

Objektorienterad programmering

Föreläsning 19

© Copyright
Mahmud Al Hakim
mahmud@dynamicos.se
www.webbacademy.se

Agenda

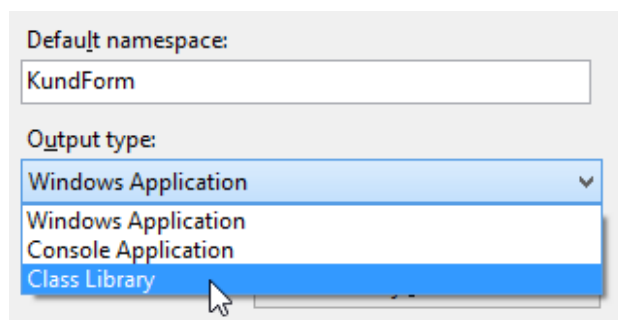
- Konstruktion av egna grafiska komponenter
 - Konstruktion av egen komponent
 - Att rita upp komponenter
 - Metoden OnPaint
 - Klassen Graphics
 - Rita en riktangel
 - Cirklar och ellipser
 - Texter
 - Bilder

Konstruktion av egen komponent

- Ibland vill man skapa en egen grafisk klass som är så generell att den kan användas i olika projekt.
- Då utvecklar man sin klass i ett separat projekt och man kan sedan referera till den från andra projekt.
- Anta att man bestämmer sig för att klassen KundForm (från förra lektionen) är så generell att den skall kunna användas i flera projekt.
- I C# skapar man ett nytt projekt som bara innehåller denna klass.

Skapa en UserControl-komponent

- Skapa först ett nytt projekt (Windows Application).
- Gå till Project - > Properties
- Ändra Output type till Class Library



Den nya klassen är subclass till UserControl

- Ta bort filen program.cs
- Ändra namnet på filen Form1.cs till KundForm.cs
- Ändra så att klassen ärver från UserControl istället för Form

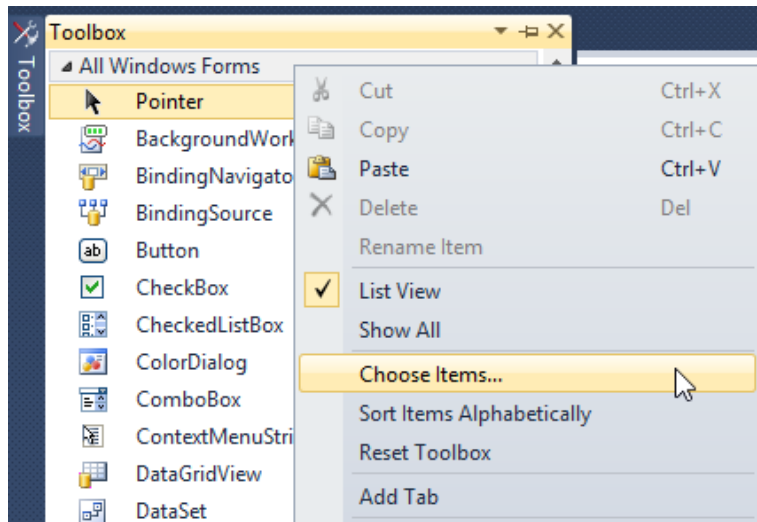
```
namespace KundForm
{
    public partial class KundForm : UserControl ←
    {
        public KundForm()
        {
            InitializeComponent();
        }
    }
}
```

Kompilera (F7) En DLL-fil skapas (KundForm.dll)

Projects > KundForm > KundForm > bin > Debug



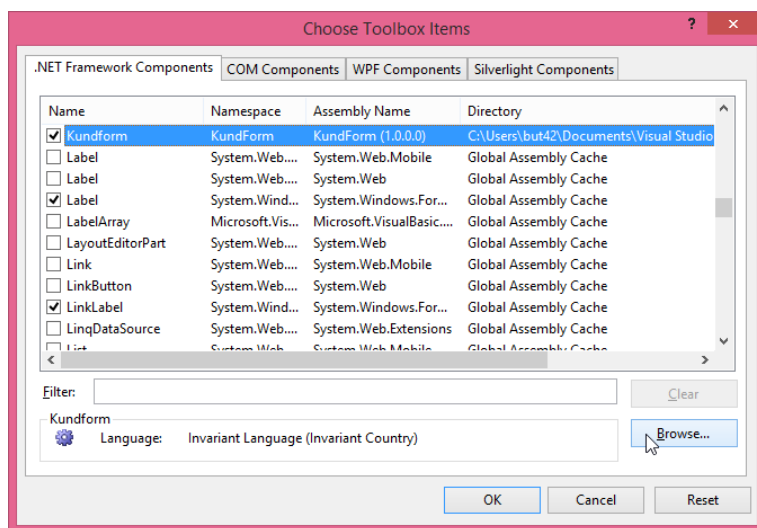
Användning av egen komponent i ett annat projekt



Copyright 2015 - Mahmud Al Hakim www.webacademy.se

7

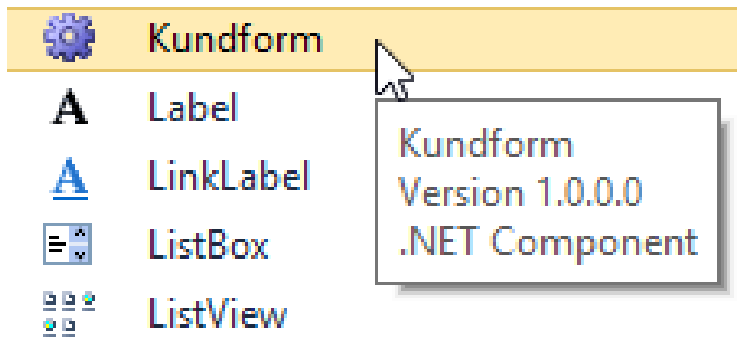
Bläddra efter den nya komponenten



Copyright 2015 - Mahmud Al Hakim www.webacademy.se

8

Nu hittar du den nya komponenten alltid bland dina verktyg



Att rita upp komponenter

- Det vanligaste sättet som används för att rita upp grafiska komponenter är att skapa en egen subclass till en standardklass som erbjuder en tom area t.ex. Form, UserControl eller Panel.
- I subclassen skriver man en alternativ version till den ärvda virtuella metoden **OnPaint**.

```
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);
}
```

Metoden OnPaint

- OnPaint anropas automatiskt varje gång komponenten behöver ritas om.
- OnPaint får som argument en referens till ett objekt av typen **PaintEventArgs**.
- Detta objekt har två egenskaper ClipRectangle och Graphics.
- Egenskapen ClipRectangle anger exakt vilken del av komponenten som behöver ritas om.
- Egenskapen Graphics är referens till ett objekt som tillhör en klass med samma namn, alltså **Graphics**.

Klassen Graphics

- I klassen Graphics deklarereras ett flertal metoder vilka kan användas för att rita olika slags figurer och texter.
- OBS! Graphics har ingen konstruktor.
- Några metoder
 - DrawRectangle
 - FillRectangle
 - DrawEllipse
 - FillEllipse
 - DrawString
 - DrawImage



Rita en riktangel

e refererar till ett objekt av klassen **PaintEventArgs**
Vi behöver en referens till ett Graphics-objekt
som hämtas från egenskapen Graphics för e

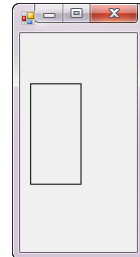
```
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);
    Graphics g = e.Graphics;
```

Först måste basklassens
metod anropas

g används för att komma
åt olika ritmetoder

Rita en riktangel - fort.

Skapa en rektangel
(x, y, bredd, höjd)



```
Rectangle r1 = new Rectangle(10, 50, 50, 100);
Pen svartPenna = new Pen(Color.Black);
g.DrawRectangle(svartPenna, r1);
```

För att kunna rita
kantlinjerna behöver
vi en penna

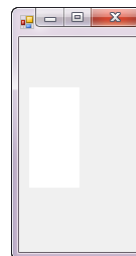
Metoden DrawRectangle
ritar rektangelns kantlinjer

Mer om rektanglar

```
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);
    Graphics g = e.Graphics;
    Rectangle r1 = new Rectangle(10, 50, 50, 100);

    Brush vitPensel = new SolidBrush(Color.White);

    g.FillRectangle(vitPensel, r1);
}
```



Metoden FillRectangle fyller rektangeln med färg med hjälp av penseln

För att fylla rektangeln med färg behöver vi en pensel

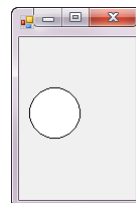
Cirklar och ellipser

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

    Pen svartPenna = new Pen(Color.Black);
    Brush vitPensel = new SolidBrush(Color.White);

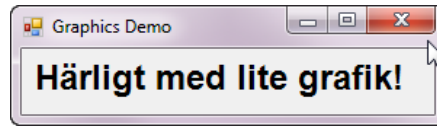
    Rectangle r2 = new Rectangle(10, 50, 50, 50);

    g.FillEllipse(vitPensel, r2);
    g.DrawEllipse(svartPenna, r2);
}
```



Rita en cirkel med hjälp av en rektangel

Texter



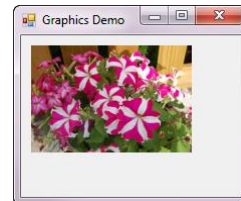
```
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);
    Graphics g = e.Graphics;

    Font f = new Font("Arial", 18, FontStyle.Bold);
    Brush svartPensel = new SolidBrush(Color.Black);
    g.DrawString("Härligt med lite grafik!", f, svartPensel, 5, 5);
}
```

Metoden DrawString
ritar text

Metoden behöver
font, pensel, x och y

Bilder



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

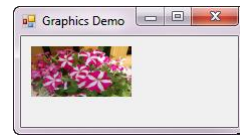
    Image bild = Image.FromFile("blommor.jpg");

    g.DrawImage(bild, 10, 10);
}
```

Metoden DrawImage ritar en
bild på positionen x, y

bild refererar på en
bild som hämtas från
en fil

Skala om bildens storlek



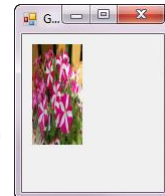
```
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);
    Graphics g = e.Graphics;

    Image bild = Image.FromFile("blommor.jpg");

    g.DrawImage(bild, 10, 10, 100, 50);
}
```

Här anges position x och y
Samt bredd och höjd

Bild i en rektangel



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

    Image bild = Image.FromFile("blommor.jpg");

    Rectangle r = new Rectangle(10, 10, 50, 100);
    g.DrawImage(bild, r);
}
```

DrawImage finns också i en version där man anger
en rektangel som parameter
då får bilden rektangelns x , y , bredd och höjd

Skriva på en bild

Varje bild har en egen rityta av typen Graphics. Man kan få en referens till denna genom att anropa metoden FromImage

```
base.OnPaint(e);
Graphics g = e.Graphics;
Image bild = Image.FromFile("blommor.jpg");
Graphics g2 = Graphics.FromImage(bild);
Font f2 = new Font("Verdana", 14, FontStyle.Italic);
Brush gulPensel = new SolidBrush(Color.Yellow);
g2.DrawString("Blommor", f2, gulPensel, 0, 0);
g.DrawImage(bild, 100, 120);
```

OBS!
g2 refererar till ritytan på själva bilden medan g refererar till ritytan i den grafiska komponenten i konstruerar.

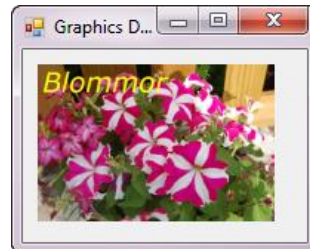
Copyright 2015 - Mahmud Al Hakim www.webacademy.se

21

Tips

```
base.OnPaint(e);
Graphics g = e.Graphics;
Image bild = Image.FromFile("blommor.jpg");
Graphics g2 = Graphics.FromImage(bild);
Font f2 = new Font("Verdana", 14, FontStyle.Italic);
Brush gulPensel = new SolidBrush(Color.Yellow);
g2.DrawString("Blommor", f2, gulPensel, 0, 0);
g.DrawImage(bild, 100, 120);

bild.Save("blommor2.jpg");
```



Vi kan spara en bild efter det att man ritat på den!

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

22