

19강. 이미지



목차

- 이미지
- 비트맵
- 더블 버퍼링
- 리소스 이미지

이미지

- GDI+에서 다룰 수 있는 이미지 형식
bmp, jpg, png, gif, tiff, exif 등
- 이미지 다루는 클래스

Image : 추상 기본 클래스 (abstract base class)

Bitmap or MetaFile의 상위 클래스

System.Object

System.MarshalByRefObject

System.Drawing.Image

System.Drawing.Bitmap

System.Drawing.Imaging.Metafile

이미지

- 이미지 로딩 메서드

```
public static Image FromFile (string filename);
```

- 이미지 출력 메서드 (Graphics)

```
public void DrawImage(  
    Image image,  
    int x,  
    int y  
)
```

code

비트맵

- Bitmap 클래스
 - Image를 상속한 클래스

`System.Object`

`System.MarshalByRefObject`

`System.Drawing.Image`

`System.Drawing.Bitmap`

```
public sealed class Bitmap : Image
```

- 기능적으로 Image < Bitmap
- 다양한 형식으로 저장 가능
bmp, emf, exif, gif, icon, jpeg, png, tiff, wmf 등

비트맵

- 비트맵을 이용한 이미지 로딩

- public Bitmap(string filename)

BMP, GIF, EXIF, JPG, PNG, TIFF 로딩 가능

- Bitmap.SetPixel(), Bitmap.GetPixel()

로딩된 이미지의 픽셀 정보

public Color GetPixel(int x, int y)

public void SetPixel(int x, int y, Color color)

비트맵

- Bitmap 출력

Graphics.drawImage()

- 로딩한 이미지와 폼의 크기가 같도록 출력

code

임의의 크기의 비트맵 생성 및 클리어

- 임의의 비트맵 생성

```
public Bitmap( int width, int height )
```

- Graphics 객체로 비트맵 참조

```
public static Graphics FromImage( Image image )
```

- Graphics를 사용하여 비트맵 지우기
Graphics.Clear()

- 400x300 크기의 비트맵을 로딩하고
"C# Programming" 출력 X 10,
배경색은 Yellow 로 출력
사각형 출력

code

이미지에 출력

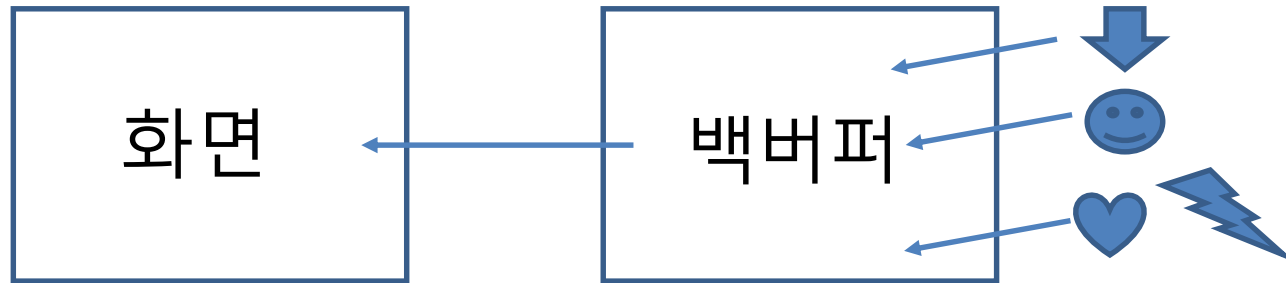


더블 버퍼링

double buffering

더블 버퍼링

- 역할
이미지 출력의 깜빡임을 줄이기 위한 처리



- 용어
백버퍼
오프스크린

더블 버퍼링 객체 구조

- 더블 버퍼링 관여 객체
 - BufferedGraphicsManager class
 - BufferedGraphicsContext class
 - BufferedGraphics class
- 더블 버퍼링 객체 구조

BufferedGraphicsManager

참조

BufferedGraphicsContext

생성

BufferedGraphics : 실체

더블 버퍼링

- 폼의 DoubleBuffered 속성 true로 설정
- DoubleBuffered 속성 코드로 설정하는 방법
`SetStyle(ControlStyles.OptimizedDoubleBuffer, true);`
- 더블 버퍼링에 출력하는 객체와 메서드
`BufferedGraphics.Graphics`
`BufferedGraphics.Graphics.DrawImage()` 그리기
`BufferedGraphics.Clear()` : 더블버퍼 지우기
- 더블 버퍼링 객체 생성
한 번만 생성

code

더블 버퍼링

- 화면에 출력
BufferedGraphics.Render()
`public void Render(Graphics target)`
- 더블버퍼링을 이용하여 이미지 출력
배경은 Yellow, 이미지 100번 출력

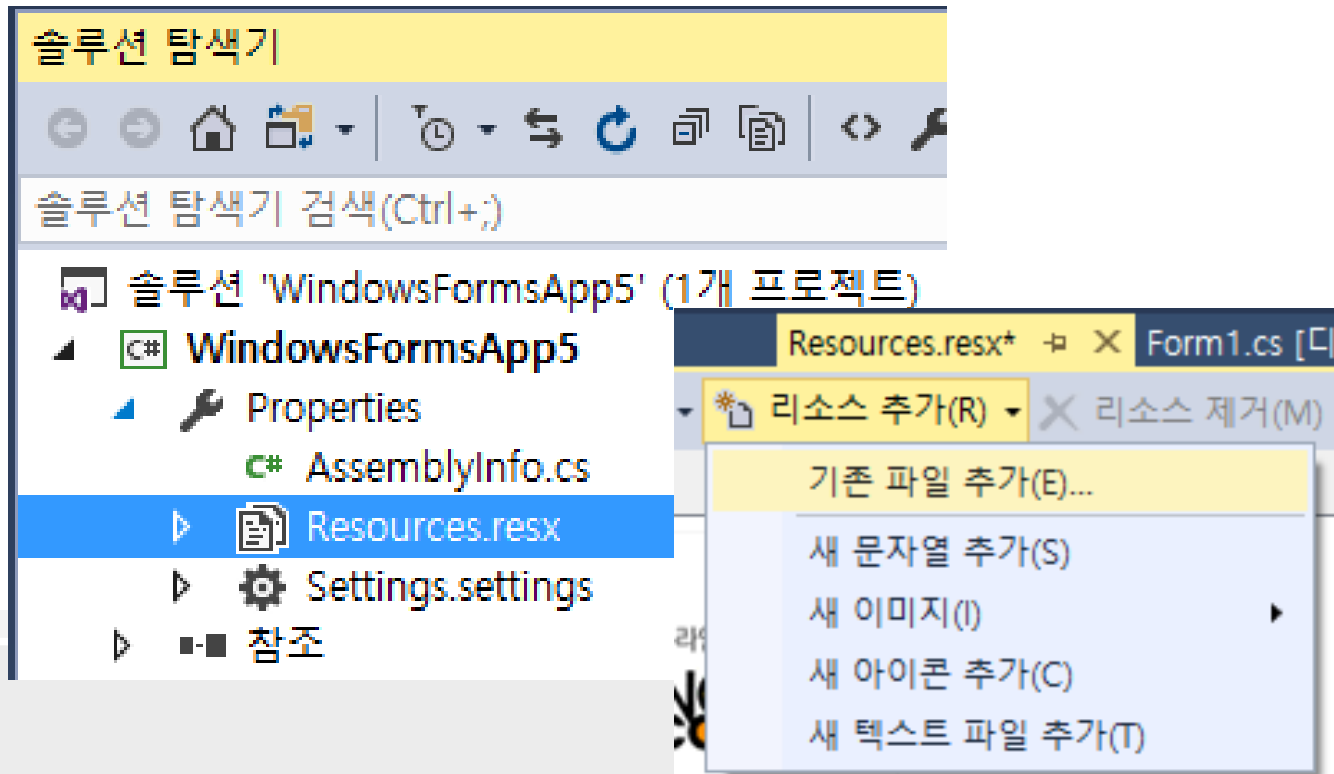
code



리소스 이미지 다루기

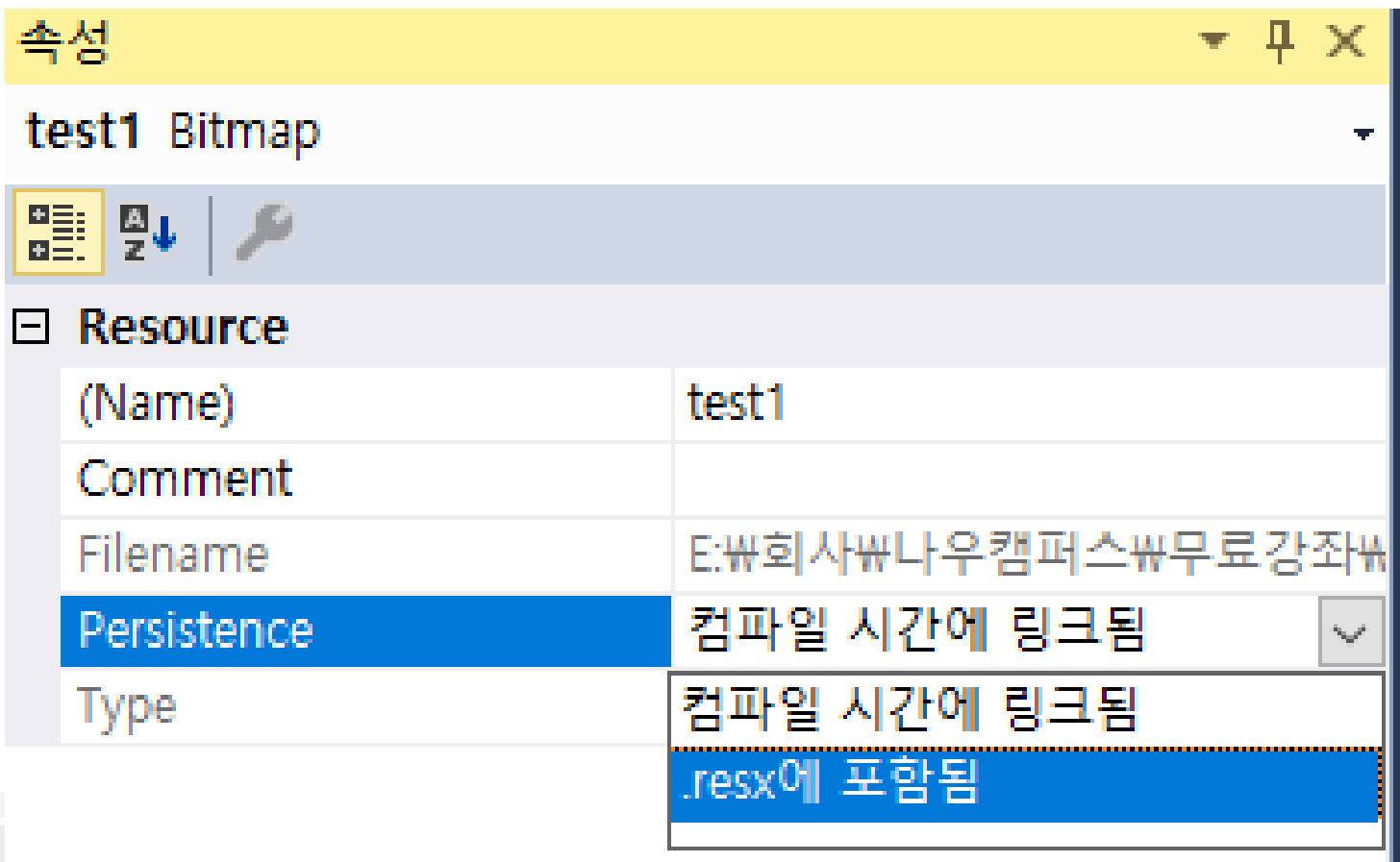
리소스 이미지

- 리소스 이미지는 실행파일에 첨가됨
➔ 실행파일의 크기가 커진다.
- 리소스 디자이너 실행
Resources.resx 더블 클릭



리소스 이미지

- 실행 파일에 포함 속성 설정
 - 이미지 Persistence 속성 변경



리소스 이미지

- 리소스 이미지 Bitmap으로 참조

Bitmap bitmap = Properties.Resource.이미지명

- 출력

Graphics.DrawImage()

code