

Objektorienterad programmering

Föreläsning 18

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

Agenda

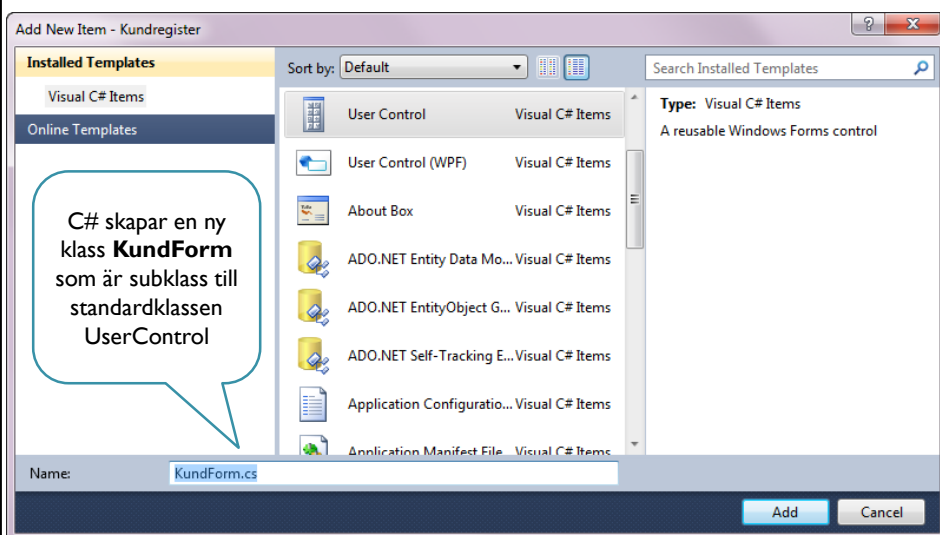
- Konstruktion av egna grafiska komponenter
 - Skapa sammansatta komponenter med UserControl
 - FlowLayoutPanel
 - Utökning av standardklasser

Sätta samman flera standardkomponenter

- Ibland har man ett antal grafiska komponenter som hör ihop och alltid bör placeras tillsammans t.ex. formulär av olika slag.
- Formulär innehåller flera textrutor och etiketter.
- Man kan då bilda en komponent som innehåller alla de olika delkomponenterna.
- Fördelen är att kunna hantera alla delkomponenter som en enhet.
- I C# finns en standardklass med namnet **UserControl** som är speciellt avsedd att användas för att skapa **sammansatta komponenter**.

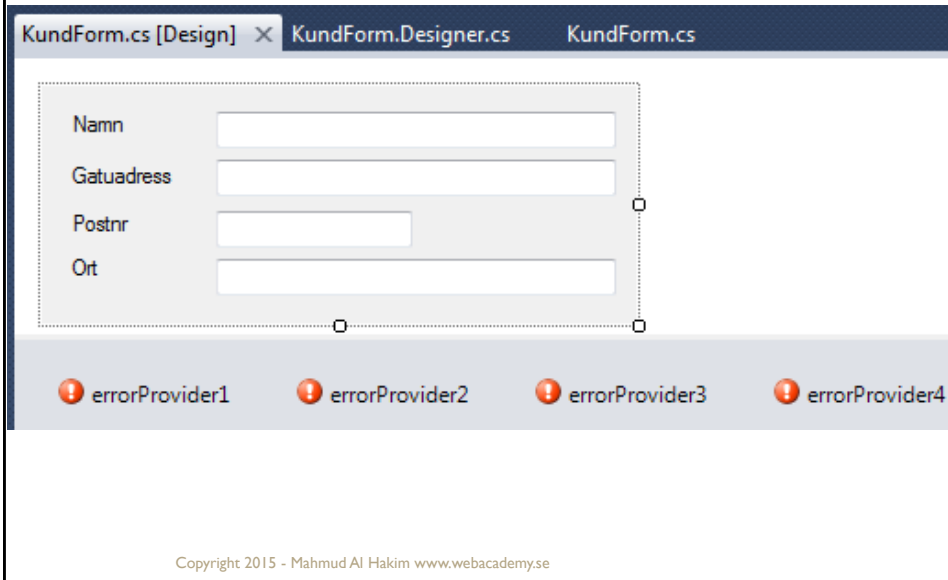
Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Att lägga till UserControl till ett projekt

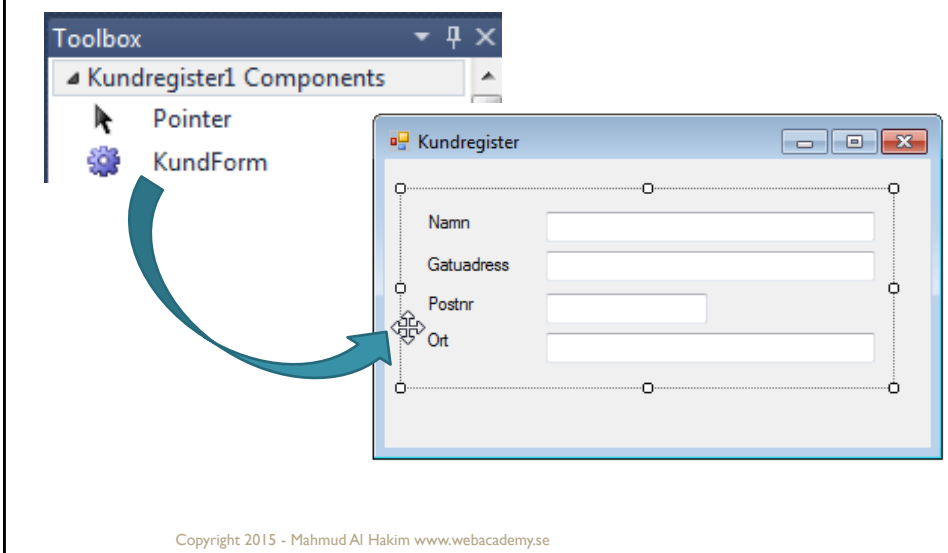


Copyright 2015 - Mahmud Al Hakim www.webacademy.se

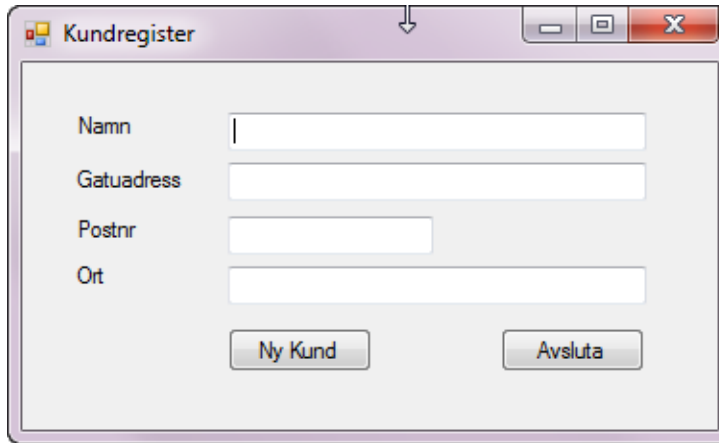
Designfönster för UserControl



En ny komponent skapas efter kompileringen!



Kundregister – Ny version



The screenshot shows a Windows application window titled "Kundregister". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area contains four text input fields stacked vertically, labeled "Namn", "Gatuadress", "Postnr", and "Ort". Below the input fields are two buttons: "Ny Kund" and "Avsluta".

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Klassen KundForm Konstruktör och egenskaper

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

    // Egenskaper
    public string Namn { get { return textBox1.Text.Trim(); } }
    public string Gata { get { return textBox2.Text.Trim(); } }
    public string Postnr { get { return textBox3.Text.Trim(); } }
    public string Ort { get { return textBox1.Text.Trim(); } }
}
```

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Klassen KundForm

Metoden TextBoxOK

```
// Metod som kontrollerar att en TextBox är ifylld
private bool TextBoxOK(TextBox b, ErrorProvider e)
{
    if (b.Text.Trim().Length > 0)
    {
        e.Clear();
        return true;
    }
    else
    {
        e.SetError(b, "Fältet måste fyllas i");
        return false;
    }
}
```

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Klassen KundForm

Metoden Kontrollera

```
// Metod som kontrollerar att alla textboxar är ifyllda
public bool Kontrollera()
{
    bool ok = true;
    ok = TextBoxOK(textBox1, errorProvider1) && ok;
    ok = TextBoxOK(textBox2, errorProvider1) && ok;
    ok = TextBoxOK(textBox3, errorProvider1) && ok;
    ok = TextBoxOK(textBox4, errorProvider1) && ok;
    return ok;
}
```

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Klassen KundForm

Metoden Rensa

```
public void Rensa()
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    ActiveControl = textBox1;
}
```

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Kundregister – Huvudprogrammet

Del I

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

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    kundFil.Close();
}

private void exitButton_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Kundregister – Huvudprogrammet Del 2

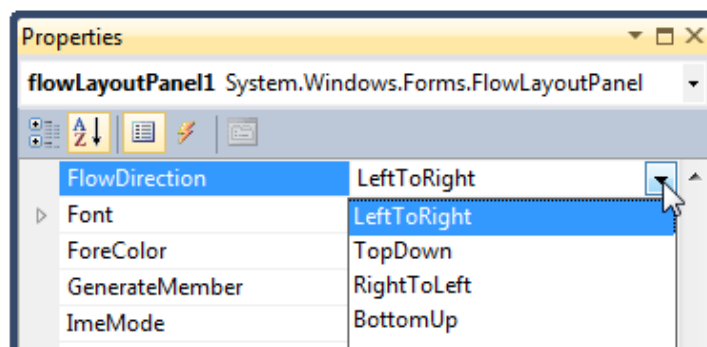
```
StreamWriter kundFil =  
    new StreamWriter("kund.txt", true, Encoding.Default);  
  
private void newButton_Click(object sender, EventArgs e)  
{  
    if (kundForm1.Kontrollera())  
    {  
        kundFil.Write(kundForm1.Name + ";");  
        kundFil.Write(kundForm1.Gata + ";");  
        kundFil.Write(kundForm1.Postnr + ";");  
        kundFil.WriteLine(kundForm1.Ort);  
        kundForm1.Rensa();  
    }  
}
```

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

FlowLayoutPanel



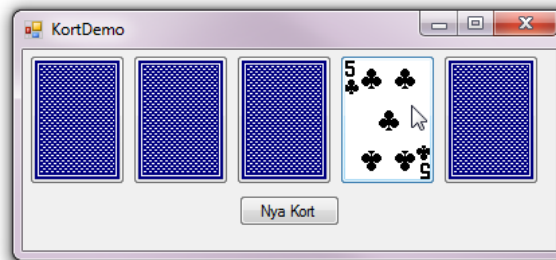
- Med hjälp av klassen FlowLayoutPanel kan man lägga ut komponenter t.ex. från vänster till höger.
- Det går också att placera ut i en annan riktning med hjälp av egenskapen FlowDirection.



Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Utökning av standardklasser

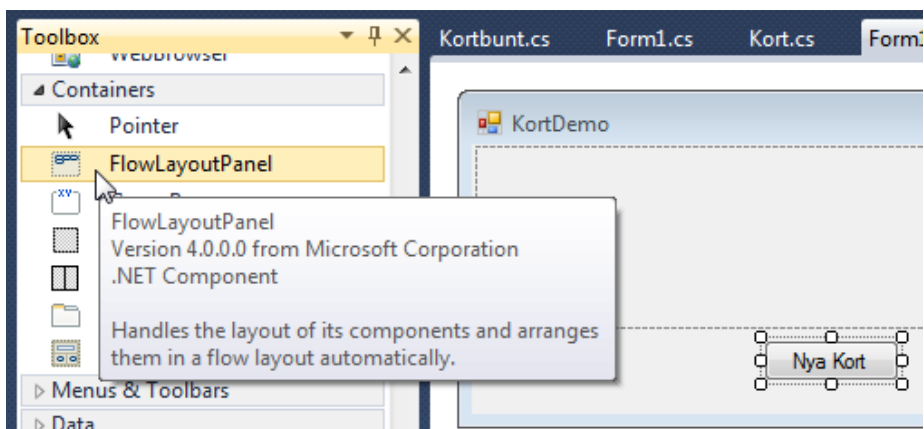
- Man kan skapa nya egna grafiska komponenter genom att skapa en subclass till en standardklass.
- T.ex. Skapa en klass som beskriver ett Kort som är subclass till klassen Button, då kommer klassen Kort att ärva alla egenskaper (samt metoder och händelser) som finns i standardklassen Button.



Copyright 2015 - Mahmud Al Hakim www.webacademy.se

KortDemo - Design

- Skapa ett fönster med två komponenter, en FlowLayoutPanel och en knapp.




Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Klassen Kort – Del 1

```
namespace KortDemo
{
    enum KortFärg { KLÖVER, RUTER, HJÄRTER, SPADER };
    enum KortValör
    {
        TVÅ = 2, TRE, FYRA, FEM, SEX, SJU, ÅTTA,
        NIO, TIO, KNEKT, DAM, KUNG, ESS
    };

    class Kort : Button
    {
    }
}
```



Klassen ärver från Button

Klassen Kort – Del 2

```
class Kort : Button
{
    KortFärg f;
    KortValör v;
    static string[] färgTab = { "Kl", "Ru", "Hj", "Sp" };
    static string[] valörTab = {"2", "3", "4", "5", "6", "7", "8",
                               "9", "10", "Kn", "D", "K", "E"};

    bool synligt;
    public static string BildMapp = @"bilder\";
    Image framsida;
    static Image baksida; // behövs bara en gång
}
```

Klassen Kort – Del 3

```
// Egenskaper
public KortFärg Färg { get { return f; } }
public KortValör Valör { get { return v; } }
public int Värde { get { return (int)v; } }
public bool Synligt
{
    get { return synligt; }
    set {
        synligt = value;
        if (synligt) this.Image = framsida;
        else this.Image = baksida;
    }
}
```

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Klassen Kort – Del 4

```
// Konstruktor med färg, valör och synligt (bool)
public Kort(KortFärg fä, KortValör va, bool synl)
{
    f = fä; v = va;
    framsida = Image.FromFile(BildMapp + färgTab[(int)f]
        + valörTab[Värde-2] + ".gif");

    if (baksida == null) // Hämtas bara en gång (static)
        baksida = Image.FromFile(BildMapp + "Bak.gif");

    this.Size = framsida.Size;
    Synligt = synl; // OBS!egenskapen Synligt
}
// Konstruktor med enbart färg, valör
public Kort(KortFärg fä, KortValör va)
: this (fä,va,false) { }
```

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

Klassen Kortbunt

- Precis samma som förut (inga ändringar).
- Finns att ladda ner här
<http://skansholm.com/skarp/exempel/Kortbunt.cs>

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

KortDemo – kod – del I

```
namespace KortDemo
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void card_Click(object sender, EventArgs e)
        {
            Kort k = (Kort)sender;
            k.Synligt = !k.Synligt; // Vänd kortet
        }
    }
}
```

En hanterare som anropas när man klickar på ett kort

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

KortDemo – kod – del 2

```
private void newbutton_Click(object sender, EventArgs e)
{
    flowLayoutPanel1.Controls.Clear(); // Ta bort gamla kort
    Kortbunt lek = new Kortbunt();
    lek.NyKortlek();
    lek.Blanda();
    for (int i = 0; i < 5; i++)
    {
        Kort k = lek.TaBortÖversta();
        flowLayoutPanel1.Controls.Add(k);
        // Koppla hanterare till kortet
        k.Click += card_Click;
    }
}
```

Copyright 2015 - Mahmud Al Hakim www.webacademy.se