

Частное учреждение образования  
«Колледж бизнеса и права»

УТВЕРЖДАЮ  
Ведущий методист  
колледжа  
\_\_\_\_\_ Е.В. Паскал  
«\_\_\_» \_\_\_\_\_ 2022

Специальность: «Программное обеспечение информационных технологий»	Учебная дисциплина: «Базы данных и системы управления базами данных»
--	--

ЛАБОРАТОРНАЯ РАБОТА № 35

Инструкционно-технологическая карта

Тема: Построение интерфейса пользователя.

Цель работы: сформировать умения осуществлять построение интерфейса пользователя.

Время выполнения: 2 часа

**Содержание работы**

1. Порядок выполнения работы
2. Пример выполнения работы
3. Контрольные вопросы
4. Литература

**1. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

1. Рассмотреть пример построения интерфейса пользователя, описанный в разделе «Пример выполнения работы» настоящей инструкционно-технологической карты.

2. Выполнить все пункты примера, представленные в настоящей инструкционно-технологической карте, и проанализировать полученные результаты.

3. Получить у преподавателя индивидуальное задание и выполнить лабораторную работу.

4. Ответить на контрольные вопросы.

**2. ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ**

ASP.NET Web Forms позволяют создавать динамическое содержимое для веб-приложения. каждая ASP.NET веб-страница создается таким же образом, как и статическая веб-страница HTML (страница, которая не включает серверную

обработку), но ASP.NET веб-страница включает дополнительные элементы, ASP.NET распознают и процессы создания HTML при выполнении страницы.

При использовании статической HTML-страницы (.htm или .html File) сервер Web выполняет запрос, считывая файл и отправляя его "как есть" в браузер. В противоположность этому, когда кто-то запрашивает ASP.NETную страницу (файла .aspx), страница выполняется как программа на веб-сервере. Во время выполнения страницы она может выполнять любую задачу, которую требует веб-сайт, включая вычисление значений, чтение или запись сведений о базе данных или вызов других программ. В качестве выходных данных страница динамически создает разметку (например, элементы в HTML-коде) и отправляет этот динамический вывод в браузер.

### Изменение пользовательского интерфейса

Необходимо изменить пользовательский интерфейс, который уже был установлен шаблоном Default.aspx по умолчанию, используемым для создания приложения. Тип изменений, которые вы будете выполнять, является типичным при создании любой веб-формы приложения. Это можно сделать, изменив заголовок, заменив некоторое содержимое и удалив ненужное содержимое по умолчанию:

1. Откройте страницу Default.aspx или переключитесь на нее.

2. Если страница отображается в режиме конструктора, переключитесь в представление исходного кода.

3. В верхней части страницы @Page директивы измените значение Title атрибута на «Добро пожаловать» (ASP.NET).

```
<%@ Page Title="Welcome" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WingtipToys._Default" %>
```

4. Кроме того, на странице Default.aspx замените все содержимое по умолчанию, содержащееся в <asp:Content> теге, таким образом, чтобы разметка была показана ниже (ASP.NET).

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent"
runat="server">
    <h1><%. Title %>.</h1>
    <h2>Wingtip Toys can help you find the perfect gift.</h2>
    <p class="lead">We're all about transportation toys. You can order
        any of our toys today. Each toy listing has detailed
        information to help you choose the right toy.</p>
</asp:Content>
```

5. Сохраните страницу Default.aspx, выбрав в меню файл пункт сохранить Default.aspx.

Полученная страница Default.aspx будет выглядеть следующим образом:

```
<%@ Page Title="Welcome" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WingtipToys._Default" %>
```

```

<asp:Content      ID="BodyContent"      ContentPlaceHolderID="MainContent"
runat="server">
    <h1><%: Title %>.</h1>
    <h2>Wingtip Toys can help you find the perfect gift.</h2>
    <p class="lead">We're all about transportation toys. You can order
        any of our toys today. Each toy listing has detailed
        information to help you choose the right toy.</p>
</asp:Content>

```

В этом примере вы установили Title атрибут @Page директивы. Когда HTML отображается в браузере, серверный код <%: Page.Title %> разрешается в содержимое, содержащееся в Title атрибуте.

На странице пример включены основные элементы, составляющие веб-страницу ASP.NET. Страница содержит статический текст, как в HTML-странице, а также элементы, характерные для ASP.NET. Содержимое, содержащееся на странице Default.aspx , будет интегрировано с содержимым главной страницы.

### **@Page Инструкции**

ASP.NET Web Forms обычно содержат директивы, позволяющие задавать свойства страницы и сведения о конфигурации для страницы. директивы используются ASP.NET в качестве инструкций по обработке страницы, но они не подготавливаются к просмотру в виде части разметки, отправляемой в браузер.

Наиболее часто используемой директивой является @Page директива, которая позволяет указать для страницы множество параметров конфигурации, включая следующие:

1. Язык программирования на сервере для кода на странице, например C#.
2. Является ли страница страницей с серверным кодом непосредственно на странице, которая называется страницей с одним файлом, или страницей с кодом в отдельном файле класса, который называется страницей кода программной части.
3. Имеет ли страница связанную главную страницу и поэтому должна рассматриваться как страница содержимого.
4. Параметры отладки и трассировки.

Если на странице не включена @Page директива или директива не содержит определенного параметра, то параметр будет унаследован из Web.config файла конфигурации или из файла конфигурации Machine.config . Файл Machine.config предоставляет дополнительные параметры конфигурации для всех приложений, выполняющихся на компьютере.

### **Серверные веб-элементы управления**

В большинстве ASP.NET Web Forms приложений вы добавите элементы управления, позволяющие пользователю взаимодействовать со страницей, например кнопками, текстовыми полями, списками и т. д. Эти серверные веб-

элементы управления похожи на кнопки HTML и элементы ввода. Однако они обрабатываются на сервере, что позволяет использовать серверный код для задания их свойств. Эти элементы управления также создают события, которые могут быть обработаны в серверном коде.

Серверные элементы управления используют специальный синтаксис, который ASP.NET распознает при выполнении страницы. имя тега для элементов управления сервера ASP.NET начинается с asp: префикса. это позволяет ASP.NET распознать и обработать эти серверные элементы управления. префикс может отличаться, если элемент управления не является частью платформы .NET Framework. в дополнение к asp: префиксу ASP.NET серверные элементы управления также включают runat="server" атрибут и ID , который можно использовать для ссылки на элемент управления в серверном коде.

При запуске страницы ASP.NET определяет серверные элементы управления и выполняет код, связанный с этими элементами управления. Многие элементы управления отображают на странице HTML или другую разметку, когда она отображается в браузере.

### **Серверный код**

Большинство ASP.NET Web Forms приложений включают код, который выполняется на сервере при обработке страницы. Как упоминалось выше, серверный код можно использовать для выполнения различных задач, таких как добавление данных в элемент управления ListView. ASP.NET поддерживает множество языков для выполнения на сервере, включая C#, Visual Basic, J# и другие.

ASP.NET поддерживает две модели для написания серверного кода для веб-страницы. В модели с одним файлом код страницы находится в элементе script, где открывающий тег содержит runat="server" атрибут. Кроме того, можно создать код для страницы в отдельном файле класса, который называется моделью кода программной части. в этом случае страница ASP.NET Web Forms обычно не содержит никакого кода сервера. @PageВместо этого директива содержит сведения, связывающие страницу ASPX со связанным файлом кода программной части.

CodeBehindАтрибут, содержащийся в @Page директиве, задает имя отдельного файла класса, а Inherits атрибут задает имя класса в файле кода программной части, соответствующем странице.

### **Обновление главной страницы**

В среде веб-форм ASP.NET главные страницы позволяют создавать устойчивый макет страниц приложения. Одна главная страница определяет внешний вид и стандартное поведение всех страниц (или группы страниц) приложения. Затем можно создать отдельные страницы содержимого, содержащие содержимое, которое необходимо отобразить, как описано выше. При поступлении запросов на страницы контента от пользователей ASP.NET объединяет их с главной страницей, чтобы получить на выходе макет главной страницы с содержимым страницы контента.

Для нового сайта требуется один логотип, отображаемый на каждой странице. Чтобы добавить этот логотип, можно изменить HTML на главной странице.

1. В Обозревателе решений найдите и откройте страницу Site.Master .
2. Если страница находится в режиме конструктора , переключитесь в представление исходного кода .
3. Обновите главную страницу, изменив или добавив разметку (ASP.NET):

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs" Inherits="WingtipToys.SiteMaster" %>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head runat="server">
```

```
<meta charset="utf-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<title><%= Page.Title %> - Wingtip Toys</title>
```

```
<asp:PlaceHolder runat="server">
```

```
<%= Scripts.Render("~/bundles/modernizr") %>
```

```
</asp:PlaceHolder>
```

```
<webopt:bundlereference runat="server" path="~/Content/css" />
```

```
<link href="~/favicon.ico" rel="shortcut icon" type="image/x-icon" />
```

```
</head>
```

```
<body>
```

```
<form runat="server">
```

```
<asp:ScriptManager runat="server">
```

```
<Scripts>
```

```
<!--To learn more about bundling scripts in ScriptManager see  
https://go.microsoft.com/fwlink/?LinkID=301884 --%>
```

```
<!--Framework Scripts--%>
```

```
<asp:ScriptReference Name="MsAjaxBundle" />
```

```
<asp:ScriptReference Name="jquery" />
```

```
<asp:ScriptReference Name="bootstrap" />
```

```
<asp:ScriptReference Name="respond" />
```

```
<asp:ScriptReference Name="WebForms.js" Assembly="System.Web"  
Path="~/Scripts/WebForms/WebForms.js" />
```

```
<asp:ScriptReference Name="WebUIValidation.js" Assembly="System.Web"  
Path="~/Scripts/WebForms/WebUIValidation.js" />
```

```
<asp:ScriptReference Name="MenuStandards.js" Assembly="System.Web"  
Path="~/Scripts/WebForms/MenuStandards.js" />
```

```
<asp:ScriptReference Name="GridView.js" Assembly="System.Web"  
Path="~/Scripts/WebForms/GridView.js" />
```

```

        <asp:ScriptReference Name="DetailsView.js" Assembly="System.Web"
Path="~/Scripts/WebForms/DetailsView.js" />
        <asp:ScriptReference Name="TreeView.js" Assembly="System.Web"
Path="~/Scripts/WebForms/TreeView.js" />
        <asp:ScriptReference Name="WebParts.js" Assembly="System.Web"
Path="~/Scripts/WebForms/WebParts.js" />
        <asp:ScriptReference Name="Focus.js" Assembly="System.Web"
Path="~/Scripts/WebForms/Focus.js" />
        <asp:ScriptReference Name="WebFormsBundle" />
        <%--Site Scripts--%>
    </Scripts>
</asp:ScriptManager>

<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" runat="server" href="~/>Wingtip Toys</a>
        </div>
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
                <li><a runat="server" href="~/>Home</a></li>
                <li><a runat="server" href="~/About">About</a></li>
                <li><a runat="server" href="~/Contact">Contact</a></li>
            </ul>
            <asp:LoginView runat="server" ViewStateMode="Disabled">
                <AnonymousTemplate>
                    <ul class="nav navbar-nav navbar-right">
                        <li><a
runat="server"
href="~/Account/Register">Register</a></li>
                        <li><a
runat="server"
href="~/Account/Login">Log
in</a></li>
                    </ul>
                </AnonymousTemplate>
                <LoggedInTemplate>
                    <ul class="nav navbar-nav navbar-right">
                        <li><a
runat="server"
href="~/Account/Manage"
title="Manage your account">Hello, <%= Context.User.Identity.GetUserName() %>
!</a></li>
                        <li>

```

```

        <asp:LoginStatus runat="server" LogoutAction="Redirect"
LogoutText="Log off" LogoutPageUrl="~/ " OnLoggingOut="Unnamed_LoggingOut"
/>

        </li>
    </ul>
    </LoggedInTemplate>
</asp:LoginView>
</div>
</div>
</div>
<div id="TitleContent" style="text-align: center">
    <a runat="server" href="~/ ">
        <asp:Image ID="Image1" runat="server" Im-
ageUrl="~/Images/logo.jpg" BorderStyle="None" />
    </a>
    <br />
</div>
<div class="container body-content">
    <asp:ContentPlaceHolder ID="MainContent" runat="server">
    </asp:ContentPlaceHolder>
    <hr />
    <footer>
        <p>&copy; <%: DateTime.Now.Year %> - Wingtip Toys</p>
    </footer>
</div>
</form>
</body>
</html>

```

В этом HTML-коде будет отображаться образ с именем logo.jpg из папки Images (изображения), которую вы добавите позже. При отображении страницы, использующей главную страницу, в браузере отображается эмблема. Если пользователь щелкнет эмблему, он вернется на страницу Default.aspx. Тег <a> привязки HTML служит оболочкой для элемента управления сервера изображений и позволяет включить изображение в состав ссылки. hrefАтрибут для тега привязки указывает корневой элемент " ~/ " веб-сайта в качестве расположения ссылки. По умолчанию страница Default.aspx отображается, когда пользователь переходит в корневую папку веб-сайта. Серверный элемент управления Image<asp:Image> содержит дополнительные свойства, такие как BorderStyle, которые отображаются в формате HTML при отображении в браузере.

### Эталонные страницы

Главная страница — это ASP.NET файл с расширением master (например, Site.master) с заранее заданным макетом, который может включать в себя статический текст, элементы HTML и серверные элементы управления. Эталонная

страница определяется специальной @Master директивой, которая заменяет @Page директиву, используемую для обычных страниц .aspx .

Кроме @Master директивы, Главная страница также содержит все элементы HTML верхнего уровня для страницы, такие как html , head и form . Например, на главной странице, добавленной выше, вы используете HTML table для макета, img элемент для эмблемы компании, статический текст и серверные элементы управления для решения общего членства в вашем сайте. в качестве части главной страницы можно использовать любые элементы HTML и ASP.NET.

В дополнение к статическому тексту и элементам управления, которые будут отображаться на всех страницах, Главная страница также включает один или несколько элементов управления ContentPlaceHolder . Эти элементы управления заполнителями определяют регионы, в которых будет отображаться заменяемое содержимое. Заменяемое содержимое, в свою очередь, определяется на страницах содержимого, таких как Default.aspx, с помощью серверного элемента управления Content .

### 3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое «Пользовательский интерфейс»?
2. Опишите @Page инструкции.
3. Охарактеризуйте эталонные страницы.

### 4. ЛИТЕРАТУРА

1. Постолиит, А. Visual Studio .NET: разработка приложений баз данных / А. В. Постолиит. – СПб.: БХВ-Петербург, 2003. – 544 с.: ил.
2. Сеть разработчиков Microsoft [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/aspnet/web-forms/overview/older-versions-security/admin/building-an-interface-to-select-one-user-account-from-many-cs>
3. Пользовательский интерфейс и навигация [Электронный ресурс]. – Режим доступа: [https://docs.microsoft.com/ru-ru/aspnet/web-forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/ui\\_and\\_navigation](https://docs.microsoft.com/ru-ru/aspnet/web-forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/ui_and_navigation)

Преподаватель

В.Ю.Купцова

Рассмотрено на заседании цикловой  
комиссии программного обеспечения  
информационных технологий №10  
Протокол № \_\_ от «\_\_» \_\_\_\_\_ 202\_\_  
Председатель ЦК В.Ю.Михалевич