Databundna kontroller

Upphovsrätt för detta verk

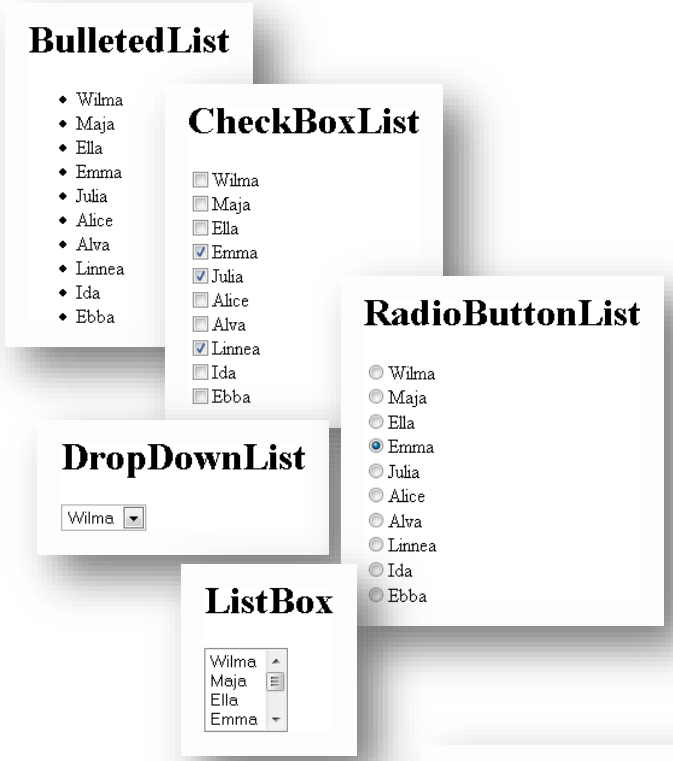
Detta verk är framtaget i anslutning till kursen ASP.NET Web Forms vid Linnéuniversitetet.

Du får använda detta verk så här:

Vad är en databunden kontroll?

- Du använder databundna kontroller till att skapa ett gränssnitt för arbete med data.
- Databundna kontroller använder du till att presentera och redigera data från databaser, data från XML-filer eller data från vilken annan typ av datakälla som helst.
- Databundna kontroller kan delas in i tre huvudgrupper efter hur de presenterar datat:
  - Listor
  - Tabeller
  - Hierarkier (tar denna kurs inte upp)

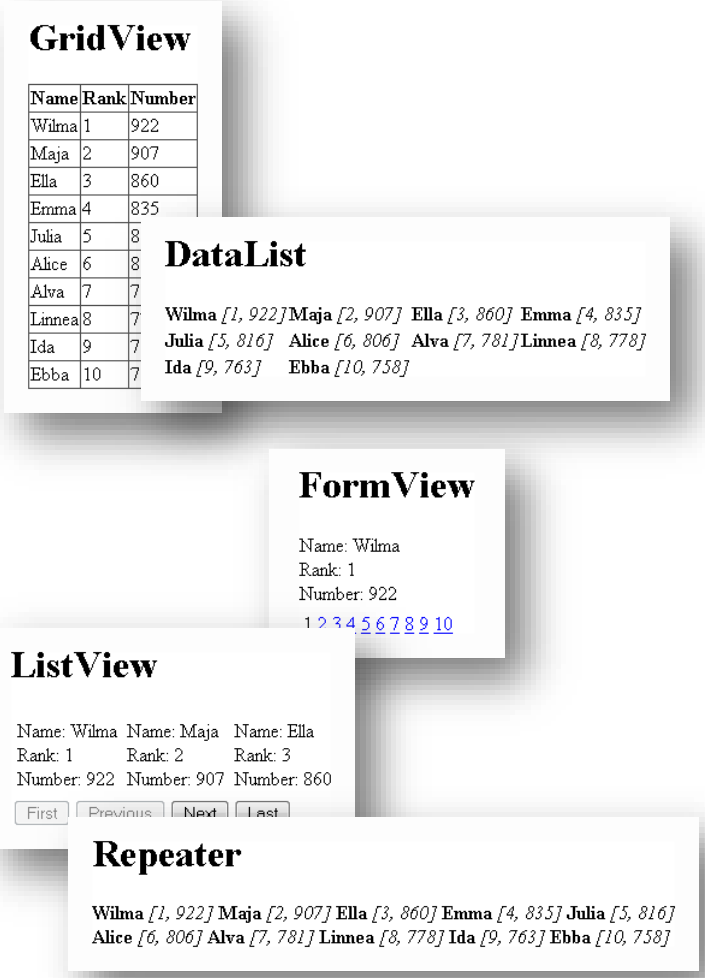
Databundna listkontroller



- Du använder listkontroller till att presentera enkla listor.
- Alla fem ärver från samma basklass – `ListControl`.
- `CheckBoxList`- och `RadioButtonList`-kontrollerna renderas som input-element i en tabell. Vill du inte ha en tabell sätter du egenskapen `RepeateLayout` till `Flow`.
- Du kan binda data, t.ex. en array, till kontrollerna programmatiskt med hjälp av egenskapen `DataSource` och metoden `DataBind`.

```
string[] girlNames =  { "Wilma", "Maja", "Ella", "Emma", "Julia",
                        "Alice", "Alva", "Linnea", "Ida", "Ebba" };
GirlNamesBulletedList.DataSource = girlNames;
GirlNamesBulletedList.DataBind();
```

Databundna tabellkontroller



- Databundna tabellkontroller behöver nödvändigtvis inte renderas som `table`-element.
- Med databundna tabellkontroller kan du presentera och modifiera data från databaser eller andra datakällor.
- Det finns sex databundna tabellkontroll som kan delas in i två huvudkategorier:
  - De som kan visa flera dataposter i taget.
    - `GridView`, `DataList`, `ListView` och `Repeater`.
  - De som kan visa en datapost i taget.
    - `DetailsView` och `FormView`.
- Du kan binda en datakälla till kontrollen deklarativt med hjälp av en `DataSource`-kontroll, t.ex. en `ObjectDataSource`-kontroll som du kopplar till en metod i en affärslagerklass.

Två sätt att binda data till en databunden kontroll

- ✓ Data som binds till en databunden kontroll kan t.ex. exponeras via metoder i klasser som exempelvis returnerar en array eller samling av referenser till objekt.
- ✓ Programmatiskt
  - Datakällan binds till kontrollen i ”code-behind”-filen i regel i samband med en ”get” av sidan.
- ✓ Deklarativt
  - Datakällan binds till kontrollen i aspx-sidan t.ex. med en `ObjectDataSource`-kontroll som använder en metod i en affärslagerklass.

Databindningsuttryck

- Databindningsuttryck är speciella uttryck som inte utvärderas då koden körs. Ett databindningsuttryck skrivs mellan `<%#` och `%>`.
- Då du binder en kontroll till data via ett `DataSource`-objekt, eller om du gör det i ”code-behind”-filen med egenskapen `DataSource` och metoden `DataBind`, så skapas databindningshändelser vilket leder till att databindningsuttryck utvärderas.
- Använder du en t.ex. en `Repeater`-kontroll måste du skapa en mall (”item template”) innehållande databindningsuttryck för att kunna presentera data från en datakälla.

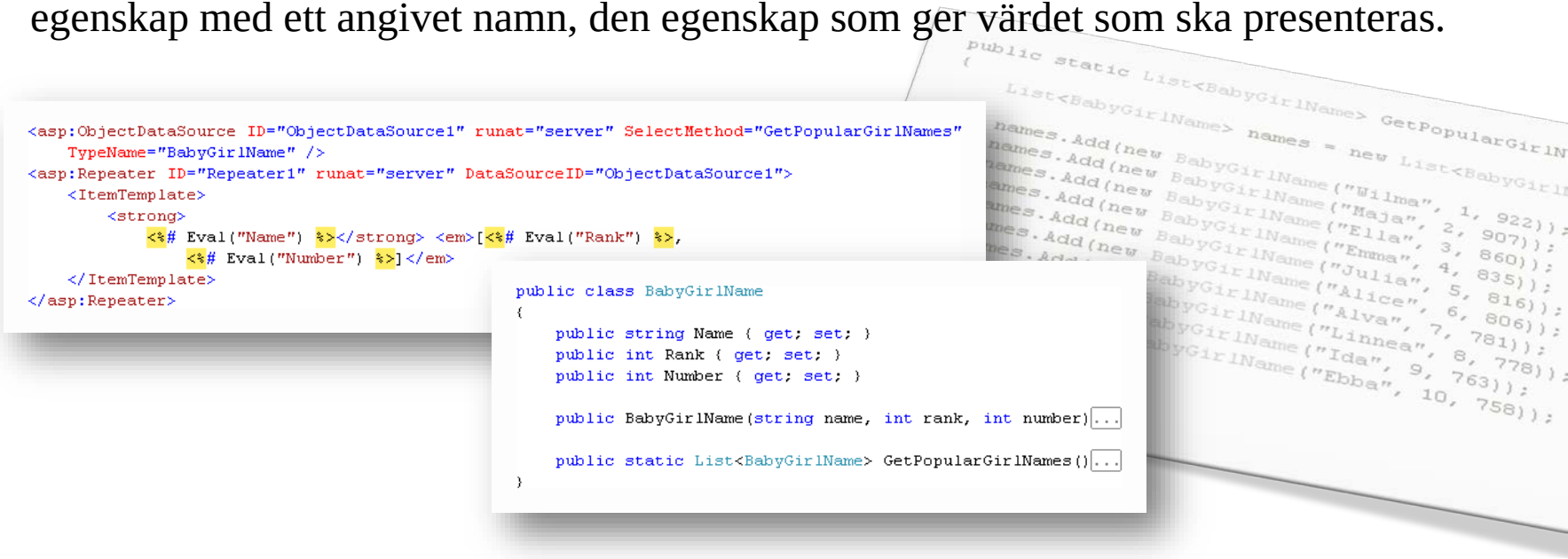
```
<asp:Repeater ID="Repeater1" runat="server" DataSourceID="ObjectDataSource1">
    <ItemTemplate>
        <strong>
            <%# Eval("Name") %></strong> <em>[<%# Eval("Rank") %>,
                <%# Eval("Number") %>]</em>
    </ItemTemplate>
</asp:Repeater>
```

Repeater

Wilma *[1, 922]* **Maja** *[2, 907]* **Ella** *[3, 860]* **Emma** *[4, 835]* **Julia** *[5, 816]*
Alice *[6, 806]* **Alva** *[7, 781]* **Linnea** *[8, 778]* **Ida** *[9, 763]* **Ebba** *[10, 758]*

Att binda data till en ”template”

- ✓ `ObjectDataSource`-kontrollen bestämmer med egenskaperna/attributen `TypeName` och `SelectMethod` vilken klass och metod som ska användas som datakälla.
- ✓ Databindningsuttrycken utvärderas för var och en av posterna som binds till kontrollen.
- ✓ Metoden `Eval` utvärderar posten (dataobjektet) som bundits för att hitta en egenskap med ett angivet namn, den egenskap som ger värdet som ska presenteras.



Formatering av databundna tabellkontroller

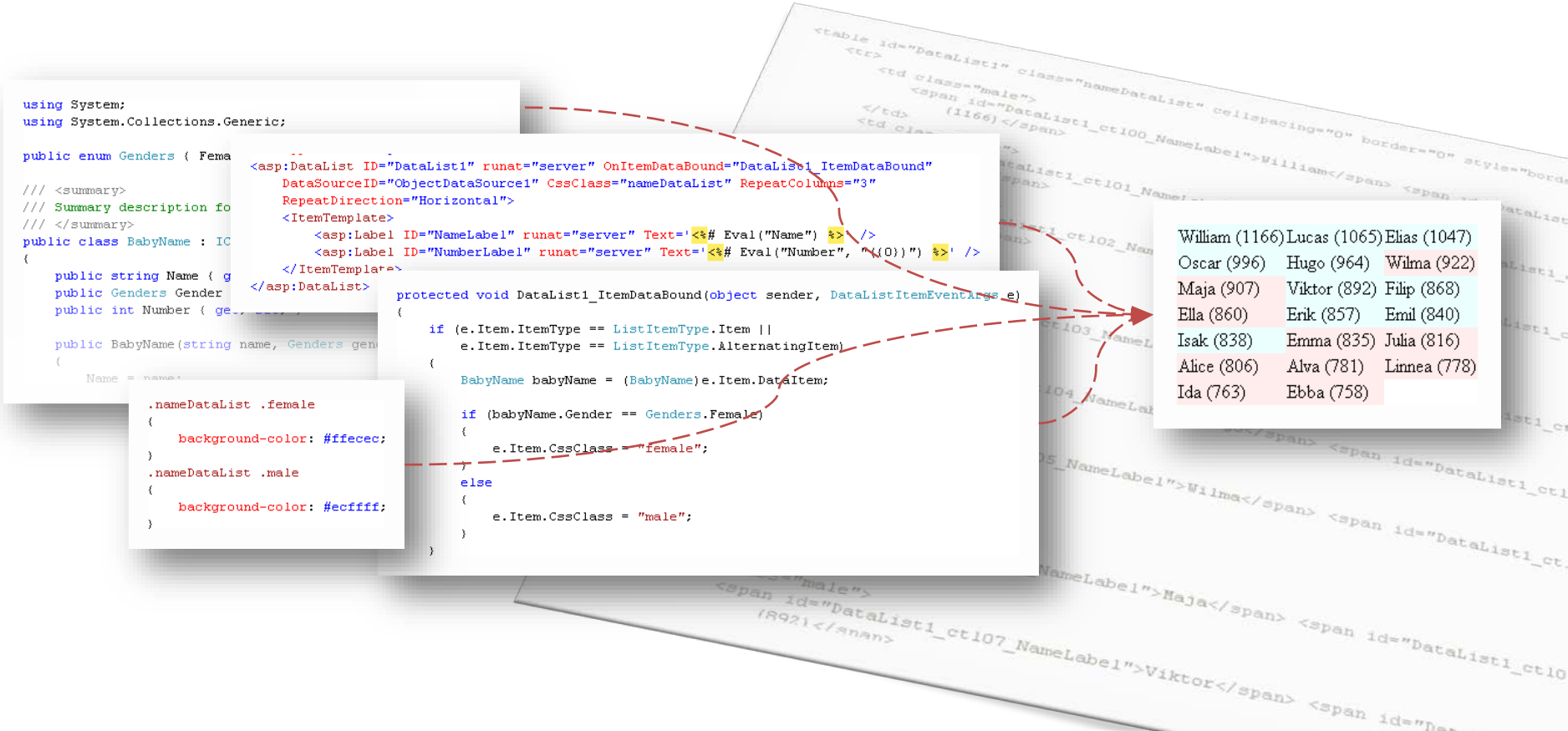
- Du har stora möjligheter att påverka formateringen av de databundna tabellkontrollerna. Kontrollerna, utom `ListView` och `Repeater`, har flera egenskaper som har med formatering att göra – **ANVÄND INTE DEM** – använd istället ”cascading style sheets” för formatering. Du kan via egenskapen `CssClass` formatera olika delar av kontrollerna.

GridView och händelsen RowDataBound

- `GridView`-kontrollen skapar händelser för varje rad som skapas och databinds – `RowCreated` och `RowDataBound`. Innehåller datakällan 20 objekt så skapas 20 `RowCreated`- och `RowDataBound`-händelser.
- Genom att skapa en hanterarmetod för händelsen `RowDataBound` kan du påverka flera olika saker för den enskilda raden i tabellen, t.ex. kan du utifrån dataobjektet som binds till raden bestämma hur raden ska formateras.

DataList, ListView och Repeater…

- ✓ …fungerar på liknande sätt som `GridView` och händelsen `RowDataBound`, men…
- ✓ …händelsen heter `ItemDataBound` för dessa kontroller.



…och mycket mer finns att läsa…

- ✓ …om ”Data Binding” i kapitel 9.
- ✓ …om ”Rich Data Controls” i kapitel 10.