

Практическая работа №5
**Основы работы с элементами графического интерфейса ASP.NET
WebForms**

Цель – изучение свойств элементов разметки, серверных элементов управления HTML, элементов управления web-сервера, способов применения различных методов обработки событий для серверных элементов управления; применение аргументов обработчиков событий для решения задач пользователя; формирование навыков обработки и представления информации средствами Web.

Информационные материалы:

Курс лекций «Компьютерные сети и Интернет-технологии».

Ватсон К., С#. – М: Изд-во "Лори", 2005. – 861 с.

Эспозито Д. Microsoft ASP.NET 2.0 Базовый курс. Мастер-класс /Пер.с англ. – М.: Изд-во «Русская редакция»; СПб.: Питер, 2007. – 668 с.

Мюллер, Дж. Visual Studio 6. Полное руководство: перевод с английского / Дж. Мюллер. - Киев : Ирина, 1999. - 672 с.

Серверные веб-элементы управления, [Электронный ресурс]. – Режим доступа:

[http://msdn.microsoft.com/ru-ru/library/bb386416\(v=vs.100\).aspx](http://msdn.microsoft.com/ru-ru/library/bb386416(v=vs.100).aspx)

Общие сведения об элементах управления входом, [Электронный ресурс]. – Режим доступа:

[http://msdn.microsoft.com/ru-ru/library/ms178329\(v=vs.100\).aspx](http://msdn.microsoft.com/ru-ru/library/ms178329(v=vs.100).aspx)

Программное обеспечение:

Microsoft Visual Studio 2005

Internet Explorer.

Контрольные вопросы:

1. Что такое веб-форма? Для чего используются веб-формы?
2. Какие типы серверных элементов управления используются в ASP.NET?
3. В каком пространстве имен определены элементы WebForms?
4. Для чего используется свойство PostBackUrl? Приведите примеры.
5. Опишите назначение свойств TabIndex, Focus, AccessKey. Для каких серверных элементов управления они применимы. Приведите примеры.

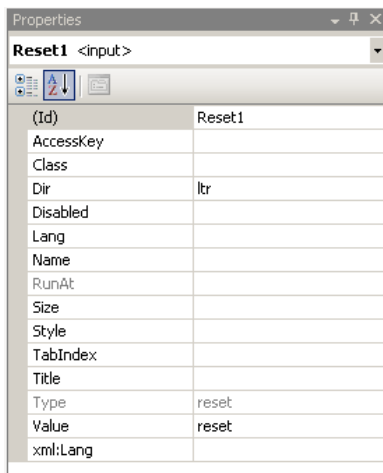
Практикум

В рамках своего Web-проекта создайте новую ASP-страницу. Разместите на ней заголовок «My-WebForms».

Упражнение. Найдите различия элементов разметки HTML, серверных элементов управления HTML и элементов управления веб-сервера в ASP.NET.

На новой aspx-странице перейдите в режим Source и между тэгами `</form>` – `</form>` вручную наберите следующий код:

```
<input id="Reset1" type="reset" value="обычный HTML" />
```



Обратите внимание, что свойство RunAt этого элемента в окне Properties недоступно.

Нажмите F5 (Debug) – Ваш элемент успешно появится в окне проводника.

Перейдите в режим Design, откройте окно элементов Toolbox, выберите группу элементов HTML и переместите на страницу новый (уже серверный элемент управления HTML) Input (Reset).

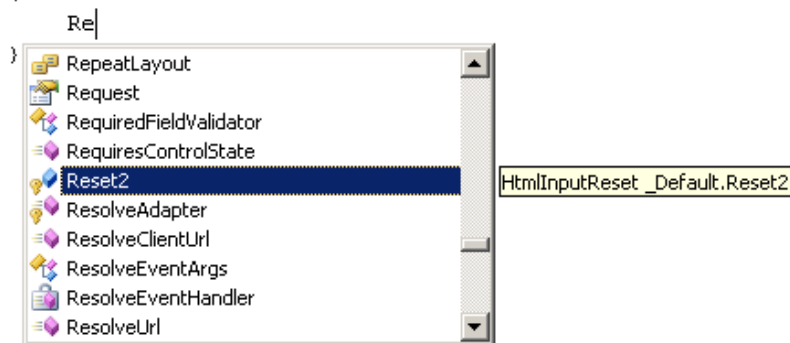
В окне Properties по умолчанию устанавливается свойство ID – Reset2, определяем свойство Value как "Это HTML ASP.NET" и в исходном HTML-тэге добавляем атрибут `runat="server"`.

Нажимаем F5 (Debug) – наши элементы отрабатываются в окне проводника без видимых различий.

Вместе с тем, атрибут `runat="server"` дает колоссальную разницу. HTML-теги сервер отображает как есть, а элементом управления HTML можно манипулировать в коде.

Для того, чтобы убедиться в этом, достаточно перейти на страницу программного кода и попытаться изменить свойства элемента разметки "Обычный HTML".

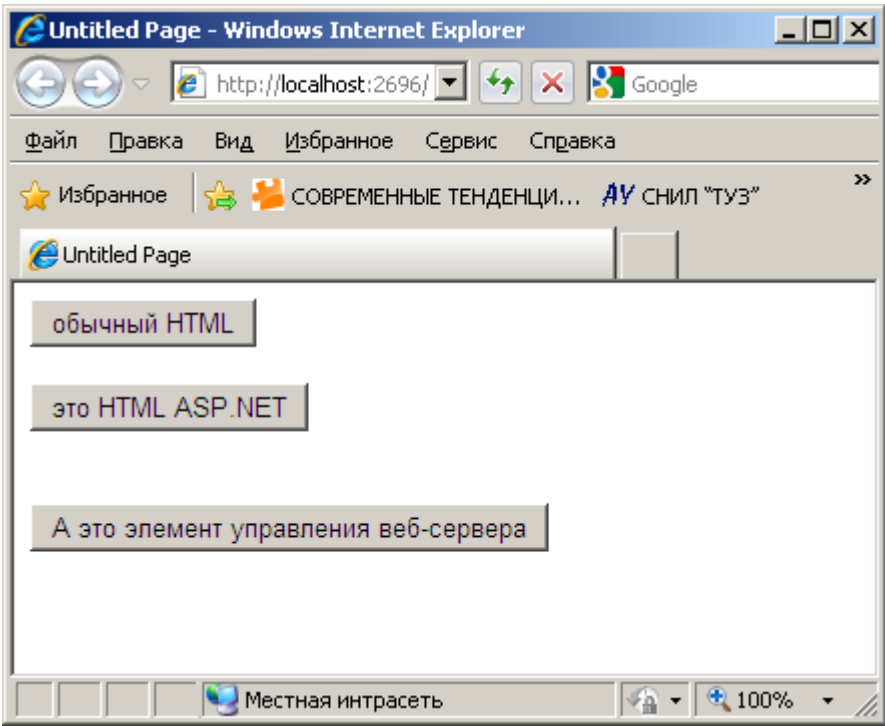
```
protected void Page_Load(object sender, EventArgs e)
{
```



Как видим, Reset1 не существует! Внести в него изменения можно только в HTML-тэгах, а чтобы придать ему свойства управления необходимо добавить `runat="server"`.

Добавим элемент управления веб-сервера, которые размещаются во вкладке Standart меню Toolbox.

Выберем уже знакомый для Вас элемент Button, в окне свойств которого присвоим ему имя "А это элемент управления веб-сервера". Нажимаем F5 (Debug) – для пользователя все элементы в окне проводника идентичны.



Вернитесь в режим просмотра исходного HTML-тэгов этих трех элементов и выделите ключевые служебные слова, которые определяют, как будут обрабатываться эти элементы при загрузке страницы.

Задание 1. Разместите в таблице (файл отчета) следующие HTML-тэги в зависимости от типа используемого элемента управления:

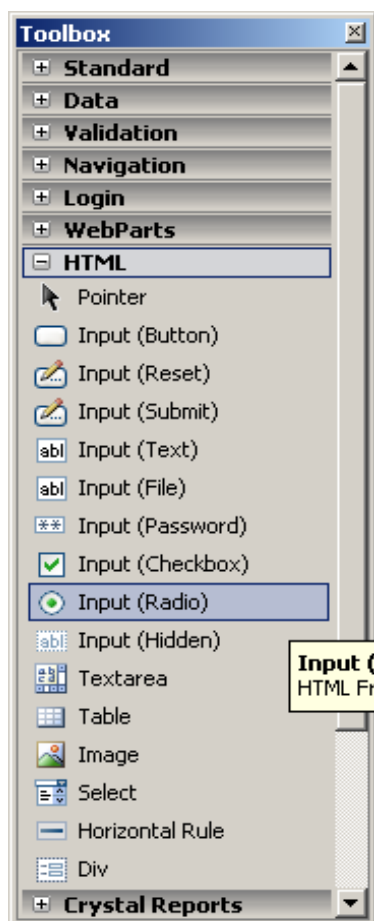
```
<input id="Password" type="password" runat="server" />
<input id="Reset" type="reset" value="Unit1" />
<asp:Image ID="Image1" runat="server" />
<input id="Reset" type="reset" value="Unit2" runat="server"/>
<input id="Checkbox" type="checkbox" />
<asp:Button ID="Button" runat="server" Text="Unit3" />
<input id="Button" type="button" value="OK" runat="server"/>
<asp:RadioButton ID="RadioButton1" runat="server" />

```

Элемент разметки HTML	Серверный элемент управления HTML	Элемент управления веб-сервера

Рассмотрим принципы работы с разными типами серверных элементов управления.

1. Элементы управления HTML



Рассматриваемая группа элементов управления является наследниками класса `System.Web.UI.HtmlControls.HtmlControl`

На странице в режиме Design размещаем четыре элемента `Input (Radio)`, свойства `Value` которых назначаем:

- "Отлично"
- "Хорошо"
- "Удовлетворительно"
- "Неудовлетворительно".

И элемент `Input (Submit)` со свойством `Text` – "Оценка".

Добавим проверочный тэг:

```
<p id="ans" runat="server"></p>
```

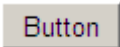

В программный код события `Submit` добавляем возможную реакцию на действия пользователя:

```
if (Radio1.Checked == true)
    ans.InnerHtml = "Ваши знания по ASP.NET отлично";
if (Radio2.Checked == true)
    ans.InnerHtml = "Ваши знания по ASP.NET хорошо";
if (Radio3.Checked == true)
    ans.InnerHtml = "Ваши знания по ASP.NET удовлетворительно";
if (Radio4.Checked == true)
    ans.InnerHtml = "Ваши знания по ASP.NET неудовлетворительно";
```

Проверим работоспособность нашего приложения (F5).

Задание 2. Самостоятельно разработайте обучающую страницу "ASP-сорбонка"¹ с использованием шаблона `Web Application` и серверных элементов управления `HTML` (`Image`). Событие – `OnClick`.

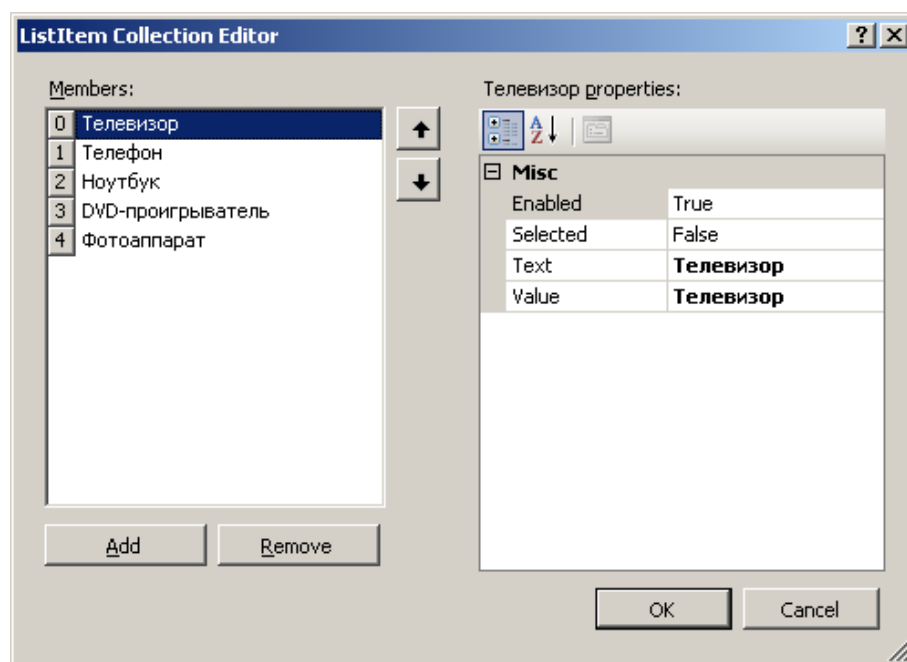
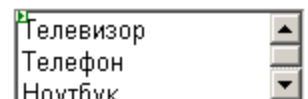
¹ *Сорбонка* — карточка небольшого размера, с одной стороны которой рисунок, а с другой – термин. Сорбонки, получившие свое название от университета г.Сорбонны, известны с древних времен и предназначены для активной тренировки памяти

Сторона-вопрос (добавьте рисунок-ассоциацию)	Сторона-ответ
<code>System.Web.UI.Control</code>	
	
4096 байт	
<code>System.Web.MailMessage</code>	
	

2. Элементы управления веб-сервера

Все серверные элементы управления находятся в пространстве имен `System.Web.UI.Control` и наследуются от класса `System.Web.UI.WebControls.WebControl`.

Например, выберем элемент `ListBox` и, используя встроенный редактор коллекции, приступим к созданию "Корзины покупателя" для Интернет-магазина.



В качестве результата события `SelectedIndexChanged` определяем возвращение информационной строки "Ваш выбор:...".

```
if (ListBox1.SelectedItem != null)
    Label1.Text = "В Вашей корзине: " + ListBox1.SelectedItem.Value;
else
    Label1.Text = "Вы выбрали: ";
```

Поясните в отчете о выполнении работы, с какой целью в этом задании введен оператор условного перехода.

Задание 3. Создайте веб-форму, на которой размещаются названия городов-побратимов г.Минска и при выборе пользователем появляется герб соответствующего города.

Рассмотрим возможности элемента управления веб-сервера **Calendar** (абстрактный класс `System. Globalization. Calendar`), который можно использовать для предоставления пользователю выбора даты (диапазона дат) в удобном формате.

С помощью Toolbox размещаем элемент Calendar на странице.

Например, нам необходимо выделить текущую дату на веб-странице. Для этого, в окне Properties выбираем свойство `TodayDayStyle` и указываем:

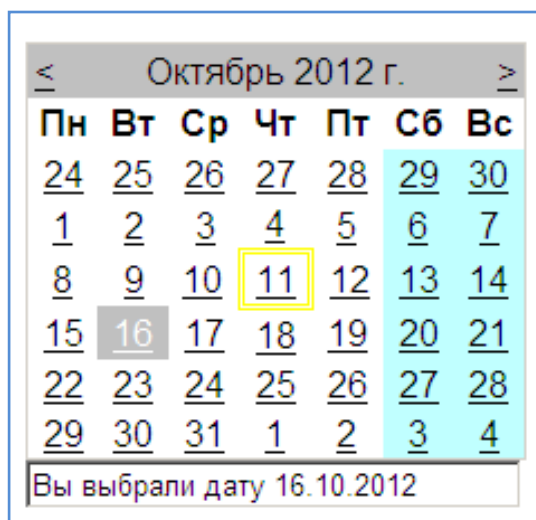
`BorderColor – Yellow;`


`BorderStyle – Double.`

Таким же образом, вы можете определить особые настройки для выходных дней (суббота, воскресенье).

Перехватить дату, выбранную пользователем, можно с использованием события `SelectionChanged`:

```
TextBox2.Text = "Вы выбрали дату  
"+Calendar1.SelectedDate.ToShortDateString();
```



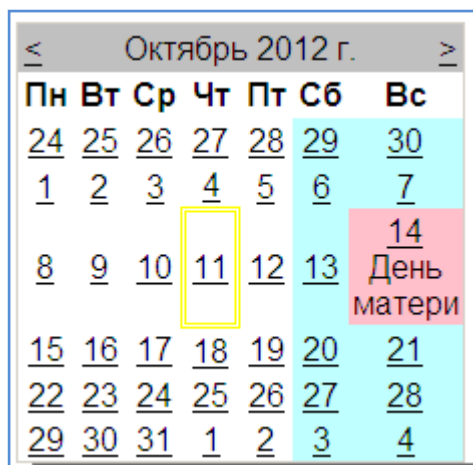
За изменение внешнего вида отдельного элемента Calendar отвечает событие `DayRender`, которое можно выбрать в окне Properties нажав значок событий элемента .

Внесем особое событие в наш календарь на 14 октября.

```

if (e.Day.Date.Day == 14&e.Day.Date.Month==10)
{
    e.Cell.BackColor = System.Drawing.Color.Pink;
    e.Cell.Controls.Add(new LiteralControl("<br />День матери"));
}

```



Задание 4. Используя элемент Calendar программно установите специальные обозначения государственных праздников Республики Беларусь, объявленных выходными днями.

Дополнительное задание (+5 баллов). На основе элемента управления Calendar создайте приложение "График отпусков" для небольшой организации численностью 7 человек. Продолжительность отпуска – 30 календарных дней, допускается разбить отпуск на две части (одна часть – не менее 14 календарных дней). Одновременно могут уйти в отпуск не более 4 работников.

Пояснение: работники вводят диапазон дат планируемого отпуска, приложение проверяет соблюдение заданных условий и представляет решение в табличной форме:

Ф.И.О. работника	Трудовой отпуск	
	дата начала	дата окончания

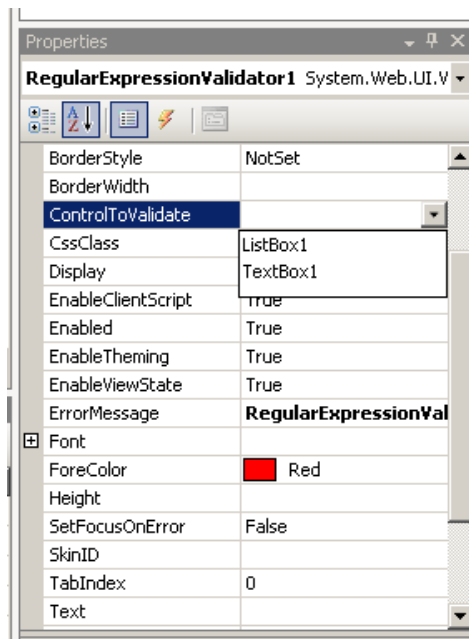
3. Элементы проверки вводимых данных

Классы элементов проверки данных (валидаторов) определяются абстрактным классом `BaseValidator`, который является наследником класса `Label`. Все валидаторы имеют свойство `ControlToValidate`.

Все проверочные элементы сгруппированы в разделе Validation меню Toolbox.

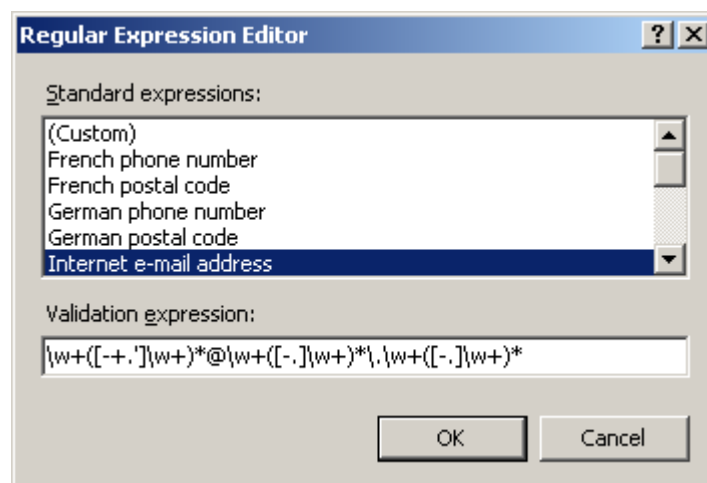
Для проверки правильности вводимого пользователем адреса электронной почты размещаем элемент управления TextBox и выбираем проверочный Regular Expression Validator.

В окне Properties выбираем контролируемый элемент TextBox (свойство ControlToValidate).



Свойство ErrorMessage позволяет определить текстовое сообщение, получаемое пользователем в результате его некорректных действий при вводе информации.

Формат адреса электронной почты является общепринятым и его шаблон включен в редактор стандартных выражений, поэтому свойство Validation Expression устанавливаем с использованием стандарта.



Не забываем добавить в программный код оператор для считывания данных вводимых пользователем.

В результате выполнения Вашего приложения ошибка пользователя должна быть установлена

Введите адрес электронной почты

125@253

Это не адрес электронной почты

Задание 5. Создайте web-форму "Демографическая карточка пользователя", в которой фиксируются возраст (лет), пол (мужской, женский), образование (среднее, высшее) и род занятий (работаю, учусь, пенсионер, отдыхаю). Добавьте необходимые валидаторы.

Дополнительное задание (+2 балла). Разработайте web-форму, в которой считывается имя пользователя. Используя элемент проверки вводимых данных установите, является ли пользователь студентом УИР. Если пользователь является студентом УИР, направляем его на страницу дисциплины "Компьютерные сети и Интернет-технологии", в иных случаях открывается главная страница Академии управления.