

Internettechnologie

academiejaar 2012–2013

Practicum 1: ASP.NET en ADO.NET

14 februari 2013

Pieterjan De Potter

Multimedia Lab

Vakgroep Elektronica en Informatiesystemen
Faculteit Ingenieurswetenschappen en Architectuur
Universiteit Gent

Overzicht

- Algemene inleiding
 - inleiding
 - overzicht practica
 - praktische afspraken
- Inleiding practicum 1
 - statische versus dynamische webpagina's
 - .NET-raamwerk
 - Programmeertaal C#
 - ASP.NET
 - ADO.NET
- Werken met Visual Web Developer 2008 Express
- Opgave practicum 1

Algemene inleiding

- Inleidingslessen
 - uitleg bij de oefeningen
 - één inleidingsles per practicum
 - steeds in auditorium A1
- Practica
 - computeroefeningen
 - op te lossen in de PC-klas (zaal Konrad Zuse)
- Aard practica
 - kennismaking met ASP.NET en ADO.NET (C#)
 - implementatie van dynamische webapplicaties

Algemene inleiding

- Verantwoordelijke begeleider
 - Pieterjan De Potter
 - Multimedia Lab
vakgroep Elektronica en Informatiesystemen (ELIS)
Gaston Crommenlaan 8 bus 201 (Zuiderpoort)
9050 Ledeberg-Gent
 - Bureau 2.24 (tweede verdieping)
 - pieterjan.depotter@ugent.be
 - Gelieve steeds [itech] en uw groepsnummer in het onderwerp te vermelden
- Medebegeleiders
 - Laurens De Vocht
 - Miel Vander Sande

Algemene inleiding

- Oefeningen moeten gemaakt worden in groepen van twee
 - groepsindeling blijft dezelfde doorheen alle practica
 - elke groep heeft een eigen nummer
 - beoordeling gebeurt per groep
 - bij problemen steeds dit groepsnummer vermelden
- Alle informatie beschikbaar op Minerva
 - <https://minerva.ugent.be>
 - Groepen inschrijven via Minerva:
“Groepen > Practica > inschrijven”
 - Merk op: eens ingeschreven liggen de groepen vast.

Algemene inleiding

- Informatie op Minerva
 - centrale website voor het vak
 - presentaties en opgaven
 - registratie van groepen
 - ad valvas
 - punten
 - links naar extra informatie
 - standaarden of specificaties
 - tutorials, help-pagina's, ...

Overzicht practica

- 3 practica
- Elk practicum wordt voorafgegaan door een inleidingsles
 - vandaag: algemene inleiding + ASP.NET en ADO.NET
 - do 14 maart (10u): MPEG-21 en W3C webservices
 - do 18 april (10u): Semantisch Web
- Inleidingslessen gaan door in Auditorium A1
- Werken aan oefeningen in PC-klas 'Konrad Zuse'
 - elke dinsdag van 8u30 tot 11u30

Overzicht practica

- Oplossingen practica moeten ingediend worden vóór de deadlines
 - practicum 1: donderdag 21 maart, om 14u00 STIPT
 - practicum 2: donderdag 18 april, om 14u00 STIPT
 - practicum 3: donderdag 16 mei, om 14u00 STIPT

te laat = nul

- Elke oefening wordt in detail bekeken (resultaat en broncode)
- Criteria verbetering (in volgorde van belangrijkheid):
 - stiptheid, correctheid, volledigheid, leesbaarheid (broncode)

Overzicht practica

- Indienen oefeningen
 - .zip file via de Minerva “Drop Box”
 - **bestandsnaam met groepnummer (zie opgaves)**
 - bevat bestandsnamen zoals beschreven in opgaven
 - Versturen naar:
 - Peter Lambert **en** Pieterjan De Potter
 - **NIET** naar ‘iedereen’

Praktische afspraken (1/5)

- Inschrijvingen en groepsindeling
 - bij de groepsindeling is het verplicht om zich in te schrijven per twee studenten
 - uitzonderingen kunnen toegestaan worden
 - bv., GIT, werkstudenten of aantal studenten niet even
 - hoofdbegeleider per e-mail contacteren
 - er zal handmatig een groep aangemaakt worden
- deadline inschrijvingen
 - **donderdag 7 maart 2012 om 14u00**
 - elke groep moet ingeschreven zijn vóór dit afgesproken tijdstip

Praktische afspraken (2/5)

- Kopiëren van oefeningen
 - indien vastgesteld wordt dat groepen oplossingen van elkaar kopiëren, dan worden de behaalde punten verdeeld onder de ‘deelnemende’ groepen
 - er wordt om uitleg gevraagd bij de betrokken groepen om tot een eerlijke verdeling te komen
 - indien dit niet lukt, dan worden de punten gelijk verdeeld

Praktische afspraken (3/5)

- Te laat indienen
 - oplossingen die te laat ingediend worden (na de vermelde deadline) worden gequoteerd met nul
- Correcte benamingen ingediende oefeningen
 - de ingediende oplossingen moeten een correcte naamgeving hebben en met vermelding van het groepsnummer geüpload worden via drop box
 - indien niet, dan worden deze oplossingen niet teruggevonden door onze scripts en dus als niet-opgelost beschouwd (nul)

Praktische afspraken (4/5)

- Bijhouden van backups (bv. persoonlijke folder op eduserv)
 - indien oplossingen als backup bijgehouden worden op de account op eduserv, dan dient men erop toe te zien dat anderen geen toegang hebben tot de eigen account
 - controleer leesrechten
 - kan oorzaak zijn van onvrijwillig meewerken aan kopiëren
- een gelijkaardige opmerking voor het opslaan van oplossingen op een PC van de PC-klassen
 - verwijder alle lokale bestanden

Praktische afspraken (5/5)

- In geval van onregelmatigheden
 - alle gevallen van vermoedelijke inbreuken tegen deze regels zullen afzonderlijk en in detail behandeld worden
 - door de verantwoordelijke voor de oefeningen
 - door de titularis van het opleidingsonderdeel
 - er zal steeds overleg met de betrokken studenten georganiseerd worden, zodat zij de redenen voor de evt. inbreuken kunnen toelichten
 - de bedoeling van dit alles is om te komen tot een zo eerlijk mogelijke quoterings van de oefeningen

Overzicht

- Algemene inleiding
 - inleiding
 - overzicht practica
 - praktische afspraken
- Inleiding practicum 1
 - statische versus dynamische webpagina's
 - .NET-raamwerk
 - Programmeertaal C#
 - ASP.NET
 - ADO.NET
- Werken met Visual Web Developer 2008 Express
- Opgave practicum 1

Statische versus dynamische webpagina's

- Statische webpagina's
 - vaste en onveranderlijke tekst (HTML)
- Dynamische webpagina's
 - pagina's die zich aanpassen (intelligent gedrag)
 - gebruikscontext
 - bijvoorbeeld: PC vs. smartphone
 - verwerking formulieren
 - interactie databank
 - resultaat verwerking
 - (aangepaste) tekst (HTML) + client-side logica
 - logica voor realisatie aanpassing
 - beschikbaar op server en/of client

Statische versus dynamische webpagina's

- Client-side verwerking
 - aanpassing webpagina gebeurt door client (browser-omgeving)
 - indien mogelijk
 - technologieën
 - JavaScript, VBScript, JScript, CSS, XSLT
- Server-side verwerking
 - aanpassing gebeurt door component webserver
 - client ontvangt resultaat van aanpassing
 - HTML (+ client side logica)
 - technologieën
 - Classic ASP, PHP, JSP, CF, CGI ("1^e generatie")
 - ASP.NET, JavaServer Faces ("2^e generatie")

Statische versus dynamische webpagina's

- Mogelijke technologieën voor dynamische webpagina's
 - JavaScript, VBScript (client side)
 - validatie van invoer via formulieren
 - tonen, verbergen en aanpassen HTML-code
 - Classic ASP, JSP, PHP, CF, Ruby (server side)
 - scripting talen verweven in HTML-code
 - Classic ASP: VBScript/JScript in HTML-code
 - ASP.NET, JavaServer Faces (server side)
 - expliciete scheiding opmaak en code
 - gebruik van volwaardige achterliggende programmeertalen
 - ASP.NET: C# en HTML-code

Inleiding practicum 1 – overzicht

- Statische versus dynamische webpagina's
- .NET-raamwerk
- Programmeertaal C#
- ASP.NET
- ADO.NET

.NET-raamwerk

- Architectuur/filosofie voor het bouwen van applicaties
 - focus op mobiliteit en webtoepassingen
 - gebruik van webstandaarden
 - HTTP: communicatieprotocol
 - XML: formaat voor uitwisseling informatie
 - SOAP, WSDL: web services (zie later)
- Overkoepelende benaming voor bundel van technologieën
 - .NET-talen (managed languages)
 - C# (C Sharp), VB.NET, managed C++, Perl, Python
 - .NET-raamwerk
 - concrete infrastructuur voor realisatie .NET-filosofie
 - aanbieden elementaire systeemservices
 - steeds meer geïntegreerd in Windows-besturingssysteem

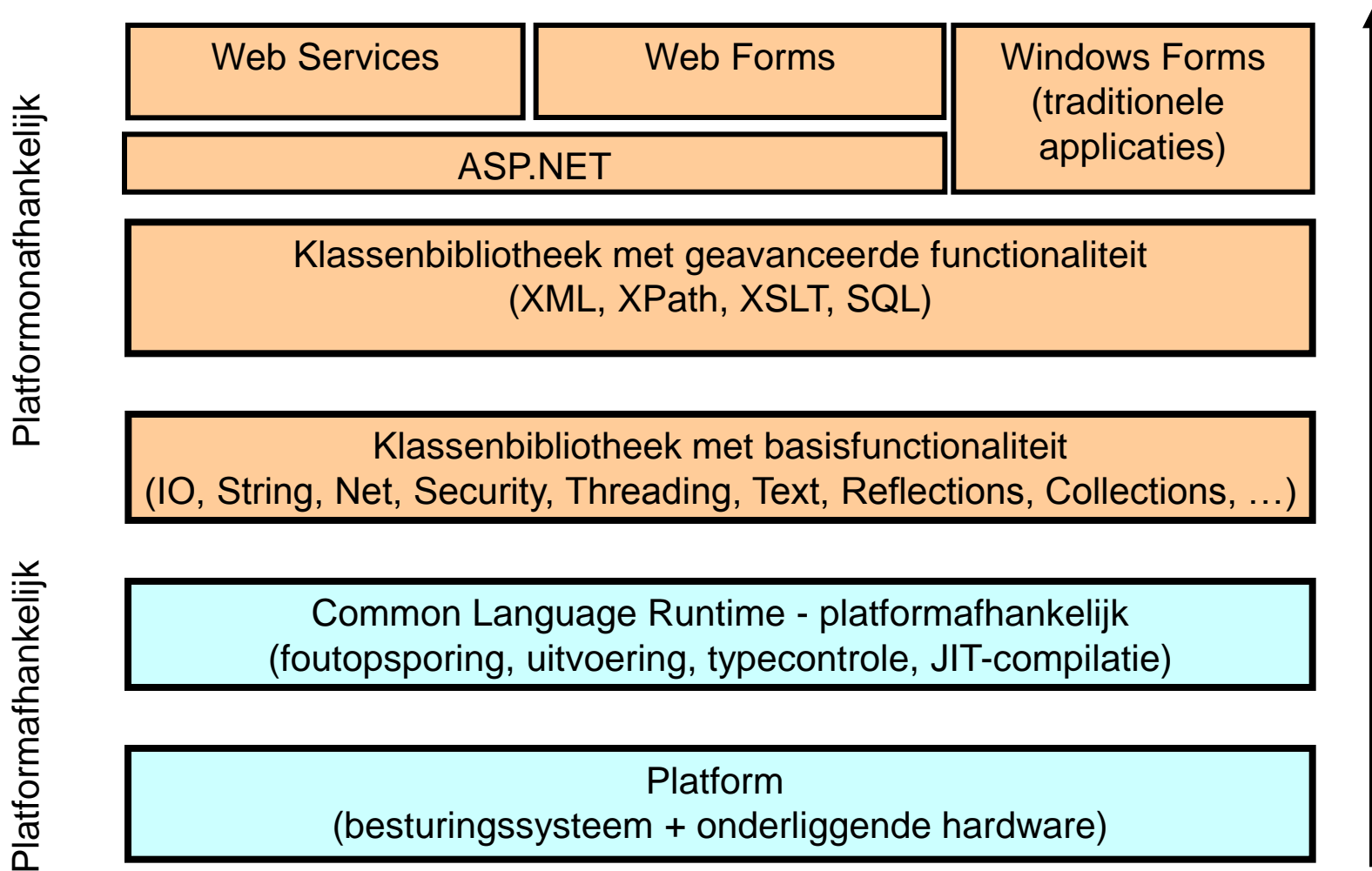
.NET-raamwerk

- Common Language Specification (CLS)
 - beschrijft deelverzameling van gegevenstypen
 - ondersteund door .NET-runtime
 - alle .NET-talen hebben deze gemeen
- .NET-talen worden gecompileerd naar een tussentaal
 - Intermediate Language (IL)
 - vergelijkbaar met bytecode in Java
 - platformonafhankelijkheid
- .NET-talen worden beheerde talen genoemd
 - IL-code wordt beheerd door runtime
 - automatisch geheugenbeheer, ...
- Componenten geschreven in managed code: assemblies
 - uitvoerbare bestanden (*.exe)
 - bibliotheken (*.dll)

.NET-raamwerk

- Common Language Runtime (CLR)
 - compileert IL-code naar native code (x86, x64, ...)
 - bij installatie of bij eerste uitvoering (Just-In-Time)
 - beheert uitvoering van .NET-code (IL-code)
 - geheugenbeheer, debugging, exceptions
 - vergelijkbaar met Java Virtual Machine (JVM)
- Klassenbibliotheek (Framework Class Library – FCL)
 - hiërarchische verzameling .NET-klassen die ontwikkelaars in eigen applicatie kunnen gebruiken
 - bevat zowel basisfunctionaliteit als geavanceerde logica
 - I/O, threading, verwerking XML, ...
 - klassen zijn ondergebracht in naamruimten
 - bijvoorbeeld: `System.Web`

.NET-raamwerk



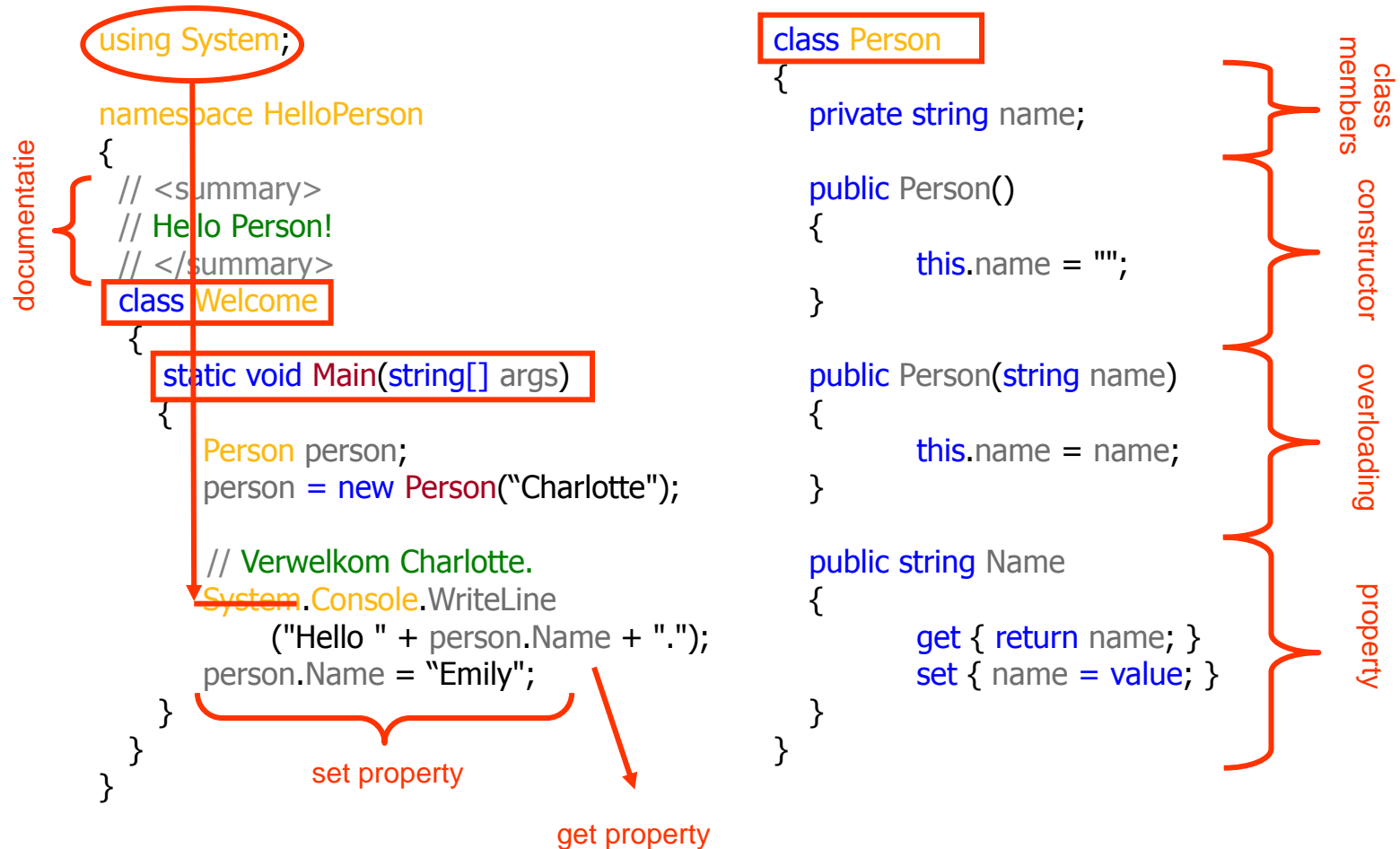
Inleiding practicum 1 – overzicht

- Statische versus dynamische webpagina's
- .NET-raamwerk
- Programmeertaal C#
- ASP.NET
- ADO.NET

C#

- Geïnspireerd op C++ en Java
 - belangrijkste taal in .NET
 - gestandaardiseerd door
 - ECMA (ECMA-334)
 - ISO/IEC (ISO/IEC 23270:2006)
- Enkele eigenschappen
 - volledig objectgeoriënteerd
 - klassen, overerving, interfaces, ...
 - sterk getypeerd
 - tijdens compilatie controle op consistent gebruik van datatypes en uitdrukkingen

C# - voorbeeld 'Hello Charlotte'



C# - enkele taalaspecten

- Statements voor flow control analoog aan Java en C++
 - if-then-else
 - for
 - while
 - switch-case (ook op strings)
- Datatype bool
 - object zonder geheelwaardige voorstelling
 - *if (aString.Length == 0) { // ... } : ok*
 - *if (aString.Length) { // ... } : niet ok*
 - (in de praktijk: *if (aString.IsNullOrEmpty)*)

C# - enkele taalaspecten

- Arrays
 - syntax
 - `datatype [] naam = new datatype[aantalElementen];`
 - voorbeeld
 - `int [] getallen = new int[10];`
 - gebruik
 - `foreach (int eenGetal in getallen) { // ... }`
 - `for (int i = 0; i < getallen.Length; i++)`
`{`
`int eenGetal = getallen[i];`
`// Verdere verwerking.`
`}`
 - opmerking
 - indexering vanaf 0

Inleiding practicum 1 – overzicht

- Statische versus dynamische webpagina's
- .NET-raamwerk
- Programmeertaal C#
- **ASP.NET**
 - situering
 - besturingselementen en gebeurtenisafhandeling (controls en events)
 - overzicht besturingselementen
 - webpagina's die naar zichzelf posten
 - serverobjecten
- ADO.NET

ASP.NET – situering

- Situering
 - onderdeel van het .NET-raamwerk
 - opvolger Classic ASP
- Eigenschappen
 - gebeurtenisgedreven webpagina's (event-driven)
 - uitgebreide interactie met client én server
 - mogelijkheid tot scheiding opmaak en logica (code)
 - Classic ASP: veelal spaghetti-code
 - realisatie logica via volwaardige programmeertalen
 - C#, VB.NET
 - Classic ASP: scripting (VBScript/JScript)
 - eenvoudig in gebruik te nemen

ASP.NET – situering

- Concept 1
 - een ASP.NET-webpagina is opgebouwd uit verschillende besturingselementen (controls) die ondersteuning bieden voor het afhandelen van gebeurtenissen (event handling)
- Concept 2
 - een ASP.NET-webpagina is één groot formulier dat naar zichzelf post
 - postback
 - praktische realisatie event handling uit eerste concept

Inleiding practicum 1 – overzicht

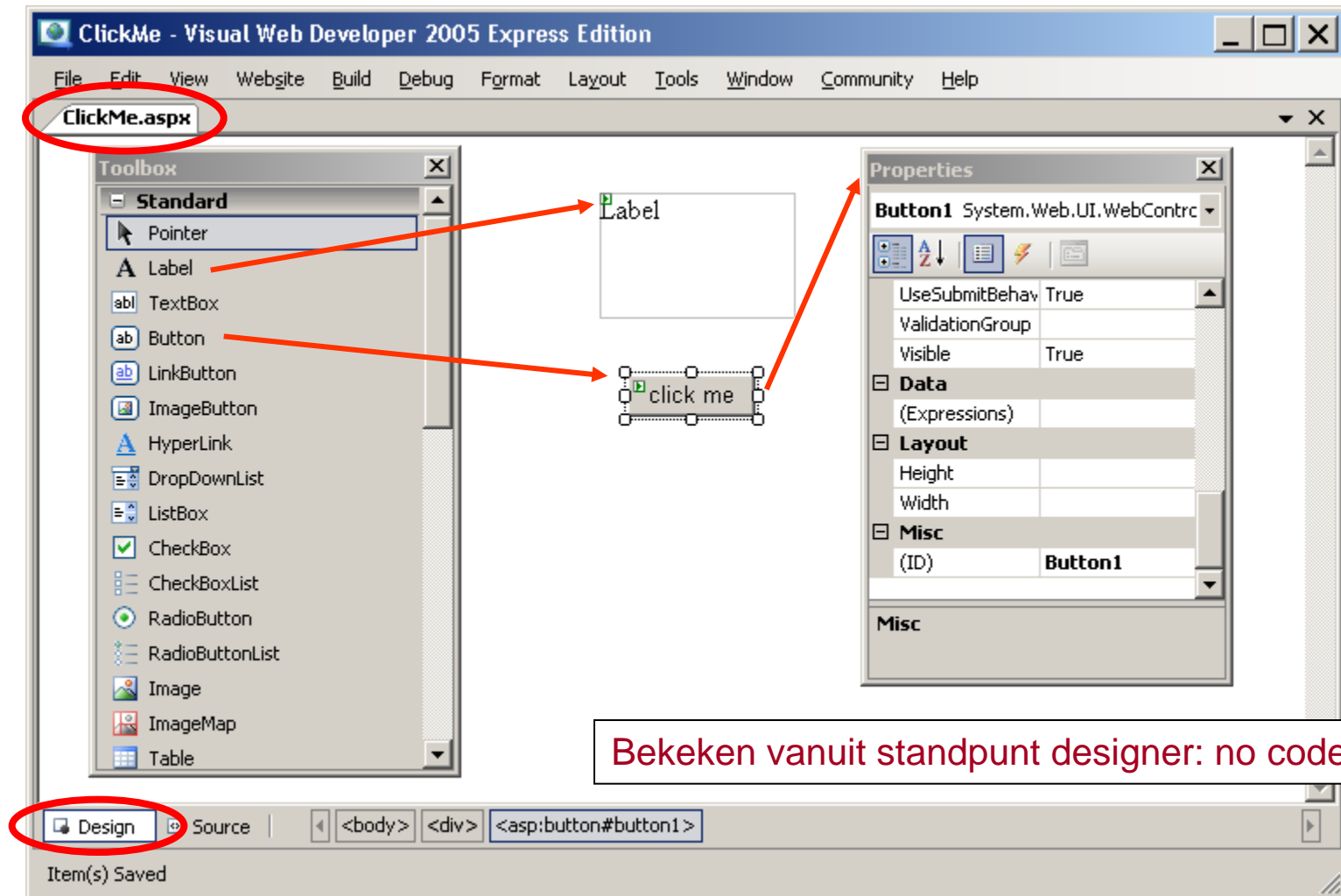
- Statische versus dynamische webpagina's
- .NET-raamwerk
- Programmeertaal C#
- ASP.NET
 - situering
 - besturingselementen en gebeurtenisafhandeling (controls en events)
 - overzicht besturingselementen
 - webpagina's die naar zichzelf posten
 - serverobjecten
- ADO.NET

ASP.NET – controls en events (concept 1)

- Besturingselementen of controls
 - verzorgen opbouw en opmaak webpagina
 - knoppen, tekstvak, keuzelijsten, ...
 - kunnen gebeurtenissen of events genereren bij client
 - indrukken knop, intikken tekst, ...
 - worden meestal doorgestuurd naar server voor verdere afhandeling
 - ook lokale verwerking op client is mogelijk
- Afhandeling gebeurtenissen (events) via “code-behind”
 - extra pagina met code (C#)
 - complementair met opmaakpagina
 - niet noodzakelijk: `<script />`

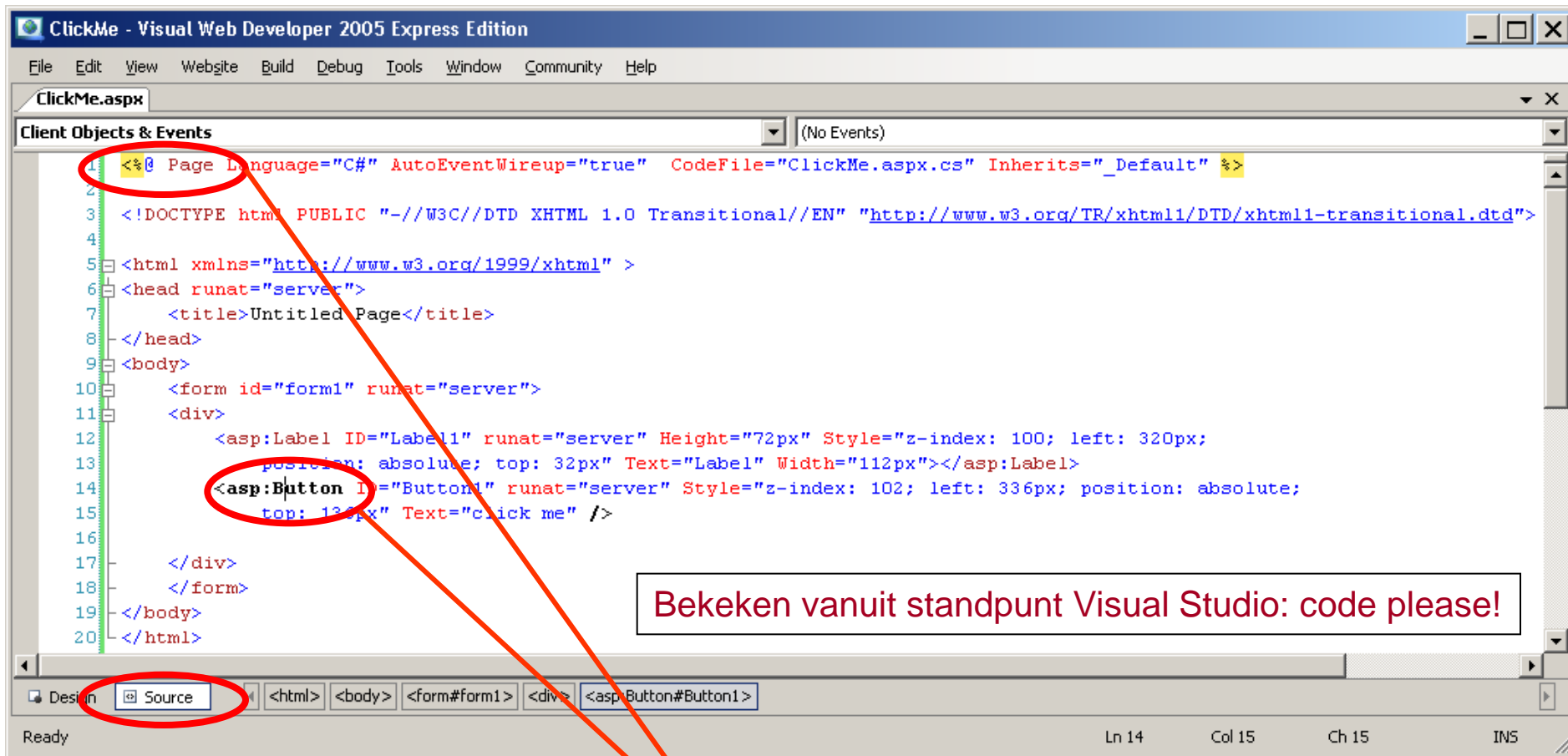
Bemerk scheiding tussen opmaak en code!

ASP.NET – voorbeeld (design view)



Bekeken vanuit standpunt designer: no code please!

ASP.NET – voorbeeld (source view)



ClickMe - Visual Web Developer 2005 Express Edition

File Edit View Website Build Debug Tools Window Community Help

ClickMe.aspx

Client Objects & Events (No Events)

```

1 <%@ Page Language="C#" AutoEventWireup="true" CodeFile="ClickMe.aspx.cs" Inherits="_Default" %>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" >
6 <head runat="server">
7   <title>Untitled Page</title>
8 </head>
9 <body>
10   <form id="form1" runat="server">
11     <div>
12       <asp:Label ID="Label1" runat="server" Height="72px" Style="z-index: 100; left: 320px;
13         position: absolute; top: 32px" Text="Label" Width="112px"></asp:Label>
14       <asp:Button ID="Button1" runat="server" Style="z-index: 102; left: 336px; position: absolute;
15         top: 136px" Text="click me" />
16     </div>
17   </form>
18 </body>
19 </html>

```

Design Source <html> <body> <form#form1> <div> <asp:Button#Button1>

Ready Ln 14 Col 15 Ch 15 INS

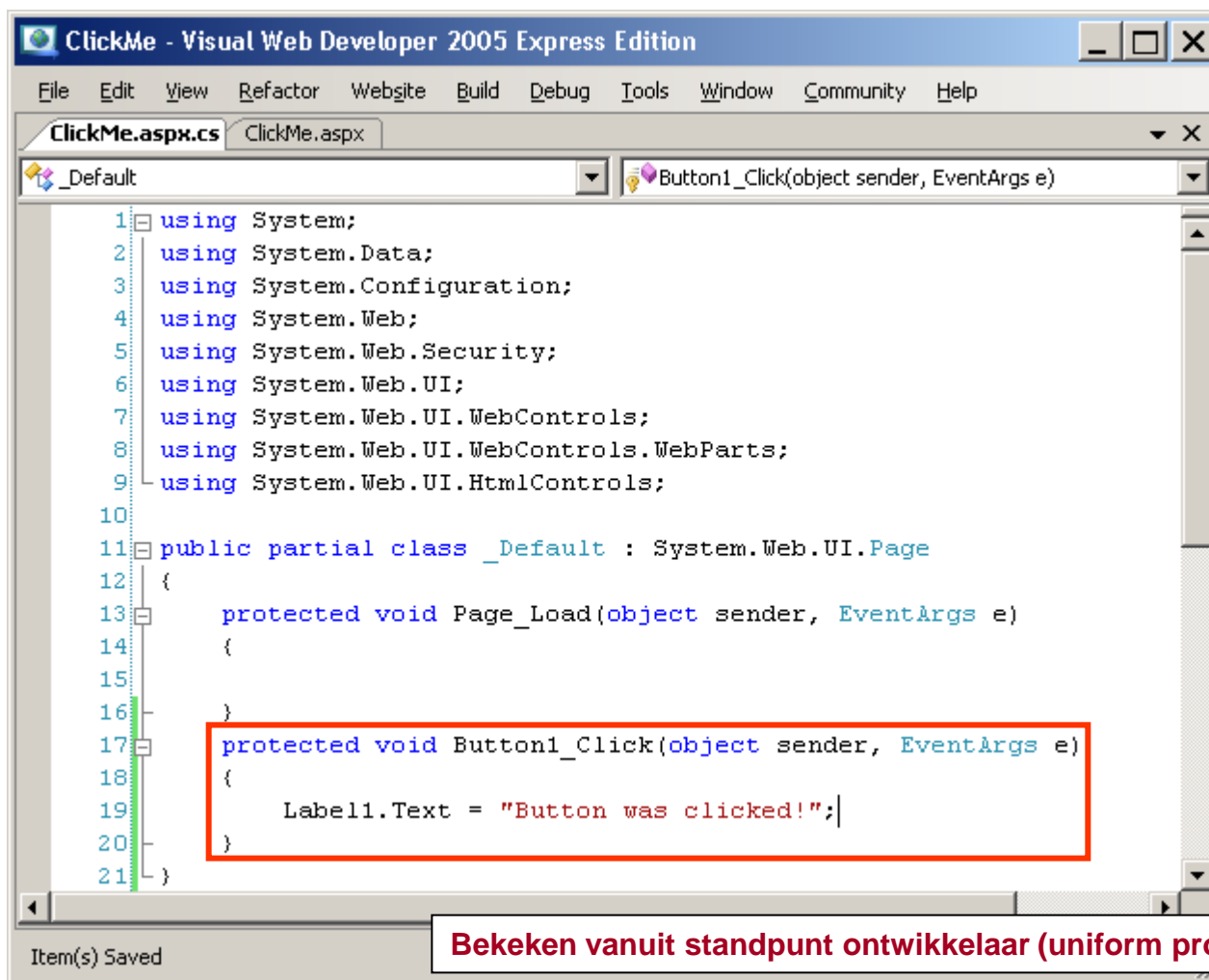
Bekeken vanuit standpunt Visual Studio: code please!

ingebbede verwerkingsinstructies & placeholders

ASP.NET – voorbeeld

- Werkwijze
 - designer verzorgt opmaak en opbouw ASP.NET-webpagina aan de hand van besturingselementen
 - o.a. drag & drop
 - Visual Web Developer genereert pseudo-HTML
 - HTML met verwerkingsinstructies en plaatshouders
- Placeholders voor besturingselementen
 - kunnen gemanipuleerd worden vanuit logica in code-behind
 - worden later automatisch vertaald naar HTML en/of JavaScript

ASP.NET – voorbeeld (code-behind)

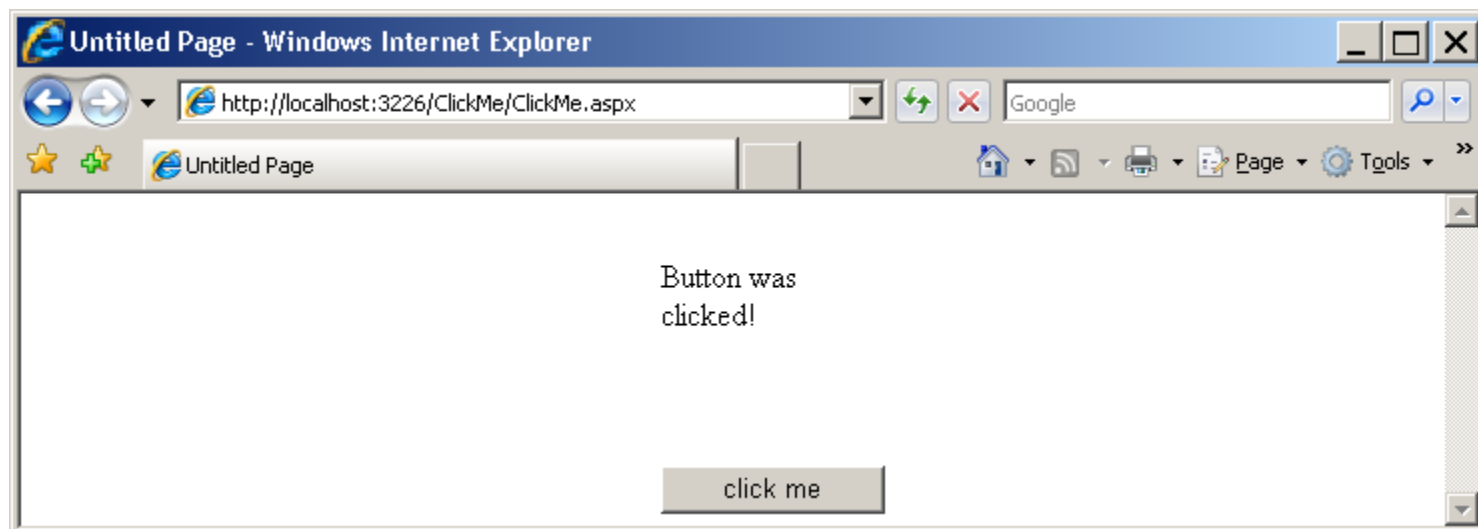
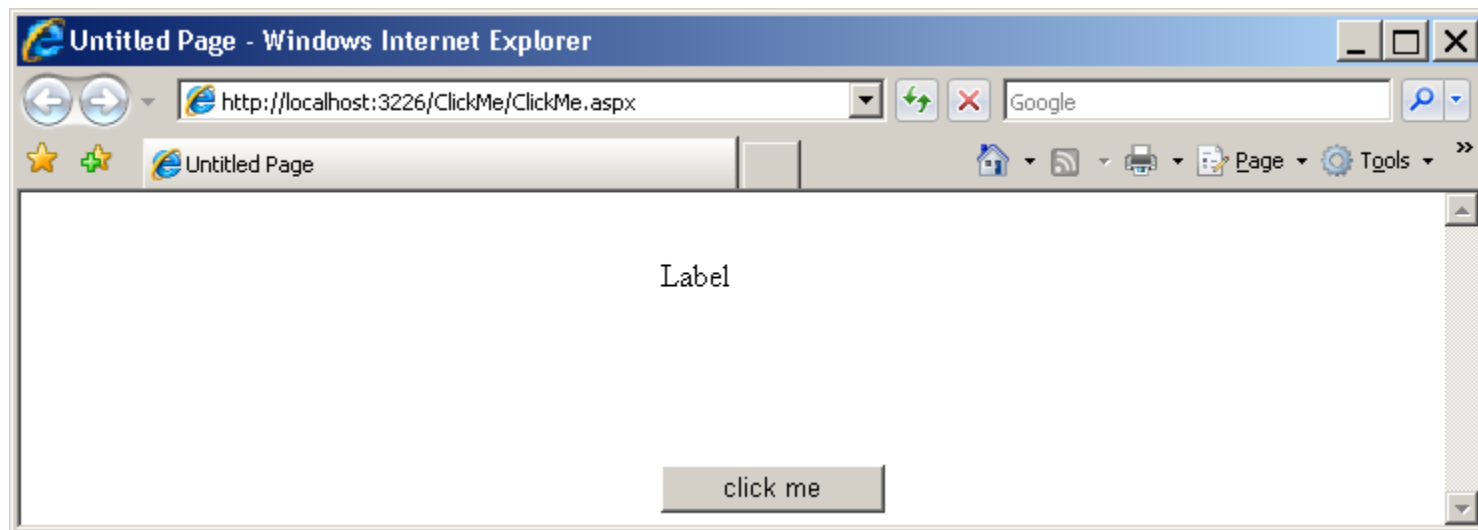


```
ClickMe - Visual Web Developer 2005 Express Edition
File Edit View Refactor Website Build Debug Tools Window Community Help
ClickMe.aspx.cs ClickMe.aspx
_Default Button1_Click(object sender, EventArgs e)
1 using System;
2 using System.Data;
3 using System.Configuration;
4 using System.Web;
5 using System.Web.Security;
6 using System.Web.UI;
7 using System.Web.UI.WebControls;
8 using System.Web.UI.WebControls.WebParts;
9 using System.Web.UI.HtmlControls;
10
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Page_Load(object sender, EventArgs e)
14     {
15     }
16
17     protected void Button1_Click(object sender, EventArgs e)
18     {
19         Label1.Text = "Button was clicked!";
20     }
21 }
```

Item(s) Saved

Bekeken vanuit standpunt ontwikkelaar (uniform programmeermodel).

ASP.NET – voorbeeld (resultaat bij client)



ASP.NET – voorbeeld (HTML-code bij client)

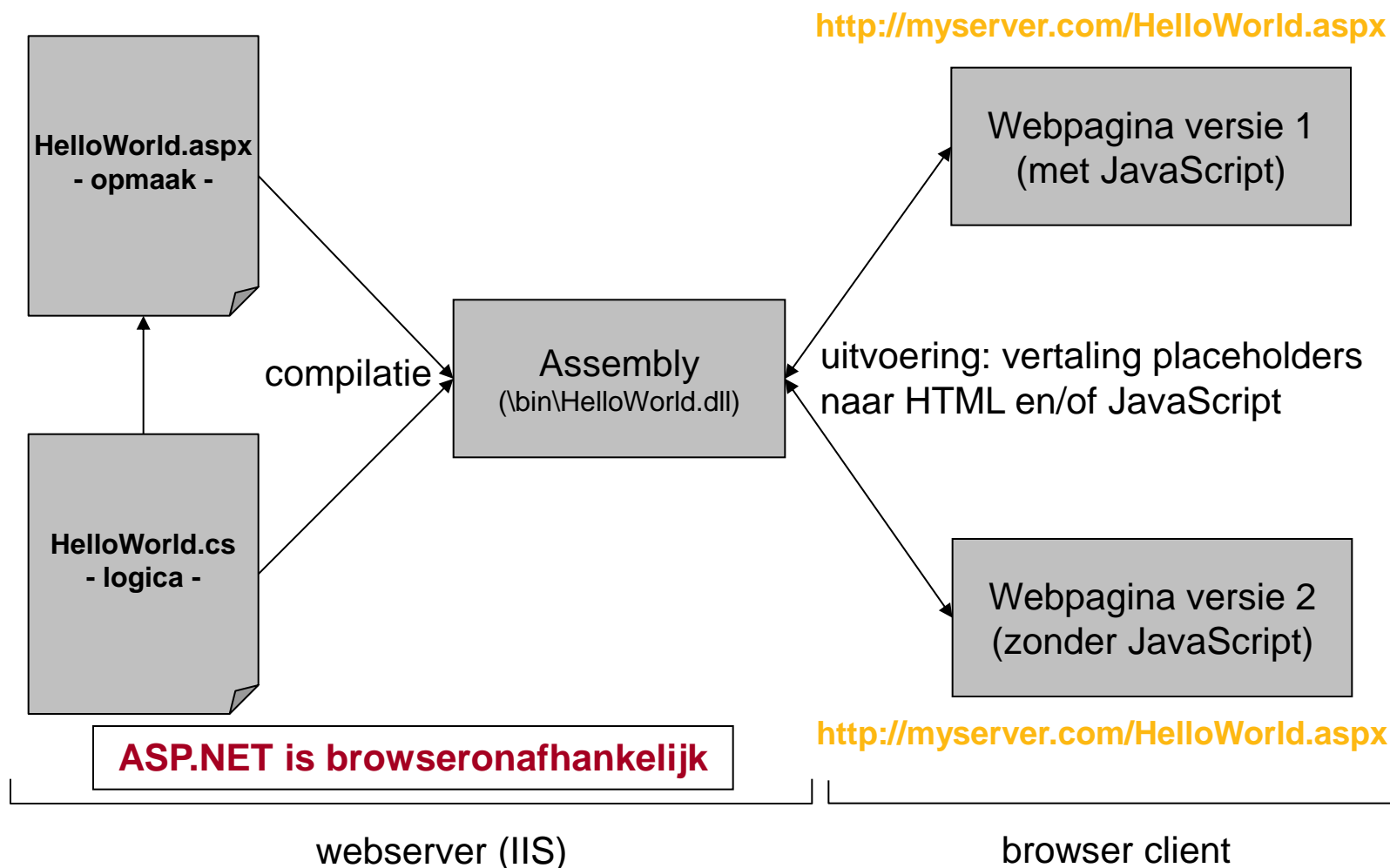
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head><title>
    Untitled Page
</title></head>
<body>
    <form name="form1" method="post" action="ClickMe.aspx" id="form1">
<div>
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPdwwUKLTE2MjY5MTY1NQ9kFgICAw9kFgICAwQ8PFglEeBFRleHQFE0JldHRvbGZlbnRlc2VudCBkaXNjaWEyOGEyaDZlZjE=" />
<div>
        <span id="Label1" style="display:inline-block;height:72px;width:112px;z-index: 100; left: 320px; position: absolute; top: 32px;">Button was clicked!</span>
        <input type="submit" name="Button1" value="click me" id="Button1" style="width:112px;z-index: 102; left: 320px; position: absolute; top: 136px" />
    </div>
</div>
<div>
    <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWAgKV/rOVcQKM54rGBizbC6bUHDnsvyOPReVe1DzEjF" />
</div></form>
</body>
</html>
```

ASP.NET – voorbeeld (besluit)

- Voldoet aan eisen verschillende spelers
 - designer
 - geen code
 - Visual Studio .NET (Visual Web Developer)
 - kan semi-automatisch code genereren met mogelijkheden tot optimalisatie
 - ontwikkelaar
 - kan werken met uniform programmeermodel
 - ervaring die lijkt op die van een krachtige clientapplicatie zoals geschreven in VB/VC++
 - eindgebruikers
 - ontvangen webpagina die (in principe) op maat gemaakt is voor de mogelijkheden van hun terminal

ASP.NET – interactie client/server



Inleiding practicum 1 – overzicht

- Statische versus dynamische webpagina's
- .NET-raamwerk
- Programmeertaal C#
- **ASP.NET**
 - situering
 - besturingselementen en gebeurtenisafhandeling (controls en events)
 - **overzicht besturingselementen**
 - webpagina's die naar zichzelf posten
 - serverobjecten
- ADO.NET

ASP.NET – controls

- Worden toegevoegd aan webpagina
- Drie types
 - statische HTML-opmaakcodes
 - user controls
 - niet belangrijk voor deze practica
 - server controls
 - HTML-controls
 - webcontrols of ASP.NET-controls

ASP.NET – controls

- Statische HTML-opmaakcodes
 - standaard-HTML-elementen
 - `<p>...</p>`
 - `<table>...</table>`
 - worden naar browser client doorgestuurd zoals vermeld in het bronbestand
 - letterlijke besturingselementen (geen vertaling)
- User controls
 - hergebruik van presentatiecode
 - HTML en/of ASP.NET-code
 - vergelijkbaar met include-bestanden (in Classic ASP)
 - code in bestand met extensie .ascx
 - niet gebruikt in deze practica

ASP.NET – controls

- HTML-controls
 - standaard HTML-elementen maar uitgebreid met twee attributen
 - id-attribuut: kan gerefereerd worden vanuit code
 - runat="server"-attribuut: verwerking op server
 - voorbeeld
 - `<p id="tekst" runat="server">Hallo</p>`
 - beschikbaar
 - `System.Web.UI.HtmlControls`
- Webcontrols of ASP.NET-controls
 - geen rechtstreekse mapping op HTML-elementen
 - rijkere functionaliteit
 - voorbeeld: Calendar
 - beschikbaar
 - `System.Web.UI.WebControls`

ASP.NET – webcontrols

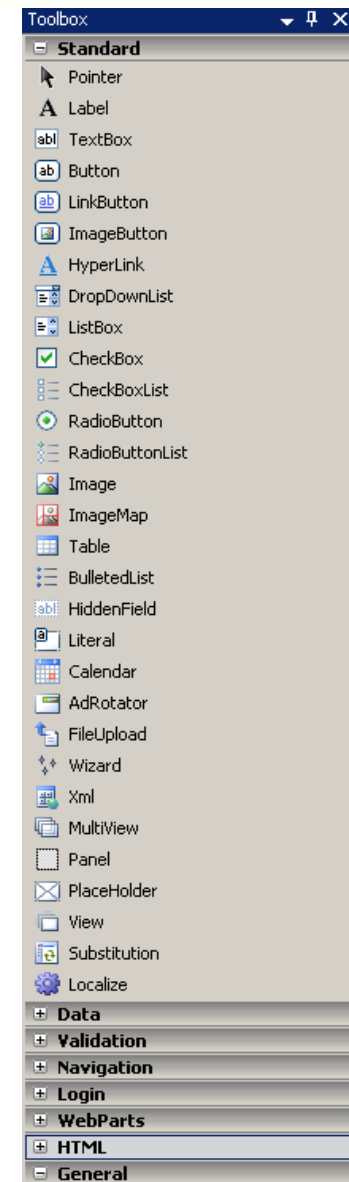
- Eigenschappen Webcontrols
 - syntax
 - `<asp:label id="myLabel" text="Hallo" runat="server" />`
 - omvat controls zoals beschikbaar in VB/VC++
 - label, textbox, dropdownbox, button, link, ...
 - omvat nieuwe controls met rijkere functionaliteit
 - Calendar, AdRotator
 - consistent met het .NET-programmeermodel
 - ondergebracht in eigen naamruimte ("asp")
 - afgebeeld op objecten
 - bijvoorbeeld: attributen corresponderen met properties
 - (!) browser op client ziet deze controls of placeholders nooit
 - worden door server vertaald naar HTML-code en/of JavaScript

ASP.NET – webcontrols

- Voor volledigheid
 - alle webcontrols erven over van gemeenschappelijke interface
 - `System.Web.UI.Control.Webcontrol`
 - uitzondering: Repeater-control
 - veel gemeenschappelijke eigenschappen
 - AccessKey, BackColor, BorderColor, BorderStyle, BorderWidth, CausesValidation, CssClass, Enabled, Font, ForeColor, Height, ID, ToolTip, Visible, Width

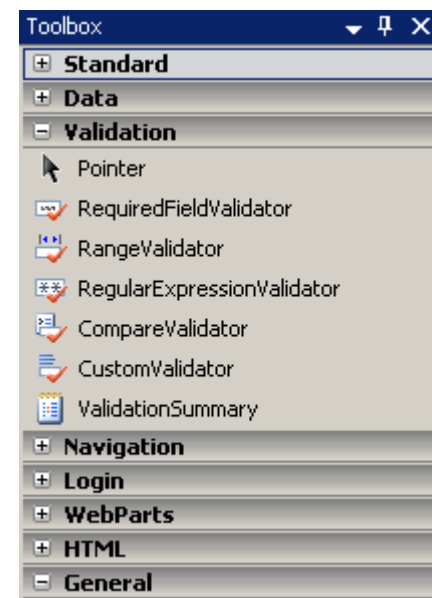
ASP.NET – overzicht controls

- Zie “toolbox” in ontwikkelomgeving
- Gedetailleerde uitleg en voorbeelden via ‘Help’ (F1)
- Enkele veel gebruikte controls
 - Label, TextBox, Button, Hyperlink
 - CheckBox, RadioButton
 - CheckBoxList, RadioButtonList, ListBox, DropDownList
 - Table, TableRow, TableHeaderCell, TableCell
 - Panel
 - bevat andere besturingselementen (container)
 - nuttig voor zichtbaarheid en opmaak v/e groep controls
 - vereenvoudigt dynamisch toevoegen van controls
 - Image, AdRotator, Calendar



ASP.NET – overzicht controls

- Webcontrols voor validatie
 - verschillende soorten validatie op andere controls
 - client-side: indien mogelijk
 - server-side: altijd
 - voorbeelden
 - RequiredFieldValidator: veld moet ingevuld zijn
 - CompareValidator: twee velden vergelijken
 - RangeValidator: veldwaarde in bepaald bereik
 - RegularExpressionValidator: veldwaarde voldoet aan een gegeven reguliere expressie
 - CustomValidator: zelfgemaakt validatiescript
 - ValidationSummary: samenvatting resultaat validatie



Inleiding practicum 1 – overzicht

- Statische versus dynamische webpagina's
- .NET-raamwerk
- Programmeertaal C#
- **ASP.NET**
 - situering
 - besturingselementen en gebeurtenisafhandeling (controls en events)
 - overzicht besturingselementen
 - webpagina's die naar zichzelf posten
 - serverobjecten
- ADO.NET

ASP.NET –PostBack (concept 2)

- Term “postback”
 - beschrijft mogelijkheid van een ASP.NET-webpagina om een HTTP-verzoek naar zichzelf te versturen
 - bijvoorbeeld: reactie op klik van de gebruiker op een knop
 - realisatie via een formulier
 - verstuurt HTTP-verzoek van het type POST
- Nut postback
 - maakt afhandelen gebeurtenissen mogelijk op server
 - voorbeeld: invullen postcode toont lijst gemeenten
 - maakt toestandsbewuste webpagina's mogelijk
 - controls in webpagina onthouden hun toestand en kunnen deze doorgeven aan de server
- Controle op postback via property `IsPostBack`

Inleiding practicum 1 – overzicht

- Statische versus dynamische webpagina's
- .NET-raamwerk
- Programmeertaal C#
- **ASP.NET**
 - situering
 - besturingselementen en gebeurtenisafhandeling (controls en events)
 - overzicht besturingselementen
 - webpagina's die naar zichzelf posten
 - serverobjecten
- ADO.NET

ASP.NET – serverobjecten

Object	Omschrijving
Response	gebruikt om "output" van de server naar de client te zenden (tekst toevoegen, cookies zetten, ...)
Request	toegang tot informatie van de client (cookies, certificaten, querystring, formulier, ...)
Server	informatie over de webserver
Application	toegang tot methoden op applicatieniveau (i.e., de website) en gemeenschappelijke data over sessies
Session	informatie over de huidige gebruikerssessie
Trace	debuginformatie en feedback

ASP.NET – Session-object

- Webserver volgt verschillende gebruikers
 - HTTP-protocol is toestandeloos
 - HTTP heeft geen methode waarmee webserver kan vaststellen of reeks verzoeken van dezelfde gebruiker afkomstig is
 - gebruikersidentificatie
 - via cookies ("session cookies")
 - via QueryString-parameters
- Elke gebruiker heeft juist één sessie-object
- Levensduur
 - creatie wanneer gebruiker eerste pagina opvraagt
 - vernietigd 20min (default) nadat laatste pagina werd opgevraagd
- Nut
 - bijhouden informatie over gebruiker (sessiestatus)
 - sleutel/waarde-paren
 - informatie doorgegeven tussen opeenvolgende pagina's

ASP.NET – Session-object

- Syntax
 - set
 - `Session[key] = value`
 - get
 - `value = Session[key]`
 - opmerking
 - key is een string of integer (index)
 - value is een object
 - eventueel nood aan 'casting' bij ophalen van Sessie-informatie (get)
- Voorbeeld
 - `Session["voornaam"] = txtVoornaam.Text;`
`txtVoornaam.Text = Session["voornaam"];`

ASP.NET – Application-object

- **Situering**
 - één applicatie-object voor alle gebruikers
- **Levensduur**
 - wordt aangemaakt zodra de eerste pagina van een website wordt opgevraagd
 - website: bestanden in application folder op IIS
 - wordt vernietigd wanneer de applicatie stopt
 - lees: wanneer server stopt
- **Nut**
 - manier om informatie tussen sessies uit te wisselen
 - applicatiestatus: totaal van gegevens die gedeeld worden tussen verschillende gebruikers van een webapplicatie
 - globale variabelen

ASP.NET – Application-object

- Syntax
 - set
 - Application[key] = value
 - get
 - value = Application[key]
 - belangrijk
 - slechts één gebruiker tegelijkertijd toegang geven
 - Lock() en UnLock() van het Applicatie-object
- Voorbeeld
 - Application.Lock(); // Thread safe werken.
Application["voornaam"] = txtVoornaam.Text;
Application.Unlock();

ASP.NET – Response-object

- Object om informatie naar de client te sturen
 - Response.Write(): uitschrijven van tekst
 - Response.Redirect(): stuurt bezoeker naar andere pagina
 - Response.OutputStream: binaire output wegschrijven in HTTP body
- Voorbeeld
 - ```
private void Page_Load(// ...)
{
 Response.Write("U wordt dadelijk doorverwezen...");
 // Stuur tekst naar client.
 Response.Flush();
 // Client doorsturen.
 Response.Redirect("index.html");
}
```

# ASP.NET – Request-object

- Object om informatie te ontvangen van de client
  - cookies of informatie toegevoegd aan URL
    - <http://multimedialab.elis.ugent.be/hello.aspx?naam=leonardo>
  - informatie ingevoerd via invulformulieren
- Syntax voor informatie toegevoegd aan URL
  - get
    - `value = Request.QueryString[key]`
      - key: string of integer (index)
      - value: string
  - set
    - niet mogelijk
  - Voorbeeld
    - `string naam = (string)Request.QueryString["naam"];`
    - `string naam = (string)Request.Form["naam"];`

# Inleiding practicum 1 – overzicht

- Statische versus dynamische webpagina's
- .NET-raamwerk
- Programmeertaal C#
- ASP.NET
  - situering
  - besturingselementen en gebeurtenisafhandeling (controls en events)
  - overzicht besturingselementen
  - webpagina's die naar zichzelf posten
  - serverobjecten
- ADO.NET

# ADO.NET

- Sleutelprobleem in webapplicaties
  - opslag en ophalen van gegevens over de tijd heen
    - forum
    - e-commerce (webwinkel)
    - rapportering (monitoring van systemen)
- ADO.NET (ActiveX Data Objects)
  - verzameling van .NET-klassen voor gegevenstoegang
    - relationele databanken: SQL Server, Oracle, ...
    - uitgebreide ondersteuning voor XML
    - naamruimte: `System.Data`
  - eenvoudige koppeling met ASP.NET
    - ASP.NET-besturingselementen kunnen integreren met ADO.NET via data binding

# ADO.NET – databanktoegang

- Microsoft SQL Server 2005 voor practica
  - volwaardige en krachtige relationele databank
  - draait op de server vstudent.elis.ugent.be
  - is *remote* toegankelijk
    - UGentNet
  - bevraging via SQL (*Structured Query Language*)
- Praktisch
  - elke groep heeft een eigen databank (itechX)
  - elke databank is beveiligd met een paswoord
    - het paswoord zal in het forum van iedere groep gepost worden
  - reeds 1 tabel aanwezig ("MuziekCataloog")

# ADO.NET – databanktoegang

- Databank bekijken (browsen)
  - via ASP.NET, ADO.NET, en SQL-commando's
    - zie later in de presentatie
  - grafisch via Visual Web Developer 2008 Express
    - geen "design"-functionaliteit
      - wél in VS .NET 2008
    - selecteer "Server Explorer" via menu "view"
    - rechtermuisklik op "Data Connections" en klik "Add Connection"
    - selecteer Microsoft SQL Server
    - connectiegegevens ingeven
    - paswoord ingeven
    - in venster "database explorer" rechtermuisklik op tabel en selecteer "show table data"

# ADO.NET – databanktoegang

The screenshot illustrates the steps to configure a database connection in SQL Server Enterprise Manager:

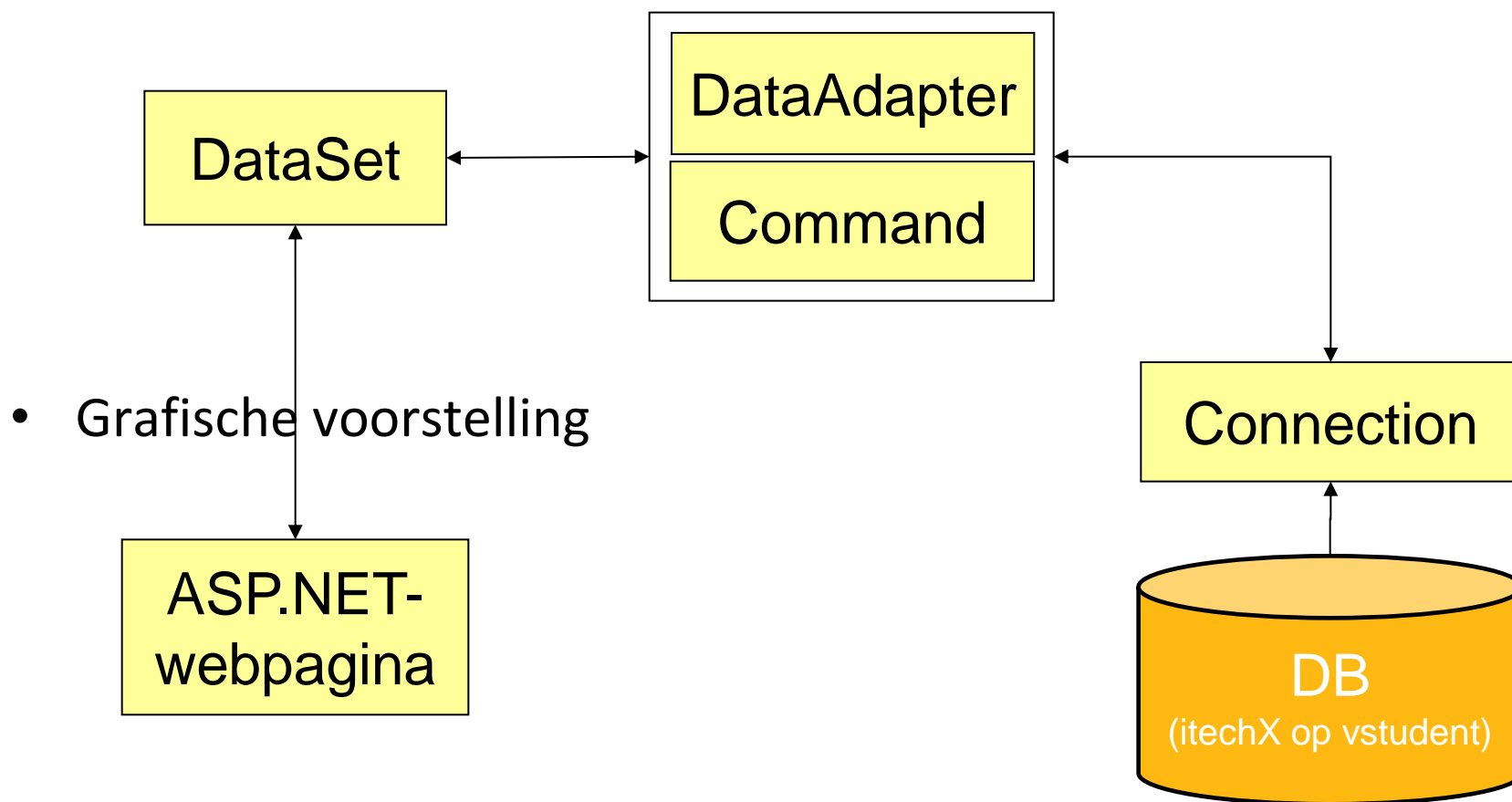
- Database Explorer:** Shows the 'Data Connections' folder expanded, with 'vstudent\students.itechX.dbo' selected. The 'Tables' folder is also expanded, showing 'MuziekCatalog' and its tables: 'id', 'Tracknr', 'Titel', 'Uitvoerder', 'Album', and 'Jaar'.
- Add Connection Dialog:**
  - Data source:** 'Microsoft SQL Server (SqlClient)'.
  - Server name:** 'vstudent.elis.ugent.be' (highlighted with a red circle).
  - Log on to the server:**
    - ☒ Use SQL Server Authentication.
    - User name:** 'itechX' (highlighted with a red circle).
    - Password:** '\*\*\*\*\*' (highlighted with a red circle).
    - ☐ Save my password.
  - Connect to a database:**
    - ☒ Select or enter a database name: 'itechX' (highlighted with a red circle).
    - ☐ Attach a database file: (with a 'Browse...' button).
  - Buttons:** 'Test Connection', 'OK', 'Cancel', and 'Advanced...'.
- Choose Data Source Dialog:**
  - Data source:** 'Microsoft SQL Server' (highlighted with a red circle).
  - Data provider:** '.NET Framework Data Provider for SQL Server'.
  - ☒ Always use this selection.
  - Buttons:** 'Continue' and 'Cancel'.



# ADO.NET – objectmodel

- Toegang tot databank via een aantal objecten
  - Connection Object
    - bevat info over connectie met databank
  - DataSet Object
    - tijdelijke opslag en manipulatie gegevens (uit databank)
  - DataAdapter Object
    - verbinding tussen DataSet en de databank
    - opvullen van een DataSet met gegevens
  - Command Object
    - uitvoeren SQL-opdracht op bepaalde gegevensbron
      - commando's sturen naar databank
- Elk van deze objecten beschikt in het .NET-raamwerk over 2 varianten
  - een generieke (elke OLE-compatibele databank: OleDbConnection,...)
  - een SQL Server-specifieke (SqlConnection,...)

# ADO.NET – objectmodel



# ADO.NET – objectmodel

- DataAdapter
  - verbinding tussen DataSet en databank
- DataSet
  - gebruikt om informatie uit gegevensbron (databank, XML-document) op te halen of bij te werken
  - een wrapper voor een onderliggend XML-document
    - structuur en beperkingen beschreven via XML Schema
  - in praktijk: vaak gedisconnecteerde representatie van databank
    - formaatagnostisch
    - laat bewerkingen toe in geheugen
    - kan synchroniseren met databank via DataAdapter
    - kan serialiseren naar XML-document

# ADO.NET – objectmodel

- DataSet-object verstrekt toegang tot andere objecten
  - DataTable
    - tabel met gegevens
    - DataSet kan meerdere DataTable-objecten bevatten
      - opgeslagen als array
  - DataRow
    - elke DataTable in DataSet bevat verzameling van nul of meerdere DataRow-objecten
      - opgeslagen als array
  - DataColumn

# ADO.NET – SQL

- SQL = Structured Query Language
  - ISO/IEC 9075
  - ISO/IEC 9075-1:1999 ondersteund in SQL Server 2005
- oefeningen vereisen enkel eenvoudige statements
  - `SELECT * FROM databank`
  - `SELECT * FROM databank WHERE kolom1 = 'waarde1' AND kolom2 = 'waarde2'`
  - `INSERT INTO data(kolom1, kolom2) VALUES ('waarde1', 'waarde2')`

# ADO.NET – voorbeeldcode

- Connection string
  - spreekt stuurprogramma voor databank aan
- Voorbeeld
  - // Bouw de connectiestring op.  
`string connectionString = "Server=VSTUDENT.ELIS.UGENT.BE;Integrated Security=false;Database=itechX;user id=itechX;password=pass";`
- Mogelijk veiligheidslek
  - paswoord is onderdeel van de connection string
  - zorg er altijd voor dat bronbestanden waarin het paswoord voorkomt voldoende zijn afgeschermd

# ADO.NET – voorbeeldcode

- DataAdapter en DataSet

Namespace System.Data.SqlClient

- Voorbeeld

```
// Bouw SQL-statement op
string getData = "SELECT * FROM data";
// Creëer DataAdapter, maak koppeling met databank
SqlDataAdapter dataAdapter =
new SqlDataAdapter(getData, connectionString);
// Maak een nieuw DataSet-object aan.
DataSet dataSet = new DataSet();
// Vul het DataSet-object op met gegevens.
dataAdapter.Fill(dataSet);
```

# ADO.NET – voorbeeldcode

- DataSet en DataTable

- Voorbeeld

```
// Haal de tabel op.
```

```
DataTable dataTable = dataSet.Tables[0];
```

```
if (dataTable.Rows.Count > 0)
```

```
{
```

```
 // Verwerking van de data.
```

```
}
```

- Algemene opmerking
  - toegang tot en weergeven van databankgegevens kan ook via de grafische interface van Visual Studio



## ADO.NET – voorbeeldcode

- Command

- Voorbeeld

// SQL-statement om een record toe te voegen aan de tabel 'data'

```
string strSQL = "INSERT INTO data(kolom1, kolom2)
VALUES ('waarde1', 'waarde2')";
```

// Creëer SqlConnection- en SqlCommand-object.

```
SqlConnection myConn = new SqlConnection(connectionString);
```

```
SqlCommand myCmd = new SqlCommand(strSQL, myConn);
```

```
myConn.Open(); // Open connectie.
```

```
myCmd.ExecuteNonQuery(); // Uitvoeren SQL-statement.
```

```
myConn.Close(); // Sluiten connectie.
```

# Overzicht

- Algemene inleiding
  - inleiding
  - overzicht practica
  - praktische afspraken
- Inleiding practicum 1
  - statische versus dynamische webpagina's
  - .NET-raamwerk
  - Programmeertaal C#
  - ASP.NET
  - ADO.NET
- Werken met Visual Web Developer 2008 Express
- Opgave practicum 1

# Ontwikkelen in .NET

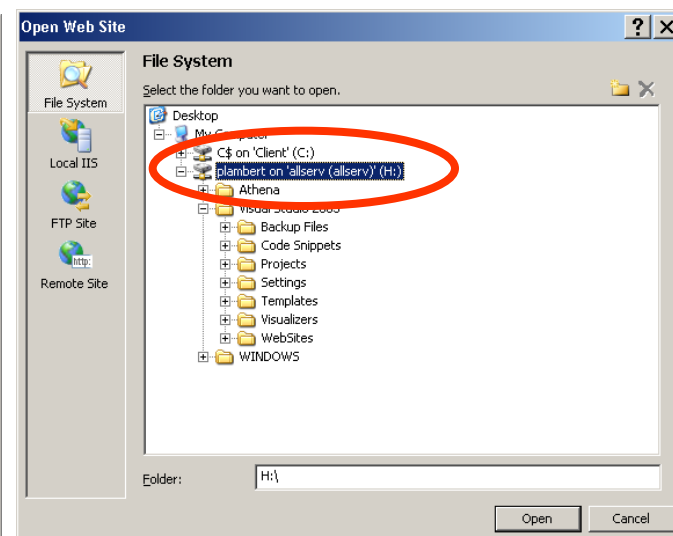
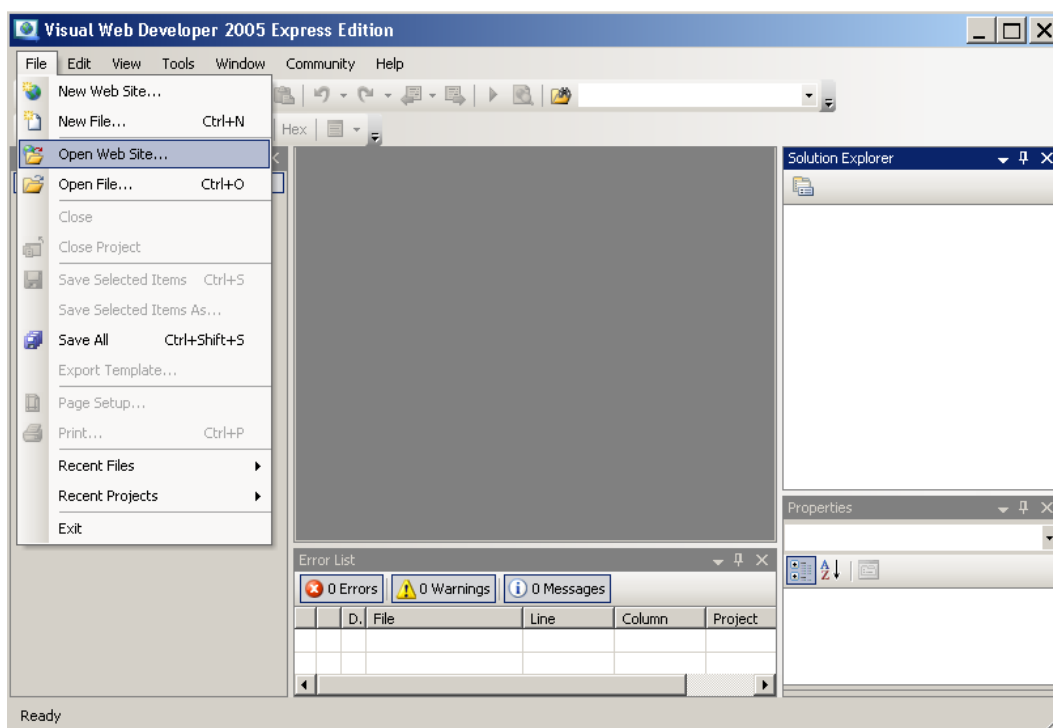
- Ontwikkeling in favoriete editor (kladblok, vi, ...)
  - compilers zijn gratis beschikbaar
    - commandline
  - tijdrovend
- Ontwikkeling in Visual Web Developer 2008 Express
  - Integrated Development Environment (IDE)
    - syntax highlighting
    - debugging
    - documentatie
  - specifiek gericht op het ontwikkelen van webapplicaties
    - bv. geïntegreerde webserver
  - interface is dezelfde als Visual Studio 2008
    - populair in industrie

# Werken met Visual Web Developer 2008 Express

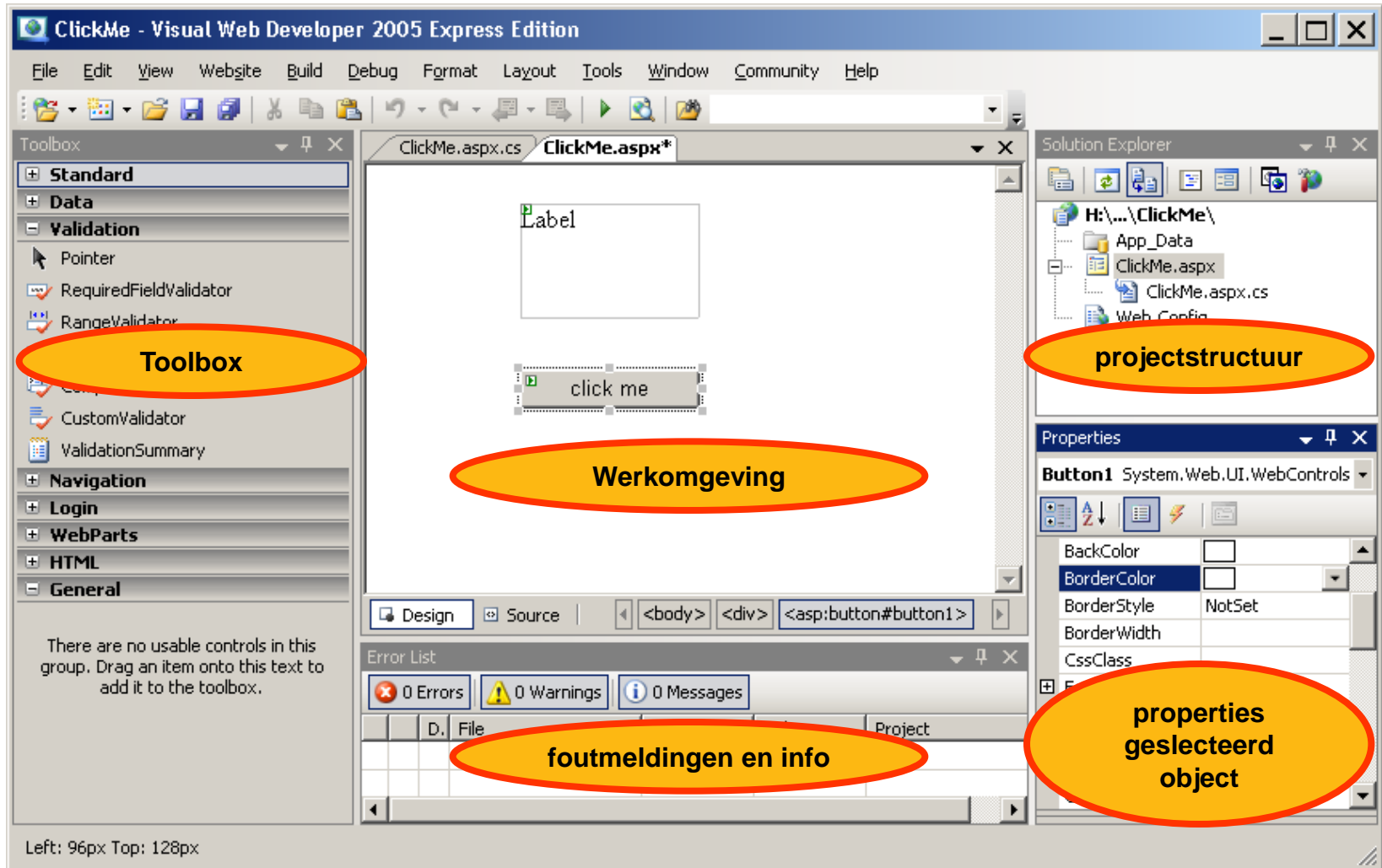
- Visual Web Developer 2008 Express
  - beschikbaar via Athena
    - <http://athena.ugent.be>
    - verleen toegang tot lokale schijven met Citrix
  - plaats de projectfolder op “H:” (project en broncode)
    - gekoppeld aan UGent-account
    - bereikbaar via Citrix-omgeving (Athena)
    - mount eventueel deze drive op lokale PC
      - zie ook volgende slide
    - niet nodig indien volledig lokaal gewerkt wordt

# Werken met Visual Web Developer 2008 Express

- Mounten van de “H:”-schijf
  - normaalgezien: samba1.ugent.be
  - de precieze locatie kan als volgt terug gevonden worden

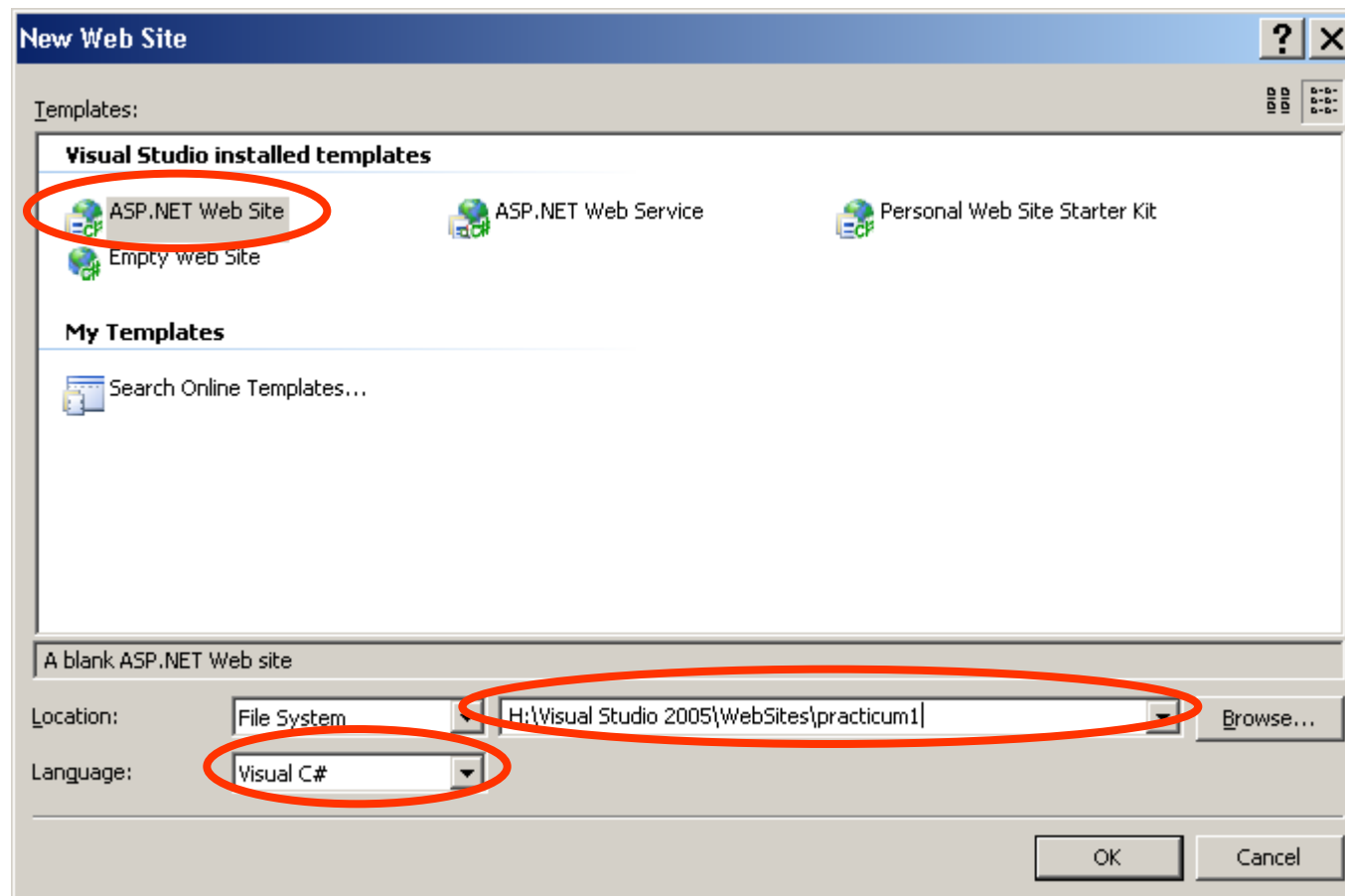


# Werken met Visual Web Developer 2008 Express



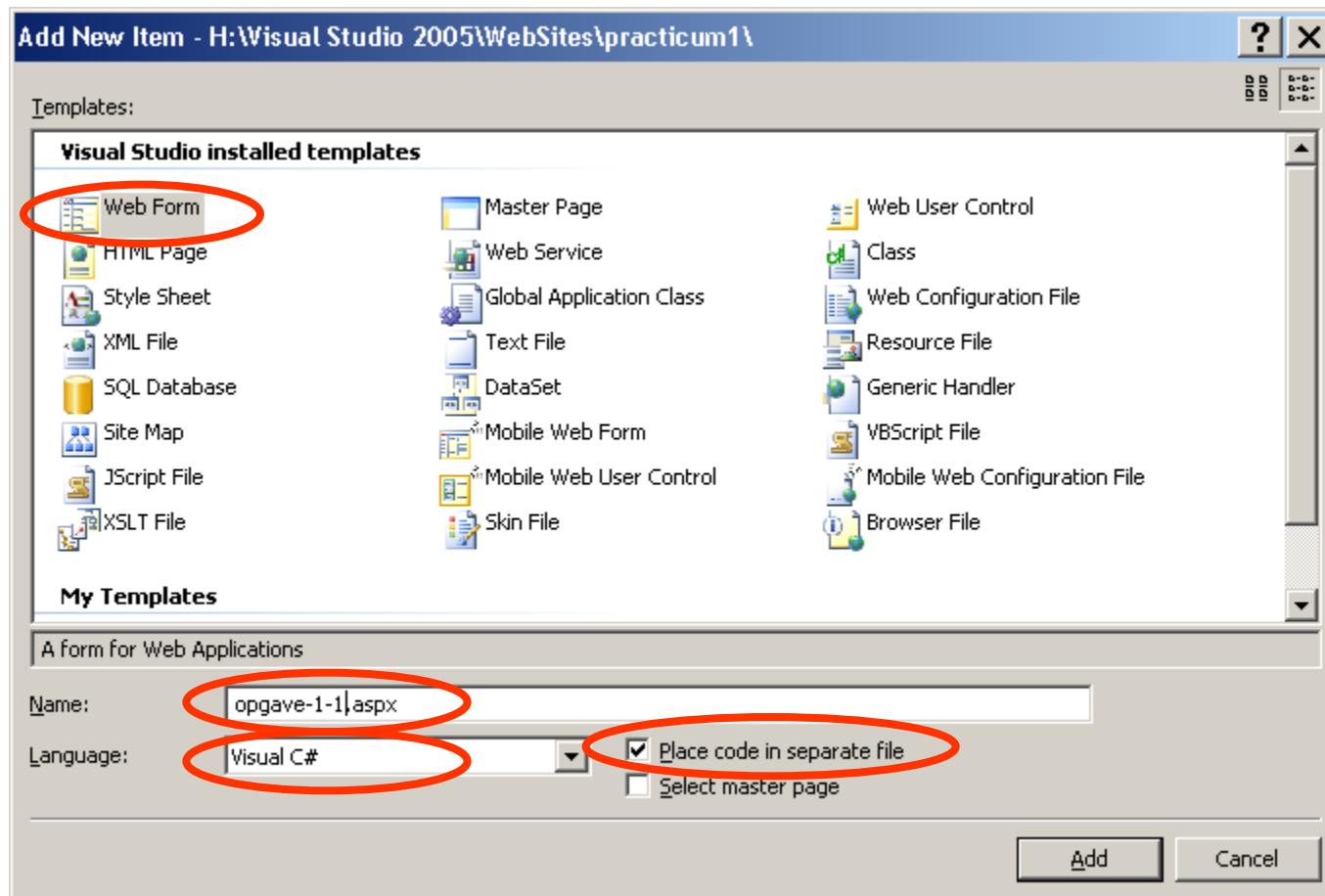
# Werken met Visual Web Developer 2008 Express

- Een nieuwe webapplicatie starten (File -> New Web Site)



# Werken met Visual Web Developer 2008 Express

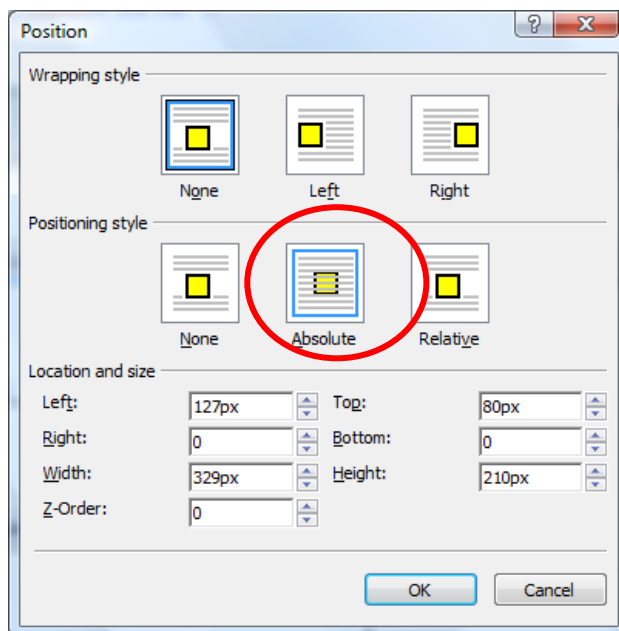
- Een pagina toevoegen (Website -> Add New Item)





# Werken met Visual Web Developer 2008 Express

- Absolute positionering van controls
  - Format -> position



- Opgelet: dit werkt niet voor <span> en <a> gebaseerde controls. Hiervoor moet style="display: block" worden toegevoegd (of moet direct style="position: absolute" worden toegevoegd)

<http://blogs.msdn.com/b/webdev/archive/2008/03/11/absolute-and-relative-positioning-in-visual-web-developer-2008-designer.aspx>

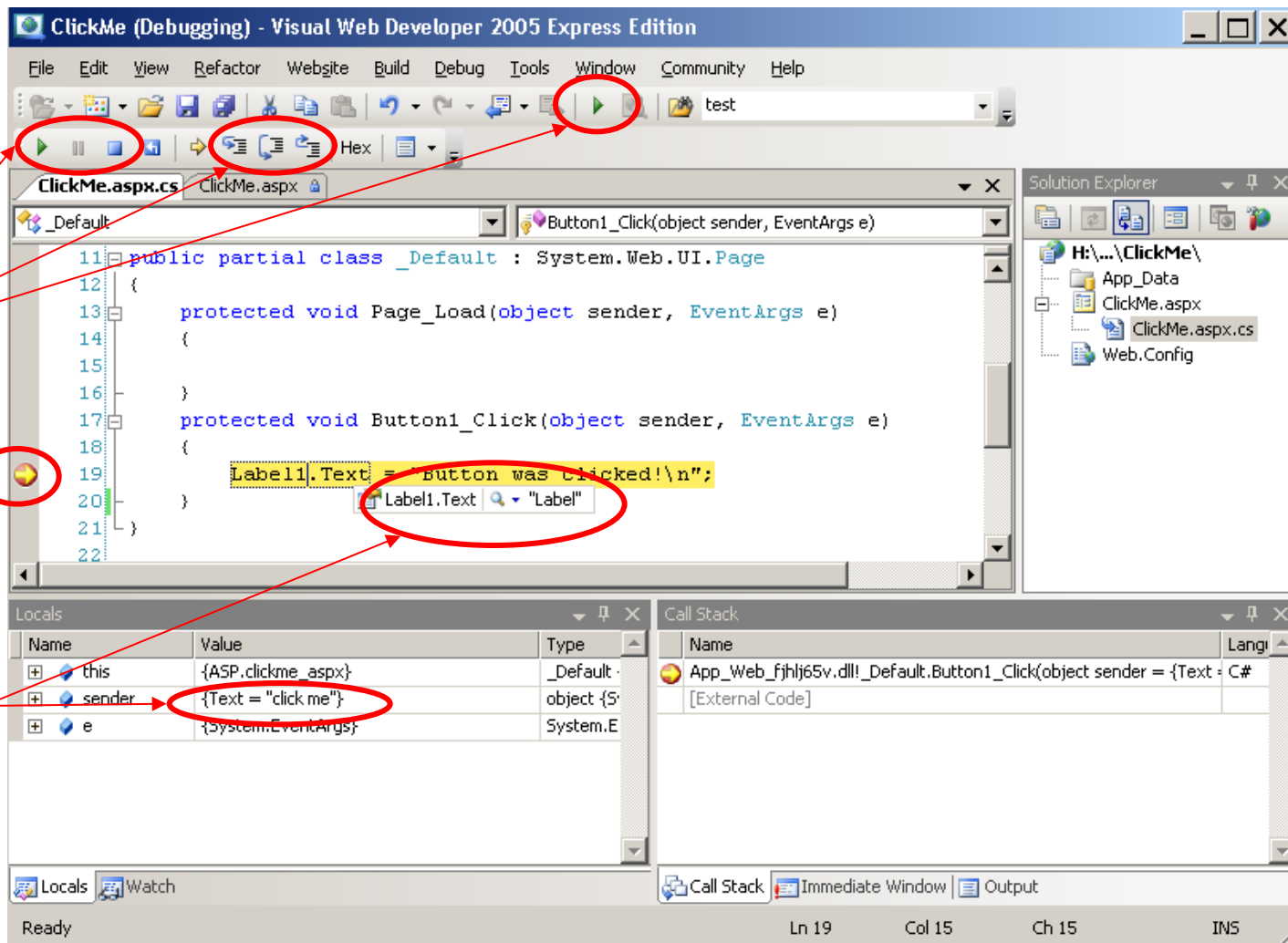
# Werken met Visual Web Developer 2008 Express

- Debuggen

debuggen

breakpoints

debug-informatie

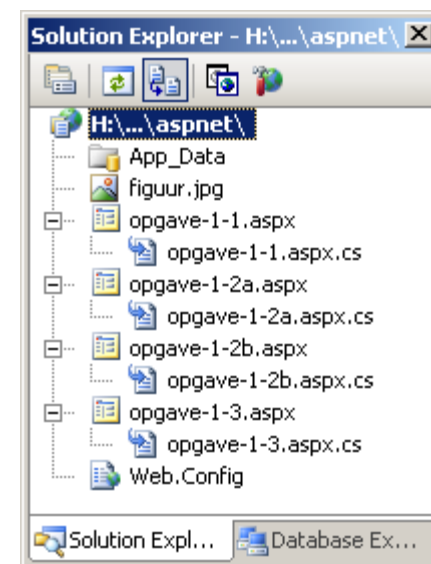


# Overzicht

- Algemene inleiding
  - inleiding
  - overzicht practica
  - praktische afspraken
- Inleiding practicum 1
  - statische versus dynamische webpagina's
  - .NET-raamwerk
  - Programmeertaal C#
  - ASP.NET
  - ADO.NET
- Werken met Visual Web Developer 2008 Express
- Opgave practicum 1

# Opgave practicum 1

- Opgave
  - 3 oefeningen
  - Minerva > Documenten > Practica:
    - ITech\_-\_opgave\_practicum\_1\_-\_ASP.NET.pdf
- Indienen via Minerva “Dropbox”
  - Versturen naar:
    - Peter Lambert **en** Pieterjan De Potter
    - **NIET** naar ‘iedereen’
  - correcte bestandsnamen (!), zie opgave
  - **aspnet\_groep[x].zip**
- Oplossingen maken
  - PC-klas ‘Konrad Zuse’



# Opgave practicum 1

- Tips
  - gebruik de help-functie van de ontwikkelomgeving
    - druk “F1” na het selecteren van een control, veld, ...
  - gebruik de online documentatie
    - <http://msdn2.microsoft.com/en-us/asp.net/default.aspx>
    - wordt ook gebruikt in de ontwikkelomgeving
  - Gebruik de debugger
  - neem opmerkingen in opgave door!