

Wykład 11

Grafika



Grafika GDI+

GDI+ - Graphics Device Interface– jeden z trzech podstawowych komponentów (razem z jądrem i API Windows) interfejsu użytkownika w Microsoft Windows. GDI+ odpowiedzialne jest za przedstawianie obiektów graficznych i przesyłanie ich do urządzeń wyjściowych, takich jak monitory i drukarki.

Graphics - obiekt reprezentujący GDI+, używany do tworzenia obrazów graficznych.

Aby uzyskać możliwość rysowania linii i kształtów, renderowania tekstu, lub wyświetlania obrazów i manipulowania nimi za należy utworzyć obiekt Graphics.

Grafika

Klasy przestrzeni nazw System.Drawing używane w rysowaniu na formie

Klasa	Opis
Graphics	Klasa, z pomocą której możemy rysować i nie tylko poprzez GDI+
Bitmap	bitmapa
Brush	Pędzel – rodzaj kolor wypełnienia
Pen	Ołówek - kontur
BufferedGraphics	Buforowane grafiki,
Font	Czcionka
Image	Abstrakcyjna podstawa dla pochodnych typów obrazowych np. bitmapy

Grafika

Podstawowe metody klasy Graphics:

Metoda	Opis
DrawArc	Łuk
DrawBezier	Krzywa Beziera
DrawCurve	Krzywa
DrawEllipse	Elipsa
DrawIcon	Ikona
DrawImage	Obraz
DrawLine	Linia
DrawPath	Ścieżka
DrawPolygon	Wielokąt
DrawRectangle	Prostokąt
DrawString	Tekst
DrawPie	Wycinek koła

Niektóre z tych metod mają odpowiedniki zaczynające się od **Fill(..)** i służące do rysowania figur wypełnionych.

Metoda **Clear(Color)**, wypełnia cały obszar wskazanym kolorem



Grafika

Struktury (zmienne złożone) przydatne w operowaniu grafiką

Struktura	Opis
Color	Reprezentuje kolor (w postaci ARGB). Zawiera kolory predefiniowane
Point	Punkt
Rectangle	Prostokąt
Size	Wielkość (np. długość czy wysokość)



Rysowanie bezpośrednio na formie – klasa Graphics

Klasa Graphics

Pierwszym krokiem jest uzyskanie obiektu klasy Graphics elementu na którym będziemy rysować.

```
public partial class Form1 : Form
{
    Graphics g;

    public Form1()
    {
        InitializeComponent();
        g = pictureBox1.CreateGraphics();
    }
}
```

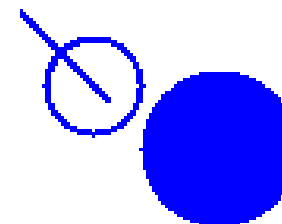
Przykład dla rysowania bezpośrednio po komponencie Form I

Klasa Graphics

Przykład rysowania:

```
private void button2_Click(object sender, EventArgs e)
{
    Color k = Color.Brown; //Ustawienie koloru
    Brush b = Brushes.Blue; //Ustawienie pędzla (wypełnienia)
    Pen p = new Pen(b, 2); //Ustawienie piska (konturu)

    g.DrawEllipse(p, new Rectangle(10, 10, 30, 30));
    //Rysowanie elipsy
    g.DrawLine(p, new Point(0, 0), new Point(30, 30));
    //Rysowanie linii
    g.FillEllipse(b, new Rectangle(40, 20, 50, 50));
}
```



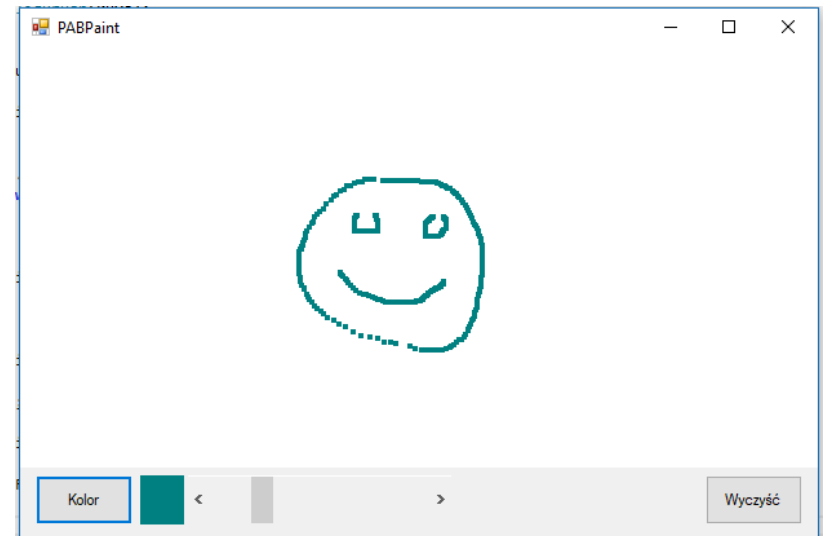


Rysowanie na bitmapie



Przykład PAINT:

Program umożliwia rysowanie za pomocą myszy – Można regulować kolor i grubość pędzla.



Przykład – PAINT

Przygotowanie obiektu klasy Graphics, pisaka i wyczyszczenie ekranu

```
namespace Paint
{
    public partial class Form1 : Form
    {
        Color kTla = Color.White;
        Color kRys = Color.Green;
        Graphics g;
        Brush b;
        bool md = false;

        public Form1()
        {
            InitializeComponent();
            g = pictureBox1.CreateGraphics();
            g.Clear(kTla);
            panel2.BackColor = kRys;
            b = new SolidBrush(kRys);
        }
    }
}
```



Przykład – PAINT

Wybór koloru pędzla

```
private void button1_Click(object sender, EventArgs e)
{
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        kRys = colorDialog1.Color;
        panel2.BackColor = kRys;
        b = new SolidBrush(kRys);
    }
}
```

Przykład – PAINT

Rysowanie - po naciśnięciu przycisku myszy (zmienna `md = true`) rysujemy, po puszczaniu (zmienna `md = false`) przestajemy rysować.

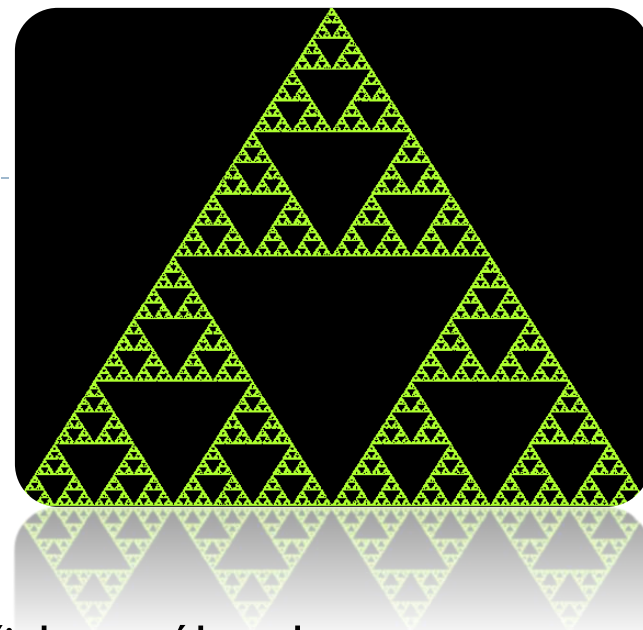
```
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    md = true;
}
private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    md = false;
}
private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
{
    if (md) g.FillEllipse(b, e.X, e.Y, hScrollBar1.Value, hScrollBar1.Value);
}
```

Pozycję kursora pobieramy ze zdarzenia **e**.

Rysujemy koło, którego średnica regulowana jest za pomocą suwaka



Przykład:



Dywan Sierpińskiego:

1. Wybieramy 3 punkty tworzące trójkąt (ich współrzędne zapisujemy w tablicach).
2. Ustawiamy punkt (x,y) - na środku ekranu.
3. Losujemy jeden z trzech wierzchołków.
4. Rysujemy punkt (pixel) w połowie odcinka pomiędzy punktem (x,y) a wierzchołkiem. Staje się on nowym punktem (x,y) .
5. Punkty 3 i 4 powtarzamy np. 100 tys razy.

Przykład – dywan Sierpińskiego

Rozpoczynamy od przygotowania zmiennej typu Bitmap – na której będziemy rysować.

```
namespace GrafikaBitmapa
{
    public partial class Form1 : Form
    {
        private Bitmap bufor = null;
        Random random = new Random();

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

W metodzie zdarzenia Paint obiektu (jego rysowanie) umieszczamy kod programu.

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    //Pobieramy obiekt Graphics, który jest tłem okna
    Graphics g = e.Graphics;
    //Tworzymy bitmapę o rozmiarach okna (ClientSize)
    bufor = new Bitmap(this.ClientSize.Width, this.ClientSize.Height);
    //wypełniamy bitmapę kolorem czarnym - punkt po punkcie
    for (int x = 0; x < this.ClientSize.Width; x++)
        for (int y = 0; y < this.ClientSize.Height; y++)
            bufor.SetPixel(x, y, Color.Black);
    //Tablice zawierają współrzędne wierzchołków trójkąta
    int[] punktX = new int[] { 0, this.ClientSize.Width / 2, this.ClientSize.Width - 1 };
    int[] punktY = new int[] { this.ClientSize.Height - 1, 0, this.ClientSize.Height - 1 };
    int los;
    //Ustawiamy punkt startowy
    int px = this.ClientSize.Width / 2;
    int py = this.ClientSize.Height / 2;
    // w pętli losujemy jeden z wierzchołków i znajdujemy nowy punkt startowy
    // - w połowie drogi pomiędzy poprzednim punktem (x,y) a wylosowanym wierzchołkiem
    for (long i = 0; i < 150000; i++)
    {
        los = random.Next(3);
        px = (px + punktX[los]) / 2;
        py = (py + punktY[los]) / 2;
        bufor.SetPixel(px, py, Color.GreenYellow);
    }
    //wyświetlenie bitmapy jako tła okna
    g.DrawImage(bufor, 0, 0);
}
```

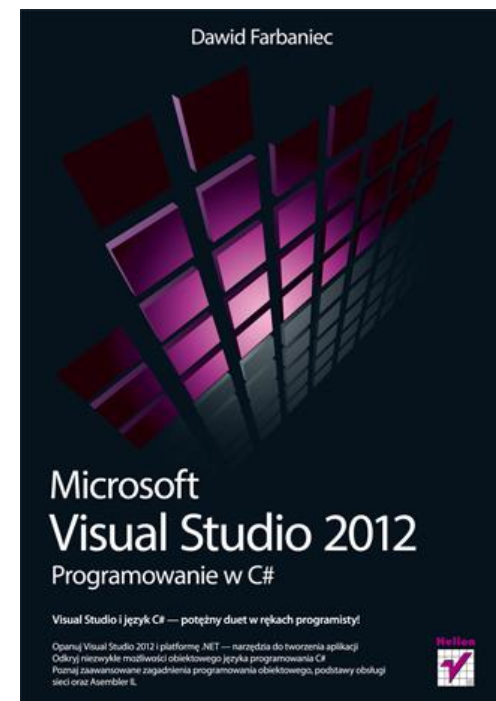
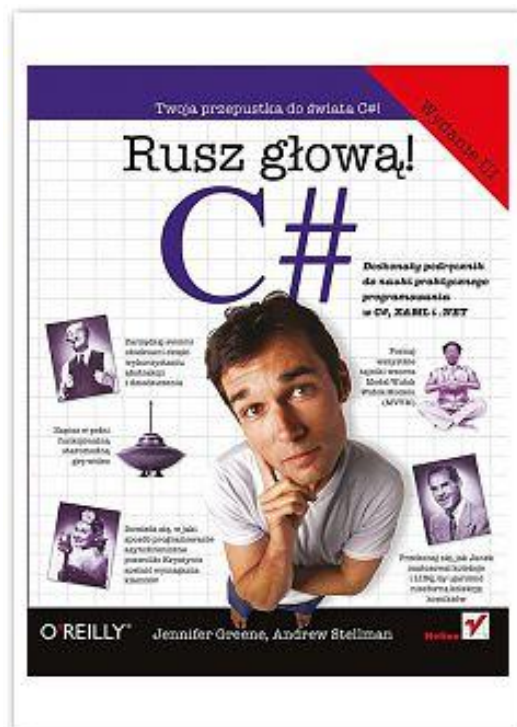
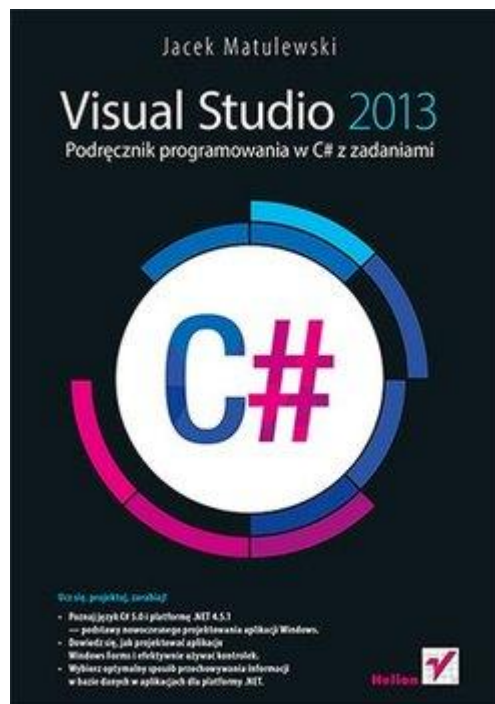



Przykład – dywan Sierpińskiego

Możemy też zapisać do pliku naszą bitmapę:

```
bufor.Save("nazaPliku.jpeg");
```

Literatura:



Użyte w tej prezentacji tabelki pochodzą z książki: Visual Studio 2013. Podręcznik programowania w C# z zadaniami
Autor: Matulewski Jęcek, Helion