

+

## Лабораторная работа №7. Первый взгляд на технологию ASP.Net. Состояние представления (ViewState) и скрытые поля (Hidden fields).

1. Откройте среду разработки Visual Studio и создайте новый проект (Меню *Файл-Создать-Проект*).
2. В левой части окна выберите шаблон «Веб». В средней колонке выберите пункт «Пустое веб-приложение ASP.Net».
3. В нижней части окна задайте имя проекта и нажмите кнопку «ОК» (Рис.1).

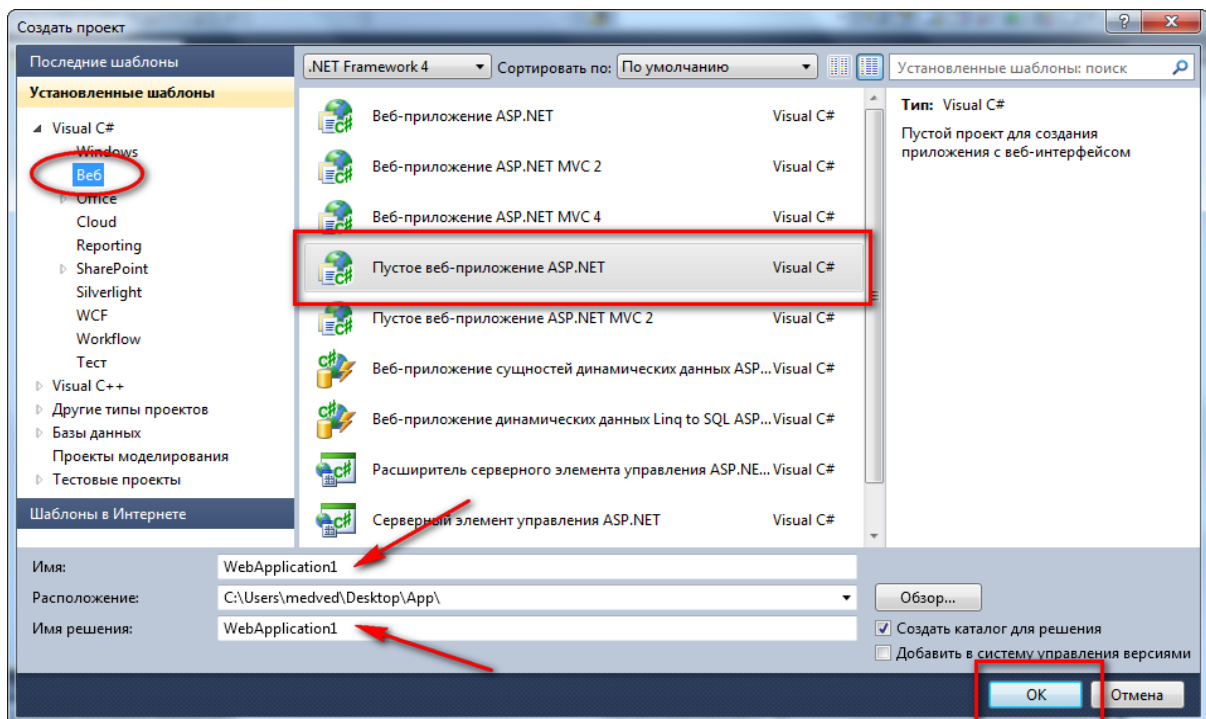


Рис.1.

Созданный проект показан ниже (Рис.2):

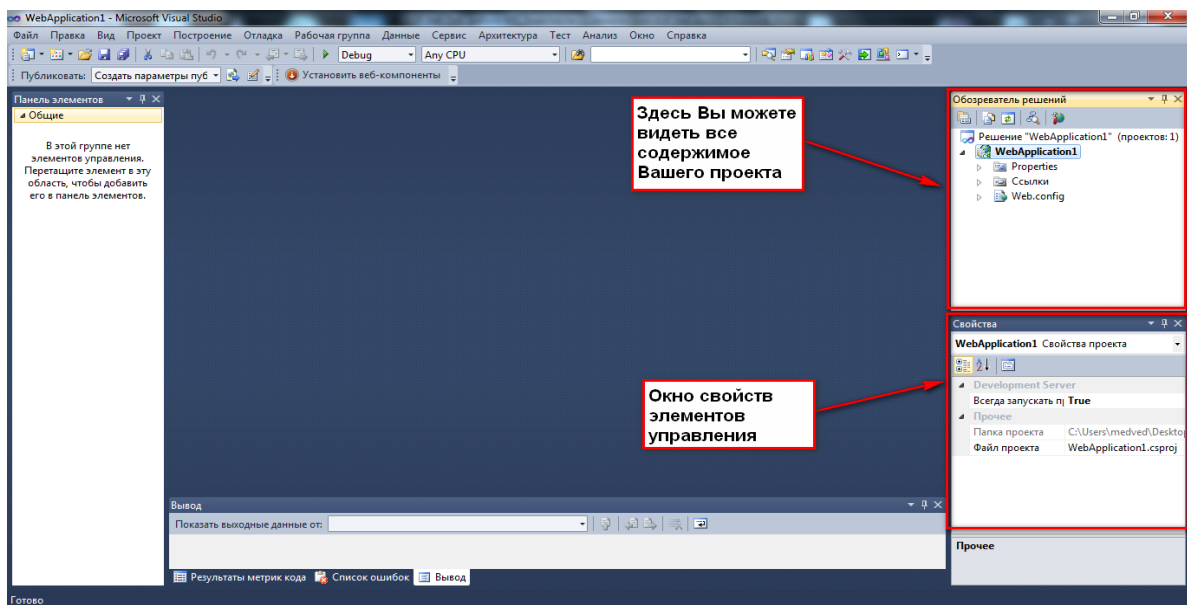


Рис.2.

4. Создайте первую страницу проекта. Для этого нажмите правой кнопкой мыши на названии проекта (WebApplication1). Выберите пункты «Добавить-Создать элемент».
5. В появившейся форме выберите пункт «Форма WebForm» и нажмите кнопку «Добавить» (Рис.3).

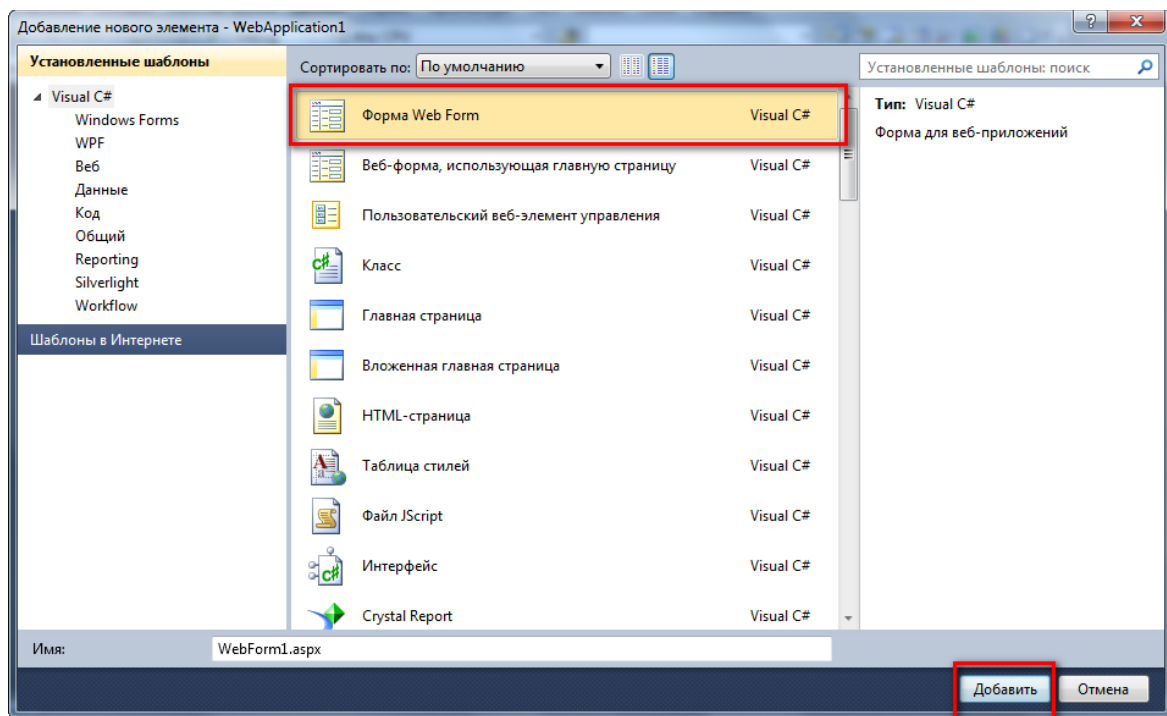


Рис.3.

6. В созданной форме уже присутствует html-код (Рис.4).

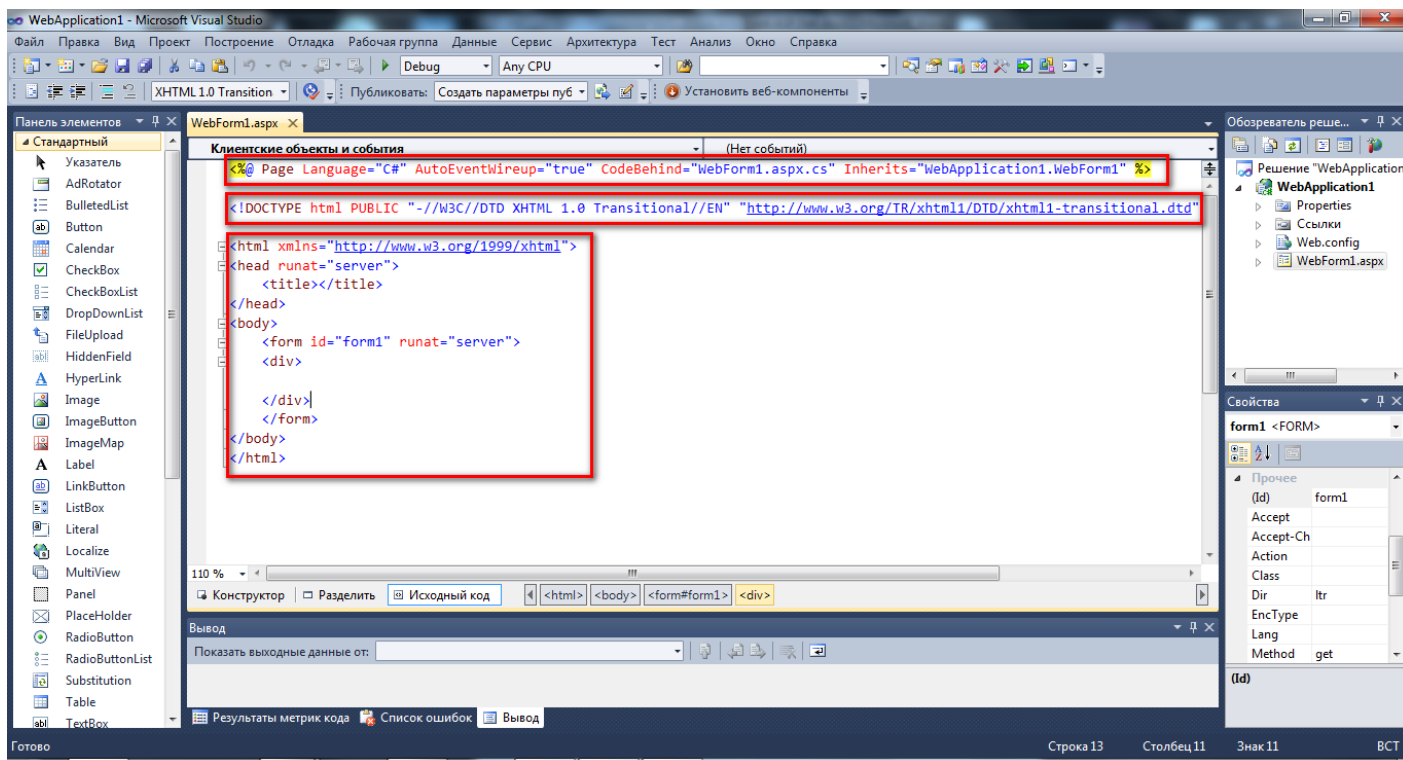


Рис. 4.

О том, что страница является частью приложения ASP.Net нам говорит первая строка. Директива Page содержит в себе указания на язык программирования, используемый в файле логики страницы, название файла логики и некоторые данные о наследовании. Далее располагаются стандартные элементы html-страницы. Но есть и отличия от стандартных элементов. У некоторых тегов прописано свойство runat= "server". Это свойство говорит о том, что для каждого такого элемента создается серверный компонент, который будет осуществлять некую логику работы с этим элементом. И основной тег, который есть на странице – это тег <form>. Данная форма является серверной (так как задано свойство runat= "server"). Серверная форма на странице может только одна.

7. Добавьте на страницу элементы управления. Все элементы формы необходимо размещать внутри тега <form>. Из списка элементов, который находится слева от формы (если его нет, необходимо выбрать в главном меню «Вид» - «Панель элементов») мышкой перетащите на страницу между тегами <div> и </div> кнопку (button) и выпадающий список (dropDownList).
8. Переключитесь в визуальный режим представления страницы. Для этого нажмите на вкладку «Конструктор» в нижней части окна.

9. Выделите *одним щелчком* мыши «Кнопку» на форме и нажмите F4. Откроется окно свойств данного элемента управления. Задайте самостоятельно цвет фона для кнопки (BackColor) и текст на кнопке (Text).
10. Добавьте в элемент «Выпадающий список» два новых элемента списка. Для этого выделите элемент на странице. У выделенного элемента справа появится иконка треугольника. Нажмите на нее. Далее откроется форма редактирования элементов выпадающего списка. Добавьте два элемента и закройте форму.
11. Запустите проект на выполнение (Ctrl+F5). Нажмите на «Кнопку» и посмотрите, как меняется Ваша страница (на данном этапе Вы увидите только перегрузку страницы, так как никакого функционала мы еще не прописали).
12. Посмотрите на код страницы (правая кнопка мыши – Посмотреть код страницы). Элементы управления «Кнопка» и «Выпадающий список» с тегами <asp> отработали на странице как стандартные элементы html - input и select. Но при этом на странице появились два новых элемента input со свойством type = "hidden". Это так называемые *Скрытые поля*, которые хранят в себе состояние элементов управления на странице. Протокол HTTP не передает данные о состоянии элементов, находящихся на странице. В технологии ASP.Net с помощью скрытых полей на сервер передается состояние элементов управления (ViewState) и информация о событиях на форме (EventValidation). Сервер после обработки запроса генерирует для клиента (браузера) html-код страницы. При этом состояние элементов управления не сбрасывается на начальное, а остается в том виде, в котором было передано в скрытых полях при запросе на сервер (Рис.5).

```

1
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><title>
7
8 </title></head>
9 <body>
10 <form method="post" action="WebForm1.aspx" id="form1">
11 <div class="aspNetHidden">
12 <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
13 value="/wEPDwUKLTQwNDcxMTQ5NWRkDycMR6xDnuzgT20mH6nPjpxv9+pYCVxv+bb+16mB4KM=" />
14 </div>
15 <div class="aspNetHidden">
16
17 <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION"
18 value="/wEdAAIhGBItggA4jILD0expTOhtzf78Z8BXhXifTCAVkevd27HtECVSvpPvByHMSX4sGxzTyNACjOE30dNI6f9w7qc" />
19 </div>
20 <div>
21 <input type="submit" name="Button1" value="Button" id="Button1" />
22 <select name="DropDownList1" id="DropDownList1">
23
24 </select>
25 </div>
26 </form>
27 </body>
28 </html>

```

Рис. 5.

13. Добавьте на форму еще один элемент управления «Метка» (Label) под выпадающим списком.

14. Создайте обработчик события для кнопки на форме. Для этого в режиме Конструктора сделайте двойной щелчок мышью по кнопке. Среда разработки автоматически сгенерирует C#-код обработчика события Click.

15. Далее пропишите следующий код в обработчике (Рис.6):

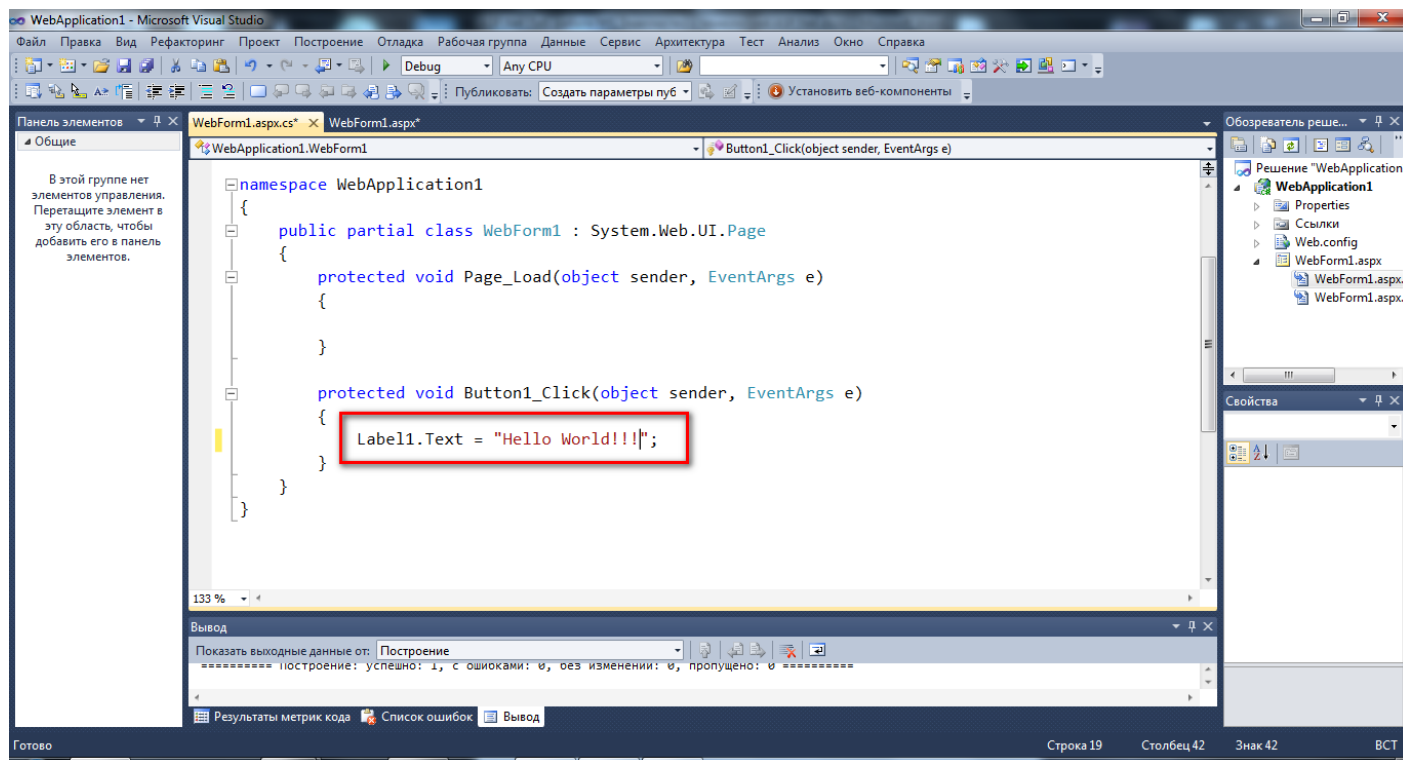


Рис. 6.

16. Запустите проект на выполнение (Ctrl+F5) и нажмите на «Кнопку». На странице отработает обработчик события нажатия на кнопке и под выпадающим списком появится текст, который мы передали в качестве значения свойству Text элемента управления Label1.
17. Установите для элемента управления DropDownList1 свойство AutoPostBack в true (после установки данного свойства в true, любые действия с данным элементом управления будут вызывать обращение на сервер).
18. Создайте обработчик события DropDownList1\_SelectedIndexChanged (событие обработки изменения значения в списке). Для этого дважды щелкните мышью по элементу управления. Среда разработки перейдет в файл логики страницы и сгенерирует код для обработчика события.
19. Добавьте в обработчик события следующий код:

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    Label1.Text = DropDownList1.SelectedValue;
}
```

20. Запустите проект на выполнение (Ctrl+F5).
21. Откройте код страницы. Можно заметить, что в html-коде появился javascript, который для DropDownList1 вызывает обращение на сервер при смене пункта в списке.
22. Самостоятельно добавьте на форму элемент управления CheckBox. Задайте значение true для свойства AutoPostBack данного элемента. Создайте обработчик события CheckBox1\_CheckedChanged.
23. Далее в обработчике события пропишите условие: Если CheckBox выбран, то свойству Label1.Text присвоить значение CheckBox.
24. Запустите проект на выполнение. Проверьте работу всех элементов на странице.

### ***Ответьте на вопросы:***

1. *Приведите примеры элементов управления ASP.Net. Какие события могут быть обработаны для элементов управления Button (кнопка) и DropDownList (выпадающий список).*

2. *Что такое ViewState (состояние элементов управления) и для чего оно используется?*
3. *Объясните назначение свойства AutoPostBack. Что происходит при выставленном значении данного свойства в true?*