

본 강의에서 수업자료로 이용되는 저작물은
저작권법 제25조 수업목적 저작물 이용 보상금제도에 의거,
한국복제전송저작권협회와 약정을 체결하고 적법하게 이용하고 있습니다.
약정범위를 초과하는 사용은 저작권법에 저촉될 수 있으므로
수업자료의 재 복제, 대중 공개·공유 및 수업 목적 외의 사용을 금지합니다.

2024. 8. 30.

부천대학교·한국복제전송저작권협회

C#

6주차 3차시

12장 그리기(2)

6주차 학습 내용

* 1차시

- * 5주차 과제 피드백(6주차 2, 3차시 수업 완료 후 피드백 점검)
- * 11장 고급 컨트롤
 - * 리스트 뷰
 - * 프로그래스바
 - * 타이머 컴포넌트

* 2차시

- * 12장 그리기
 - * 그리기 개요
 - * 그리기 관련 자료형

* 3차시

- * 12장 그리기
 - * 도형 그리기
 - * 문자열 그리기
 - * 이미지 그리기

C# 실습



제 12장 그리기



목차

- 그리기 개요
- 그리기 관련 자료형
- 도형 그리기
- 문자열 그리기
- 이미지 그리기



도형 그리기(그리기 메서드)

- DrawXxx() 메소드 계열
 - 선을 그리기 위한 **펜**을 첫 번째 매개변수로 받아 외곽선을 그리는 메소드.
- FillXxx() 메소드
 - 영역을 칠하기 위한 **브러시**를 첫 번째 매개변수로 받아 도형의 내부 영역만을 칠하는 메소드.



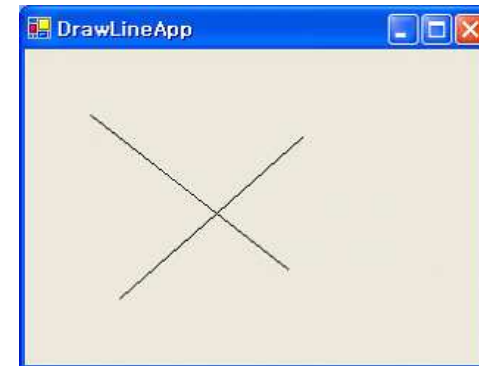
선 그리기 1 – DrawLine() 메소드

■ 형식

```
public void DrawLine(Pen p, Point pt1, Point pt2);
```

■ 예제

```
Pen p = new Pen(Color.Black);  
Point startPoint = new Point(45, 45);  
Point endPoint = new Point(180, 150);  
g.DrawLine(p, startPoint, endPoint);  
g.DrawLine(p, new Point(190, 60), new Point(65, 170));  
p.Dispose();
```





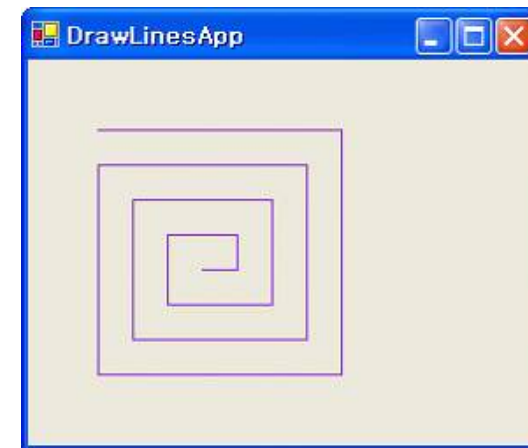
선 그리기 2 - DrawLines() 메소드

■ 형식

```
public void DrawLines(Pen p, Point[] pts);
```

■ 예제

```
Point[] pts = {  
    new Point(40, 40), new Point(180, 40),  
    new Point(180, 180), new Point(40, 180),  
    new Point(40, 60), new Point(160, 60),  
    new Point(160, 160), new Point(60, 160),  
    new Point(60, 80), new Point(140, 80),  
    new Point(140, 140), new Point(80, 140),  
    new Point(80, 100), new Point(120, 100),  
    new Point(120, 120), new Point(100, 120)  
};  
g.DrawLines(new Pen(Color.BlueViolet), pts);
```





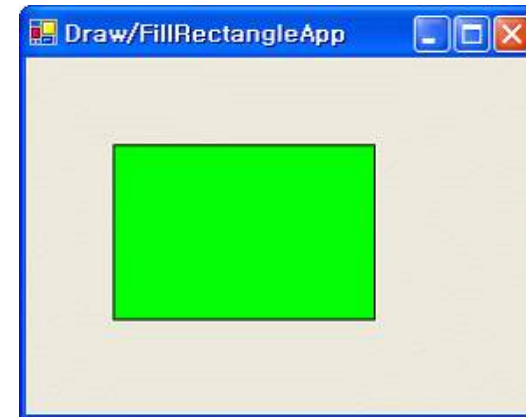
사각형 그리기 1 – DrawRectangle()

■ 형식

```
public void DrawRectangle(Pen p, Rectangle r);  
public void FillRectangle(Brush b, Rectangle r);
```

■ 예제

```
Rectangle r = new Rectangle(50, 50, 150, 100);  
g.FillRectangle(Brushes.Lime, r);  
g.DrawRectangle(new Pen(Color.Black), r);
```





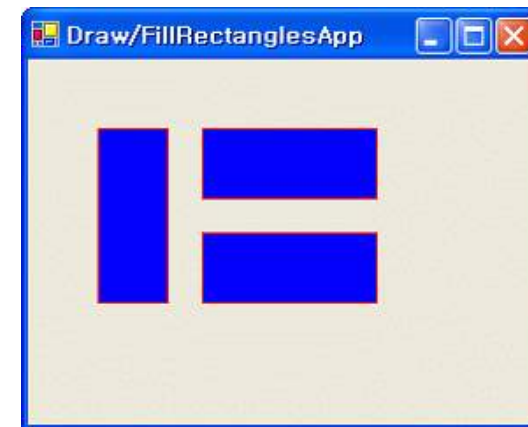
사각형 그리기 2 – DrawRectangles()

■ 형식

```
public void DrawRectangles(Pen p, Rectangle[] rects);  
public void FillRectangles(Brush b, Rectangle[] rects);
```

■ 예제

```
Rectangle[] rects = {  
    new Rectangle(40, 40, 40, 100),  
    new Rectangle(100, 40, 100, 40),  
    new Rectangle(100, 100, 100, 40)  
};  
g.FillRectangles(Brushes.Blue, rects);  
g.DrawRectangles(Pens.Red, rects);
```





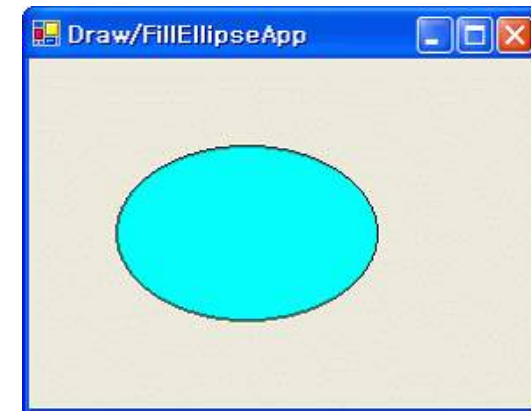
타원 그리기

■ 형식

```
public void DrawEllipse(Pen p, Rectangle r);  
public void FillEllipse(Brush b, Rectangle r);
```

■ 예제

```
Rectangle r = new Rectangle(50, 50, 150, 100);  
g.FillEllipse(Brushes.Cyan, r);  
g.DrawEllipse(Pens.Black, r);
```





호 그리기

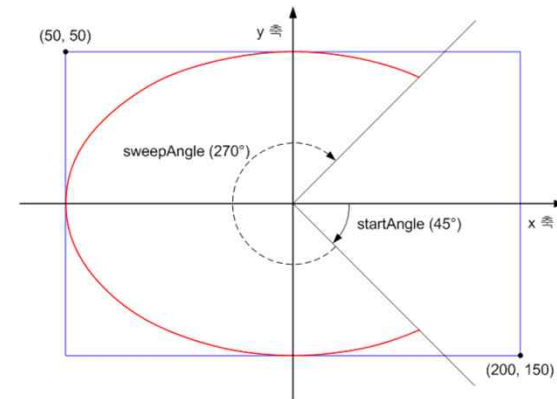
■ 형식

```
public void DrawArc(Pen p, Rectangle r, int startAngle, int sweepAngle);
```

■ 예제

- 기준점이 (50, 50)이고 폭과 높이가 각각 150, 100
- 시작 각도는 X축을 기준으로 45도, 호각이 270도

```
Rectangle r = new Rectangle(50, 50, 150, 100);  
g.DrawArc(Pens.Red, r, 45, 270);
```





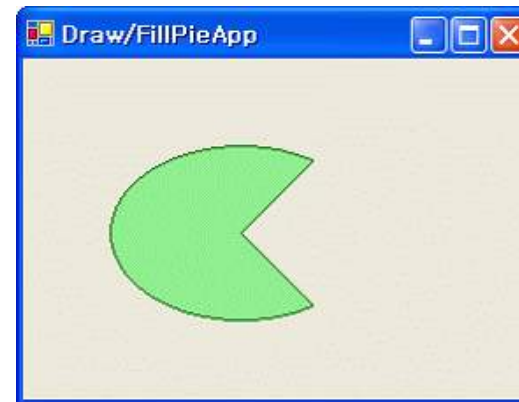
부채꼴 그리기

■ 형식

```
public void DrawPie(Pen p, Rectangle r, int startAngle, int sweepAngle);  
public void FillPie(Brush b, Rectangle r, int startAngle, int sweepAngle);
```

■ 예제

```
Rectangle r = new Rectangle(50, 50, 150, 100);  
g.FillPie(Brushes.LightGreen, r, 45, 270);  
g.DrawPie(Pens.DarkGreen, r, 45, 270);
```





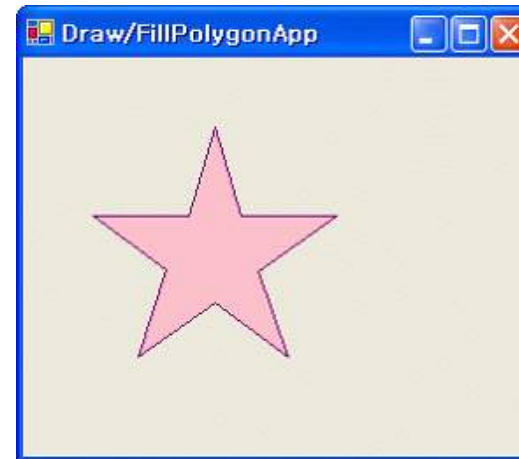
다각형 그리기

■ 형식

```
public void DrawPolygon(Pen p, Point[] pts);  
public void FillPolygon(Brush b, Point[] pts);
```

■ 예제

```
Point[] pts = {  
    new Point(110, 40), new Point(125, 91),  
    new Point(180, 91), new Point(135, 123),  
    new Point(152, 172), new Point(110, 141),  
    new Point(66, 172), new Point(82, 122),  
    new Point(40, 91), new Point(95, 91)  
};  
g.FillPolygon(Brushes.Pink, pts);  
g.DrawPolygon(Pens.Purple, pts);
```





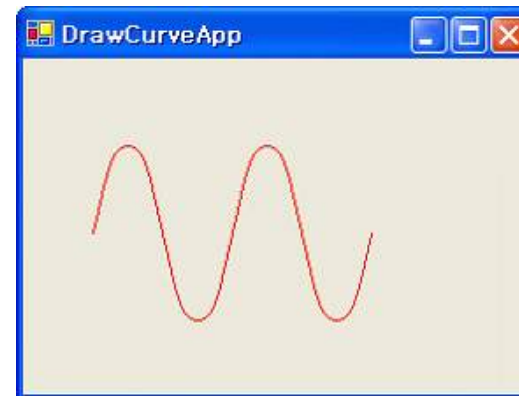
곡선 그리기

■ 형식

```
public void DrawCurve (Pen p, Point[] pts);
```

■ 예제

```
Point[] pts = {  
    new Point(40, 100), new Point(50, 60),  
    new Point(60, 50), new Point(70, 60),  
    new Point(80, 100), new Point(90, 140),  
    new Point(100, 150), new Point(110, 140),  
    new Point(120, 100), new Point(130, 60),  
    new Point(140, 50), new Point(150, 60),  
    new Point(160, 100), new Point(170, 140),  
    new Point(180, 150), new Point(190, 140),  
    new Point(200, 100)  
};  
g.DrawCurve(Pens.Red, pts);
```





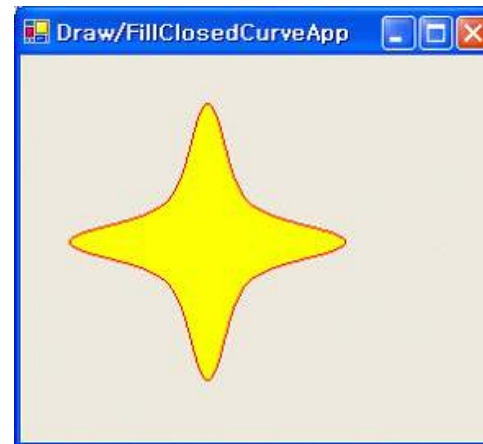
폐곡선 그리기

■ 형식

```
public void DrawClosedCurve(Pen p, Point[] pts);  
public void FillClosedCurve(Brush b, Point[] pts);
```

■ 예제

```
Point[] pts = {  
    new Point(115, 30), new Point(140, 90),  
    new Point(200, 115), new Point(140, 140),  
    new Point(115, 200), new Point(90, 140),  
    new Point(30, 115), new Point(90, 90)  
};  
g.FillClosedCurve(Brushes.Yellow, pts);  
g.DrawClosedCurve(Pens.Red, pts);
```





Font 클래스

- 문자열을 그릴 때 사용하는 글꼴의 모양과 크기를 나타내는 클래스

- 생성자

```
public Font(string name, float size);  
public Font(string name, float size, FontStyle fs);
```

- FontStyle 열거형 상수

- Bold: 굵은(또는 진한) 텍스트.
- Italic: 기울임꼴 텍스트.
- Regular: 일반 텍스트.
- Strikeout: 중간에 줄이 그어진 텍스트.
- Underline: 밑줄이 있는 텍스트.



Font 클래스의 주요 프로퍼티

- Name: 글꼴의 이름.
- Size: 글꼴의 크기.
- Height: 글꼴의 높이.
- Bold: 글꼴 모양이 굵은지 여부.
- Italic: 글꼴 모양이 기울임꼴인지 여부.
- Strikeout: 글꼴 모양이 중간에 줄이 있는지 여부.
- Underline: 글꼴 모양이 밑줄이 있는지 여부.



문자열 그리기

- DrawString() 메소드
 - 문자열을 그리기 위해서 사용되는 메소드
- MeasureString() 메소드
 - 어떤 문자열을 그리는데 필요한 영역의 크기를 계산해 주는 메소드



DrawString() 메소드 [1/3]

- 폼에 문자열을 출력하기 위해서 사용하는 메소드.
- 형식

```
public void DrawString(string s, Font f, Brush b, float x, float y);  
public void DrawString(string s, Font f, Brush b, PointF p);  
public void DrawString(string s, Font f, Brush b, Rectangle r, [StringFormat sf]);
```



DrawString() 메소드 [2/3]

```
public void DrawString(string s, Font f, Brush b, float x, float y);  
public void DrawString(string s, Font f, Brush b, PointF p);  
public void DrawString(string s, Font f, Brush b, Rectangle r, [StringFormat sf]);
```

■ 예제

```
//Graphics g = CreateGraphics();  
//g.Clear(SystemColors.Control);  
Font f = new Font("Tahoma", 15);  
g.DrawString("Hello World!", f, Brushes.Black, 10, 10);  
f.Dispose();
```



```
string s = "This string is long enough to wrap.";  
s += " With a 250px-width rectangle, ";  
s += "it requires six lines to display the string in its entirety.";  
Font f = new Font("Tahoma", 15);  
Rectangle r = new Rectangle(10, 10, 250, 150);  
g.DrawRectangle(Pens.Black, r);  
g.DrawString(s, f, Brushes.Black, r);  
f.Dispose();
```

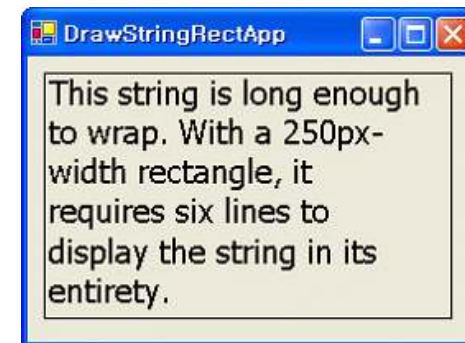




Image 클래스

■ 이미지

■ 래스터 기반 이미지

- 색상을 가진 점(pixel) 단위로 표현
- 사진과 같이 복잡하고 일정한 형태를 가지지 않은 이미지를 표현하기에 적합

■ 벡터 기반 이미지

- 점, 선, 도형과 같은 그래픽 단위로 표현
- 일정한 형태로 구성된 이미지를 표현하기에 적합
- 이미지의 변환 작업을 쉽게 할 수 있음.

■ Image 클래스

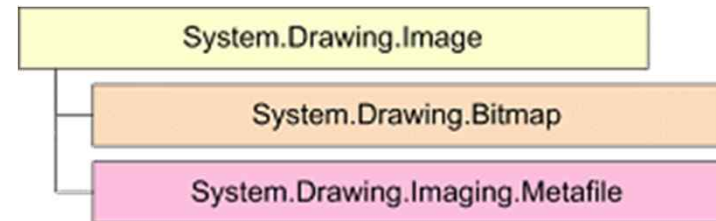
■ 이미지를 나타내는 클래스

■ 파생 클래스

- **Bitmap 클래스**: 래스터 기반 이미지를 표현.
- **Metafile 클래스**: 벡터 기반 이미지를 표현.

■ 주요 프로퍼티

- Size: 이미지 크기.
- Width: 이미지의 폭.
- Height: 이미지의 높이.





Bitmap 클래스

- 래스터 기반 이미지를 나타내는 클래스.
 - 메모리에서만 존재하는 가상의 이미지를 나타낼 때도 사용

- 생성자

```
public Bitmap(string fileName); // 이미지 파일 객체  
public Bitmap(int w, int h);    // 가상 이미지 객체
```

- Bitmap 클래스에서 지원하는 파일 형식
 - BMP: Bit-mapped graphics format의 의미.
 - EXIF: EXchangeable Image File의 약자.
 - GIF: Graphics Interchange Format의 약자.
 - JPEG: Joint Photographic Experts Group의 약자.
 - PNG: Portable Network Graphics의 약자.
 - TIFF: Tag Image File Format의 약자.



Metafile 클래스

- 벡터 기반 이미지를 나타내는 클래스
 - **그리기 연산**들을 수행하면 생성된 그래픽 객체에 그러한 연산이 기록.
 - 그려진 결과가 저장되는 것이 아니라 그려지는 과정을 기록.
- System.Drawing.Imaging 네임스페이스에 포함



이미지 그리기

■ 형식

```
public void DrawImage(Image, int, int);  
public void DrawImage(Image, Point);  
public void DrawImage(Image, int, int, int, int);  
public void DrawImage(Image, Rectangle);  
public void DrawImage(Image, Point[]);  
public void DrawImage(Image, Rectangle, Rectangle, GraphicsUnit);
```

■ 예제

```
Image img = new Bitmap("naver.gif");  
Graphics g = CreateGraphics();  
g.Clear(SystemColors.Control);  
g.DrawImage(img, 0, 0); // g.DrawImage(img, new Point(0, 0));
```



```
Image img = new Bitmap("naver.gif");  
Graphics g = CreateGraphics();  
g.Clear(SystemColors.Control);  
g.DrawImage(img, ClientRectangle);
```



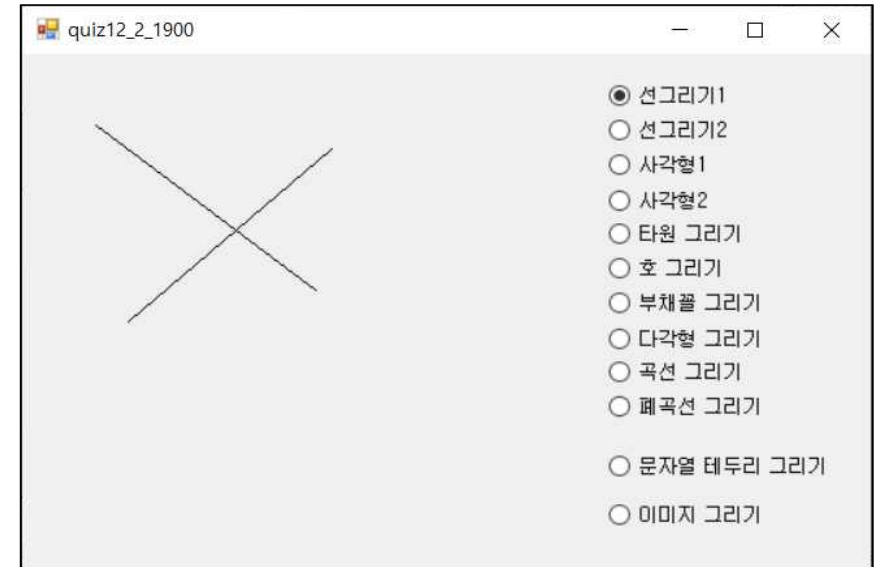


도형 메서드와 이미지 그리기

quiz 12.2 [w6_quiz12_2_xxxx] : 도형 메서드와 이미지 그리기

```
private void Form1_Click(object sender, System.EventArgs e)
{
    if (radioButton1.Checked == true)
    {
        Graphics g = CreateGraphics();
        g.Clear(SystemColors.Control);
        Pen p = new Pen(Color.Black);
        Point startPoint = new Point(45, 45);
        Point endPoint = new Point(180, 150);
        g.DrawLine(p, startPoint, endPoint);
        g.DrawLine(p, new Point(190, 60), new Point(65, 170));
        p.Dispose();
    }
    ...

    if (radioButton3.Checked == true)
    {
        Graphics g = CreateGraphics();
        g.Clear(SystemColors.Control);
        Rectangle r = new Rectangle(50, 50, 150, 100);
        g.FillRectangle(Brushes.Lime, r);
        g.DrawRectangle(new Pen(Color.Black), r);
    }
    ...
}
```



6주차 학습 내용 정리

* 1차시

- * 5주차 과제 피드백(6주차 2, 3차시 수업 완료 후 피드백 점검)
- * 11장 고급 컨트롤
 - * 리스트 뷰 => quiz 11.1 [w6_quiz11_1_xxxx] : 리스트 뷰 퀴즈
 - * 프로그래스바
 - * 타이머 컴포넌트

* 2차시

- * 12장 그리기
 - * 그리기 개요 => 예제 12.1.2 [w6_ex12_1_2_xxxx] : 그리기 객체 방법1,방법2
 - * 그리기 관련 자료형 => quiz 12.1 [w6_quiz12_1_xxxx] : 그리기 관련 자료형

* 3차시

- * 12장 그리기
 - * 도형 그리기
 - * 문자열 그리기
 - * 이미지 그리기 => quiz 12.2 [w6_quiz12_2_xxxx] : 도형 그리기와 이미지 그리기