

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 2차시

12장 그리기(1)

# 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



## 그리기 개요

### ■ Graphics(System.Drawing.Graphics) 클래스

- System.Drawing 네임스페이스에 포함
- 선, 사각형, 타원 등과 같은 도형을 그리는데 필요한 기본적인 메소드들이 존재

### ■ 그래픽 객체

- 도형을 그리기 위해 필요한 Graphics 클래스의 객체
- 그리기판이 되는 대상
- 객체를 얻거나 생성하는 방법
  - ① Paint 이벤트 처리기의 매개변수
  - ② Control 클래스의 CreateGraphics() 메소드
  - ③ Graphics.FromImage() 메소드

### 차이점

- Paint 이벤트의 Graphics
  - 화면이 다시 그려질 필요가 있을 때 즉 Paint 이벤트가 발생할 때마다 매번 그려진다.
- CreateGraphics()
  - 그리는 작업을 할 때에만 그려진다.
  - 화면이 다시 그려지면 CreateGraphics()를 이용해서 그린 내용은 사라진다.



## Paint 이벤트의 매개변수

- Paint 이벤트의 매개변수
  - Paint 이벤트: 폼을 다시 그려야 할 때 발생하는 이벤트.
  - 처리기의 두 번째 매개변수에 **그래픽 객체**가 들어 있음.
  - 여기에 그리기 작업을 하면 폼에 그려짐.

- Paint 이벤트 처리기의 메소드 형식

```
private void Form1_Paint(object sender, System.Windows.Forms.PaintEventArgs e)
{
    Graphics g = e.Graphics;
    // 그래픽 객체를 이용한 그리기 작업
    // ...
}
```

- PaintEventArgs 클래스
  - **Graphics** 프로퍼티: 그리기에 필요한 Graphics 클래스의 객체
  - **ClipRectangle** 프로퍼티는 새로 그려야 하는 영역



## CreateGraphics() 메소드

### ■ Control 클래스의 CreateGraphics() 메소드

- Paint 이벤트 처리기가 아닌 다른 곳에서 그리기를 하고자 할 때 사용.
- 메소드를 사용하여 그래픽 객체를 생성
- Control 클래스의 메소드이기 때문에 파생된 모든 클래스에서 그래픽 객체를 만들 수 있음.

### ■ 사용 방법

```
private void DrawPrivateObject() {  
    // 단계 1: 그래픽 객체를 생성한다.  
    Graphics g = CreateGraphics();  
    // ...  
    // 단계 2: 그래픽 객체를 이용하여 그리기를 한다.  
    // ...  
}
```





## Graphics.FromImage() 메소드

### ■ Graphics.FromImage() 메소드

- 이미지에서 만들어진 그래픽 객체이기 때문에 그리기 작업의 결과가 화면에 곧바로 나타나지 않음.
- 결과를 화면에 표시하려면 이미지 객체를 화면에 출력해야 함.
- 그리기 도중에 발생할 수 있는 **깜박거림을 제거하기 위해 사용.**

### ■ 사용 방법

```
private void DrawOffScreenImage() {  
    // 단계 1: 이미지 객체로부터 그래픽 객체를 생성한다.  
    Image img = new Bitmap(w, h);  
    Graphics bg = Graphics.FromImage(img);  
    // ...  
    // 단계 2: bg에 그리기 작업을 한다.  
    // ...  
    // 단계 3: 결과 이미지를 표시하기 위한 그래픽 객체를 생성한다.  
    Graphics fg = CreateGraphics();  
    // ...  
    // 단계 4: 이미지를 화면에 출력한다.  
    fg.DrawImage(img, 0, 0);  
}
```

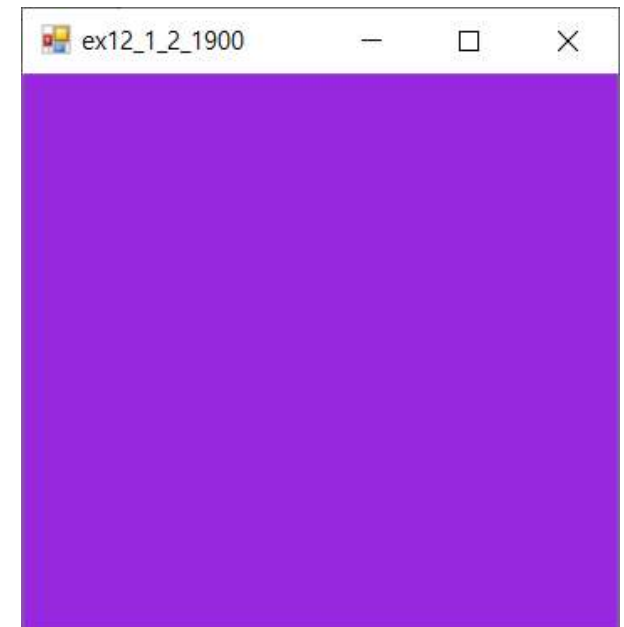


## Paint 이벤트의 매개변수

예제 12.1.2 [w6\_ex12\_1\_2\_xxxx] : 그리기 객체 방법1, 방법2

```
private void Form1_Paint(object sender, System.Windows.Forms.PaintEventArgs e)
{
    Graphics g = e.Graphics;
    Random r = new Random();
    Color c = Color.FromArgb(r.Next(0, 255), r.Next(0, 255), r.Next(0, 255));
    g.FillRectangle(new SolidBrush(c), e.ClipRectangle);
}

private void Form1_Click(object sender, EventArgs e)
{
    Graphics g = CreateGraphics();
    Random r = new Random();
    Color c = Color.FromArgb(r.Next(0, 255), r.Next(0, 255), r.Next(0, 255));
    g.FillRectangle(new SolidBrush(c), ClientRectangle);
}
```





## Graphics.FromImage() 메소드

[예제 12\_3 – FromImageApp.cs]

```
private void Form1_Paint(object sender, System.Windows.Forms.PaintEventArgs e)
{
    Size s = ClientRectangle.Size;
    Image img = new Bitmap(s.Width, s.Height); //1

    Graphics bg = Graphics.FromImage(img);
    Random r = new Random();
    Color c = Color.FromArgb(r.Next(0, 255), r.Next(0, 255), r.Next(0, 255));
    bg.FillRectangle(new SolidBrush(c), ClientRectangle); //2

    Graphics fg = e.Graphics;
    fg.DrawImage(img, 0, 0); //3
}
```



## 그리기 관련 자료형

- 색상을 나타내는 Color 구조체
- 좌표와 관련있는 Point 구조체와 Size 구조체, Rectangle 구조체
- 도형을 그리거나 채우는데 사용되는 Pen 클래스와 Brush 클래스
- 글꼴을 나타내는 Font 클래스
- 이미지를 나타내는 Image 클래스



## Color 구조체 [1/3]

### ■ 설명

- 색을 RGB(Red, Green, Blue) 형식으로 나타낸 구조체
- RGB마다 각각 0~255 값을 가짐.( 0: 완전 투명, 255: 불투명)
- 투명도 값인 A(Alpha)을 가질 수도 있음.( 0: 완전 투명, 255: 불투명)

### ■ 메서드

```
Color c = Color.FromArgb(R, G, B);  
Color c = Color.FromArgb(A, R, G, B);  
Color c = Color.FromKnownColor(KnownColor.Member);  
Color c = Color.FromName("ColorName");
```



## Color 구조체 [2/3]

### ■ KnownColor 열거형

- 시스템 색상: 윈도우의 구성 요소의 색을 나타내는 색상.

ActiveBorder	ActiveCaption	ActiveCaptionText	AppWorkspace	Control
ControlDark	ControlDarkDark	ControlLight	ControlLightLight	ControlText
Desktop	GrayText	Highlight	HighlightText	HotTrack
InactiveBorder	InactiveCaption	InactiveCaptionText	Info	InfoText
Menu	MenuText	ScrollBar	Window	WindowFrame
WindowText				

### ■ 그 외 색상

AliceBlue	AlightWhite	Aqua	Aquamarine	Azure
Beige	Bisque	Black	BlanchedAlmond	Blue
BlueViolet	Brown	BurlyWood	CadetBlue	Chartreuse
Chocolate	Coral	CorianderBlue	Cornsilk	Crimson
Cyan	DarkBlue	DarkCyan	DarkGoldenrod	DarkGray
DarkGreen	DarkKhaki	DarkMagenta	DarkOliveGreen	DarkOrange
DarkOrchid	DarkRed	DarkSalmon	DarkSeaGreen	DarkSlateBlue
DarkSlateGray	DarkTurquoise	DarkViolet	DeepPink	DeepSkyBlue
DimGray	DodgerBlue	Firebrick	FloralWhite	ForestGreen
Fuchsia	Gainsboro	GhostWhite	Gold	Goldenrod
Gray	Green	GreenYellow	Honeydew	HotPink
IndianRed	Indigo	Indigo	Khaki	Lavender
LavenderBlush	LawnGreen	LemonChiffon	LightBlue	LightCoral
LightCyan	LightGoldenrodYellow	LightGray	LightGreen	LightPink
LightSalmon	LightSeaGreen	LightSkyBlue	LightSlateGray	LightSteelBlue
LightYellow	Lime	LimeGreen	Linen	Magenta
Maroon	MediumAquamarine	MediumBlue	MediumOrchid	MediumPurple
MediumSeaGreen	MediumSlateBlue	MediumSpringGreen	MediumTurquoise	MediumVioletRed
MistyRose	MintCream	MistyRose	Moccasin	NavyWhite
New	OldLace	Olive	OliveDrab	Orange
OrangeRed	Orchid	PaleGoldenrod	PaleGreen	PaleTurquoise
PaleVioletRed	PapayaWhip	PeachPuff	Pink	Pink
Plum	PowderBlue	Purple	RosyBrown	RosyBrown
PowderBlue	SaddleBrown	Salmon	SandyBrown	SeaGreen
SeaShell	Sienna	Silver	SlateBlue	SlateBlue
SlateGray	Sliver	SpringGreen	SteelBlue	Tan
Teal	Thistle	Tomato	Transparent	Turquoise
Violet	Wheat	White	WhiteSmoke	Yellow
YellowGreen				



## Color 구조체 [3/3]

### ■ 사용법

```
// ① 정적 프로퍼티를 이용하여 객체 생성.  
Color c = Color.White;  
// ② 미리 정의된 열거형 상수를 이용하여 객체 생성.  
Color c = Color.FromKnownColor(KnownColor.White);  
// ③ 미리 정의된 색상 이름을 이용하여 객체 생성.  
Color c = Color.FromName("White");  
// ④ RGB 값을 이용하여 객체 생성.  
Color c = Color.FromArgb(255, 255, 255);
```

### ■ 프로퍼티

- R: 빨간색(red)의 값 (0~255).
- G: 녹색(green)의 값 (0~255).
- B: 파란색(blue)의 값 (0~255).
- A: 투명도(Alpha)의 값 (0~255).( 0: 완전 투명, 255: 불투명)
- IsEmpty: 빈 구조체 여부.



## Point 구조체

### ■ 설명

- 평면상의 **한 점**을 표시하기 위한 자료형
- x 좌표와 y 좌표의 형식으로 위치를 표시
- 그리기뿐만 아니라 폼이나 컨트롤에서 위치를 지정하는데도 사용

### ■ 생성자

```
Point pt = new Point();      // (0, 0)을 나타냄.  
Point pt = new Point(x, y);  // (x, y) 좌표를 나타냄.
```

### ■ 주요 프로퍼티

- X: X 좌표 또는 수평 위치.
- Y: Y 좌표 또는 수직 위치.
- IsEmpty: 빈 구조체 여부.

### ■ 관련 구조체

- PointF 구조체: 값을 실수로 표현.

### ■ PointF -> Point 변환 메소드

```
public static Point Ceiling(PointF value);    // 올림  
public static Point Round(PointF value);      // 반올림  
public static Point Truncate(PointF value);   // 내림
```





## Size 구조체

### ■ 설명

- 사각형 모양을 갖는 영역의 크기를 나타내기 위해서 사용되는 구조체
- **영역의 크기는 폭(width)과 높이(height)로 나타냄.**
- 그리기뿐만 아니라 폼이나 컨트롤에서 사각형 모양으로 표시되는 영역의 크기를 나타낼 때도 사용

### ■ 생성자

```
Size area = new Size();  
Size area = new Size(Width, Height);
```

### ■ 주요 프로퍼티

- Width: 가로 폭.
- Height: 세로 높이.
- IsEmpty: 빈 구조체 여부.

### ■ 관련 구조체

- **SizeF 구조체**: 값을 실수로 표현.

### ■ SizeF -> Size 변환 메소드

```
public static Size Ceiling(SizeF value);    // 올림  
public static Size Round(SizeF value);    // 반올림  
public static Size Truncate(SizeF value);  // 내림
```



## Rectangle 구조체 [1/2]

### ■ 설명

- 사각형 모양을 갖는 영역의 위치와 크기를 나타내기 위해서 사용
- Point 구조체와 Size 구조체의 개념을 모두 가지고 있는 구조체
- 사각형이나 또는 폼과 컨트롤의 위치와 크기를 동시에 나타내기 위해서 사용

### ■ 생성자

```
Rectangle r = new Rectangle();  
Rectangle r = new Rectangle(Point, Size);  
Rectangle r = new Rectangle(X, Y, Width, Height);
```

### ■ 주요 프로퍼티

- X: 사각 영역의 왼쪽 상단의 X 좌표.
- Y: 사각 영역의 왼쪽 상단의 Y 좌표.
- Width: 사각 영역의 가로 폭.
- Height: 사각 영역의 세로 높이.
- Left/Top: 사각 영역의 왼쪽 X 좌표/위쪽의 Y 좌표
- Right/Bottom: 사각 영역의 오른쪽 X 좌표/아래쪽의 Y 좌표
- Location: 사각 영역의 시작 위치 (Point(X, Y))
- Size: 사각 영역의 크기 (Size(Width, Height))
- IsEmpty: 빈 구조체 여부.



## Rectangle 구조체 [2/2]

### ■ 관련 구조체

- RectangleF 구조체: 값을 실수로 표현.

### ■ RectangleF -> Rectangle 변환 메소드

```
public static Rectangle Ceiling(RectangleF value);    // 올림  
public static Rectangle Round(RectangleF value);    // 반올림  
public static Rectangle Truncate(RectangleF value);  // 내림
```

### ■ 주요 메소드

- **Contains()**: 점이나 영역의 포함 여부를 알려준다.
- **Inflate()**: 지정된 크기만큼 영역을 확장한다.
- **Intersect()**: 두 영역의 교차 영역을 구한다.
- **IntersectsWith()**: 두 영역의 교차 여부를 알려준다.
- **Offset()**: 영역을 지정된 크기만큼 이동한다.
- **Union()**: 두 영역을 포함한 최소 영역을 구한다.



## Pen 클래스

### ■ 설명

- 직선이나 도형을 그릴 때 색상이나 굵기 등 선에 관한 정보를 가지는 펜을 위한 클래스
- Graphics의 메서드 중 Draw로 시작하는 메서드에서 공통적으로 펜을 인수로 요구

### ■ 생성자=> 색상, 무늬(선이 굵을 때) 선택

```
Pen p = new Pen(Brush b);  
Pen p = new Pen(Color c);  
Pen p = new Pen(Brush b, float width);  
Pen p = new Pen(Color c, float width);
```

### ■ Pens 클래스

- 미리 정의된 펜들을 가진 프로퍼티로 가진 클래스
- 프로퍼티 이름은 KnownColor 열거형 상수(시스템 색상 이름을 제외)의 이름과 동일
- 펜의 두께는 1.0이고 이름과 같은 색상을 가진 펜을 정의.



## Pen 클래스 – 주요 프로퍼티

- **Width** 프로퍼티
  - 펜의 굵기를 나타내는 프로퍼티
    - `public float Width { get; set; }`
- **DashStyle** 프로퍼티
  - 점선이나 파선처럼 선의 모양을 나타내는 프로퍼티
    - `public DashStyle DashStyle { get; set; }`
  - DashStyle 열거형
    - System.Drawing.Drawing2D 네임스페이스에 포함.
- **StartCap/EndCap** 프로퍼티
  - 선 시작과 끝의 모양을 나타내는 프로퍼티
    - `public LineCap StartCap { get; set; }`
    - `public LineCap EndCap { get; set; }`
  - LineCap 열거형
    - System.Drawing.Drawing2D 네임스페이스에 포함.
- **DashCap** 프로퍼티
  - 점선의 경우 선中间的 대시 모양 지정



## Brush 클래스

### ■ 설명

- 면을 채울 때 사용하며 Graphics클래스의 Fill로 시작하는 메서드에서 사용
- 도형과 같이 일정한 면적을 차지하는 부분을 색이나 모양 패턴, 혹은 그림으로 채울 때 사용하는 객체가 브러시
- 브러시를 위한 클래스

### ■ Brush 클래스의 파생 클래스

- SolidBrush: 단색으로 된 브러시.
- TextureBrush: 이미지(BMP, JPEG 등)로 된 브러시.
- HatchBrush: 모양 패턴을 사용하는 브러시.
- LinearGradientBrush: 선형 색전환(gradient)을 사용하는 브러시.
- PathGradientBrush: 지정된 경로에 따른 색전환을 사용하는 브러시.



## SolidBrush 클래스

- 단색으로 된 브러시 객체를 위한 클래스

- 생성자

```
SolidBrush b = new SolidBrush(Color c);
```

- 예제

```
SolidBrush b = new SolidBrush(Color.Lime);  
g.FillRectangle(b, ClientRectangle);  
b.Dispose();
```





# TextureBrush 클래스

- 이미지로 된 브러시 객체를 위한 클래스

- 생성자

```
TextureBrush b = new TextureBrush(Image img);
```

- 예제

```
Image img = new Bitmap("plac.jpg");    // 이미지 객체 생성
TextureBrush b = new TextureBrush(img); // 이미지로 된 브러시 객체 생성
g.FillRectangle(b, ClientRectangle);   // 사용자 영역을 이미지로 채움
img.Dispose();                          // 이미지 객체를 해제
b.Dispose();                            // 브러시 객체를 해제
```





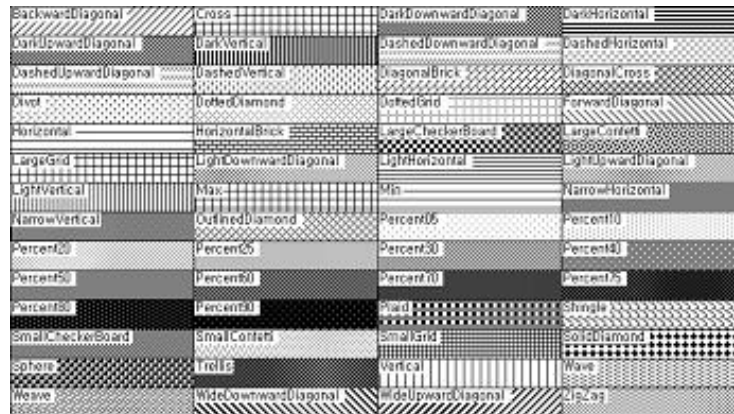


# HatchBrush 클래스

- 모양 패턴으로 된 브러시 객체를 위한 클래스
- `System.Drawing.Drawing2D` 네임스페이스에 포함
- 생성자

```
HatchBrush b = new HatchBrush(HatchStyle hs, Color c);
```

- HatchStyle 열거형





## 그리기 관련 자료형

quiz 12.1 [w6\_quiz12\_1\_xxxx] : 그리기 관련 자료형

```
private void Form1_Click(object sender, EventArgs e)
{
    Graphics g = CreateGraphics();
    Random r = new Random();

    Color c = Color.FromArgb(r.Next(0, 255), r.Next(0, 255), r.Next(0, 255));

    Point p = new Point(10, 20);
    Size s = new Size(80, 40);
    Rectangle rect = new Rectangle(p, s);

    Pen pe = new Pen(Color.Red);
    pe.Width = 10;

    g.DrawEllipse(pe, rect);
    //g.DrawEllipse(pe, , );

    g.FillEllipse(new SolidBrush(c), rect);

    Rectangle rect1 = new Rectangle(100, 20, 80, 40);
    g.FillRectangle(new SolidBrush(c), rect1)
}
```

