

Лабораторна робота.

Розробка web-сторінки з використанням серверних елементів управління

Мета: знайомство зі засобами автоматизації розробки web-додатків в ASP.NET у вигляді серверних елементів управління WebForm.

Основні відомості

Одна з основних переваг ASP.NET полягає в тому, що при її використанні різко спрощується створення інтерфейсу користувача на web-сторінках. Елементи управління WebForm (або серверні елементи управління (Server Controls), або елементами управління web (Web Controls)) визначено в просторі імен System.Web.UI.WebControls. Їх основним призначенням є позбавлення від трудомісткого створення елементів управління HTML на web-сторінці вручну.

Прийнято виділяти три типи серверних елементів управління:

- серверні елементи управління HTML – звичайні теги HTML;
- елементи управління web-сервера – нові теги ASP.NET;
- серверні елементи управління для перевірки даних (валідації) – застосовуються для валідації вхідних даних від клієнтського додатку (зазвичай веб-браузера).

Переваги від використання таких елементів при розробці web-додатків:

- скорочується кількість коду, написаного вручну (що особливо помітно для складних елементів документа). Елемент просто «перетягується» з панелі інструментів, після чого виконується налаштування його параметрів в спеціальному вікні. При цьому всі зміни автоматично заносяться безпосередньо в *.aspx файл;
- кожному елементу управління відповідає певний клас в бібліотеці базових класів .NET, що дозволяє писати для них такий же код як і для будь-яких інших класів;
- для будь-якого елемента управління WebForm визначений набір подій, які обробляються на web-сервері;
- для будь-якого елемента управління WebForm надається можливість для перевірки введення даних користувачем.

За замовчуванням серверні елементи управління HTML в ASP.NET файлах розглядаються як текст. Крім того, всі серверні елементи управління HTML мають бути розміщені усередині області дії тегу `<form>`, що також має атрибут `runat="server"`. Подібно до елементів управління HTML елементи управління web-сервера створюються на web-сервері і передбачають додавання атрибута `runat="server"`. Проте вони можуть і не відповідати конкретним елементам HTML і представляти складніші елементи. Загальний синтаксис для опису таких елементів:

`<asp: <тип елемента> ID=" ідентифікатор " runat="server"></asp: <тип елемента> >`

Наприклад, `<asp:Label ID="Label1" runat="server" Text="Hello World"></asp:Label>`

Форми ASP.NET відрізняються від звичайних форм наявністю властивості `runat="server"`, вони обробляються на сервері. Форми є одним з полів сторінки. На сторінці знаходяться елементи управління. Більшість з них повинні бути розташовані усередині форми. ASP.NET і дозволяють запам'ятовувати стан елементів управління, тобто текст, який було введено в текстове поле, передається на сервер і назад на сторінку після її оновлення:

`<form ID="FormVote" runat="server"></form>`

Всі форми обробляються методом POST.

Другий обов'язковий атрибут – це його ідентифікатор або ID. Він потрібний, щоб звертатися до елемента в програмі, тобто це ім'я елемента сторінки за який його можна ідентифікувати.

Елементи управління HTML є похідними класу System.Web.UI.HtmlControls.HtmlControl. Вони безпосередньо відображаються у вигляді елементів розмітки HTML. Їх відображення не залежить від типу браузера. Властивості таких елементів повністю відповідають атрибутам тегів HTML.

Порівняємо звичайний тег: `<input id="Reset" type="reset" value="reset" />` з елементом управління HTML `<input id="Reset" runat="server" type="reset" value="reset" />`

Різниця полягає тільки в наявності атрибуту `= "server"`. Але він дає колосальну різницю. Теги сервер

відображає як є, а елементом управління можна маніпулювати в коді. Тільки у другому випадку у методі сторінки можна написати

```
Reset.Value = "АСП";
```

що рівносильно

```
this.Reset.Value = "АСП";
```

Отже, **Reset** стає одним з членів класу сторінки.

Елементи управління HTML можна розміщувати на одній сторінці упереміш з серверними елементами.

Серверні елементи більш гнучкі, тому що вони прив'язані не до розмітки, а до функціоналу, який потрібно забезпечити. Багато елементів не мають аналогів в HTML, наприклад, календар. Їх зображення повністю контролюється ASP.NET. Перехоплюючи події **PreRender**, **Init**, **Load**, можна керувати цим процесом. Властивості серверних елементів управління строго типізуються, на відміну від HTML-елементів. У цій таблиці приведені web-елементи управління, які мають пару серед тегів HTML (їх значно більше).

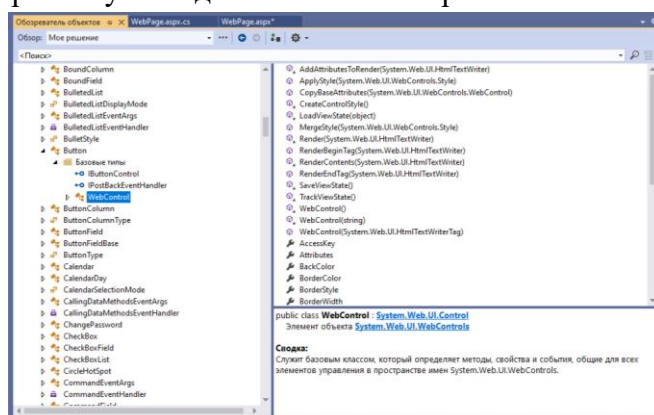
Таблиця відповідності деяких серверних елементів управління тегам HTML

Елемент управління ASP.NET	Відповідний тег HTML	Призначення
<asp:Label>		відобразити текст
<asp:ListBox>	<Select>	список вибору
<asp:DropDownList>	<Select>	випадний список
<asp:TextBox>	<Input Type="Text"> <Input Type="Password"> <Textarea>	рядок редагування поле редагування
<asp:HiddenField>	<Input Type="Hidden">	невидиме поле
<asp:RadioButton> <asp:RadioButtonList>	<Input Type="Radio">	перемикач, список перемикачів
<asp:CheckBox> <asp:CheckBoxList>	<Input Type="CheckBox">	прапорець, список прапорців
<asp:Button>	<Input Type="button"> <Input Type="submit">	командна кнопка
<asp:Image>		зображення
<asp:ImageButton>	<input type="image">	кнопка-зображення
<asp:Table>	<Table>	таблиця
<asp:Panel>	<Div>	контейнер
<asp:BulletedList>	, 	маркований список
<asp:HyperLink>	<A Href>	гіперпосилання

Сервер не обов'язково генерує ті ж самі теги HTML для серверних елементів управління. Все залежить від типу браузера, який використовує клієнт.

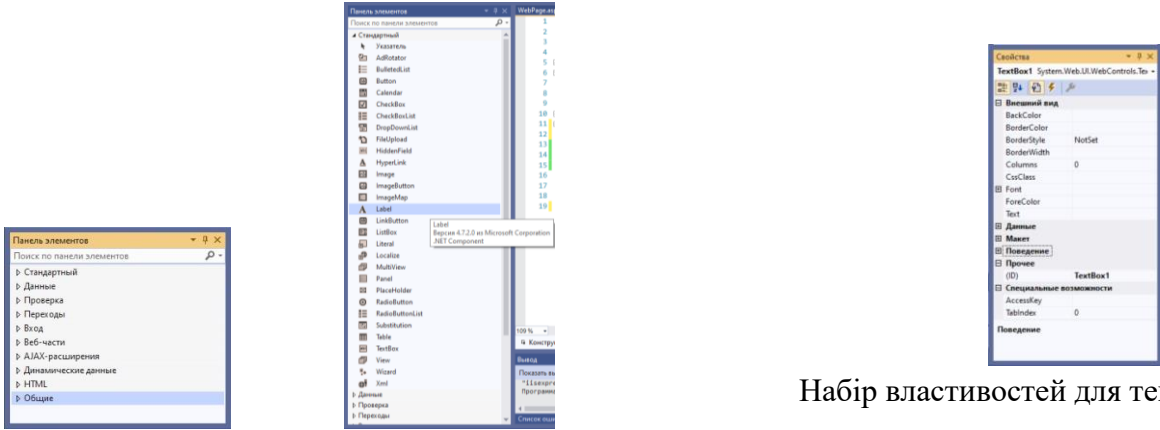
Всі серверні елементи управління знаходяться в просторі імен **System.Web.UI.WebControls** і успадковуються від класу **System.Web.UI.WebControls.WebControl**.

Всі існуючі класи можна проглянути за допомогою «Обозреватель объектов» (Class Browser).



За допомогою властивостей, методів і подій серверних елементів управління можна абстрагуватися від деталей HTML-кода і працювати зі сторінкою та її елементами як з об'єктами.

Набір серверних елементи управління містить «Панель елементів» (Toolbox).



а)

б)

Набір елементів управління WebForm:

а – всі розділи; б – розділ «Стандартные»
(Standard)

Набір властивостей для текстового поля

При внесенні змін для елемента управління у вікні властивостей всі зміни відразу ж записуються безпосередньо в файл *.aspx. Наприклад, якщо для текстового поля TextBox1 змінити (задати) значення властивостей BackColor, BorderColor, BorderStyle, BorderWidth і ToolTip, в тезі <asp: TextBox> з'являться нові пари «ім'я-значення»:

```
<asp:TextBox ID="TextBox1" runat="server" BackColor="White" BorderColor="White" BorderStyle="None"
BorderWidth="1" ToolTip="Прізвище, ім'я та по батькові" Style="background-color: #FFFFCC"></asp:TextBox>
```

І клас Control, і клас WebControl надають похідним класам кожен своїм набором членів.

Властивості базового класу Control

Властивість	Опис
ID	дозволяє отримати або встановити ідентифікатор елемента управління
MaintainState	дозволяє працювати зі станом перегляду для елемента управління
Page	повертає об'єкт Page, тобто, сторінку, на якій знаходиться цей елемент управління
Visible	дозволяє отримати або визначити видимість елемента управління (тобто, буде він виводитися на сторінці чи ні)

Властивості базового класу WebControl

Властивість	Опис
BackColor	дозволяє отримати/встановити колір фону елемента управління
BorderColor	дозволяє отримати/встановити колір рамки навколо елемента управління
BorderStyle	дозволяє отримати/встановити стиль рамки навколо елемента управління
BorderWidth	дозволяє отримати/встановити товщину рамки навколо елемента управління
Enabled	дозволяє отримати/встановити значення, що визначає доступність елемента управління
Font	дозволяє отримати інформацію про використаний шрифт
ForeColor	дозволяє отримати/встановити колір переднього плану (зазвичай колір напису) для елемента управління
Height/Width	визначають висоту/ширину елемента управління
TabIndex	дозволяє отримати/встановити значення TabIndex (черговості переходу по клавіші табуляції Tab)
ToolTip	дозволяє отримати/встановити значення ToolTip (підказка, що з'являється при наведенні на елемент управління миші)

Базові елементи управління WebForm

Елемент управління	Опис
Button ImageButton	різновиди кнопок
CheckBox CheckBoxList	прапорець вікно списку з декількома прапорцями
DropDownList ListBox ListItem	ці типи призначені для створення різних видів списків (спадаючий, звичайний і елемент списку відповідно)
Image Panel Label	ці типи представляють контейнери для статичного тексту і зображень (а також засіб для їх групування)
RadioButton RadioButtonList	стандартний перемикач вікно списку з набором перемикачів
TextBox	Текстове вікно для введення даних користувачем. Може бути налаштоване для прийому одного або декількох рядків тексту

Приклад групи перемикачів.

```
<asp:RadioButtonList ID="RadioButtonList1" runat="server" BackColor="#E0E0E0"
  BorderColor="Silver" Width="494px" BorderStyle="Outset" BorderWidth="3px">
  <asp:ListItem Selected="True">Відмінно</asp:ListItem>
  <asp:ListItem>Добре</asp:ListItem>
  <asp:ListItem>Задовільно</asp:ListItem>
  <asp:ListItem>Незадовільно</asp:ListItem>
</asp:RadioButtonList>
```

Завдання

Створити web-форму для реєстрації/авторизації користувача сайту. Перевірити коректність введення даних на формі. Побудувати регулярний вираз, що описує шаблон для перевірки номера телефону та електронної пошти. Для внесення персональних даних (при реєстрації) використати різні види кнопок (у тому числі групування), зображення, спискові елементи управління, наприклад, DropDownList.

Контрольні питання

1. У чому полягають основні відмінності web-інтерфейсу від інтерфейсу windows-додатку?
2. Якими перевагами володіє web-інтерфейс порівняно з інтерфейсом windows-додатку?
3. Якими недоліками володіє web-інтерфейс порівняно з інтерфейсом windows-додатку?
4. У яких випадках доцільно застосовувати web-інтерфейс?
5. Які елементи інтерфейсу можуть використовуватися при побудові web-інтерфейсу?
6. Чи відрізняються елементи web-інтерфейсу від відповідних елементів windows-додатку?
7. Яким чином виконується обробка подій для елементів web-інтерфейсу?
8. Яку роль грає HTML в побудові web-інтерфейсу?
9. Яким чином виконується перевірка даних, що вводяться користувачем у web-додатку?
10. У чому полягають відмінності даного способу перевірки від перевірки даних в windows-додатку?