Form  
{  
    public TextureBrushExam2()  
    {  
        this.Text = "TextureBrush 예제";  
    }  
  
    static void Main()  
    {  
        Application.Run(new TextureBrushExam2());  
    }  
  
    protected override void OnPaint(PaintEventArgs pea)  
    {  
        Graphics g = pea.Graphics;  
  
        Bitmap bmp = new Bitmap("logo.bmp");  
        TextureBrush tb = new TextureBrush(bmp);  
        tb.WrapMode = WrapMode.Tile; // 영역을 채우는  
        방법(title선택)  
        //tb.WrapMode = WrapMode.TileFlipX;  
        //tb.WrapMode = WrapMode.TileFlipXY;  
        //tb.WrapMode = WrapMode.TileFlipY;  
  
        g.FillRectangle(tb, this.ClientRectangle);  
  
        bmp.Dispose();  
        tb.Dispose();  
    }  
}
```

- LinearGradientBrush

```
using System;  
using System.Drawing;
```

```

using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class LinearGradientBrushExam : Form
{
    public LinearGradientBrushExam()
    {
        this.Text = "LinearGradientBrush 예제";
    }

    static void Main()
    {
        Application.Run(new LinearGradientBrushExam());
    }

    protected override void OnPaint(PaintEventArgs pea)
    {
        Graphics g = pea.Graphics;

        Point pt1 = new Point(0, 0);
        Point pt2 = new Point(30, 30);
        Rectangle rect = new Rectangle(0, 0, 50, 50);

        // (0,0) 빨간색 좌표 (30,30) 파란색 설정
        // 두 점사이의 직선을 따라 빨간색에서 파란색으로 그라데이션 효과를
        // 주는 브러쉬로 영역을 채움
        LinearGradientBrush lgb1 = new LinearGradientBrush(pt1, pt2,
Color.Red, Color.Blue);
        LinearGradientBrush lgb2 = new LinearGradientBrush(rect, Color.Yellow,
Color.Magenta, 45.0f);

        g.FillEllipse(lgb1, 30, 50, 100, 100);
        g.FillRectangle(lgb2, 150, 50, 100, 100);

        lgb1.Dispose();
        lgb2.Dispose();
    }
}

```

– PathGradientBrush

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class PathGradientBrushExam : Form
{
    public PathGradientBrushExam()
    {
        this.Text = "PathGradientBrush 예제";
    }
}

```

```

static void Main()
{
    Application.Run(new PathGradientBrushExam());
}

protected override void OnPaint(PaintEventArgs pea)
{
    Graphics g = pea.Graphics;

    GraphicsPath gp = new GraphicsPath();
    gp.AddLine(150, 10, 300, 100);
    gp.AddLine(300, 100, 270, 250);
    gp.AddLine(270, 250, 150, 300);
    gp.AddLine(150, 300, 50, 250);
    gp.AddLine(50, 250, 30, 150);
    gp.AddLine(30, 150, 50, 70);
    gp.AddLine(50, 70, 150, 10);
    gp.CloseFigure();

    PathGradientBrush pgb = new PathGradientBrush(gp);

    // 그라데이션의 가운데 색상을 설정
    pgb.CenterColor = Color.White;

    // 색 배열을 설정
    pgb.SurroundColors = new Color[] { Color.Red, Color.Orange,
    Color.Yellow, Color.Green, Color.Blue, Color.Indigo, Color.Magenta };

    g.FillPath(pgb, gp);
    g.DrawPath(Pens.Black, gp);

    pgb.Dispose();
    gp.Dispose();
}
}

```

#### - HatchBrush

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class HatchBrushExam : Form
{
    public HatchBrushExam()
    {
        this.Text = "HatchBrush 예제";
    }
}

```

```

static void Main()
{
    Application.Run(new HatchBrushExam());
}

protected override void OnPaint(PaintEventArgs pea)
{
    Graphics g = pea.Graphics;

    HatchBrush hb = new HatchBrush(HatchStyle.Divot, Color.Red,
Color.Blue);
    g.FillRectangle(hb, this.ClientRectangle);

    hb = new HatchBrush(HatchStyle.Cross, Color.Orange, Color.Green);
    g.FillRectangle(hb, 50, 50, 100, 100);

    hb = new HatchBrush(HatchStyle.DiagonalBrick, Color.Pink,
Color.Cyan);
    g.FillRectangle(hb, 150, 50, 100, 100);

    hb = new HatchBrush(HatchStyle.Wave, Color.Yellow, Color.Magenta);
    g.FillRectangle(hb, 50, 150, 100, 100);

    hb = new HatchBrush(HatchStyle.ZigZag, Color.White, Color.Black);
    g.FillRectangle(hb, 150, 150, 100, 100);

    hb.Dispose();
}
}

```

- TextBrush : 브러시를 이용한 글자 출력

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;

```

```

public class TextBrushExam : Form
{
    public TextBrushExam()
    {
        this.Text = "TextBrush 예제";
    }
}

```

```

static void Main()
{
    Application.Run(new TextBrushExam());
}

```

```
protected override void OnPaint(PaintEventArgs pea)
```

```
{  
    Graphics g = pea.Graphics;  
    Bitmap bmp = new Bitmap("logo.bmp");  
    Rectangle rect = new Rectangle(0, 0, 20, 20);  
    Font font = new Font("궁서체", 30, FontStyle.Bold);  
  
    Brush brush = new TextureBrush(bmp);  
    g.DrawString("TexTureBrush...", font, brush, 10, 20);  
    brush = new LinearGradientBrush(rect, Color.Yellow, Color.Magenta,  
        45.0f);  
    g.DrawString("LinearGradientBrush...", font, brush, 10, 70);  
    brush = new HatchBrush(HatchStyle.Cross, Color.Orange, Color.Green);  
    g.DrawString("HatchBrush...", font, brush, 10, 120);  
  
    brush.Dispose();  
  
}  
}
```

## 01 GraphicsPath

### • GraphicsPath 클래스

- 다각형을 정의하고 테두리(Draw) 영역(Fill) 을 채우는 GDI+ 도구 클래스임
- 시작과 끝을 갖는 패스를 정의( 열린 그림 : 시작과 끝이 다름, 닫힌 그림 : 시작과 끝이 같음)

### • 시작하고 닫을 때 사용하는 3가지 메서드

메서드	설명
StartFigure()	<ul style="list-style-type: none"><li>- 현재 그림을 닫지 않고 새 그림을 시작</li></ul>
CloseFigure()	<ul style="list-style-type: none"><li>- 현재 그림을 닫고 새 그림을 시작</li><li>- 연결된 일련의 선과 곡선이 현재 그림에 있으면 메서드가 끝점에서 시작점으로 선을 연결하여 루프를 닫음</li></ul>
CloseAllFigure()	<ul style="list-style-type: none"><li>- 이 경로에 있는 열린 그림을 모두 닫고 새 그림을 시작</li><li>- 열린 그림의 각 끝점에서 시작점으로 선을 연결하면 해당 그림이 닫힘</li></ul>

[19]

 비트교육센터

### 사용예제

#### • GraphicsPath 사용하기

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class GraphicsPathExam1 : System.Windows.Forms.Form
{
    public GraphicsPathExam1()
    {
        this.Text = "GraphicsPathExam1";
        this.Size = new Size(500, 500);
    }

    static void Main()
    {
        Application.Run(new GraphicsPathExam1());
    }

    protected override void OnPaint(PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        Pen pen = new Pen(Brushes.Black, 3);

        // 개체 생성
```

```

GraphicsPath gp = new GraphicsPath();

    // 첫번째 그림
    gp.AddLine(10, 10, 100, 100);
    gp.AddEllipse(50, 50, 100, 100);
    // 현재 그림을 닫음
    gp.CloseFigure();

    // 두번째 그림
    gp.StartFigure();
    gp.AddArc(300, 300, 50, 50, 45, 180);
    gp.AddLine(250, 200, 370, 350);
    // 지금까지 열린 그림을 닫음
    gp.CloseAllFigures();

    // 세번째 그림
    gp.AddPie(30, 300, 70, 70, 90, 270);

    // 실제 도형을 그려주는 부분
    g.DrawPath(pen, gp);
    //g.FillPath(Brushes.Black, gp);

    pen.Dispose();
    gp.Dispose();
}
}

```

- Graphics객체 사용하기

- 패스를 닫는 방법 1 : 그림이 끝나는 점과 시작점을 동일하게 처리
- 패스를 닫는 방법 2 : CloseFigure() 사용

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class GraphicsPathExam2 : System.Windows.Forms.Form
{
    public GraphicsPathExam2()
    {
        this.Text = "GraphicsPathExam2";
        this.Size = new Size(250, 150);
    }
    static void Main()
    {
        Application.Run(new GraphicsPathExam2());
    }

    protected override void OnPaint(PaintEventArgs e)
    {

```

```
Graphics g = e.Graphics;
Pen pen = new Pen(Brushes.Black, 3);

GraphicsPath gp = new GraphicsPath();

gp.AddLine(10, 10, 10, 110);
gp.AddLine(10, 110, 110, 110);
gp.AddLine(110, 110, 110, 10);
gp.CloseFigure();

gp.AddLine(120, 10, 120, 110);
gp.AddLine(120, 110, 220, 110);
gp.AddLine(220, 110, 220, 10);

g.DrawPath(pen, gp);

pen.Dispose();
gp.Dispose();
}

}
```

## 01 Region

### • Region 클래스

- GraphicsPath와 비슷한 기능을 갖고 있음
- 도형의 영역을 정의하고 내부를 채우거나 정의된 영역의 일부를 잘라내는 등의 작업에 사용됨
- GraphicsPath와 달리 AddXXX 형태의 메서드가 없음

### • 사용법

- Region클래스의 생성자에 Rectangle, RectangleF 또는 GraphicsPath 객체를 인자로 넣어서 만듬

[20]

 비트교육센터

## 영역처리

### • 두 영역에 대한 처리

- r1.Intersect(r2) : r1과 r2영역의 교집합 영역
- r1.Union(r2) : r1과 r2영역의 합집합 영역
- r1.Xor(r2) : r1과 r2영역의 배타적 합집합(합집합에서 교집합 뺀)영역
- r1.Complement(r2) : r1과 r2의 합집합에서 r1에 해당되는 부분을 뺀 영역
- r1.Exclude(r2) : r1영역 중 r2에 해당하는 영역을 뺀 영역

## 사용예제

### • Region 객체 조합

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class RegionExam1 : Form
{
    Rectangle rect1, rect2;
    Region reg1, reg2;

    public RegionExam1()
    {
        this.Text = "Region 객체 조합 예제";
        this.Size = new Size(700, 200);
    }
    static void Main()
```

```

{
    Application.Run(new RegionExam1());
}

private void DrawRect(Graphics g, string str)
{
    g.DrawRectangle(Pens.Black, rect1);
    g.DrawRectangle(Pens.Black, rect2);
    g.FillRegion(Brushes.Black, reg1);
    g.DrawString(str, this.Font, Brushes.Black, rect2.X + 20, rect1.Y + 130);
}

protected override void OnPaint(PaintEventArgs e)
{
    Graphics g = e.Graphics;

    rect1 = new Rectangle(50, 20, 50, 110);
    rect2 = new Rectangle(20, 50, 110, 50);

    reg1 = new Region(rect1);
    reg2 = new Region(rect2);
    reg1.Complement(reg2);
    this.DrawRect(g, "Complement");

    rect1.X += 130;
    rect2.X += 130;
    reg1 = new Region(rect1);
    reg2 = new Region(rect2);
    reg1.Exclude(reg2);
    this.DrawRect(g, "Exclude");

    rect1.X += 130;
    rect2.X += 130;
    reg1 = new Region(rect1);
    reg2 = new Region(rect2);
    reg1.Intersect(reg2);
    this.DrawRect(g, " Intersect");

    rect1.X += 130;
    rect2.X += 130;
    reg1 = new Region(rect1);
    reg2 = new Region(rect2);
    reg1.Union(reg2);
    this.DrawRect(g, " Union");

    rect1.X += 130;
    rect2.X += 130;
}

```

```

        reg1 = new Region(rect1);
        reg2 = new Region(rect2);
        reg1.Xor(reg2);
        this.DrawRect(g, " Xor");

        reg1.Dispose();
        reg2.Dispose();
    }
}

```

- GraphicsPath와 Region 활용

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class RegionExam2 : Form
{
    public RegionExam2()
    {
        this.Text = " 원 모양 품";
    }

    static void Main()
    {
        Application.Run(new RegionExam2());
    }

    protected override void OnPaint(PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        GraphicsPath shape = new GraphicsPath();
        shape.AddEllipse(0, 0, 100, 100);
        shape.AddRectangle(new Rectangle(100, 0, this.Width - 100, this.Height));
        shape.AddLine(100, 100, 0, 200);
        shape.AddLine(0, 200, 200, 200);
        shape.CloseFigure();
        this.Region = new Region(shape);
        shape.Dispose();

        g.DrawString("GraphicsPath와 Region 활용", this.Font, Brushes.Black, 120, 30);
    }
}

```

## 01 Clipping

- Clipping

- 그리기 표면에서 그리기 연산이 실제 수행되는 영역

[21]

 비트교육센터

### 사용예제

- 클리핑 영역 출력

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class ClippingExam1 : System.Windows.Forms.Form
{
    static void Main()
    {
        Application.Run(new ClippingExam1());
    }

    protected override void OnPaint(PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        Pen p = new Pen(Color.Black);
        p.DashStyle = DashStyle.Dot;
        g.DrawRectangle(p, 100, 100, 100, 100);
        g.DrawRectangle(p, 10, 10, 50, 50);
        g.DrawRectangle(p, 150, 150, 200, 200);

        // 클리핑 영역 지정
        g.SetClip(new Rectangle(100, 100, 100, 100));
```

```

        // 클리핑 영역에서만 출력이 이루어지게 됨
        g.FillRectangle(Brushes.Black, 10, 10, 50, 50);
        g.FillRectangle(Brushes.Red, 150, 150, 200, 200);
    }
}

```

- 텍스트 클리핑 예제

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;

public class ClippingExam2 : System.Windows.Forms.Form
{
    public ClippingExam2()
    {
        this.Text = "ClippingExam2";
        this.Size = new Size(600, 300);
    }

    static void Main()
    {
        Application.Run(new ClippingExam2());
    }

    protected override void OnPaint(PaintEventArgs e)
    {
        Bitmap bmp = new Bitmap("태극기.gif");
        FontFamily ff = new FontFamily("궁서체");

        Graphics g = e.Graphics;
        g.FillRectangle(Brushes.Pink, this.ClientRectangle);
        GraphicsPath gp = new GraphicsPath();
        gp.AddString("국기", ff, (int)FontStyle.Bold, 150, new Point(5, 20),
StringFormat.GenericDefault);

        // 글자의 영역만 클리핑 설정
        g.SetClip(gp);
        // 설정된 영역에 그림 출력
        g.DrawImage(bmp, this.ClientRectangle);
    }
}

```

- 새로 그려질 클리핑 영역 출력

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;

```

```

public class ClippingExam3 : System.Windows.Forms.Form
{
    static void Main()
    {
        Application.Run(new ClippingExam3());
    }
    protected override void OnPaint(PaintEventArgs e)
    {
        Console.WriteLine("Clipping 영역 : {0}", e.ClipRectangle);
    }
}

```

- ClipRectangle을 이용한 그리기 작업 최소화 예

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class ClippingExam4 : System.Windows.Forms.Form
{
    static void Main()
    {
        Application.Run(new ClippingExam4());
    }

    protected override void OnPaint(PaintEventArgs e)
    {
        Console.WriteLine("Clipping 영역 : {0}", e.ClipRectangle);
        Rectangle rect = new Rectangle(0, 0, 200, 100);
        Graphics g = e.Graphics;

        g.DrawRectangle(Pens.Black, rect);
        g.SetClip(rect);

        // 폼을 다른 윈도우로 가려보면 클리핑 영역에서는 그리기 연산이
        // 일어나지만 클리핑 영역이 아닌 경우에는 DrawString()
        // 메서드의 호출이 오지 않음
        //if (g.VisibleClipEmpty)
        if (rect.Intersects(e.ClipRectangle))
        {
            g.DrawString("그리기 작업 수행", this.Font, Brushes.Black, 20, 10);
            Console.WriteLine("그리기 작업 실행됨!!!");
        }
        else
        {
            Console.WriteLine("아무 작업도 안함!!!!");
        }
    }
}

```

# **06**

Chapter

## 이미지와 비트맵

- 비트맵 관련 클래스

- Object <-- MarshalByRefObject <-- Image(추상 클래스)
- System.Drawing 에 정의

클래스	설명
Image 클래스	- 다양한 이미지 처리 메서드와 속성 제공
Bitmap 클래스	- 픽셀 단위로 이미지를 다룰 때 자주 사용 - 닷넷에서 주로 사용하는 클래스
Metafile 클래스	- 벡터 이미지를 다룰 때 사용 - GDI+를 이용 그리기 연산 수행시 해당 정보가 벡터 이미지의 큐에 저장되며 이러한 벡터 정보 이미지를 기록 및 재생 할 수 있음

### Win OS에서 메타 파일 처리를 위한 파일 형식

- WMF(Windows 메타 파일)
- EMF(확장 메타 파일)
- \*) Metafile클래스로 손쉽게 제어 가능

### 사용예제

- 이미지를 GDI+화면에 출력

```
using System;
using System.Drawing;
using System.Windows.Forms;
```

```
public class LoadImage : System.Windows.Forms.Form
{
    static void Main()
    {
        Application.Run(new LoadImage());
    }

    protected override void OnPaint(PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        Image bmp = Image.FromFile("f15.jpg");
        //Bitmap bmp = new Bitmap("f15.jpg");

        this.Height = bmp.Height;
```

```

        this.Width = bmp.Width;

        g.DrawImage(bmp, 0, 0);
    }
}

```

- 메타 파일 다루기

```

using System;
using System.Drawing;
using System.Windows.Forms;
using System.Drawing.Imaging;
using System.Runtime.InteropServices;

```

```

public class MetafileExam : System.Windows.Forms.Form
{
    private Metafile metafile;
    private int index;
    private Graphics.EnumerateMetafileProc metafileDelegate;
    private Point destPoint;

    public MetafileExam()
    {
        metafile = new Metafile("test.wmf");
        index = 0;

        metafileDelegate = new Graphics.EnumerateMetafileProc(MetafileCallback);
        destPoint = new Point(20, 10);
        this.Text = "Metafile 이미지 출력";
    }

    static void Main()
    {
        Application.Run(new MetafileExam());
    }

    protected override void OnPaint(PaintEventArgs e)
    {
        e.Graphics.EnumerateMetafile(metafile, destPoint, metafileDelegate);
    }

    private bool MetafileCallback(
        EmfPlusRecordType recordType,
        int flags,
        int dataSize,
        IntPtr data,
        PlayRecordCallback callbackData)
    {
        byte[] dataArray = null;

```

```
        if (data != IntPtr.Zero)
        {
            dataArray = new byte[dataSize];
            Marshal.Copy(data, dataArray, 0, dataSize);
        }

        metafile.PlayRecord(recordType, flags, dataSize, dataArray);
        Console.WriteLine("{0} -> type:{1}, flags:{2}, size:{3}", this.index++,
recordType, flags, dataSize);

        return true;
    }
}
```

**07**

Chapter

타이머

## 01 Timer

### • 서버 기반 타이머

- 밀리초 단위로 타이머 호출 시간 설정 가능
- 다른 타이머에 비해 정확도가 높음
- 주로 24시간 가동하며, 연속된 데이터를 처리하는 웹 서버등에서 사용됨
- Elapsed 이벤트가 발생되는 반복 간격을 Interval 속성을 이용해 지정함
- Enabled 속성 : Elapsed 이벤트 활성화 여부를 설정할 때 사용
- Start() / Stop() 메서드를 이용해 속성 값 변경
- Elapsed 이벤트는 지정된 시간(Interval)이 경과하면 ElapsedEventHandler에 등록된 메서드를 호출하게 됨

[25]

 비트교육센터

### 사용 예제

- 서버 기반 타이머

```
using System;
using System.Timers;
public class ServerTimer
{
    public static void Main()
    {
        System.Timers.Timer serverTimer = new System.Timers.Timer();
        serverTimer.Elapsed += new ElapsedEventHandler(OnTimedEvent);
        serverTimer.Enabled = true; // 서버 타이머 실행
        serverTimer.Interval = 1000;

        Console.WriteLine("예제를 종료하려면 q를 입력하세요.");
        while (Console.Read() != 'q') ;
    }

    // 1초에 한번씩 호출하는 메서드
    private static void OnTimedEvent(object sender, ElapsedEventArgs e)
    {
        Console.WriteLine("서버 타이머 이벤트 발생");
        Console.WriteLine("발생 시간 : {0}", e.SignalTime);
    }
}
```

## 01 Timer

### • 윈도우 기반 타이머

- 서버 타이머에 비해 작고 가벼움
- 서버타이머가 지정된 시간마다 Elapsed 이벤트에 등록한 메서드를 호출하는 것에 비해, 윈도우 타이머는 Tick 이벤트에 등록한 메서드를 호출함
- Tick 이벤트는 EventHandler 대리자 형태의 메서드를 Interval 간격마다 호출하게 됨
- Interval과 Enabled 속성이 있음
- 서버와 달리 Enabled속성을 이용하기 보다는 Start() / Stop()메서드를 이용하여 타이머를 제어하도록 코드를 작성함

[26]

 비트교육센터

### 사용 예제

#### • 윈도우 기반 타이머

```
using System;
using System.Drawing;
using System.Windows.Forms;
public class WinTimer : System.Windows.Forms.Form
{
    Timer time;
    int timecount = 0;

    public WinTimer()
    {
        this.Text = "10초후에 닫히는 창 예제";
        time = new Timer();
        time.Interval = 1 * 1000;
        time.Tick += new EventHandler(time_Tick);

        this.Load += new EventHandler(WinTimer_Load);
    }

    private void WinTimer_Load(object obj, EventArgs e)
    {
        System.Console.WriteLine("타이머 시작!!!");
        time.Start(); // 타이머 시작
    }
```

```
private void time_Tick(object sender, EventArgs ea)
{
    if (timecount < 10)
    {
        System.Console.WriteLine("{0} Tick...", DateTime.Now);
        this.Text = (10 - timecount) + " 초 남았습니다...";
        timecount++;
    }
    else
    {
        time.Stop();
        MessageBox.Show("프로그램을 종료합니다.", "10초 타이머");
        this.Close();
    }
}

public static void Main()
{
    Application.Run(new WinTimer());
}
```

## 01 Timer

### • 스레드 기반 타이머

- 생성자로 스레드 타이머 객체를 생성함
- 타이머 실행되는 도중에 타이머의 시작시간, 메서드 호출 간격 변경 사용가능
- 반드시 개체 사용이 끝나면 Dispose() 메서드로 타이머 리소스를 해제해야 함

```
Timer( TimerCallback callback, // 호출될 메서드  
       object state, // 콜백 메서드에서 사용할 정보가 포함된 개체  
       int dueTime, // TimerCallback에 등록된 메서드를 호출하기 전에  
                   // 타이머 지연 시간을 밀리초 단위로 설정  
       int period); // 호출 간격
```

[27]

 비트교육센터

### 사용 예제

- 스레드 기반 타이머

```
using System;  
using System.Threading;  
  
class TimerExampleState  
{  
    public int counter = 0;  
    public Timer timer;  
}  
  
class ThreadTimer  
{  
    public static void Main()  
    {  
        TimerExampleState s = new TimerExampleState();  
  
        TimerCallback timerDelegate = new  
        TimerCallback(CheckStatus);  
  
        Timer timer = new Timer(timerDelegate, s, 2000,  
                               1000);  
  
        s.timer = timer;  
    }  
}
```

```

        while(s.timer != null);

    }

    private static void CheckStatus(Object state)
    {
        TimerExampleState ts =(TimerExampleState)state;
        ts.counter++;
        if(ts.counter == 3)
        {
            Console.WriteLine("{0}:스레드 체
크 :{1}.", ts.counter, DateTime.Now);
            (ts.timer).Change(5000,2000);
            Console.WriteLine("스레드 상태 변
경...");

            Console.WriteLine("5초후에 2초 간
격으로 다시 시작됩니다.");
        }
        else if(ts.counter == 4)
        {
            Console.WriteLine("스레드 상태 변
경 => 2초 간격으로 시작");
            Console.WriteLine("{0}:스레드 체
크 :{1}.", ts.counter, DateTime.Now);
        }
        else if(ts.counter == 7)
        {
            Console.WriteLine("타이머를 종료합
니다...");
            ts.timer.Dispose();
            ts.timer = null;
        }
        else
        {
            Console.WriteLine("{0}:스레드 체
크 :{1}.", ts.counter, DateTime.Now);
        }
    }
}

```

## 간단한 사용 예제

- 화면 보호기 흘내내기

```

using System;
using System.Drawing;
using System.Windows.Forms;

```

```

public class ScreenExam : System.Windows.Forms.Form
{
    Random rnd = new Random();
    public ScreenExam()
    {

```

```

{
    this.Text = "타이머를 이용한 화면 보호기";
    this.Bounds = Screen.PrimaryScreen.Bounds;
    Timer time = new Timer();
    time.Interval = 50;          // 0.05초 간격
    time.Tick += new EventHandler(time_Tick);
    time.Enabled = true;
}

protected void time_Tick(object obj, EventArgs ea)
{
    Graphics grfx = this.CreateGraphics();

    // 난수 발생
    int shape = rnd.Next(3);
    switch (shape)
    {
        case 0:
            // 원 그리기
            grfx.DrawEllipse(Pens.Red, rnd.Next(this.Width),
                rnd.Next(this.Height), rnd.Next(100), rnd.Next(100));
            break;
        case 1:
            // 사각형 그리기
            grfx.DrawRectangle(Pens.Blue, rnd.Next(this.Width),
                rnd.Next(this.Height), rnd.Next(100), rnd.Next(100));
            break;
        case 2:
            // 호 그리기
            grfx.DrawArc(Pens.Green, rnd.Next(this.Width),
                rnd.Next(this.Height), 30, 30, rnd.Next(360), rnd.Next(360));
            break;
    }
}

static void Main()
{
    Application.Run(new ScreenExam());
}

```

# **08**

Chapter

GDI+ 좌표 변환

## 01 GDI+ 좌표

### • GDI+ 3가지의 좌표계

좌표계	설명
영역 좌표	- 그리기 메서드에서 사용하는 형식
페이지 좌표	- 클라이언트 영역의 왼쪽 모서리를 기반으로 Graphics 클래스의 PageScale과 PageUnit 속성을 사용해 크기 및 단위(Pixel, Inch, Millimeter, Point 등)를 변경할 때 사용
장치 좌표계	- 실제 출력 장치(모니터, 프린터) 등에서 사용되는 좌표계

### • 페이지 좌표계의 속성

속성	자료형	설명
PageUnit	GraphicsUnit	- 페이지 단위 설정
PageScale	float	- 페이지 배율 설정

## 01 GDI+ 좌표

### • 영역 좌표계가 제공하는 변환

메서드	설명
ResetTransform	- 변환된 좌표계를 초기화 시킴
RotateTransform	- 점(0,0)을 중심으로 지정된 각도만큼 회전
ScaleTransform	- 영역 변환 배율을 설정
TranslateTransform	- 좌표계의 원점을 변경
MultiplyTransform	- 지정된 Matrix를 곱해 좌표계를 변환

[30]

 비트교육센터

### 사용 예제

- 영역 좌표계 배율 변환

```
using System;
using System.Drawing;
using System.Windows.Forms;
public class TransExam1 : System.Windows.Forms.Form
{
    public TransExam1()
    {
        this.Text = "배율(Scaling) 변환";
    }
    static void Main()
    {
        Application.Run(new TransExam1());
    }
    protected override void OnPaint(PaintEventArgs e)
    {
        Rectangle rect = new Rectangle(0, 0, 100, 100);
        Graphics g = e.Graphics;

        g.DrawRectangle(Pens.Black, rect);
        g.DrawString("VC#", this.Font, Brushes.Black, 10, 50);

        // 가로 3배, 세로 2.5배 확대
        g.ScaleTransform(3.0f, 2.5f);
```

```

        g.DrawRectangle(Pens.Red, rect);
        g.DrawString("VC#", this.Font, Brushes.Red, 10, 50);
    }
}

```

- 영역 좌표계의 이동

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class TransExam2 : System.Windows.Forms.Form
{
    public TransExam2()
    {
        this.Text = "이동(Translation) 변환";
    }
    static void Main()
    {
        Application.Run(new TransExam2());
    }
    private void DrawGrid(Graphics g)
    {
        Pen pen = new Pen(Color.Blue, 0.1f);
        pen.DashStyle = DashStyle.Dot;

        for (int i = 0; i < this.Width; i += 10)
            g.DrawLine(pen, i, 0, i, this.Height);

        for (int j = 0; j < this.Height; j += 10)
            g.DrawLine(pen, 0, j, this.Width, j);

        pen.Dispose();
    }
    protected override void OnPaint(PaintEventArgs e)
    {
        Graphics g = e.Graphics;

        // 화면에 격자 문양 출력
        this.DrawGrid(g);

        Font font = new Font("궁서체", 20);

        g.DrawString("VC#", font, Brushes.Black, 10, 20);
        g.DrawLine(Pens.Black, 0, 100, this.Width, 100);
        g.DrawLine(Pens.Black, 100, 0, 100, this.Height);

        // X좌표 , Y좌표 변환
        g.TranslateTransform(100, 100);
    }
}

```

```
g.DrawString("VC#", font, Brushes.Red, 10, 20);
```

```
    font.Dispose();  
}  
}
```

- 회전 변환

```
using System;  
using System.Drawing;  
using System.Drawing.Drawing2D;  
using System.Windows.Forms;  
public class TransExam3 : System.Windows.Forms.Form  
{  
    public TransExam3()  
    {  
        this.Text = "회전(Rotation) 변환";  
        this.Size = new Size(300, 300);  
    }  
    static void Main()  
    {  
        Application.Run(new TransExam3());  
    }  
    private void DrawGrid(Graphics g)  
    {  
        Pen pen = new Pen(Color.Blue, 0.1f);  
        pen.DashStyle = DashStyle.Dot;  
  
        for (int i = 0; i < this.Width; i += 10)  
            g.DrawLine(pen, i, 0, i, this.Height);  
  
        for (int j = 0; j < this.Height; j += 10)  
            g.DrawLine(pen, 0, j, this.Width, j);  
  
        pen.Dispose();  
    }  
    protected override void OnPaint(PaintEventArgs e)  
    {  
        Graphics g = e.Graphics;  
        this.DrawGrid(g);  
  
        Font font = new Font("궁서체", 20);  
  
        g.DrawString("회전 시키기전", font, Brushes.Black, 50, 20);  
        g.DrawLine(Pens.Black, 0, 0, this.ClientSize.Width, this.ClientSize.Height);  
  
        //회전 각도  
        g.RotateTransform(45);
```

```
g.DrawString("회전시킨후", font, Brushes.Red, 50, 20);
```

```
    font.Dispose();
}
}
```

- 기울기 변환

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class TransExam4 : System.Windows.Forms.Form
{
    public TransExam4()
    {
        this.Text = "기울기(Shearing) 변환";
        this.Size = new Size(300, 300);
    }
    static void Main()
    {
        Application.Run(new TransExam4());
    }
    private void DrawGrid(Graphics g)
    {
        Pen pen = new Pen(Color.Blue, 0.1f);
        pen.DashStyle = DashStyle.Dot;

        for (int i = 0; i < this.Width; i += 10)
            g.DrawLine(pen, i, 0, i, this.Height);

        for (int j = 0; j < this.Height; j += 10)
            g.DrawLine(pen, 0, j, this.Width, j);

        pen.Dispose();
    }
    protected override void OnPaint(PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        this.DrawGrid(g);

        Font font = new Font("굴서체", 20);

        g.DrawString("기울기 변환전", font, Brushes.Black, 50, 20);
        g.DrawRectangle(Pens.Black, 50, 20, 170, 22);

        Matrix matrix = new Matrix();

        // 가로 기울기 , 세로 기울기
        matrix.Shear(0.5f, 0.5f);
```

```

// 가로 세로를 0.5 도씩 기울여 출력
g.MultiplyTransform(matrix);

matrix.Dispose();

g.DrawString("Youngjin.com", font, Brushes.Red, 50, 20);
g.DrawRectangle(Pens.Red, 50, 20, 170, 22);

font.Dispose();
}
}

```

- 누적 변환

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class TransExam5 : System.Windows.Forms.Form
{
    public TransExam5()
    {
        this.Text = "누적 변환";
        this.Size = new Size(300, 300);
    }
    static void Main()
    {
        Application.Run(new TransExam5());
    }
    private void DrawGrid(Graphics g)
    {
        Pen pen = new Pen(Color.Blue, 0.1f);
        pen.DashStyle = DashStyle.Dot;

        for (int i = 0; i < this.Width; i += 10)
            g.DrawLine(pen, i, 0, i, this.Height);

        for (int j = 0; j < this.Height; j += 10)
            g.DrawLine(pen, 0, j, this.Width, j);

        pen.Dispose();
    }
    protected override void OnPaint(PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        this.DrawGrid(g);

        // 좌표계의 기준 원점을 10번 이동시킴
    }
}

```

```

        for (int i = 0; i < 10; i++)
        {
            g.DrawEllipse(Pens.Black, new Rectangle(10, 10, 50, 70));

            // 원점을 10, 10씩 x, y축으로 이동시킴
            g.TranslateTransform(10, 10);
        }
    }
}

```

- 비누적 변환

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class TransExam6 : System.Windows.Forms.Form
{
    public TransExam6()
    {
        this.Text = "비누적 변환";
        this.Size = new Size(300, 300);
    }
    static void Main()
    {
        Application.Run(new TransExam6());
    }
    private void DrawGrid(Graphics g)
    {
        Pen pen = new Pen(Color.Blue, 0.1f);
        pen.DashStyle = DashStyle.Dot;

        for (int i = 0; i < this.Width; i += 10)
            g.DrawLine(pen, i, 0, i, this.Height);

        for (int j = 0; j < this.Height; j += 10)
            g.DrawLine(pen, 0, j, this.Width, j);

        pen.Dispose();
    }
    protected override void OnPaint(PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        this.DrawGrid(g);

        // 원점을 기준으로 반지름 100인 파란색 원을 그림
        g.DrawEllipse(Pens.Black, new Rectangle(10, 10, 100, 100));

        // 원점을 50, 50 x축 y축 방향으로 이동
        g.TranslateTransform(50, 50);
    }
}

```

```

        // 반지름 100인 빨간색 타원형을 그림
        g.DrawEllipse(Pens.Red, new Rectangle(10, 10, 100, 100));

        // 원점을 0으로 되돌린 후
        g.ResetTransform();

        // 가로 세로가 100인 파란색 사각형을 그림
        g.DrawRectangle(Pens.Blue, new Rectangle(10, 10, 100, 100));
    }
}

```

• 복합 변환

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;
public class TransExam7 : System.Windows.Forms.Form
{
    public TransExam7()
    {
        this.Text = "복합 변환";
        this.Size = new Size(350, 350);
    }
    static void Main()
    {
        Application.Run(new TransExam7());
    }
    private void DrawGrid(Graphics g)
    {
        Pen pen = new Pen(Color.Blue, 0.1f);
        pen.DashStyle = DashStyle.Dot;

        for (int i = 0; i < this.Width; i += 10)
            g.DrawLine(pen, i, 0, i, this.Height);

        for (int j = 0; j < this.Height; j += 10)
            g.DrawLine(pen, 0, j, this.Width, j);

        pen.Dispose();
    }
    protected override void OnPaint(PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        this.DrawGrid(g);

        Font font = new Font("궁서체", 18);
        Pen pen = new Pen(Brushes.Red, 2);
    }
}

```

```
// 좌표계의 원점을 원도우 화면의 중앙으로 이동
g.TranslateTransform(this.Width / 2, this.Height / 2);

for (int i = 0; i < 360; i += 45)
{
    // 45도 단위로 좌표축을 회전시키면서 문자열 출력
    g.RotateTransform(i);
    g.DrawRectangle(pen, 0, 0, 300, 24);
    g.DrawString(".NET 3.0", font, Brushes.Blue, 30, 0);
}

pen.Dispose();
font.Dispose();
}
}
```

# **09**

Chapter

마우스와 키보드

- 마우스

- System.Windows.Forms 네임스페이스의 SystemInformation 클래스 이용

속성	설명
MouseWheelPresent (bool)	- 마우스 휠이 있으면 true반환
MouseWheelScrollDelta(int)	- 마우스 휠 회전 증분에 대한 델타값 크기 반환
MouseWheelScrollLines(int)	- 마우스 휠이 회전될때 스크롤할 줄의 수
MouseButtons(int)	- 마우스에 있는 버튼 수 반환
MouseButtonsSwapped(bool)	- 마우스 왼쪽 및 오른쪽 버튼의 기능이 바뀌었으면 true, 아니면 false반환
MouseSpeed(int)	- 현재 마우스 속도 반환
DoubleClickTime(int)	- 더블 클릭이 이루어지기 위해 첫번째 클릭 이후 두번째 클릭할때 까지의 제한 시간
DoubleClickSize(size)	- 사용자 입력이 더블 클릭으로 간주되기 위한 영역의 최소 크기

### • 마우스 이벤트 처리

속성	설명
MouseEnter	- 마우스커서가 컨트롤 위를 지나갈 때 발생
MouseLeave	- 마우스 커서가 컨트롤 영역을 벗어날 때 발생
MouseHover	- 마우스 커서가 컨트롤 영역으로 들어와 움직임을 멈춘 후 발생 - MouseEnter와 MouseLeave 사이에 반드시 한번은 발생함

### • 마우스 이벤트의 발생 순서

- 1) MouseEnter : 특정 컨트롤 영역 진입
- 2) MouseMove : 마우스 포인터 이동
- 3) MouseHover / MouseDown / MouseWheel : 이동 또는 버튼 조작
- 4) Mouseup : 버튼 놓음
- 5) MouseLeave : 특정 컨트롤 영역에서 빠져 나옴

[33]

 비트교육센터

## 사용 예제

- 마우스 이벤트를 이용한 이미지 뷰어

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Drawing.Drawing2D;

namespace ImageViewer
{
    public partial class Form1 : Form
    {
        private Bitmap m_OriginalBmp; // 원본 이미지
        private bool m_breadBmp; // 이미지 읽기 확인
        private Bitmap m_SmallBmp; // 오른쪽에 출력될 이미지
        private Rectangle m_region; // 마우스가 가리키는 영역
        private bool m_bMouseDown; // 마우스 버튼이 눌렸는지 유무
        private int m_Ratio; // 이미지 확대/축소 비율

        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

```

// 초기화
this.m_SmallBmp = new Bitmap(this.m_SmallPanel.Width,
this.m_SmallPanel.Height);
this.m_Ratio = 0;
}

// 파일 열기 메뉴 선택
private void menuItem_FileOpen_Click(object sender, EventArgs e)
{
    OpenFileDialog fd = new OpenFileDialog();
    fd.InitialDirectory = "C:\\"; // 수정된 부분
    fd.Filter = "Image Files | *.JPG;*.GIF;*.PNG;*.TIF;*.BMP;*.*";
    if (fd.ShowDialog() == DialogResult.OK)
    {
        this.m_OriginalBmp = new Bitmap(fd.FileName);
        this.m_breadBmp = true;
        m_region = new Rectangle(this.m_OriginalPanel.Width / 2 - 100,
this.m_OriginalPanel.Height / 2 - 100, 100, 100);
        this.m_OriginalPanel.Invalidate();
    }
}

// 종료 메뉴 선택
private void menuItem_Exit_Click(object sender, EventArgs e)
{
    m_SmallBmp.Dispose();
    m_OriginalBmp.Dispose();
    this.Close();
}

// 왼쪽 Panel 영역의 Paint 이벤트(마우스 드래그 영역에 노란색 점선 그리기)
private void m_OriginalPanel_Paint(object sender, PaintEventArgs e)
{
    if (this.m_breadBmp)
    {
        Graphics g = e.Graphics;
        g.FillRectangle(Brushes.White, this.m_OriginalPanel.ClientRectangle);
        Rectangle rect = new Rectangle(0, 0, this.m_OriginalBmp.Width,
this.m_OriginalBmp.Height);
        g.DrawImage(this.m_OriginalBmp, rect);

        // 노란색 점선 그리기
        Pen pen = new Pen(Brushes.Gold, 3);
        pen.DashStyle = DashStyle.Dash;
        g.DrawRectangle(pen, m_region);
        pen.Dispose();
    }
}

```

```

// 왼쪽 Panel에서 MouseDown이벤트 처리
private void m_OriPanel_MouseDown(object sender, MouseEventArgs e)
{
    this.m_bMouseDown = true;
}

// 왼쪽 Panel에서 MouseUp이벤트 처리
private void m_OriPanel_MouseUp(object sender, MouseEventArgs e)
{
    this.m_bMouseDown = false;
}

// 왼쪽 Panel에서 MouseMove이벤트 처리
private void m_OriPanel_MouseMove(object sender, MouseEventArgs e)
{
    if (this.m_bMouseDown)
    {
        m_region.X = e.X;
        m_region.Y = e.Y;

        this.m_SmallBmp.Dispose();

        // 오른쪽 패널의 새로운 비트맵 생성 및 그리기
        this.m_SmallBmp = new Bitmap(this.m_SmallPanel.Width,
        this.m_SmallPanel.Height);
        Graphics gi = Graphics.FromImage(this.m_SmallBmp);
        gi.DrawImage(this.m_OriBmp, this.m_SmallPanel.ClientRectangle,
        this.m_region, GraphicsUnit.Pixel);
        gi.Dispose();
    }
}

// 왼쪽 , 오른쪽 패널 갱신
this.m_OriPanel.Invalidate();
this.m_SmallPanel.Invalidate();
}

// 오른쪽 패널의 Paint이벤트 처리
private void m_SmallPanel_Paint(object sender, PaintEventArgs e)
{
    // 읽어 들인 이미지 확인
    if (this.m_breadBmp)
    {
        Graphics g = e.Graphics;

        // 패널을 흰색으로 칠함
        g.FillRectangle(Brushes.White, this.m_SmallPanel.ClientRectangle);
        // 왼쪽 패널에서 설정한 이미지 출력
        g.DrawImage(this.m_SmallBmp, 0, 0);
    }
}

```

```

}

// 오른쪽 패널에 출력할 이미지 크기 조정
private void ResizeSmallImage()
{
    // 이미지 배율
    float ratio = (float)(1.0 + this.m_Ratio * .25);
    int w = (int)(this.m_SmallPanel.Width * ratio);
    int h = (int)(this.m_SmallPanel.Height * ratio);

    Rectangle rect = new Rectangle(0, 0, w, h);

    Bitmap tempBmp = new Bitmap(w, h);

    Graphics gi = Graphics.FromImage(tempBmp);
    gi.DrawImage(this.m_SmallBmp, rect);

    this.m_SmallBmp.Dispose();
    this.m_SmallBmp = new Bitmap(w, h);
    this.m_SmallBmp = tempBmp.Clone(rect, tempBmp.PixelFormat);

    gi.Dispose();
    tempBmp.Dispose();

    this.m_SmallPanel.Invalidate();
}

// + 버튼 클릭 이벤트
private void btn_Plus_Click(object sender, EventArgs e)
{
    if (this.m_Ratio < 3)
    {
        this.m_Ratio++;
        this.ResizeSmallImage();
    }
}

// - 버튼 클릭 이벤트
private void btn_Minus_Click(object sender, EventArgs e)
{
    if (this.m_Ratio > -3)
    {
        this.m_Ratio--;
        this.ResizeSmallImage();
    }
}
}

```

- 키 이벤트처리

- 키보드는 활성화 폼에서만 입력 가능
- 4가지 종류의 그룹으로 구분

종류	설명
문자키	- 문자, 숫자, <Space>, <Tab>, <BackSpace>, <Esc>
비문자키	- 방향키, 함수키, 삭제, 삽입 키들로 문자 입력과는 직접적인 연관이 없음
토글키	- <Caps Lock>, <Num Lock>, <Scroll Lock>, <Insert>
শিফ트키	- <Shift> <Alt> <Ctrl> 등으로 다른 키나 마우스키 등과 조합해서 사용

- 키 이벤트

종류	설명
KeyDown이벤트	- 컨트롤에 포커스가 있을때 키를 누르면 발생
KeyUp이벤트	- 컨트롤에 포커스가 있을 때 키를 놓으면 발생
KeyPress이벤트	<ul style="list-style-type: none"> <li>- 컨트롤에 포커스가 있을 때 키를 누르면 발생</li> <li>- 문자키가 눌릴때만 발생</li> </ul>

### 사용예제

- 키 이벤트 처리

```

public class KeyExam : Form
{
    public static void Main()
    {
        Application.Run(new KeyExam());
    }

    string strdata = " "; // 화면에 출력할 문자
    public KeyExam()
    {
        this.Text = "키보드 입력 예제";
        this.BackColor = Color.Black;
        this.ForeColor = Color.White;
    }
    protected override void OnKeyDown(KeyEventArgs key)
    {
        if (key.KeyCode == Keys.Back) // 백스페이스 키이면, 입력한 글자 지움
        {
            if (strdata.Length > 0)
            {
                strdata = strdata.Remove(strdata.Length - 1); // 마지막 문자 지움
            }
            Invalidate();
        }
    }
}

```

```

        }

        else if (key.KeyCode == Keys.Enter)
        {
            strdata += "WrWn";
            Invalidate();
        }

        else if (key.KeyCode == Keys.Space)
        {
            strdata += " ";
            Invalidate();
        }

        // 0~9 사이의 키가 들어오면 KeyCode는 D0~D9로 들어오게 됨: 숫자로 변환
        // 숫자 코드 48부터 0으로 정의되어 있음
        else if (key.KeyCode <= Keys.D9 && key.KeyCode >= Keys.D0)
        {
            strdata += (key.KeyValue - 48);
            Invalidate();
        }

        else if (key.KeyCode == Keys.F1)
        {
            MessageBox.Show("프로그램 도움말!");
        }

        else if ((key.KeyCode == Keys.F10) && (key.Control)) // ctrl+F10은 종료
        {
            MessageBox.Show("프로그램을 종료합니다.");
            this.Close();
        }

        else
        {
            // A~Z 사이의 대문자만 정상 출력 가능
            strdata += key.KeyCode;
            Invalidate();
        }
    }

protected override void OnPaint(PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.DrawString(strdata, new Font("궁서체", 20, FontStyle.Bold), Brushes.White, 20,
    30);
}

```

# **10**

Chapter

인쇄하기

- 인쇄 관련 클래스

분류	클래스	설명
Dialog 관련 클래스	PageSetupDialog	<ul style="list-style-type: none"> <li>- 페이지 설정을 처리하는 대화상자</li> <li>- 용지의 크기 및 공급, 방향, 여백 등을 설정</li> </ul>
	PrintDialog	<ul style="list-style-type: none"> <li>- 컴퓨터에 설치된 프린터를 선택할 수 있는 대화상자</li> <li>- 프린터 속성과 인쇄 범위 및 매수를 설정할 수 있음</li> </ul>
	PrintPreviewDialog	<ul style="list-style-type: none"> <li>- 프린터에 인쇄할 내용을 미리 화면에 보여주는 대화상자</li> <li>- 인쇄될 내용을 확대/축소해 출력하는 기능과 스크롤링 등의 인쇄</li> </ul>
인쇄 과정 제어 클래스	PageSettings	<ul style="list-style-type: none"> <li>- 인쇄될 특정 페이지에 대한 속성을 설정</li> <li>- 속성을 제어하는 PageSetupDialog 클래스의 속성으로 사용됨</li> </ul>
	PrintDocument	<ul style="list-style-type: none"> <li>- 인쇄될 내용을 담고 있는 개체</li> <li>- 이 개체에 담긴 내용은 인쇄나 미리보기 기능에 사용됨</li> </ul>
	PrintSettings	<ul style="list-style-type: none"> <li>- 프린터의 색, 급지 방식, 가로 인쇄 등의 프린터 속성을 설정</li> <li>- PageSetupDialog 또는 PrintDialog 클래스의 속성으로 사용됨</li> </ul>

- 인쇄 진행 과정

- 인쇄 과정 자체는 System.Drawing.Printing.PrintDocument 클래스에 의해 처리
- 이 클래스에 인쇄 방법을 설명하는 속성을 설정하고, 인쇄 프로세스를 시작하는 Print() 메서드를 호출하면 바로 인쇄가 시작
- Print 메서드는 PrintPage 이벤트를 처리하고 PrintEventArgs에 포함된 Graphics 를 이용해 인쇄할 페이지 영역에 앞서, GDI+그리기 내용을 출력 가능.

- 1) BeginPrint 이벤트 : 인쇄 중에 사용되는 글꼴, 파일 스트림 및 기타 리소스를 초기화 함
- 2) QueryPageSettings 이벤트 : 현재 인쇄할 페이지에 관련된 프린터 설정 변경
- 3) PrintPage 이벤트 : 인쇄할 내용을 PrintEventArgs 인자를 갖고 출력 페이지에 실제 인쇄를 진행하는 부분
- 4) EndPrint 이벤트 : 인쇄 중에 사용된 글꼴, 파일 스트림 및 기타 리소스를 해제

[38]

 비트교육센터

### 사용 예제

- 6개의 클래스를 모두 사용한 간단한 일기장 프로그램

```

using System;
using System.Drawing;
using System.Windows.Forms;
using System.Drawing.Printing;

namespace MyDiary
{
    public partial class Form1 : Form
    {
        private Font m_MainFont = null;
        private Font m_SubFont = null;
        private Font m_SmallFont = null;
        private PageSettings m_PageSetting = null;
        private Bitmap m_backbmp = null;
        private Bitmap[] m_weather = new Bitmap[4];

        public Form1()
        {
            InitializeComponent();

            m_MainFont = new Font("궁서체", 15, FontStyle.Bold);
            m_SubFont = new Font("굴림체", 13);
            m_SmallFont = new Font("바탕체", 9);
        }
    }
}

```

```

m_backbmp = new Bitmap(GetType(), "background.jpg");

for (int i = 0; i < 4; i++)
{
    string str = string.Format("weather{0}.gif", i + 1);
    m_weather[i] = new Bitmap(GetType(), str);
}
cb_Weather.SelectedIndex = 0;
}

private void btn_PageSetup_Click(object sender, EventArgs e)
{
try
{
    PageSetupDialog psetup = new PageSetupDialog();
    if (this.m_PageSetting == null)
        this.m_PageSetting = new PageSettings();
    psetup.PageSettings = this.m_PageSetting;
    psetup.ShowDialog();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void btn_Preview_Click(object sender, EventArgs e)
{
try{
    PrintDocument pd = new PrintDocument();
    pd.PrintPage += new PrintPageEventHandler(this.PrintPageEvent);

    if (this.m_PageSetting != null)
        pd.DefaultPageSettings = this.m_PageSetting;

    PrintPreviewDialog pdlg = new PrintPreviewDialog();
    pdlg.Document = pd;
    pdlg.ShowDialog();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void btn_Print_Click(object sender, EventArgs e)
{
try{

```

```

PrintDocument pd = new PrintDocument();
pd.PrintPage += new PrintPageEventHandler(this.PrintPageEvent);

if (this.m_PageSetting != null)
    pd.DefaultPageSettings = this.m_PageSetting;

PrintDialog pdlg = new PrintDialog();
pdlg.Document = pd;
if (pdlg.ShowDialog() == DialogResult.OK)
{
    pd.Print();
}
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

protected void PrintPageEvent(Object obj, PrintPageEventArgs pea)
{
    Graphics g = pea.Graphics;
    PaintDocument(g);
    pea.HasMorePages = false;
}

private void PaintDocument(Graphics g)
{
    g.FillRectangle(Brushes.White, 100, 50, 800, 600);
    g.DrawImage(m_backbmp, 100, 50); // 바탕이미지 채우기

    g.DrawImage(m_weather[cb_Weather.SelectedIndex], 410, 230); // 날씨 아이
    콘 출력
    g.DrawString(cb_Weather.SelectedItem.ToString(), this.m_MainFont,
    Brushes.Brown, 450, 237);
    g.DrawString(this.dtp_Date.Text, this.m_SmallFont, Brushes.Brown, 370, 280);

    StringFormat sf = new StringFormat();
    sf.Alignment = StringAlignment.Center;
    sf.LineAlignment = StringAlignment.Center;
    Rectangle rect = new Rectangle(100, 50, 400, this.m_MainFont.Height * 3);
    g.DrawString(this.txt_Title.Text, this.m_MainFont, Brushes.Black, rect, sf);

    rect = new Rectangle(110, 130, 400, this.m_SubFont.Height * 10);
    g.DrawString(this.txt_Content.Text, this.m_SubFont, Brushes.Black, rect);
}

}
}

```

# 11

Chapter

버튼, 라벨, 스크롤,  
컨테이너 클래스

- **ButtonBase 클래스**
  - 버튼 기본 클래스
  - 파생클래스 (Button, CheckBox, RadioButton)
  
- **ButtonBase 클래스 상속받은 버튼 계열 클래스 사용**

- 1) 객체 선언
- 2) 객체 생성 및 초기화
- 3) 이벤트 등록
- 4) 등록한 이벤트가 발생하면 호출될 메서드 구현

### 사용예제

- 버튼 예제

```
using System;
using System.Drawing;
using System.Windows.Forms;

public class ButtonExam : Form
{
    Button btn;
    public ButtonExam()
    {
        btn = new Button();
        btn.Parent = this;
        btn.Text = "클릭";
        btn.Location = new Point(100, 100);
        btn.Click += new EventHandler(btn_Click);
    }
    private void btn_Click(object sender, EventArgs e)
    {
        Graphics g_form = this.CreateGraphics();
        Graphics g_button = btn.CreateGraphics();

        g_form.FillRectangle(Brushes.Green, this.ClientRectangle);
        g_button.FillRectangle(Brushes.Red, btn.ClientRectangle);
    }
}
```

```
public static void Main()
{
    Application.Run(new ButtonExam());
}
```

- CheckBox 컨트롤

- 특정 항목을 선택/해제 할 때 사용하는 컨트롤

속성	자료형	설명
Checked	bool	- 체크 선택 여부 반환, 기본값은 false
AutoCheck	bool	- true : 자동 토글( 기본값 true)

- RadioButton 컨트롤

- 하나만 설정 가능
- 속성은 CheckBox와 동일

[41]

 비트교육센터

### 사용 예제

- 체크, 라디오버튼 예제

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

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

        private void button1_Click(object sender, EventArgs e)
        {
            string message = "취미는";
            if(checkBox1.Checked)
            {
                message += "Win -" + checkBox1.Text;
            }
        }
    }
}

```

```
        }

        if (checkBox2.Checked)
        {
            message += "Wn -" + checkBox2.Text;
        }

        if (checkBox3.Checked)
        {
            message += "Wn -" + checkBox3.Text;
        }

        message += "Wn성별은 ";
        if(radioButton1.Checked)
        {
            message += radioButton1.Text;
        }
        else if(radioButton2.Checked)
        {
            message += radioButton2.Text;
        }

        MessageBox.Show(message);
    }
}
```

## 01 Label

### • Label 클래스

- 텍스트를 출력하기 위해 만들어진 컨트롤
- 출력할 텍스트의 길이가 컨트롤의 폭보다 길 경우 텍스트를 여러 줄로 출력하게 됨

속성	자료형	설명
BorderStyle	BorderStyle	- Label의 테두리 스타일을 가져오거나 설정
Text	string	- Label에 출력된 문자열을 가져오거나 설정
TextAlign	ContentAlignment	- Label의 텍스트 맞춤을 가져오거나 설정

[42]

 비트교육센터

### 사용 예제

- 버튼과 라벨을 이용한 이용 예제(계산기)

```
using System;
using System.Drawing;
using System.Windows.Forms;
public class Calculator : System.Windows.Forms.Form
{
    Button[] btn = new Button[16];
    Label lbl_display;
    int iResult;
    int option = 0;
    int iButtonSize = 50;
    public Calculator()
    {
        int i = 0;
        this.Text = "간단한 계산기";
        lbl_display = new Label();
        lbl_display.Text = "0";
        lbl_display.BackColor = Color.Yellow;
        lbl_display.TextAlign = ContentAlignment.MiddleRight;
        lbl_display.SetBounds(50, 10, 200, 20);
        this.Controls.Add(lbl_display);
        //lbl_display.Parent = this;
        for (i = 0; i < 10; i++)
    {
```

```

        btn[i] = new Button();
        btn[i].Text = i.ToString();
        btn[i].SetBounds(iButtonSize + (i % 3 * iButtonSize),
                         iButtonSize + (i / 3 * iButtonSize), iButtonSize, iButtonSize);
        this.Controls.Add(btn[i]);
        //btn[i].Parent = this;
    }

    // 사칙 연산 버튼 문자열
    string[] str = { "+", "-", "*", "/" };
    for (i = 0; i < 4; i++)
    {
        btn[i + 10] = new Button();
        btn[i + 10].Text = str[i];
        btn[i + 10].SetBounds(4 * iButtonSize, iButtonSize + (i * iButtonSize),
                             iButtonSize, iButtonSize);
        this.Controls.Add(btn[i + 10]);
    }

    // 클리어 초기화 버튼 생성
    btn[14] = new Button();
    btn[14].Text = "CLR";
    btn[14].SetBounds(2 * iButtonSize, 4 * iButtonSize, iButtonSize, iButtonSize);
    this.Controls.Add(btn[14]);
    btn[15] = new Button();

    // 결과 버튼 생성
    btn[15].Text = "ANS";
    btn[15].SetBounds(3 * iButtonSize, 4 * iButtonSize, iButtonSize, iButtonSize);
    this.Controls.Add(btn[15]);
    for (i = 0; i < 16; i++)
    {
        btn[i].Click += new EventHandler(Calculator_Click);
    }
}

private void Calculator_Click(object sender, EventArgs e)
{
    Button obj = (Button)sender;
    if (obj.Text == "+")
    {
        iResult = Int32.Parse(lbl_display.Text);
        option = 1;
        obj.ForeColor = Color.Red;
        lbl_display.Text = "";
    }
    else if (obj.Text == "-")
    {
        iResult = Int32.Parse(lbl_display.Text);
        option = 2;
    }
}

```

```

        obj.ForeColor = Color.Red;
        lbl_display.Text = "";
    }
    else if (obj.Text == "*")
    {
        iResult = Int32.Parse(lbl_display.Text);
        option = 3;
        obj.ForeColor = Color.Red;
        lbl_display.Text = "";
    }
    else if (obj.Text == "/")
    {
        iResult = Int32.Parse(lbl_display.Text);
        option = 4;
        obj.ForeColor = Color.Red;
        lbl_display.Text = "";
    }
    else if (obj.Text == "CLR")
    {
        btn[9 + option].ForeColor = Color.Black;
        iResult = 0;
        lbl_display.Text = "";
    }
    else if (obj.Text == "ANS")
    {
        if (option == 1)
        {
            iResult = iResult + Int32.Parse(lbl_display.Text);
            btn[10].ForeColor = Color.Black;
        }
        else if (option == 2)
        {
            iResult = iResult - Int32.Parse(lbl_display.Text);
            btn[11].ForeColor = Color.Black;
        }
        else if (option == 3)
        {
            iResult = iResult * Int32.Parse(lbl_display.Text);
            btn[12].ForeColor = Color.Black;
        }
        else if (option == 4)
        {
            iResult = iResult / Int32.Parse(lbl_display.Text);
            btn[13].ForeColor = Color.Black;
        }
        this.lbl_display.Text = iResult.ToString();
    }
    else
    {

```

```
        this.lbl_display.Text += obj.Text;
    }
}

static void Main()
{
    Application.Run(new Calculator());
}
}
```

- ScrollBar 클래스

- 추상화 클래스이므로 이를 직접 사용하지 않고 파생된 HScrollBar 나 VScrollBar로 스크롤 기능을 구현

속성	자료형	설명
Minimum	int	- 스크롤 값의 하한 값을 가져오거나 설정
Maximum	int	- 스크롤 값의 상한 값을 가져오거나 설정
SmallChange	int	- 스크롤을 상자를 조금 움직였을때 value에 추가하거나 뺄값을 가져오거나 설정
LargeChange	int	- 스크롤을 상자를 많이 움직일 때 Value에 추가하거나 뺄값을 가져오거나 설정
Value	int	- 스크롤바에 있는 스크롤 상자의 현재 위치를 나타내는 숫자를 가져오거나 설정

## 01 ScrollBar

### • ScrollBar 클래스 이벤트

- 추상화 클래스이므로 파생된 HScrollBar 나 VScrollBar로 스크롤 기능을 구현

이벤트	메서드	설명
ValueChanged	OnValueChanged	<ul style="list-style-type: none"><li>- 프로그램에서 value속성이 변경될 때 발생</li><li>- 스크롤바를 조작할 때 발생하지 않음</li></ul>
Scroll	OnScroll	<ul style="list-style-type: none"><li>- 마우스 또는 키보드 동작으로 스크롤 상자가 움직일 때 발생(Value값이 변할 때는 발생하지 않음)</li></ul>

### • ScrollEventArgs 클래스 속성

속성	자료형	설명
NetValue	int	<ul style="list-style-type: none"><li>- 스크롤 막대의 새 Value값을 가져오거나 설정</li></ul>
OldValue	int	<ul style="list-style-type: none"><li>- 변경 전 Value속성에 포함된 숫자 값</li></ul>
Type	ScrollEventType	<ul style="list-style-type: none"><li>- 스크롤 이벤트를 발생시키는데 사용된 동작의 종류 반환</li></ul>
ScrollOrientation	ScrollOrientation	<ul style="list-style-type: none"><li>- Scroll이벤트를 발생시킨 스크롤바의 막대 방향 반환</li></ul>

[44]

 비트교육센터

## 사용 예제

### • 스크롤바 예제

```
using System;
using System.Drawing;
using System.Windows.Forms;

public class ScrollExam : Form
{
    Panel top_panel = null;
    Panel panel = null;
    Label[] lbl = new Label[6];
    VScrollBar[] vscroll = new VScrollBar[3];
    HScrollBar hscroll = null;
    string[] strcolor = { "빨강", "초록", "파랑" };
    int r, g, b;

    public ScrollExam()
    {
        r = g = b = 0;
        int cx = this.ClientSize.Width;
        int cy = this.ClientSize.Height;

        top_panel = new Panel();
        top_panel.Parent = this;
        top_panel.Location = new Point(cx / 2, 0);

        top_panel = new Panel();
```

```

top_panel.Size = new Size(cx / 2, 100);
top_panel.BackColor = Color.Blue;

hscroll = new HScrollBar();
hscroll.Parent = top_panel;
hscroll.Minimum = 0;
hscroll.Maximum = 255;
hscroll.TabStop = true;
hscroll.ValueChanged += new EventHandler(ScrollExam_ValueChanged);
hscroll.Scroll += new ScrollEventHandler(ScrollExam_Scroll);
hscroll.Location = new Point(20, 30);
hscroll.Size = new Size(cx / 2 - 30, 20);

panel = new Panel();
panel.Parent = this;
panel.Location = new Point(0, 0);
panel.Size = new Size(cx / 2, cy);
panel.BackColor = Color.Yellow;

for (int i = 0; i < 3; i++)
{
    lbl[i * 2] = new Label();
    lbl[i * 2].Parent = panel;
    lbl[i * 2].ContentAlignment = ContentAlignment.MiddleCenter;
    lbl[i * 2].Text = strcolor[i];
    lbl[i * 2].Location = new Point(i * cx / 6, 5);
    lbl[i * 2].Size = new Size(cx / 6, 15);

    vscroll[i] = new VScrollBar();
    vscroll[i].Parent = panel;
    vscroll[i].SmallChange = 1;
    vscroll[i].LargeChange = 15;
    vscroll[i].Minimum = 0;
    vscroll[i].Maximum = 255;
    vscroll[i].ValueChanged += new EventHandler(ScrollExam_ValueChanged);
    vscroll[i].TabStop = true;
    vscroll[i].Location = new Point((4 * i + 1) * cx / 24, 24);
    vscroll[i].Size = new Size(cx / 12, cy - 4 * 12);

    lbl[2 * i + 1] = new Label();
    lbl[2 * i + 1].Parent = panel;
    lbl[2 * i + 1].ContentAlignment = ContentAlignment.MiddleCenter;
    lbl[2 * i + 1].Text = "0";
    lbl[2 * i + 1].Location = new Point(i * cx / 6, cy - 3 * 6);
    lbl[2 * i + 1].Size = new Size(cx / 6, 15);
}
}

private void ScrollExam_ValueChanged(object sender, EventArgs e)

```

```

{
    ScrollBar obj = (ScrollBar)sender;
    if (obj == vscroll[0])           //빨강색을 처리하는 VScrollBar에서 이벤트 발생
    {
        r = obj.Value;
        lbl[1].Text = r.ToString();
    }
    else if (obj == vscroll[1])//초록색을 처리하는 VScrollBar에서 이벤트 발생
    {
        g = obj.Value;
        lbl[3].Text = g.ToString();
    }
    else if (obj == vscroll[2])//파랑색을 처리하는 VScrollBar에서 이벤트 발생
    {
        b = obj.Value;
        lbl[5].Text = b.ToString();
    }
    else if (obj == hscroll)//HScrollBar에서 ValueChanged이벤트가 발생
    {
        this.Text = "알파(투명도)값 :" + obj.Value;

    }
    this.BackColor = Color.FromArgb(r, g, b); //윈도우의 배경색을 설정된 값으로 변경
}

// HScrollBar에서 Scroll 이벤트가 발생하면 호출되는 메서드
private void ScrollExam_Scroll(object sender, ScrollEventArgs e)
{
    string str = String.Format("NewValue = {0}, OldValue = {1}, ScrollEventType = {2}, ScrollOrientation = {3}", e.NewValue, e.OldValue, e.Type, e.ScrollOrientation);
    this.Text = str;
}

public static void Main()
{
    Application.Run(new ScrollExam());
}
}

```

## 01 컨테이너 클래스

### • 컨테이너 클래스

- 다른 컴포넌트를 담을 수 있는 클래스
- 컨테이너 안에 포함된 컨트롤들은 컨테이너가 움직일 때 같이 움직임
- 출력할 컨트롤이 많고 컨트롤 간의 구성이 복잡할 때 사용하게 됨

종류	설명
Panel	<ul style="list-style-type: none"><li>- 다른 컨트롤을 구별할 수 있도록 그룹화하는데 사용</li><li>- Panel 컨트롤은 스크롤 막대가 있음</li></ul>
GroupBox	<ul style="list-style-type: none"><li>- 다른 컨트롤을 구별할 수 있도록 그룹화하는데 사용</li><li>- GroupBox 컨트롤은 캡션을 표시</li></ul>
TabControl	<ul style="list-style-type: none"><li>- 여러 개의 탭을 이용해 Panel을 겹쳐놓은 것과 같은 효과를 주는 컨트롤</li></ul>
SplitContainer	<ul style="list-style-type: none"><li>- 이동 가능한 막대로 구분된 두 개의 패널로 구성</li></ul>
FlowLayoutPanel	<ul style="list-style-type: none"><li>- 내용을 가로 또는 세로로 동적 배열하는 패널</li></ul>
TableLayoutPanel	<ul style="list-style-type: none"><li>- 행과 열로 구성된 표 형태로 내용을 동적으로 배열하는 패널</li></ul>

[45]

 비트교육센터

### 사용예제

- 컨테이너 클래스중 가장 많이 사용되는 컨트롤 : Panel , GroupBox

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace PanelExam
{
    public partial class Form1 : Form
    {
        int index_color = 0;
        int index_border = 0;
        public Form1()
        {
            InitializeComponent();
        }

        private void button11_Click(object sender, EventArgs e)
        {
            // Panel 바탕색 바꾸기
            if(index_color == 0){
```

```

        this.panel.BackColor = Color.Red;
        this.panel.BackgroundImage = null;
    }
    else if(index_color == 1){
        this.panel.BackColor = Color.Yellow;
    }
    else if (index_color == 2){
        this.panel.BackColor = Color.Blue;
    }
    else if (index_color == 3){
        this.panel.BackgroundImage = Image.FromFile("회색.bmp");
    }

    if(index_color > 3){
        index_color = 0;
    }
    else{
        index_color++;
    }
}

private void button12_Click(object sender, EventArgs e)
{
    // Panel BoarderStyle 바꾸기
    if(index_border == 0){
        this.panel.BorderStyle = BorderStyle.None;
    }
    else if(index_border == 1){
        this.panel.BorderStyle = BorderStyle.FixedSingle;
    }
    else if(index_border == 2){
        this.panel.BorderStyle = BorderStyle.Fixed3D;
    }

    if (index_border > 2){
        index_border = 0;
    }
    else{
        index_border++;
    }
}
}
}

```

# **12**

Chapter

텍스트 박스, 리스트 박스, 콤보박스

## 01 TextBox

### • TextBoxBase 클래스

- TextBoxBase는 텍스트 컨트롤의 기본 기능이 구현되어 있는 추상화 클래스
- 이 클래스를 상속받은 TextBox, MaskedTextBox, RichTextBox 클래스를 이용하여 문자열 입출력을 쉽게 다룸

종류	설명
TextBox	<ul style="list-style-type: none"><li>- 한 줄 또는 여로 줄짜리 문자열을 입력받을 때 사용되며 텍스트 박스 컨트롤 중 가장 많이 사용됨</li></ul>
MaskedTextBox	<ul style="list-style-type: none"><li>- 입력받는 문자에 대해 마스크(예 *, #...) 또는 특정 형식(달력, 통화 입력 등)으로 값을 입력 받으며, 지정된 형식을 벗어난 입력이 들어오면 MaskInputRejected 이벤트 발생</li></ul>
RichTextBox	<ul style="list-style-type: none"><li>- 서식 있는 RTF(Rich Text Format)와 아스키 텍스트 파일을 다룰 때 사용되는 컨트롤</li><li>- 다양한 폰트를 사용한 글꼴 효과 및 OLE 객체 삽입 등의 고급 기능을 사용할 수 있음</li></ul>

[47]

 비트교육센터

### 사용 예제

#### • TextBox 예제

```
using System;
using System.Drawing;
using System.Windows.Forms;

public class TextBoxExam : Form
{
    Label lbl_txt = new Label();
    TextBox txt1 = new TextBox(); // 단일 라인
    TextBox txt2 = new TextBox(); // 멀티 라인
    MaskedTextBox mask_txt = new MaskedTextBox();
    ToolTip toolTip1 = new ToolTip();

    public TextBoxExam()
    {
        lbl_txt.Text = "TextBox 예제";
        lbl_txt.Parent = this;
        lbl_txt.BorderStyle = BorderStyle.Fixed3D;
        lbl_txt.SetBounds(10, 10, 500, 20);

        txt1.Parent = this;
        txt1.TextChanged += new EventHandler(txt_TextChanged);
        txt1.BorderStyle = BorderStyle.Fixed3D;
        txt1.SetBounds(10, 50, 500, 20);
    }
}
```

```

txt1.BackColor = Color.Black;
txt1.ForeColor = Color.White;

txt2.Parent = this;
txt2.TextChanged += new EventHandler(txt_TextChanged);
txt2.BorderStyle = BorderStyle.Fixed3D;
txt2.Multiline = true;           // 멀티라인
txt2.ScrollBars = ScrollBars.Both; // 스크롤바 설정
txt2.SetBounds(10, 100, 500, 100);

mask_txt.Parent = this;
mask_txt.TextChanged += new EventHandler(txt_TextChanged);
mask_txt.BorderStyle = BorderStyle.Fixed3D;
mask_txt.SetBounds(10, 220, 500, 20);
mask_txt.PasswordChar = '*'; // 패스워드 문자를 등록
mask_txt.Mask = "0000/00/00"; // 마스크 형식 지정
mask_txt.MaskInputRejected += new
MaskInputRejectedEventHandler(maskedTextBox1_MaskInputRejected);
mask_txt.KeyDown += new KeyEventHandler(maskedTextBox1_KeyDown); // 키 입력 이벤트

this.Size = new Size(550, 300);
}

private void txt_TextChanged(object sender, EventArgs e)
{
    string str = "";
    if ((TextBoxBase)sender == txt1) {
        this.Text = "단일라인 문자열 입력";
        str = txt1.Text;
    }
    else if ((TextBoxBase)sender == txt2) {
        this.Text = "멀티라인 문자열 입력";
        str = txt2.Text;
    }
    else if ((TextBoxBase)sender == mask_txt) {
        this.Text = "마스크 문자열 입력";
        str = mask_txt.Text;
    }
    lbl_txt.Text = str;
}

void maskedTextBox1_MaskInputRejected(object sender,
MaskInputRejectedEventArgs e)
{
    if (mask_txt.MaskFull) {
        toolTip1.ToolTipTitle = "너무 많은 데이터 입력!";
        toolTip1.Show("입력 가능한 데이터는 8개까지입니다.", mask_txt,
mask_txt.Location.X, 20, 5000);
    }
}

```

```

        else {
            toolTip1.ToolTipTitle = "입력 문자가 잘못됨";
            toolTip1.Show("입력 가능한 문자는 0~9사이입니다.", mask_txt,
mask_txt.Location.X, 20, 5000);
        }
    }

void maskedTextBox1_KeyDown(object sender, KeyEventArgs e)
{
    toolTip1.Hide(mask_txt); // 키 입력이 발생하면 툴팁 지움
}

public static void Main()
{
    Application.Run(new TextBoxExam());
}

```

- RichTextBox 예제

```

using System;
using System.Drawing;
using System.Windows.Forms;

public class RichTextBoxExam : Form
{
    RichTextBox richTextBox;
    public RichTextBoxExam()
    {
        richTextBox = new RichTextBox();
        richTextBox.Parent = this;
        richTextBox.Dock = DockStyle.Fill;
        this.Load += new EventHandler(RichTextBoxExam_Load);
    }

    private void RichTextBoxExam_Load(object sender, EventArgs e)
    {
        richTextBox.LoadFile("song.rtf");      // RTF 파일 읽기
        richTextBox.Find("소나무 츄갑을", RichTextBoxFinds.MatchCase); // 검색 후 반전

        richTextBox.SelectionFont = new Font("궁서체", 30, FontStyle.Bold); // 선택된 문
자열 글꼴 변경
        richTextBox.SelectionColor = Color.Blue; // 선택된 문자열 색깔 변경
    }

    public static void Main()
    {
        Application.Run(new RichTextBoxExam());
    }
}

```

## 01 ListBox

### • ListControl 클래스

- 리스트 항목 관리 사용
- Items 컬렉션, 리스트의 인덱스를 설정할 때 사용하는 SelectedIndex 속성
- 리스트 항목을 선택할 때 발생하는 SelectedIndexChanged 이벤트 등을 제공함
- 제한된 공간에서 여러 아이템을 사용할 수 있는 기능 제공

종류	설명
ListBox	- 리스트 항목 관리 사용
CheckedListBox	- 각 항목의 왼쪽 확인란에 체크 박스를 표시함 - 리스트 박스에 포함된 아이템 항목들의 체크 유무를 관리하는 CheckedItems 속성을 제공
ComboBox	- 리스트박스와 텍스트 박스를 결합한 컨트롤임

[48]

 비트교육센터

### 사용 예제

#### • ListBox 예제

```
using System;
using System.Drawing;
using System.Windows.Forms;

public class ListBoxExam : Form
{
    private CheckedListBox checkedListBox1;
    private ListBox listBox1;
    private TextBox txt_info;
    string[] str1 = { "딸기", "바나나", "포도", "귤", "호두", "사과", "수박", "체리" };
    string[] str2 = { "서울", "경기", "충청", "전라", "경상", "제주" };

    public ListBoxExam()
    {
        listBox1 = new ListBox();
        checkedListBox1 = new CheckedListBox();
        txt_info = new TextBox();

        listBox1.Parent = this;
        listBox1.SetBounds(10, 10, 50, 100);
        listBox1.SelectedIndexChanged += new EventHandler(SelectedIndexChanged);
        checkedListBox1.Parent = this;
        checkedListBox1.SetBounds(70, 10, 50, 100);
```

```

checkedListBox1.SelectedIndexChanged += new
EventHandler(SelectedIndexChanged);

txt_info.Parent = this;
txt_info.SetBounds(10, 120, 300, 120);
txt_info.Multiline = true;

for (int i = 0; i < str1.Length; i++)
    listBox1.Items.Add(str1[i]);

for (int j = 0; j < str2.Length; j++)
    checkedListBox1.Items.Add(str2[j]);
}

private void SelectedIndexChanged(object sender, EventArgs e)
{
    ListControl obj = (ListControl)sender;
    if (obj == listBox1)
    {
        txt_info.Text += "WrWn [listBox 선택] : " + listBox1.SelectedItem;
    }
    else if (obj == checkedListBox1)
    {
        string str = "";
        for (int i = 0; i < checkedListBox1.CheckedItems.Count; i++)
        {
            str += "Wt" + (i + 1).ToString() + " : " +
checkedListBox1.CheckedItems[i].ToString();
        }
        txt_info.Text += "WrWn [CheckListBox 선택] : " + str;
    }
}

public static void Main()
{
    Application.Run(new ListBoxExam());
}
}

```

- ComboBox 예제

```

using System;
using System.Drawing;
using System.Windows.Forms;

public class ComboBoxExam : Form
{
    private ComboBox cbb;
    public ComboBoxExam()
    {

```

```

        cbb = new ComboBox();
        cbb.Parent = this;
        cbb.SelectedValueChanged += new EventHandler(cbb_SelectedValueChanged);
        cbb.SelectionChangeCommitted += new
EventHandler(cbb_SelectionChangeCommitted);
        cbb.Location = new Point(100, 100);
        cbb.Text = "색을 선택하세요";
        cbb.Items.Add("빨강");
        cbb.Items.Add("노랑");
        cbb.Items.Add("파랑");
        cbb.Items.Add("검정");
        cbb.Items.Add("회색");
    }

private void cbb_SelectedValueChanged(object sender, EventArgs e)
{
    MessageBox.Show("SelectedValueChanged 이벤트 발생");
    Color color = Color.White;
    ComboBox obj = (ComboBox)sender;
    switch (obj.SelectedIndex)
    {
        case 0:
            color = Color.Red;
            break;
        case 1:
            color = Color.Yellow;
            break;
        case 2:
            color = Color.Blue;
            break;
        case 3:
            color = Color.Black;
            break;
        case 4:
            color = Color.White;
            break;
    }
    this.BackColor = color;
}

private void cbb_SelectionChangeCommitted(object sender, EventArgs e)
{
    MessageBox.Show("SelectionChangeCommitted 이벤트 발생");
    this.Text = ((ComboBox)sender).SelectedText;
}

public static void Main()
{
    Application.Run(new ComboBoxExam());
}

```

# **13**

Chapter

대화상자

- 모달(Modal)

- 모달이 띄워지면 대화상자를 종료하지 않는 한 다른 윈도우에 접근이 불가능
- 어느 하나의ダイ얼로그창이 프로그램의 모든 제어권을 독점
- 이ダイ얼로그창이 종료되기 전에는 다른 작업 불가능( 다른 창 활성화가 불가능)

- 모달리스 (Modeless)

- 모달리스 대화상자가 띄워져도 다른 윈도우에 접근이 가능
- 어느 하나의ダイ얼로그창이 프로그램의 제어권을 독점하지 않음
- 다른 작업이 가능( 다른 창 활성화가 가능 )

- 구현 방법

- 모달 : Form 클래스가 제공하는 ShowDialog() 호출
- 모달리스 : Form 클래스가 제공하는 Show() 메서드 호출

- 공통 대화상자

종류	기능	설명
OpenFileDialog	열기	- 파일을 열기 위해 사용하는 대화상자
SaveFileDialog	저장	- 파일 저장을 위한 대화상자
ColorDialog	색	- 색상을 선택하는 대화상자
FontDialog	글꼴	- 글꼴 크기와 스타일을 선택하는 대화상자
FolderBrowserDialog	찾아보기	- 파일 디렉토리를 찾기 위한 대화상자

### 사용 예제

- 공통 대화상자

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

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

        private void btn_FileOpen_Click(object sender, EventArgs e)
        {
            OpenFileDialog dlg = new OpenFileDialog();
            dlg.Filter = "RichText files (*.rtf)|*.rtf|All files (*.*)|*.*";
            dlg.InitialDirectory = "c:\\";
            dlg.Title = "파일 열기 대화상자 예제";
        }
    }
}

```

```

        if(dlg.ShowDialog() == DialogResult.OK)
        {
            this.textBox1.Text = dlg.FileName + " [[파일열기]]";
            this.richTextBox1.LoadFile(dlg.FileName,
RichTextBoxStreamType.RichText);
        }
    }

private void btn_FileSave_Click(object sender, EventArgs e)
{
    SaveFileDialog dlg = new SaveFileDialog();

    dlg.Title = "파일 저장 대화상자 예제";
    dlg.CreatePrompt = true;
    dlg.OverwritePrompt = true;
    dlg.FileName = "default";
    dlg.DefaultExt = "rtf";
    dlg.InitialDirectory = "c:\\WW";
    dlg.Filter = "RichText files (*.rtf)|*.rtf";

    System.IO.MemoryStream memstream = new System.IO.MemoryStream();
    this.richTextBox1.SaveFile(memstream, RichTextBoxStreamType.RichText);

    if (dlg.ShowDialog() == DialogResult.OK)
    {
        try
        {
            System.IO.Stream fs = dlg.OpenFile();
            memstream.Position = 0;
            memstream.WriteTo(fs);
            fs.Close();
            this.textBox1.Text = dlg.FileName + " [[파일저장]]";
        }
        catch(Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

private void btn_Font_Click(object sender, EventArgs e)
{
/*
    FontDialog dlg = new FontDialog();
    if(dlg.ShowDialog() == DialogResult.OK)
    {
        this.richTextBox1.SelectionFont = dlg.Font;

```

```

        this.textBox1.Text = "글꼴을 " + dlg.Font + " 로 변경!!!";
    }
    */
    FontDialog dlg = new FontDialog();
    dlg.ShowApply = true;
    dlg.ShowColor = true;

    if (dlg.ShowDialog() == DialogResult.OK)
    {
        this.richTextBox1.SelectionFont = dlg.Font;
        this.textBox1.Text = "글꼴을 " + dlg.Font + " 로 변경!!!";
    }
}

private void btn_Color_Click(object sender, EventArgs e)
{
    ColorDialog dlg = new ColorDialog();
    if(dlg.ShowDialog() == DialogResult.OK)
    {
        this.richTextBox1.SelectionColor = dlg.Color;
        this.textBox1.Text = "색상을 " + dlg.Color + " 로 변경!!!";
    }
}

private void btn_FolderBrowser_Click(object sender, EventArgs e)
{
    FolderBrowserDialog dlg = new FolderBrowserDialog();
    dlg.Description = "디렉토리를 지정하세요!";
    dlg.ShowNewFolderButton = true;
    dlg.RootFolder = Environment.SpecialFolder.MyComputer;

    if(dlg.ShowDialog() == DialogResult.OK)
    {
        this.textBox1.Text = "선택한 폴더: " + dlg.SelectedPath;
    }
}
}

```

# **14**

Chapter

드래그 앤 드롭, 클립보드

- 드래그 앤 드롭

- 마우스로 끌어서 프로그램을 실행하는 것
- 관련 이벤트

종류	설명
DragEnter	- 컨트롤이나 폼의 AllowDrop 속성이 true로 설정되었을 때, 어떤 개체를 드래깅하는 마우스 포인터가 처음으로 컨트롤이나 폼의 클라이언트 영역에 들어왔을 때 발생
DragOver	- DragEnter 이벤트 발생 후에는 마우스 컨트롤이나 폼 클라이언트 영역안에서 움직일 때마다 발생
DragLeave	- 마우스가 컨트롤이나 폼의 클라이언트 영역밖으로 나가면 발생
DragDrop	- 마우스 커서가 컨트롤이나 폼이 클라이언트 영역위에 있을 때, 마우스를 놓으면 발생

- 클립 보드

- 원도우 운영체제가 제공하는 강력하고 유용한 임시 기억 공간

[53]

 비트교육센터

## 사용 예제

- 드래그 앤 드롭 예제

```

using System;
using System.Drawing;
using System.Text;
using System.IO;
using System.Windows.Forms;

namespace DragandDropExam
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // 드래그 앤 드롭 기능 사용 설정(폼 드롭 기능 활성화)
            this.AllowDrop = true;

            // 드래그 앤 드롭이 가능한 텍스트 박스 컨트롤 구현
            this.txt_box.AllowDrop = true; // TextBox 드롭 기능 활성화
            // DragOber : 마우스가 컨트롤이나 폼 안에 들어와서 움직일 때마다 호출
            this.txt_box.DragOver += new DragEventHandler(txt_box_DragOver);
            // 마우스가 컨트롤이나 폼 안에 존재할 때, 마우스를 놓으면 발생
            this.txt_box.DragDrop += new DragEventHandler(txt_box_DragDrop);
        }
    }
}

```

```

// 드래그 앤 드롭이 가능한 리치에디트박스 컨트롤 구현
this.rich_txt_box.AllowDrop = true; // RichTextBox 드롭 기능 활성화
// DragEnter : 드래깅 하는 마우스 포인터가 처음으로 컨트롤이나 폼에 들어왔을
경우
this.rich_txt_box.DragEnter += new
DragEventHandler(rich_txt_box_DragEnter);
this.rich_txt_box.DragDrop += new
DragEventHandler(rich_txt_box_DragDrop);

this.DragOver += new DragEventHandler(pic_box_DragOver); //DragOver 이벤
트
this.DragDrop += new DragEventHandler(pic_box_DragDrop); //DragDrop 이
벤트
}

// TextBox 위에 마우스가 위치하면
void txt_box_DragOver(object sender, DragEventArgs dea)
{
    // GetDataPresent : 확인할 형식에 해당되는 데이터가 Data 속성에
    // 포함되어 있는지를 알려줌
    // 형식) 21가지의 형식 : Bitmap, Dib, FileDrop(파일 놓기)...

    // 드래그한 데이터가 파일이거나 문자열 타입이라면
    if (dea.Data.GetDataPresent(DataFormats.FileDrop) ||
        dea.Data.GetDataPresent(DataFormats.StringFormat))
    {
        // 드래그한 데이터가 Move 기능을 허용한다면
        if ((dea.AllowedEffect & DragDropEffects.Move) != 0)
            // 끌기 소스의 데이터가 놓기 대상으로 이동
            dea.Effect = DragDropEffects.Move;// 코드 존재 유무

        // 드래그한 데이터가 Ctrl 키가 눌린 상태에서 Copy를 허용한다면
        // keyState 1 : 마우스 왼쪽 버튼
        //      2 : 마우스 오른쪽 버튼
        //      4 : Shift 키
        //      8 : Ctrl 키
        //     16 : 마우스 중간 버튼
        //     32 : Alt 키
        if(((dea.AllowedEffect & DragDropEffects.Copy) != 0) &&
           ((dea.KeyState & 8) != 0))
            dea.Effect = DragDropEffects.Copy;// 데이터가 놓기 대상에 복사됨

    }
}

// 마우스버튼을 놓을 때 발생
void txt_box_DragDrop(object sender, DragEventArgs dea)
{

```

```

if (dea.Data.GetDataPresent(DataFormats.FileDrop))
{
    string[] fname = (string[])dea.Data.GetData(DataFormats.FileDrop);
    TextReader tr = new StreamReader(fname[0]); // 파일 읽기
    txt_box.Text = tr.ReadToEnd(); // 텍스트 박스에 출력
    tr.Close(); // 파일 닫기
}
}

void rich_txt_box_DragEnter(object sender, DragEventArgs dea)
{
    // 끌기 소스가 파일이나 문자열일 경우
    if (dea.Data.GetDataPresent(DataFormats.FileDrop) ||
        dea.Data.GetDataPresent(typeof(string)))
    {
        // 드래그한 개체의 Effect 속성을 Move 속성으로 변경
        if ((dea.AllowedEffect & DragDropEffects.Move) != 0)
            dea.Effect = DragDropEffects.Move;

        // Copy 속성 및 Ctrl 키
        // - 1 : 마우스 왼쪽 버튼
        // - 2 : 마우스 오른쪽 버튼
        // - 4 : Shift 키
        // - 8 : Ctrl 키
        // - 16 : 마우스 중간 버튼
        // - 32 : Alt 키
        if (((dea.AllowedEffect & DragDropEffects.Copy) != 0) &&
            ((dea.KeyState & 8) != 0)) // Ctrl 키가 눌린 경우
            dea.Effect = DragDropEffects.Copy;
    }
}

void rich_txt_box_DragDrop(object sender, DragEventArgs dea)
{
    if (dea.Data.GetDataPresent(DataFormats.FileDrop))
    {
        string[] fname = (string[])dea.Data.GetData(DataFormats.FileDrop);
        this.rich_txt_box.LoadFile(fname[0]);
    }
}

void pic_box_DragOver(object sender, DragEventArgs dea)
{
    if (dea.Data.GetDataPresent(DataFormats.FileDrop) ||
        dea.Data.GetDataPresent(typeof(Bitmap)))
    {
        if ((dea.AllowedEffect & DragDropEffects.Move) != 0)
            dea.Effect = DragDropEffects.Move;
    }
}

```

```

        if (((dea.AllowedEffect & DragDropEffects.Copy) != 0) &&
            ((dea.KeyState & 8) != 0))
            dea.Effect = DragDropEffects.Copy;
    }
}

void pic_box_DragDrop(object sender, DragEventArgs dea)
{
    if (dea.Data.GetDataPresent(DataFormats.FileDrop))
    {
        string [] fname = (string [])dea.Data.GetData(DataFormats.FileDrop);
        try
        {
            Image img = Image.FromFile(fname[0]);
            this.pic_box.Image = img;
        }
        catch(Exception ex)
        {
            MessageBox.Show(ex.Message);
            return;
        }
    }
}
}

```

- 클립 보드 예제

```

using System;
using System.IO;
using System.Drawing;
using System.Windows.Forms;

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

        private void btn_clipinfo_Click(object sender, EventArgs e)
        {
            if (Clipboard.ContainsImage())
            {
                txt_info.AppendText("1. 이미지 파일 포함되어 있음...WrWn");
                btn_addImage.Enabled = true;
                btn_getImage.Enabled = true;
            }
        }
}

```

```

        else
        {
            txt_info.AppendText("1. 이미지 파일이 포함되지 않음...WrWn");
            btn_addImage.Enabled = true;
        }

        if (Clipboard.ContainsAudio())
        {
            txt_info.AppendText("2. 오디오 파일 포함되어 있음...WrWn");
            btn_addAudio.Enabled = true;
            btn_getAudio.Enabled = true;
        }
        else
        {
            txt_info.AppendText("2. 오디오 파일이 포함되지 않음...WrWn");
            btn_addAudio.Enabled = true;
        }

        if (Clipboard.ContainsText())
        {
            txt_info.AppendText("3. 텍스트 포함되어 있음...WrWn");
            btn_addText.Enabled = true;
            btn_getText.Enabled = true;
        }
        else
        {
            txt_info.AppendText("3. 텍스트 포함되지 않음...WrWn");
            btn_addText.Enabled = true;
        }
    }

    private void btn_addImage_Click(object sender, EventArgs e)
    {
        OpenFileDialog dlg = new OpenFileDialog();
        dlg.Title = "클립보드에 추가할 이미지를 선택하세요~";
        dlg.Filter = "*.* | *.bmp | *.gif | *.jpg";
        if(dlg.ShowDialog() == DialogResult.OK)
        {
            Image img = Image.FromFile(dlg.FileName);
            Clipboard.SetImage(img);      // 클립보드에 이미지 추가
            txt_info.AppendText("클립보드에 " + dlg.FileName + " 이미지 추가 WrWn");
        }
    }

    private void btn_getImage_Click(object sender, EventArgs e)
    {
        if (Clipboard.ContainsImage())
        {

```

```

    Bitmap bmp = new Bitmap(Clipboard.GetImage()); // 클립보드에서 이미지
    추출
        bmp.Save("c:\\\\클립보드이미지.bmp");
        txt_info.AppendText("클립보드 이미지를 C:\\\\에 클립보드이미지.bmp로 출
        력 WrWn");
    }
}

private void btn_addAudio_Click(object sender, EventArgs e)
{
    OpenFileDialog dlg = new OpenFileDialog();
    dlg.Title = "클립보드에 추가할 오디오 파일을 선택하세요~";
    dlg.Filter = "*.* | *.wav";
    if(dlg.ShowDialog() == DialogResult.OK)
    {
        FileStream fs = File.Open(dlg.FileName, FileMode.Open);
        Clipboard.SetAudio(fs);
        fs.Close();
        txt_info.AppendText("클립보드에 " + dlg.FileName + " 오디오 추가 WrWn");
    }
}

private void btn_getAudio_Click(object sender, EventArgs e)
{
    if (Clipboard.ContainsAudio())
    {
        Stream st = Clipboard.GetAudioStream();
        FileStream fs = File.Create("C:\\\\클립보드오디오.wav");
        for(int i = 0; i < st.Length; i++)
            fs.WriteByte((byte)st.ReadByte());
        fs.Close();
    }
}

private void btn_addText_Click(object sender, EventArgs e)
{
    Clipboard.SetText("클립보드에 문자열을 추가합니다.");
    txt_info.AppendText("클립보드에 문자열 추가...WrWn");
}

private void btn_getText_Click(object sender, EventArgs e)
{
    if (Clipboard.ContainsText())
        MessageBox.Show(Clipboard.GetText());
}
}

```