

Практическая работа №2 Создание страниц ASP.NET

Цель – изучение структуры ASP.NET-страницы и взаимодействие ее элементов, "жизненный цикл" страницы, понятие Master-страницы, пространства имен библиотеки классов; освоение технологий работы со стандартными компонентами ASP.NET - страница .aspx (элементы управления Label, Textbox, Button), страница code-behind; создание проекта "Моя персональная страница"; разработка веб-страницы "Калькулятор" средствами ASP.NET; формирование навыков структурирования и размещения информации.

Информационные материалы:

Курс лекций «Компьютерные сети и Интернет-технологии».

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 : Ирида, 1999. - 672 с.

Программное обеспечение: Microsoft Visual Studio 2005, Internet Explorer.

Полезные ссылки:

[http://msdn.microsoft.com/ru-ru/library/ms379557\(v=vs.80\).aspx](http://msdn.microsoft.com/ru-ru/library/ms379557(v=vs.80).aspx)

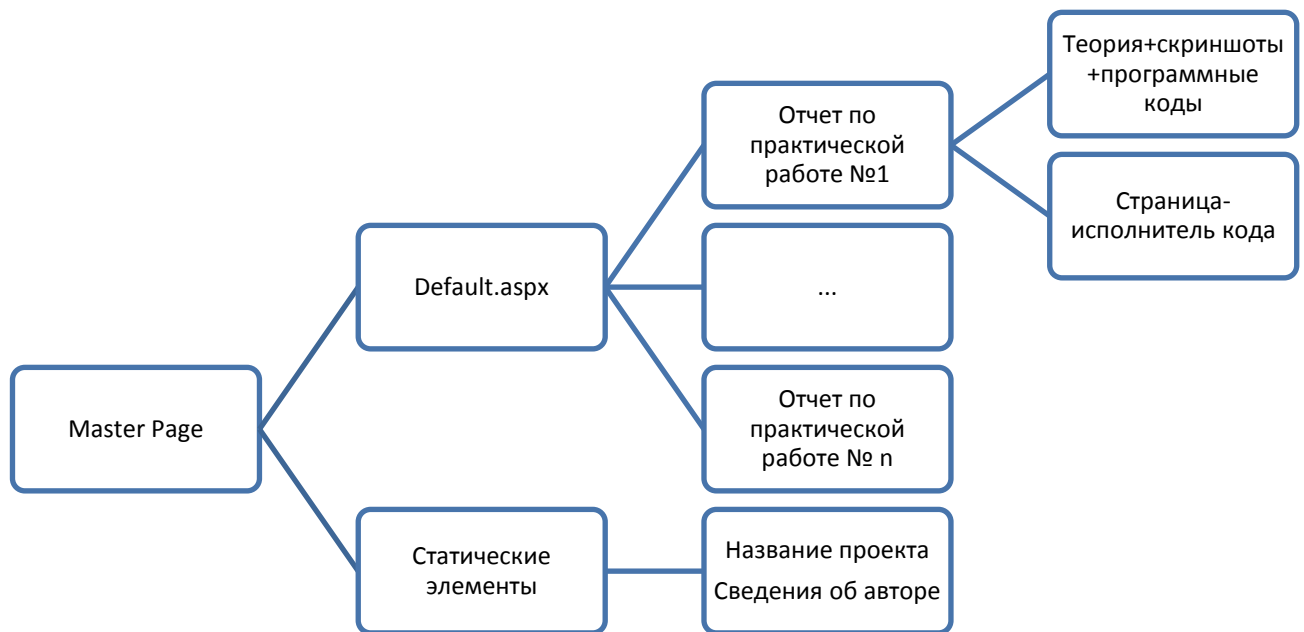
<http://cyberguru.ru/microsoft-net/asp-net/aspnet20-masterpages-basics-page2.html>

Контрольные вопросы:

1. Опишите механизм работы приложений ASP.NET.
2. Что такое Common Language Specification?
3. С какой целью создаются Master-страницы?
4. Каким образом ASP.NET поддерживает состояние серверных элементов управления?
5. К какому пространству имен принадлежат такие классы ASP.NET как Page и Literal Control?
6. Какие семь пространств имен автоматически импортируются в каждую страницу ASP.NET?
7. Что означает такое понятие как "безопасный тип"?
8. Каким образом можно узнать, какие классы имеются в библиотеке классов .NET?
9. В какой последовательности выполняются события ASP-страницы? Что такое "жизненный цикл" ASP-страницы? Какие события в него включены?
10. Какое свойство необходимо для работы форм и серверных элементов управления в ASP.NET?

Практикум

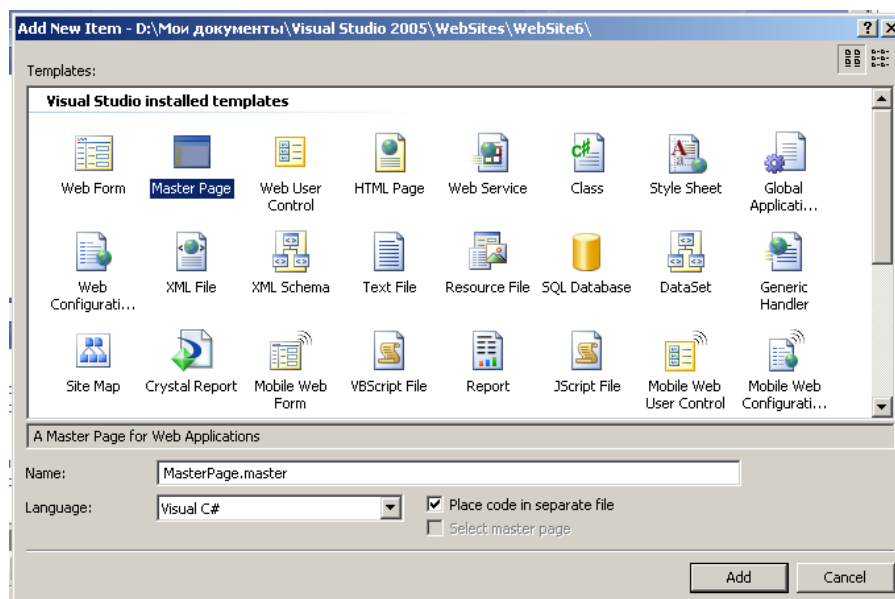
Задание 1. Создание проекта "Моя персональная страница" (см.схему)



Откройте VisualStudio, выполните команду New Web Site, выбрав в качестве шаблона (Template) пустой сайт (Empty Web Site).

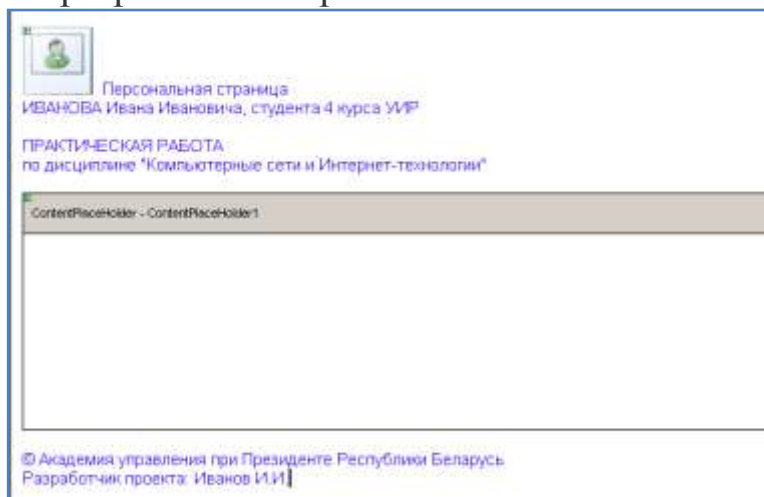
Для разработки и реализации программного кода Вы можете выбрать удобный для Вас язык программирования.

Для создания страницы шаблона для Вашего проекта на панели управления выберите значок  Add New Item и в открывшемся окне откройте страницу Master Page.



На созданной странице шаблона вне области ContentPlaceFolder разместите статические элементы:

- название проекта "Персональная страница Ф.И.О";
- Ваше фото и краткие сведения о себе;
- общий заголовок "Практическая работа по дисциплине "Компьютерные сети и Интернет-технологии";
- сведения о разработчике проекта.

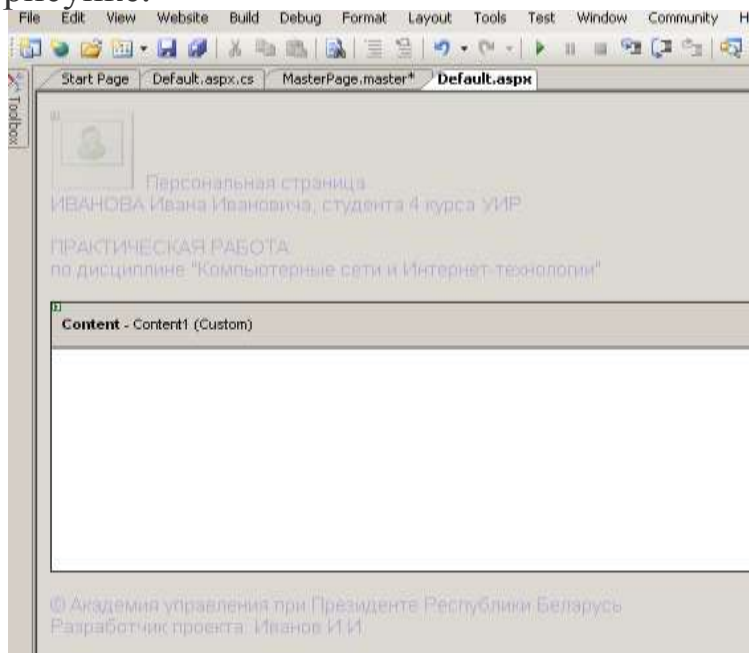


В режиме просмотра HTML измените заголовок шаблона (тег Title), например, на следующий "Блог Ф.И.О".

Область ContentPlaceFolder1 будет предназначена для размещения динамических aspx-страниц. В нашем проекте - для формирования банка Ваших отчетов по выполненным практическим заданиям.

Используя шаблон Web Form, который открывается по команде Add New Item, создайте основную страницу проекта Default.aspx. При этом не забудьте включить флажок Select Master Page, который обеспечит связь отдельных веб-страниц Вашего проекта со страницей-шаблоном.

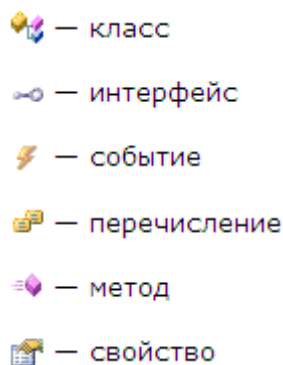
Внешний вид страницы Default.aspx, связанной с Master Page, представлен на рисунке.



Перейдите в режим просмотра HTML-кода и убедитесь, что добавлены элементы управления содержимого (Content controls), которые хранят в себе элементы и разметку, указанную вами в конструкторе.

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
Title="Блог Иванова Ивана" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    &nbsp;
</asp:Content>
```

В Visual Studio имеется специальный компонент Object Browser, с помощью которого можно просмотреть имеющиеся библиотеки классов. Все пространства имен показаны в виде дерева со следующим обозначением:



Самостоятельно создайте новую веб-страницу Default2.aspx (связанную с Master Page) и разместите подготовленный Вами отчет о выполнении предыдущей практической работы (например, в форме статического текста с рисунками).

Перейдите к странице Default.aspx и с помощью панели ToolBox создайте управляющий элемент LinkButton1.

В окне Properties:

1. Присвоим для атрибута Text следующее свойство:
 - Отчет по практической работе №1 "Базовые понятия ASP.NET".
2. Укажем URL-адрес для страницы-содержимого Default2.aspx (свойствоPostBackURL)/
3. В качестве события (Even) отметим, что загрузка страницы Default2.aspx (Page Load) осуществляется при нажатии (Click) на кнопку-ссылку LinkButton1.

Добавьте элемента навигации между страницами Вашего проекта и проверьте его работоспособность.

На основной странице проекта Default.aspx добавьте кнопку-ссылку - Отчет по практической работе №2 "Создание страниц ASP.NET" и приступайте к выполнению самостоятельных заданий.

Задание 2. Создайте простую страницу ASP.NET, в которой пользователь может ввести два числа и, щелкнув на кнопке "Умножить", увидеть их произведение.

Используйте элементы управления для двух текстовых полей, а также для кнопки "Умножить". Кроме того, используйте код стандартного обработчика событий.

Для преобразования вводимого текста в значения целого типа, которые нужны для умножения, попробуйте использовать функцию `Cint()`, например, `Cint(tbNumber1, Text)`.

Задание 3. Для предыдущего задания добавьте комментарии по алгоритму выполнения программного кода, HTML-кода и используемых пространств имен. При необходимости разбейте слишком длинные операторы на несколько строк.

Любой комментарий отделяется дескрипторами, принадлежащими одному из трех наборов:

- в HTML это `<!--...-->`;
- в VB используется `'`;
- в C# возможны два вида комментариев - `/* текст комментария */` или символы `//`, указанные в начале строки;
- для серверных комментариев применяется набор `<%--...--%>`.

Комментарии обычно используются, чтобы дать полезные разъяснения по работе кода, а в самой этой работе они не принимают никакого участия.

Чтобы разбить слишком длинные операторы на несколько строк, используйте символ продолжения кода (`_`). Но если этот разделитель надо ставить посреди цепочки символов, то тогда не забывайте вначале закрыть ту часть цепочки, которая находится слева от разделителя, а затем — добавить амперсанд.

Задание 4. На основе задания 2 разработайте веб-приложение выполнения всех арифметических операций (сложение, вычитание, умножение, деление) для двух аргументов. Для оформления используйте серверные элементы управления.