

Practicum 1

ASP.NET en ADO.NET

De bedoeling van dit practicum is vertrouwd te worden met ASP.NET (Active Server Pages) en de programmeertaal C# (spreek uit als C sharp). ASP.NET is een technologie die het mogelijk maakt om op een eenvoudige manier dynamische webpagina's te maken. Om dit te realiseren kan men onder andere gebruikmaken van C#, een volwaardige programmeertaal. Dit laatste staat in contrast met ASP Classic waar men enkel gebruik kan maken van scripttalen zoals VBScript en JScript.

In het kader van dit practicum zal in eerste instantie aandacht besteed worden aan het gebruik van ASP.NET Controls en C#. Daarna komen enkele specifieke eigenschappen van ASP.NET aan bod, zoals het uitwisselen van gegevens tussen verschillende pagina's.

Als laatste komt ADO.NET aan bod: een collectie van klassen die onderdeel zijn van het .NET-raamwerk die het mogelijk maakt om op een eenvoudige manier databanken aan te spreken vanuit webpagina's. Zodoende is het mogelijk om losgekoppelde n-lagige webapplicaties te implementeren.

Deadline

21 maart 2012, 14u00

Opmerkingen en tips

1. Wat dient er ingediend te worden op Minerva in "aspnet_groep[x].zip" voor het practicum over ASP.NET en ADO.NET?

Eenvoudig gezegd: alles. Dit betekent dat alle bestanden binnen de solution-folder gezipd moeten worden.

2. Persistent houden van variabelen en controls

Ook bij eenPostBack wordt een ASP.NET-pagina opnieuw geïnitieerd (Page_Init()) waardoor variabelen die gedeclareerd worden als member van de paginaklasse 'verloren' kunnen gaan. Er zijn een aantal mogelijkheden om variabelen en controls persistent te houden bij ASP.NET-pagina's die naar zichzelf posten (en bij uitbreiding ook bij meerdere ASP.NET-pagina's). Een overzicht kan gevonden worden op onderstaande referentie [1]. In de context van dit practicum zijn vooral het Session- en Application-object van toepassing (naast het gebruik van een databank uiteraard).

[1] <http://msdn.microsoft.com/en-us/magazine/cc300437.aspx>

Opgave 1 - Een eerste ASP.NET-pagina

In deze opgave willen we een tabel configureren volgens de wensen van een gebruiker. De laatste kan de volgende instellingen manipuleren:

- aantal rijen,
- aantal kolommen,
- achtergrondkleur van de even kolommen (keuze uit verschillende kleuren),
- achtergrondkleur van de oneven kolommen (keuze uit verschillende kleuren),
- dikte van de rand van de tabel (uitgedrukt in pixels),
- breedte van de tabel (uitgedrukt als een percentage).

Maak een ASP.NET-webpagina die bovenstaande informatie vraagt aan de eindgebruiker. Doe dit volledig in het Nederlands (ook de namen van de kleuren)! Na een druk op een knop moet de tabel onderaan het scherm getoond worden.

In de code-behind van deze webpagina *moet* er een methode terug te vinden zijn met de volgende signatuur:

```
private void maakTabel(int aantalRijen, int aantalKolommen, string
achtergrondkleurEvenKolommen, string achtergrondkleurOnevenKolommen,
int dikteRand, int breedteTabel)
```

De methode `maakTabel()` past de eigenschappen ('properties') van een statische `Table`-control aan zodat aan de wensen van de eindgebruiker voldaan is.

Opmerkingen:

- We tellen vanaf één: de eerste kolom is dus oneven.
- De kleuren waaruit de gebruiker kan kiezen, mag u zelf bepalen. De gebruiker kent enkel Nederlands.
- Hoe de wensen van de gebruiker in te lezen, is vrij te kiezen (d.w.z. kies de gepaste web controls).
- U mag ervan uitgaan dat de gebruiker geen fouten maakt bij het aanpassen van de instellingen. Validatie van de invoer is bij deze opgave dus niet (nog) belangrijk.

Tips:

- Een `Table` bestaat uit `Rows` (`TableRow`-objecten), die op hun beurt bestaan uit `Cells` (`TableCell`-objecten).
- Zet de inhoud (tekst) van een lege cel gelijk aan " " indien deze zichtbaar moet blijven.
- De modulo-operator in C# is %.
- Een `String` converteren naar een `Integer` kan via het `System.Convert`-object.
- Een `Panel` kan gebruikt worden om bij elkaar horende besturingselementen (*web controls*) te groeperen.
- De property `IsPostBack` kan gebruikt worden om te controleren of de pagina voor de eerste keer wordt geladen.
- Een besturingselement (*control*) automatisch laten posten, vereist dat de `AutoPostBack`-eigenschap van het betreffende besturingselement op `True` wordt gezet.
- Dubbelklik op een besturingselement om er code (gedrag) aan toe te voegen.

Bestandsnaam oplossing:

opgave-1-1.aspx

opgave-1-1.aspx.cs

Opgave 2 - Sessie-object en GDI+

De Graphics Device Interface (GDI(+)) van Windows laat toe een ruime set van bewerkingen uit te voeren op grafische objecten. GDI-functionaliteit is in het .NET-raamwerk toegankelijk via de `System.Drawing`-namespace en respectieve sub-namespaces. In deze opgave maken we gebruik van deze functionaliteit om een aantal operaties (herschaling/rotatie/...) uit te voeren op stilstaande beelden (BMP/JPEG/GIF/...).

Voor deze opgave moeten twee webformulieren aangemaakt worden. Op de eerste pagina zullen de noodzakelijke parameters opgegeven worden om hieruit de bewerkte figuur te genereren die zal weergegeven worden m.b.v. de tweede pagina. De nodige parameters dienen tussen beide pagina's te worden doorgegeven via het `Session`-object.

Bij het laden van de eerste pagina dient via een `TextBox` de bestandsnaam van de te bewerken figuur opgegeven te worden. Deze figuur naar keuze (BMP/JPEG/GIF/...) mag als 'existing item' aan de website worden toegevoegd (en dus als aanwezig op de server worden beschouwd). Voorzie een aantal knoppen (`Buttons`) die overeenkomen met een bepaalde bewerking. Als de gebruiker klikt op één van deze knoppen, dan worden een aantal `Controls` getoond op de webpagina. Deze `Controls` moeten aan de gebruiker toelaten om een aantal parameters van de geselecteerde bewerking in te stellen. Implementeer minimaal volgende bewerkingen, met de geschikte `Controls` voor tenminste de volgende bijhorende parameters:

| Bewerking | Parameters |
|-----------------------------|--|
| Herschaling | Relatieve grootte in x- en y-richting na herschaling |
| Rotatie | Keuze uit rotatie over 90, 180, of 270 graden |
| Omzetting naar grijswaarden | n.v.t. |

Naast de knoppen voor het kiezen van de bewerking, is er ook nog een submit-knop aanwezig. Na het 'submittten' van de parameters (i.e. het klikken op de submit-knop) wordt de gebruiker doorverwezen naar de tweede pagina. Aan de hand van de parameters in het `Session`-object wordt de aangepaste figuur gegenereerd en getoond. Voeg op de gemodificeerde figuur telkens een tekstje toe met de groepsnaam in een lettertype en -grootte naar keuze.

Opmerkingen:

- U hoeft slechts één bewerking per keer te ondersteunen. Een combinatie van bewerkingen is in principe uiteraard ook mogelijk, maar niet verplicht.
- U hoeft geen validatie van de gebruikersinvoer te voorzien.
- Er zijn verschillende opties om de gevraagde functionaliteit in deze opgave te implementeren. Alle mogelijkheden zijn in principe toegelaten zolang de minimaal vereiste functionaliteit aanwezig is.
- De gegenereerde figuur wordt niet ingebed in een webpagina, maar wordt rechtstreeks naar de client (browser) gestuurd.
- Gebruik de beschikbare help-functies om meer uitleg en voorbeeldcode (!) te vinden over bepaalde methodes en objecten. In de code kan daartoe op F1 gedrukt worden, of men kan online zoeken op MSDN.

Tips:

- Maak gebruik van het `Response`-object om de gegenereerde figuur op de juiste manier te tonen.
- Gebruik de namespaces `System.Drawing` en `System.Drawing.Imaging`
- Bepaalde methodes vereisen een absoluut pad in de folderstructuur van de webserver. Omdat dit pad niet altijd gekend kan zijn door de programmeur, kan de methode `Server.MapPath` gebruikt worden.
- Om de figuur te bewerken kunnen de objecten `Bitmap` en `Graphics` gebruikt worden.

Bestandsnamen oplossing:

opgave-1-2a.aspx
opgave-1-2a.aspx.cs
opgave-1-2b.aspx
opgave-1-2b.aspx.cs

Opgave 3 - Validatie van gebruikersinvoer en interactie met een databank

Maak een ASP.NET-webpagina die gebruikmaakt van ADO.NET om informatie uit een databank te tonen aan de bezoeker van de webpagina. Bovendien kan de gebruiker via de webpagina nieuwe informatie toevoegen aan de databank, alsook gegevens verwijderen. We gebruiken een zeer eenvoudige databank die een muziekcatalogus bevat. De betreffende databank bevat juist één tabel (`MuziekCatalog`) met de volgende velden:

- id: primaire sleutel (uniek nummer)
- Tracknr,
- Titel,
- Uitvoerder,
- Album,
- Jaar.

Iedere groep beschikt over een eigen databank op `vstudent.elis.ugent.be`. De user ID is "itechX" en het paswoord zal gepost worden in het goepsforum op minerva.

Zorg voor foutopvang indien de connectie met de databank niet zou lukken. Doe dit aan de hand van het `try-catch` mechanisme van C#, zodat er in dergelijke gevallen geen "Application Error"-pagina gegenereerd wordt door de webserver.

De webpagina moet een tabel tonen die bovenstaande informatie uit de databank ophaalt. Het tonen van de informatie in een tabel kan op verschillende manieren gebeuren. De ingediende oplossing mag meerdere manieren bevatten. Er moet gekozen kunnen worden op welk veld de afgebeelde tabel gesorteerd is. Tevens moeten de gegevens in het 'Uitvoerder'-veld aanklikbaar zijn, waardoor de gebruiker meer informatie krijgt over de uitvoerder, bijvoorbeeld een Google resultatenpagina of het betreffende artikel op Wikipedia (er moet niet gecontroleerd worden of het betreffende artikel bestaat).

Daarnaast moet de gebruiker de mogelijkheid gepresenteerd krijgen om nieuwe muzieknummers toe te voegen aan de databank (i.e. een invulveld voor elke kolom in de databank behalve voor de primaire sleutel "id"). Daarbij is het vereist dat per invulveld een `RequiredFieldValidator` toegevoegd wordt die er voor zorgt dat alle velden door de gebruiker ingevuld worden.

Controleer of de ingevulde informatie al in de databank aanwezig is.

- Indien niet: voeg de informatie toe aan de databank en informeer de gebruiker dat dit succesvol uitgevoerd werd.
- Indien wel: voeg de informatie niet toe aan de databank en waarschuw de gebruiker dat deze gegevens reeds in de databank aanwezig zijn.

Als er iets mis loopt tijdens het wegschrijven van de gegevens, zorg dan voor de nodige foutopvang.

Voorzie ook een mogelijkheid om records (rijen) uit de databank te verwijderen.

Opmerkingen:

- De tabel is een zeer eenvoudige databank die niet genormaliseerd is.
- Records (rijen) toevoegen of verwijderen kan via de grafische interface van Visual Studio (zie slides). Deze interface kan ook gebruikt worden om eventuele fouten in de code op te sporen (naast debugging natuurlijk).

- De ingevulde gegevens verder uitgebreid te controleren is niet vereist (bijvoorbeeld controleren of een ingevulde waarde wel degelijk een geheel getal is).

Tips:

- Bekijk de lijst van mogelijk *web controls* in de toolbox van Visual Studio.
- Controleer eerst of de gegevens al in de databank aanwezig zijn en schrijf deze dan pas weg naar de databank. Je mag er van uit gaan dat er geen concurrente operaties zijn op de databank (werken met 'transactions' is niet nodig).
- Gebruik `Panel`-controls

Bestandsnaam oplossing:

opgave-1-3.aspx

opgave-1-3.aspx.cs