МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» ЮРГИНСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ

> УТВЕРЖДАЮ Зам. директора по УР ЮТИ ТПУ

_____В.Л. Бибик « __ » _____ 2016 г.

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ

Методические указания к выполнению лабораторных работ по курсу «Международные информационные ресурсы и стандарты информатизации» для магистров II курса, обучающихся по направлению 230700 «Прикладная информатика в аналитической экономике»

Составитель М. А. Корчуганова

Издательство Томского политехнического университета 2016 УДК 004 ББК 32.81 П79

П79 Проектирование информационных ресурсов: методические указания к выполнению лабораторных работ по курсу «Междуна-родные информационные ресурсы и стандарты информатизации» для студентов II курса, обучающихся по специальности 230700 «Прикладная информатика в аналитической экономике». / сост.: М.А. Корчуганова; Юргинский технологический институт. – Юрга: Изд-во Юргинского технологического института (филиала) Томского политехнического университета, 2016. – 57 с.

УДК 004 ББК 32.81

Методические указания рассмотрены и рекомендованы к изданию методическим семинаром кафедры Информационные системы ЮТИ ТПУ «<u>21</u>» января 2016 г.

Зав. кафедрой ИС Кандидат технических наук, доцент ______А.А. Захарова

Председатель учебно-методической комиссии ______Е.В. Молнина

Рецензент

Кандидат технических наук, доцент ЮТИ ТПУ *Т.Ю. Чернышева*

> © Составление. ФГБОУ ВПО НИ ТПУ Юргинский технологический институт (филиал), 2016
> © Корчуганова М.А., составление, 2016

1. ЛАБОРАТОРНАЯ РАБОТА № 1

Тема работы: *Разработка дизайн-проекта веб-страницы.* **Цель работы:** Создать листинг и разработать дизайн веб-страницы **Теоретическая часть**

Веб-дизайн (*web page design*) – это процесс проектирования, планирования, моделирования и реализации доставки электронного содержимого через сеть Веб с использованием технологий (на основе языков разметки), подходящих для интерпретации и визуализации веббраузером или другим графическим веб-интерфейсом пользователя.

Очень важным является приведение веб-ресурса в соответствие стандартам W3C, что обеспечивает доступность содержания для людей с ограниченными физическими возможностями и пользователей портативных устройств, а также кроссплатформенность кода разметки ресурса.

Основными аспектами веб-дизайна (в первую очередь для коммерческих веб-сайтов) являются следующие:

- Содержимое. Информационное наполнение должно быть релевантным тематике сайта и ориентированным на целевую аудиторию сайта.
- Удобство в использовании (*usability*). Сайт должен иметь дружественный пользовательский интерфейс с простой и надежной навигацией.
- Внешний вид. Графика и текст должны быть выполнены в одном стиле и согласованы на всех страницах сайта. Стиль оформления должен демонстрировать профессионализм, привлекательность и релевантность.
- Видимость. Сайт должен легко находиться с помощью большинства поисковых систем и рекламных площадок.

По-сути, веб-сайт - это информационная система, которая содержит две основные компоненты:

- Компоненту представления (*front-end*). Видимое содержимое (разметка страниц, графика, аудио и текст).
- Компоненту реализации (*back-end*). Связана с организацией и эффективной реализацией исходных кодов. Включает в себя неотображаемые сценарии, серверные компоненты, являющиеся основой для компоненты представления.

Составлением технического задания на разработку веб-сайта для специалистов занимается менеджер проекта. Работа с заказчиком начинается с составления краткого описания, в котором излагаются пожелания заказчика по *визуальному представлению* и *структуре* сайта. С учетом возможностей программных и дизайнерских средств и исходя из краткого описания, *менеджер* составляет *техническое задание*, которое должно быть *утверждено заказчиком*.

Этапы проектирования веб-сайта зависят от объема сайта, его функциональности и многого другого.

Разработка веб-сайта включает в себя следующие этапы:

- 1. Дизайн главной и типовых страниц сайта. (выполняется обычно в графическом редакторе).
- 2. HTML-кодирование, в результате которого создаётся код, который можно просматривать с помощью браузера.
- Программирование сайта. Может осуществляться как "с чистого листа", так и с помощью специального высокоуровнего пакета системы управления сайтом (CMS).
- 4. Размещение сайта в Веб, наполнение контентом и публикация.
- 5. Оптимизация веб-сайта с целью повышения его видимости в Веб.
- 6. Сдача сайта заказчику.

Для дизайна веб-сайта и разработки веб-приложений имеется широкий спектр инструментальных средств. Компания Майкрософт имеет уже давнюю традицию создания WYSIWYG-инструментов такого рода. Среди таких инструментов можно назвать, в частности *Microsoft Visual Studio. NET* и *Microsoft Expression Web*.

Microsoft Expression Web

Microsoft Expression Web является составной частью пакета Microsoft Expression Studio.

Microsoft Expression Studio - пакет графических программ компании Майкрософт, предназначенных для разработчиков и дизайнеров. Использует программный интерфейс *WinFX*.

WinFX (теперь NET Framework 3.0) — интерфейс программирования приложений, входящий в состав операционных систем Windows Vista и Windows Server 2008.

В состав *Microsoft Expression Studio* входят следующие инструменты:

- *Microsoft Expression Blend* WYSIWYG-инструмент для проектирования пользовательского интерфейса приложений, создаваемых на основе Windows Presentation Foundation.
- *Microsoft Expression Web* WYSIWYG-инструмент для дизайна веб-сайтов и редактор HTML.
- *Microsoft Expression Media* векторно/растровый графический редактор.

- *Microsoft Expression Design* профессиональный инструмент для графического дизайна и подготовки иллюстраций для настольных и web-приложений.
- *Microsoft Expression Encoder* профессиональный мультимедиакодек для содержимого в формате VC-1 для *Microsoft Silverlight*.

Microsoft Expression Web 2 позволяет разрабатывать вебстраницы, интегрирующие в рамках веб-сайта такие технологии как *XML*, *CSS 2.1*, *ASP.NET 2.0*, *XHTML*, *XSLT* и *JavaScript.*, а также внедрять в страницы управляющие элементы ASP.NET, Microsoft Silverlight и AJAX. Для работы с этим инструментом требуется наличие установленной компоненты .*NET Framework 2.0*.

Необходимо уделить особенное внимание созданию мастерстраниц.

Поскольку страницы веб-приложения составляют некое единое целое, объединенное едиными требованиями к оформлению, содержанию, расположению элементов управления, то создаваемые вебстраницы сегодня очень редко разрабатываются "с нуля". Обычно при создании страниц используются готовые шаблоны, либо они создаются в процессе разработки веб-страниц приложения. Одним из средств решения подобных задач являются *мастер-страницы* (master pages). Они реализуют простую модель создания шаблонов форм с возможностью их повторного использования.

Для реализации данного механизма существуют такие типы страниц, как *мастер-страницы* (master pages) и *страницы содержимого* (content pages).

- Мастер-страница представляет собой шаблон страницы, который может содержать любые элементы, допустимые для обычной страницы, а также программный код. Обычно мастерстраница содержит фиксированные элементы, одинаковые для всех страниц, и заполнитель содержимого для остальной части страницы. Наиболее типичными фиксированными элементами являются верхний и нижний колонтитулы, панель навигации, панель меню и другие.
- Страница содержимого включает в себя допустимые элементы управления и с помощью их определяет содержимое, которым заполняются специальные области мастер-страниц. Каждая страница содержимого ссылается только на одну мастерстраницу из которой она получает элементы. Помимо фиксированных элементов мастер-страницы она также предоставляет и дополнительное содержимое.

Мастер-страница должна содержать элемент управления *ContentPlaceHolder*, который предназначен для определения области, куда страница содержимого может вставлять содержимое. При создании новой мастер-страницы элемент *ContentPlaceHolder* создается по умолчанию. Для того чтобы превратить обычную страницу в страницу содержимого, необходимо в качестве значения свойства *MasterPageFile* страницы указать имя мастер-страницы, а также добавить на страницу нужный элемент управления.

Ход работы:

1. Создание HTML страницы в Microsoft Visual Studio. NET.

Для дизайна веб-страниц в Visual Studio. NET необходимо выбрать раздел меню **File => New** и в открывшемся окне выбрать тип файла HTML Page (puc.1.1):

🤹 s	olution	2 - M	icros	soft \	/isual §	itudio							-		K
File	Edit	View	Pro	oject	Debug	Data	Tools	Test	Analyze	Window	Help				
16	- 55	- 💕		3	X 🗈	12. 4	7 - (*		- □↓ ▶	Debug	 Any CPU 		~ 📝	9	
: 12	ie 15	8	₽,	家日	e 1	3. A.	14	# "E	9	ф ц.		0 . 1 (S	「二字目	110	
													<u> </u>		Î
													-		
N	ew File)												? 🔀	
	ategorie	s:				Те	mplates:								
	Gen	eral					Visual S	itudio i	nstalled to	emplates					
	Perh Web Visu Sorip	. page t	that c	an inc	lude clier	nt-side O	Text Style KML S Bitma T Cursu Visua Visua	File Sheet Schema up File or File I C# Clas	55		MTML Page	ce Template ass			
													_		
												Upen •	Cano	.61	
Code	Definitio	n Wind	low											→ ₽ >	×
	No de	Tini	t10	n se	lecte	a									-
	ode Defir	nition V	/indov	v 🖃	Find Res	ults 1				-			_	>	
	ind Sund	ol Per	ubell	1 50	or Liet										
1.000 (Mar)	ing Syllic	JUINUSI	ato (1	or cor U	a outr									

Рис. 1.1. Окно выбора типа файла

Для дизайна имеется панель Toolbox, содержащая набор основных элементов html страницы (подключается через меню: **View > Toolbox**):

STUTIE Trace 7 Inter- Microsoft	P Woost Studia		
Ele Edit Mars Basiant Date	s visual studio		
	uy una ruman nuos res anapar minum rep		
N I I I I I I I I I I I I I I I I I I I	■ 2. mi k k = 2 2 [] ちゅうき 46 10 4 3 [10 14 24 [10 14 3] 14 3 [15 3 [16 3]	Style Application: Manual	• (//////// •
Toobox • # X	HTMLPage3.htm		• ×
le Pointer	Client Objects & Events 🖉 (No Events)		×
 Input (Button) 	<pre><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xht</pre> </pre>	ml1/DTD/shtml1-tro	ansitional.dtd*
🖄 Input (Reset)	<pre>ektnl xnlmg="http://www.w3.org/1999/whtml" > cheads</pre>		
🖄 Input (Submit)	<title>Untitled Page</title>		
ali Input (Text)	-		
adi Input (File)	E <000di>		1
Input (Peesword)	-		
 Input (Crecipius) Input (Radio) 	-		
Input (Hidden)			
Textarea			
Table			
🛃 Image			-
5 Select	4		3
 Horizontal Rule 			<u></u>
E General			
	1.5		3
	Gesign Split 🗄 Source 🤄 chitml> chicaty>		Þ
Code Definition Window	P		- 0 ×
No definition select	ted		-
4			2
Code Definition Window 🙀 Find Ri	lende 1		
A Find Symbol Results 🚡 Error List	A data		
Ready	in 7	Col 1 Ch 1	DNS

Рис. 1.2. Меню основных элементов

 Создание мастер-страницы в Microsoft Visual Studio. NET. Для создания мастер-страницы необходимо выбрать раздел меню
 File => New и в открывшемся окне выбрать тип файла Web > C# > Master Page:

Ø Solution2 - Microsoft Visual Studio	
File Edit View Project Dabug Data Tools Test Analyze Window Help	
🛐 • 🔄 • 🛤 🖬 🖉 V. P. 🗮 LO - D - EL EL S. Debra - Jose COU - J DA	
Rev File	et tit sue ti
Toobox Categories: Templates:	
General General Visual Studio installed templates	
Performance Web Form Masker Page Visual Basic Web Service HTML Page Visual C++ Web Service Gobal Application Class Script Web Configuration File MML File Text File Text File Mesource File Script File Script File Script File	
A Master Page for Web Applications	
Code Definition	- 9 X
No definition selected	-
¢	>
Code Definition Window 🙀 Find Results 1	

Рис. 1.3. Меню выбора файла

После этого появится окно мастер-страницы, содержащей пустую форму и два контейнера содержимого ContentPlaceHolder (в заголовке и в теле документа):

MasterPageZ.master - Micro	oseft Visual Studie	×				
File Edit View Project Debu	ag Duta Pormat Toola Taat Analyza Window Help					
187 • 🗉 • 🥔 🖬 🖉 🕺 🔤	a 🏡 ヴ・ヴ・ 🗐 + 🗓 ト Debug - Kery CPU - 😸 CP - 🛛 🖓 😁 🔹 + 🖓 😁 🐨 - 🚽	- 1				
ENDER A REALTING		12				
Teober - P X	Made and Table 1					
🗄 Standard 🧄		÷.				
R Pointer	Server Objects & Events (No Events)	×				
A Label	ste Baster Language**C#* to					
at TextBox	CINCTURE STALL BIRLING S-CONCULTED VETRIL 1. O. Transitional/CERF Entractions of and/Dicktell/PDF/shrell-transitional drifts					
He Button	CIDENTIFIC HURT FORME					
LinkButton	<pre></pre>					
InageButton						
A HyperLink	- «/seript>					
DropDownList	The characteristic sector and a sector of the sector and the sector of t					
EistBox	Contral manner "Attention" (Attention of the sector)					
CheciBox	cities Untiled Page / title>					
CheckBoxList	<asp:contentplaceholder id="head" runat="server"></asp:contentplaceholder>					
 RadioButton 						
📋 RadioButtonList	-					
📑 Inage						
🔛 InageMap	Contra los formas functions functions					
Table	<pre>dasp:ContentPlaceHolder id="ContentPlaceHolder1" runst="server"></pre>					
; BulletecList						
HiddenField	-					
E Liberal	- (11)					
🚰 Calendar	-					
AdRotator	- () body-	~				
S Fleupload	c					
\$* Wibard		1				
🔝 Xani						
MultiView						
Panel		~				
Diacet-tokler	4					
View	Georgen 🖾 Spik 🗄 Source d Chimits doody.>	E				
Code Definition Window		x				
No definition select	ed					
<		2				
Code Definition Window 🛒 Find Re	981					
🚑 Find Symbol Results 👸 Error List						
Ready	Ln 16 Col5 Cb5 DB5					

Рис. 1.4. Окно мастер страницы

3. Знакомство с Microsoft Expression Web

Microsoft Expression Web можно использовать в качестве визуального редактора документов разнообразных форматов: HTML, ASP.NET, PHP, CSS, JavaScript, XML. Поддерживается работа с шаблонами, мастер-страницами и веб-сайтами в целом. Выбрав раздел меню **File > New**, вызовите следующее диалоговое окно:



Рис. 1.5. Диалоговое окно

Выберите формат документа HTML.

В результате появится следующее окно, содержащее пустую HTML страницу:

Untitled 1 (Untitled_1.html) - Mi		
		Type a question for help
🗋 = 🥁 = 🚮 🖄 = (None)	- (Defaul Ford) - (Defaul Size) - (ローワー) B ズ 豆 (野 書 道) 旧 田 伊 伊 (田・型・山・田 田 緑 後)	
Folder List D X	Untitled_Literal	Теовен 🗆 🗙
	a dadya	- HTML
		🖂 Tags
	1 //EW "http://www.wd.org/TB/Xhtal1/DTD/Xhtal1-transition	III <dv></dv>
	Charles And the Incertation of the Construction of the Constructio	(iii) <apan></apan>
	4 chead>	4 Dreak
	<pre>{ casta content="text/html; charget=utf=6" http=equiy="Content-Type" /></pre>	Im Horizontal Line
	<pre>< <ili> </ili></pre>	🛃 Inege
		11 Inline Frame
	3 chottp	Ciflaver
	10 Parts	Paragraph
	12	E form Controls
	13	ad advanced & the
	15	The Course Barry
		T Form
		(¹⁰¹) Gran Barr
		(Direct Direct)
		Direct (Deather)
		P Input (Cheobax)
		All trace (ree)
		int Input (hoden)
		lig Input (Image)
Tag Properties D X		(in Input (Password)
25 AL 26 🛷		() Input (Radio)
E 2 + 1 = 1 22		C Input (Hesel)
<pre><body></body></pre>		Conput (Submit)
- Attributes		Mil Input (Text)
Stra.		A Label
- basig		20 Test Area
ge data		⊗ Hedia
🚰 dr		< ASPJRET Controls
19 H		Standard
gene en		🗷 Deta
2 lik		Validation
- Style		× Navigation
of the	()	a Waldhate
🖤 vink		E ATAX
😤 xmitlang		
- Events		Manage Styles 🛛 🗙
✓ ondick		A New Style Options -
/ operation		All Allack Tale These
∉ onkeyp		(20 wright other overall
🖉 onkeyup 🔛	Gibesign Bisplit Dicode <	CSS styles:
For Help, gress F1	Visual Aids: On Style Academic Auto 317 hydres: Standard	813 × 117 XHTML 1.0 T CS5 2.1

Рис. 1.6. Окно НТМL страницы

Используя панель инструментов (Toolbox), создайте веб-страницу для регистрации пользователя следующего вида:

Perietrpauea (C:\Documents an	d Settings/sav/)Moe.govywwww/(My Web Sites/Registration.html) - Microsoft Expression Web	
i Eile Edit View Insert Forma	t Tools Tyble Site Data View Task Panes Window Help	Type a question for help
🗋 = 🥁 = 🛃 🖄 = (None)	- (Detsut Ford) - (Detsut Sos) - ヴ・ワー B ズ 豆 🎫 第二日 日 課 課 🔲 - 🖄 - 🛆 - 🗂 🗉 🛃 😣 🤅	
Folder List 📴 🗋 🗆 🗙	Web Ste Registration.html ×	Теоїбен 🗆 🗙
CI/Documents and Settings'ssav(More	daatys citiva citiva citiva citiva citiva citiva data citivativa citivativa	× HTML
Registration.html		Tags
	1 < DOCTYPE http://www.wd.org/TB/Xhtml//DD/Xhtml1-transitional//EW "http://www.wd.org/TB/Xhtml1/DD/Xhtml1-transition < doctype / http://www.wd.org/TB/Xhtml1/DD/Xhtml1-transitional//EW "http://www.wd.org/TB/Xhtml1/DD/Xhtml1-transition	💷 ediyə
		(E) <apan></apan>
	4 chead>	4 Dreak
	S Casta content="text/html; charget=utf=8" http=equiv="Content-Type" /> S Casta content="text/html; charget=utf=8" http=equiv="Content-Type" />	Horizontal Line
	<pre><pre>ctude type 'text/od'></pre></pre>	🛃 Inege
	8.style1 (Inline Frame
	9 text-align: left;	Layer
	10 }	T Pwragraph
	12 c/head>	- Form Controls
	13	All Advanced By days
	14 CONT	The Deve Deve Ber
	16 <fors action*""="" methods"post"=""></fors>	17 From
	17 odivo-clabel id-"kabell">Hus c/label>cinput name-"name" type-"text" />	1 ⁽¹⁷⁾ Comp Park
	10 CdiveCabel 1d*"Imbell">Opraserseper/Kommaser (/label>Cinput name*"company" type*"text" /> 10 cdive	Coop bas
	20 Clabel id-"Label3">+-mail Cinput name-"smail" type-"text" />	Contract (Souther)
	21 <div class="atyle1"></div>	P Input (cheobax)
	dr D	Tubre (use)
	24 (/0.7)	Triple (noter)
4 3	23 <th>ing troot (mage)</th>	ing troot (mage)
Tag Properties 🛛 🗙	26 <th>(in Input (Password)</th>	(in Input (Password)
25 AL 96 14	and states	Contract (1996)
Bec 2+ 11 04	23 <th>Curput (reset)</th>	Curput (reset)
and a	30 11 c (http://	E Inper (seamy
 Attributes Attributes 	22 22 22 22 22 22 22 22 22 22 22 22 22	(all input (rext)
tune store		A LOOP
🛫 value Опправить		Les Test Area
🛫 accept		🗷 Hedia
😭 accesskey	£ 5	× ASPJIET Controls
r alon		Standard
at a started	Ина	Deta Validation
re data	(Province) and (Company)	E Navination
ge dr	Opinintal percontainint	E Login
😤 disabled	e-mail	WebParts
· 문 년	(reut)	XALA E
🚰 lang	Ompasers D	
and reachedy		Planage Styles D X
in size		A New Style Options -
🚰 arc		Al Attach Style Sheet
🛫 style		CCC abelow
🛫 tabindex 🛛 💌	Gobesign BSplit. D Code <	Cost Styles
	Ungelählte die Onle destinations die des Onle destinations die des Onle des	avairation and cost of the second

Рис. 1.7. Страница регистрации пользователя

Для удобства вставки элементов формы рекомендуется использовать в качестве контейнера элемент <div>.

После сохранения документа (**File > Save As**) можно предварительно посмотреть страницу в одном из веб-браузеров:

Perioti	грация (С:\Docum	ents	and Settings\sav\Mow.goxywermi/My Web Sites\Registration.html) - Microsoft Expression Web			
Die D	dit <u>View</u> Insert		rmat Tools Tyble Site Data View Task Panes Window Help		Type a question	for help
D Nev	w +	(2)	・ (Default Ford) ・ (Default Stor) ・ (キ・ト・) B エロ (原業業)(日日課課)(日・公・人・) B	- 12.	3.	
📑 🔐 💷 ри	en Ctrl+D	D	X Da Web The Registration.html	×	Теорон	_ × 0
👏 Орн	en Sjte	savi	Are a Constant Colored Descent Colored Colored Colored Technology		× HTHL	
Cha		1	i genta class crass cass cass caus causes	2	🖂 Tags	
200		L 1	<pre></pre>	-transition	edv>	
clo	ICE SIDE		<pre>chtal sains="http://www.ws.org/1999/shtal"></pre>		T&T <upanto< td=""><td></td></upanto<>	
Sau Sau	re Ctri+S	L 1	4 chests		Strack	
😹 Sam	re As	L 1	5 Casta content-"text/html; charpet-utf-8" http-equiy-"Content-Type" />		Havingstal Line	
64 Sam	16.00		<pre>6 <title>Permorp aggec/title></title></pre>		Citizana (
10°		1	<pre>> <style type="text/cdd"> = ======</pre></td><td></td><td>111 Index Come</td><td></td></tr><tr><td>Pyt</td><td>blish Site</td><td>L 1</td><td>9 text-aligns lefts</td><td></td><td>Charles Charles</td><td></td></tr><tr><td>107</td><td>port +</td><td>L 1</td><td>10 }</td><td></td><td>1.Lt Layer</td><td></td></tr><tr><td>Exp.</td><td>e tro</td><td>L 1</td><td>11 </style></pre>		 Paragraph 	
	ning in Romany -	0			Form Controls	
Pie	wew in Browsel +	9	1 windows atteinet explorer 7.0 P12		Advanced Button	
Etin	nt •	2	2 Windows Internet Explorer 7.0 (640 x 450)		Drop-Down Box	
Ppo	iperties	E.	3 Windows Internet Explorer 7.0 (800 x 600)		Form	
Ber	cent Files •	8	4 Windows Internet Environm 7.0 0034 y 168		1 ¹¹¹ Draw Bay	
		-	2 And a set of the set		(a) (next (B) (free)	
KAC	ceuf pares .	P	<pre>pe-mail cinput mame*"smail" type"text" /></pre>		E Insut (Chaddow)	
X Eyt	t	12	6 Mozilia Firefox 20.0.7 (640 x 480)		Diplot (Creckbort)	
		2	Z Mozilla Firefox 20.0.7 (900 x 600)		There (ne)	
		R.	8 Mozila Firefox 2.0.0.7 (1024 x 768)		Inter Englis (Hoosen)	
<		_			Input (Image)	
Tag Prope	erties		Eleview in Multiple Browsers		Ittl Input (Password)	
(m) + 1			Enview in Multiple Browsers (540 x 450)		Input (Radio)	
E. 24	1 M GT		Preview in Multiple Browsers (800 x 600)		C Input (Reset)	
sinput>			Preview in Multiple Browsers (1024 x 768)		C Input (Submit)	
- Attribe	utes				M Input (Text)	
an 19	ster aubmit	1	Edit Snowser List		A Label	
🛫 typ	pe submit				Ell Text Area	
ge val	lue Ompatirru				E Heda	
acce	apt				- riedu	
2 acce	essney		1 23 P	2	- Renderd	
100	n			^	- Stanward	
100	ched		Nast		 Validation 	
and dep	us .		Optimized and the second		 Valuation 	
🗬 dr			op an automatic at opposite and		T Login	
🛫 daa	bled		e-mail		+ WebParts	
🛫 id			100		XALA X	
🛫 lang	9		Ompasing D			
er mex	xlen				Manage Styles	
read	donly				And New Style	
in size						
90	-				Attach Style Sheet	
tabl	index				CSS styles:	
1.27 000			Create Caller Constant	2	Townset Bans	
			Visual Alds: On Style Application: Auto 823 8	ytes Standard	013 x 218 XHEML L.0 T	C55 2.1

Рис. 1.8. Окно вызывающее просмотр страницы браузером

2. Создание веб-сайта.

Для создания пустого веб-сайта выберите раздел меню File > New, затем на закладке Web Site выберите Empty Web Site и укажите расположение сайта:



Рис. 1.9. Окно указания адреса сайта

В качестве шаблона для создания страниц веб-сайта можно использовать мастер-страницу:

Ct/Inetpub/wwwroot/MHSiteDemo - Microsoft Expression Web	_ = ×
i Eile Edit View Insert Format Iools Table Site Data View Task Panes Window Help	Type a question for help
1.4.	🗆 • 🍳 • A • 🗆 🔅 🖁
Folder List	nlbox 🗆 🗆 🗙
G:Unetpublikwww.root/MHSiteDer Page Web Ske	ITHL 🔨
🕼 default.html 🔹 👘 HTML Description	lags
General ASPX Create a Master Page for Vielo	<dv></dv>
ASP.NET Disp.pp Applications.	(span)
CS CS Madder Enge	Break
C35 Layouts	el Horizontal Line
Style Sheets 🙀 JavaScript	I Inloe Frans
Terman Bonar	Laver
Province region	[Paragraph
Create from Dynamic Web Template	Form Controls
Tag Properties	El Advanced Button
00 A 00 000	Trop-Down Box
96 2* 71 49	Form
Cpoors -	Group Box
Page Editor Uptions	a) Input (Button)
Programming Language: Cgr 🐱	/ Input (Cheddox)
	nage Styles 🛛 🗙
	New Style Options 🔻 🚔
OK Cancel	Attach Style Sheet
	5 styles:
CIFolders CI Remote Web Site CI Reports Ci Hyperlaks	×

Рис. 1.10. Мастер страница

Созданная мастер-страница содержит два контейнера содержимого ContentPlaceHolder: в заголовке и в теле документа:



Рис. 1.11. Формирование контейнеров содержимого заголовка и тела документа

Все изменения, вносимые в эту мастер-страницу будут автоматически реплицироваться во всех документах веб-сайта, созданных на ее основе. Вся специфика вновь создаваемых документов должна быть локализована в пределах контейнера содержимого (ContentPlaceHolder). Например, в заголовке создадим общее заглавие "My Home Web Site Demo" для всех страниц веб-сайта:

<head runat="server">

<meta content="text/html; charset=utf-8" http-equiv="Content-Type" /> <title>My Home Site Demo</title>

<asp:ContentPlaceHolder id="head" runat="server">

<div>My Home Web Site Demo</div>

</asp:ContentPlaceHolder>

</head>

Для создания новой веб-страницы на основе мастер-страницы необходимо выбрать в панели списка файлов мастер-страницу и затем в контекстном меню (вызывается кликом правой кнопки мыши) выбрать раздел "New from Master Page":

i Ede Edit View Inset Format Tools Toble Site Data View Task Panes Window Help Prove Panes Panes Prove Panes	🛃 My Home Site D	ето	(C:\Inetpub\wwwroot\Mi	HSiteDemo\MHSDemo.master) - Microsoft Expression Web	
Cluber of the second of t	Eile Edit ⊻iew		ert Format Tools Tabl	ie <u>S</u> ite <u>D</u> ata View Tas <u>k</u> Panes <u>W</u> indow <u>H</u> elp	Type a question for help .
Pader ist Image Styles Manage Styles X C()Introduction Image Styles X Image Styles X Image Styles X Image Styles	i 🗈 • 🚅 • 🛃 🖄	-	None) • (Default F	ant) ・ (Default Size) ・ ヴ・ピ・ B Z 亚 (新著道) 旧語 讲 律	[□•型•▲•[■三县_]
Constraints Constrats Constraints Constraints Constraints Constraints	Folder List		🗋 🗖 🗙 🔛 Web Site	MHSDemo.master X	Toolbox 🗆 🗙
Tag Properties A site shore Page de"Coral" runat="server"> Tag Properties X Rename pri ContentPlaceBolder: sid="Coral" runat="server"> ************************************	C:\Inetpub\www.o default.html	ot (140-	SkiDonn i Hou Ski SkiDonn i Kondys drs Lethours Open With Lethours Den in New Window New from Master Page Piertew in growser Cut Cost Sery	The Analysian and the analysia	- HTHL - HTHL - Tags - Tags - Assylict Controls - Assylict Controls - Assylict Controls - Assylict Controls - Standard - Dota - Sola - Validation - Login - Login - WebParts - AJAX
ger accopt (A descript	Tag Properties	×	Set as Home Page Rename Delete Publish Selected Files Don't Bublish Ptoperties	<pre>id="form!" runst="server"> pp:ContentFlaceHolder1" runst="server"> spp:ContentFlaceHolder> </pre>	
27 Increase CSS styles: 2 28 reshod 2 2 2 28 reshod 2 2 2 28 reshod 2 2 2	accept accept		Cibesgn Bspi	Code (Manage Styles X Ad New Style Options Attach Style Sheet CSS styles: V

Рис. 1.12. Вызов мастер-страницы

Далее, в режиме отображения "Split" для элемента "ContentPlaceHolder1" по стрелке справа от него следует выбрать из контексного меню раздел "Create Custom Content":

It Edit Yee Prove Question for help Image Styles Image Styles Image Styles	🛃 Untitled 1 (Untitled_1.aspx) - M	crosoft Expression Web	- 8 X
Image Styles Tag Properties X ContextPlaceHolder1 (Master)	ÉEile Edit ⊻iew (nsert Forma	t Iools Tgble Site Data View Task Panes Window Help	Type a question for help
Folder List X Toolbox X C (Unetpub/www.od/N+Station) HTHL HTHL MH2Demo.master If www.od/N+Station.master HTHL MH2Demo.master If www.od/N+Station.master If ags I collabor Standard Base Insprage="C#" hasterpagefile="file:///C:/Instpub/www.col/NHSDemo.master I collabor I collabor Standard I collabor X Standard I collabor X Standard I collabor X Standard I collabor X Validation I collabor X Validation I collabor X X I collabor	📄 🔹 🎽 🔹 🛃 🖄 🔹 (None)	(Default Fort) (Default Size) (Default Size) (Default Size)	目 律 律 田 • ⁹² • ▲ •]
ContentPlaceHolder1 (Master) Content Tasks C	Folder List Image: Comparison of the	Web Ste NHSDemo.master Untitled_1.aspx* × fme:///C:/Intrpub/www.ost/NHSDemo.master fme:///C:/Intrpub/www.cot/NHSDemo.master 1 1<<% Page language="C#" masterpagefile="file:///C:/Intrpub/www.cot/NHSIteD"	Toolhox □ X - HTHL E Tags E Torn Controls E Hedia - ASP/RET Controls E Standard E Data E Validation E Iogin E Uogin E WebParts E AJAX
Construction (version)		Contract/Northaldert (Martea)	Manage Styles
		ContentPlaceHolder1 (Master)	ontent Yasks pora V reate Custom Cortext CSS styles:

Рис. 1.13. Форма добавления содержания страницы

В результате в коде документа появятся пара тегов <asp:Content></asp:Content>, между которыми может быть добавлено новое содержимое страницы.

В качестве примера рассмотрим следующее содержимое:



Рис. 1.14. Пример добавления календаря на страницу.

После сохранения документа его можно просмотреть в веб-браузере:

🌈 Календарь - Windows Ir	iternet Explorer	
💽 🗸 🖉 http://localho	st:61190/N 😪 😽 🗙 Google	₽ -
🚖 🔅 🌈 Календарь	🔄 📩 🔹 📾 🔹 🔂 Страница	* *
My Home Web Site De	mo	~
	Календарь	
	≤ Январь 2009 г. ≥	
	Пн Вт Ср Чт Пт Сб Вс	
	<u>29 30 31 1 2 3 4</u>	
	<u>5 6 7 8 9 10 11</u>	
	<u>12 13 14 15 16 17 18</u> января 076 17 18	
	<u>19 20 21 22 23 24 25</u>	
	<u>26 27 28 29 30 31 1</u>	
	<u>∠ 3 4 5 6 / 8</u>	
		<u>×</u>
javascript:doPosl	💇 Местная интрасеть 🔍	100% 🔹 🔡

Рис. 1.5. Веб-страница с календарем

Контрольные задания

- 1. Вставьте логотип (можно использовать любую картинку) на страницах веб-сайта, созданного в Microsoft Expression Web, путем венесения необходимых изменений в мастер-страницу сайта.
- 2. Создайте страницу регистрации пользователя веб-сайта, созданного в Microsoft Expression Web, на основе мастер-страницы.

2. ЛАБОРАТОРНАЯ РАБОТА № 2

Название: Создание НТМL-документа.

Цель работы: Научиться создавать файлы, содержащие элементы форматирования, предписанные языком HTML.

Теоретическая часть: НТМL-документ — это просто текстовый файл с расширением *.htm или *.html).

Вот самый простой HTML-документ: <html> <head> <title> Лабораторная 2. Пример 1. </title> </head> <body> <H1> Привет! </H1> $\langle P \rangle$ Это простейший пример HTML-документа. </P> <P> Этот html-файл может быть одновременно открыт

и в блокноте, и в Internet Explorer'е. Сохранив изменения в

блокноте,

просто нажмите кнопку Reload ('перезагрузить') в Explorer'e,

чтобы увидеть эти изменения.

</P>

</body>

</html>

Для удобства чтения в примере введены дополнительные отступы, однако в HTML это совсем не обязательно. Более того, браузеры просто игнорируют символы конца строки и множественные пробелы в HTMLфайлах. Поэтому пример мог бы выглядеть вот так:

```
<html>
<head>
<title>Лабораторная 2. Пример 1.</title>
</head>
<body>
<H1>Привет!</H1>
```

<Р>Это простейший пример HTML-документа.</Р>

<Р>Этот html-файл может быть одновременно открыт

и в блокноте, и в Internet Explorer'е. Сохранив изменения в блокно-

просто нажмите кнопку Reload ('перезагрузить') в Explorer'е,

чтобы увидеть эти изменения.</Р>

</body>

те,

</html>

Как видно из примера, вся информация о форматировании документа сосредоточена в его фрагментах, заключенных между знаками "<" и ">". Такой фрагмент (например, <html>) называется меткой (поанглийски — tag, читается "тег").

Большинство HTML-меток — парные, то есть на каждую открывающую метку вида <tag> есть закрывающая метка вида </tag> с тем же именем, но с добавлением "/".

Метки можно вводить как большими, так и маленькими буквами. Например, метки <body>, <BODY> и <Body> будут восприняты браузером одинаково.

Многие метки, помимо имени, могут содержать атрибуты — элементы, дающие дополнительную информацию о том, как браузер должен обработать текущую метку. Более подробно метки и их атрибуты рассмотрим в ходе выполнения лабораторной работы.

Ход работы:

Создание простейших файлов HTML.

1. Создайте папку, в которой Вы будете сохранять созданные Webстраницы.

2. Запустите стандартную программу Блокнот (Notepad) и наберите следующий текст с элементами форматирования:

<HTML>

<HEAD>

<TITLE> Учебный файл HTML </TITLE>

</HEAD>

<BODY>

Расписание занятий на вторник

</BODY>

</HTML>

3. Сохраните файл под именем schedule.html.

4. Для просмотра созданной Web-страницы загрузите броузер Microsoft Internet Explorer.

5. Откройте в меню броузера Файл (File), Открыть (Open), Просмотр(Обзор — Browse) и найдите созданный ранее файл *schedule.html*, загрузите его. Убедитесь, что название Web-страницы (Учебный файл *HTML*) отразилось в верхней статусной строке броузера.

Управление расположением текста на экране.

1. Внесите изменения в текст, расположив слова "Расписание", "занятий", "на вторник" на разных строках:

<HTML> <HEAD> <TITLE> Учебный файл HTML </TITLE> </HEAD> <BODY> Расписание занятий на вторник </BODY> </HTML>

2. Сохраните внесенные изменения, с помощью команд Файл (File), Сохранить (Save).

3. Просмотрите с помощью броузера Microsoft Internet Explorer новую полученную Web-страницу используя клавишу F5 или с помощью кнопки Обновить (Refresh).

Предыдущие шаги не вызвали никаких видимых изменений в документе html — при отображении гипертекста броузеры автоматически размещают текст на экране, не принимая во внимание встречающиеся в файле переводы строк и идущих подряд пробелов. Существуют специальные команды, выполняющие перевод строки и задающие начало нового абзаца. Кроме того, существует команда, которая запрещает программе броузера каким-либо образом изменять форматирование текста и позволяет точно воспроизвести на экране заданный фрагмент текстового файла. Тег перевода строки
 отделяет строку от последующего текста или графики. тег абзаца <P> тоже отделяет строку, но еще добавляет пустую строку, которая зрительно выделяет абзац. Оба тега являются одноэлементными (не требуют закрывающего тега).

2. Внесите изменения в текст файла HTML: <HTML> <HEAD> <TITLE> Учебный файл HTML </TITLE> </HEAD> <BODY> Расписание <P>занятий
на вторник </BODY> </HTML> Сохраните внесенные изменения в файле *schedule.html*. Просмотрите с помощью броузера обновленную страницу.

Выделение фрагментов текста.

Существует три тега выделения фрагментов текста: $\langle B \rangle : \langle B \rangle =$ для выделения полужирным, $\langle I \rangle : \langle I \rangle =$ для выделения курсивом, $\langle U \rangle : \langle U \rangle =$ для выделения подчеркиванием.

1. Внесите изменения в файл schedule.html:

<HTML>

<HEAD>

<TITLE> Учебный файл HTML </TITLE>

</HEAD>

<BODY>

Расписание <I>занятий</I> <U> на вторник</U>

</BODY>

</HTML>

2.Посмотрите новую полученную Web-страницу.

3. Используйте вложение тегов:

```
<I><B>Расписание</B></I> <I>занятий</I> <U> на вторник</U>
```

Изменение размера текста

1. Существует два способа управления размером текста, отображаемого броузером: использование стилей заголовка и задание размера текущего шрифта;

2. Внесите изменения в файл *schedule.html*:

<HTML>

<HEAD>

<TITLE> Учебный файл HTML </TITLE>

</HEAD>

<BODY>

<H1>Расписание</H1> <I>занятий</I> <U>на вторник</U>

</BODY>

</HTML>

3. Просмотрите обновления через браузер.

4. Внесите следующие изменения изменения в файл *schedule.html*:

<HTML>

<HEAD>

<TITLE> Учебный файл HTML </TITLE>

</HEAD>

<BODY>

Расписание занятий на вторник

</BODY>

</HTML>

5. Самостоятельно измените размер шрифта для текста "занятия на вторник", используя тег .

Гарнитура и цвет шрифта

тег предоставляет возможности управления размером, начертанием и цветом текста. Изменение гарнитуры шрифта выполняется простым добавлением к тегу атрибута FACE. Например, для изображения текста шрифтом Arial необходимо записать .

Для изменения цвета шрифта в теге можно использовать атрибут COLOR="X". Вместо X надо подставить либо название цвета (в кавычках), либо его шестнадцатеричное значение. При задании цвета шестнадцатеричным числом можно представить этот цвет разложенным на три составляющие: красную(R), зеленую (G), синюю (B), каждая из которых имеет значение от 0 до FF. Примеры записи цвета в формате RGB приведены в таблице1.

Таблица 1

Цвет	RRGGBB
black черный	000000
white белый	FFFFFF
red красный	FF0000
green зеленый	00FF00
azure бирюзовый	00FFFF
blue синий	0000FF
gray серый	A0A0A0
purple фиолетовый	FF00FF
yellow желтый	FFFF00
brown коричневый	996633
orange оранжевый	FF8000
violet лиловый	8000FF

Записи цвета в формате RGB

6. Внесите изменения в файл *schedule.html*:
 <HTML>
 <HEAD>

<TITLE> Учебный файл HTML </TITLE>

</HEAD>

<BODY>

<U><I>Pacпиcaниe</I>

занятий на вторник

 $\langle BODY \rangle$

</HTML>

7. Самостоятельно измените размер, цвет, гарнитуру, стиль текста документа.

Выравнивание текста по горизонтали

1. С помощью тегов HTML можно управлять горизонтальным выравниванием текста. Если не оговаривать способ выравнивания, все элементы в документе будут выравниваться по левому краю и иметь неровное правое поле (см. Рис.2). Современные броузеры для выравнивания текста используется атрибут ALIGN=, который встраивается в теги абзаца или заголовка.

ALIGN=CENTER — Выравнивание по центру

ALIGN=RIGHT — Выравнивание по правому краю

ALIGN=LEFT — Выравнивание по левому краю

2. Внесите изменения в файл schedule.html:

<HTML>

<HEAD>

<TITLE> Учебный файл HTML </TITLE>

</HEAD>

<BODY>

<P ALIGN=CENTER>

<FONT

COLOR="#008080"

SIZE="7">Paсписание

<I>занятий на вторник</I>

</P>

 $\langle BODY \rangle$

</HTML>

3. Просмотрите обновления

Задание цвета фона и текста

Цвета фона и текста документа устанавливаются в начале файла HTML в теге <BODY>. Атрибут BGCOLOR= задает цвет фона страницы, TEXT= определяет цвет текста для всей страницы, LINK= и VLINK= определяют цвета соответственно непросмотренных и просмотренных ссылок.

1. Внесите изменения в файл schedule.html: <HTML> <HEAD> <TITLE> Учебный файл HTML </TITLE>

```
</HEAD>
<BODY BGCOLOR="#FFFFCC" TEXT="#330066">
<P ALIGN=CENTER>
<FONT COLOR="#008080"
SIZE="7"><B>Расписание</B></FONT><BR>
```

<I>занятий на вторник</I>

```
</P>
```

```
</BODY>
```

</HTML>

Задание на самостоятельное выполнение.

1. Проведите экспериментальную работу с созданным документом. Меняя размеры окна, посмотрите, как браузер показывает текст с принудительным разрывом строк. Что происходит, когда окно уменьшается настолько, что в нем не может поместиться целиком даже одно слово?

2. Самостоятельно создайте собственный HTML-документ. Пусть это будет небольшой рассказ о себе и своих увлечениях.

Обсудите в группах полученные результаты, подготовьте ответы на контрольные вопросы.

Контрольные вопросы

- 1. Этапы создания HTML документа.
- 2. Что такое теги, метки, скрипты.
- 3. Какие атрибуты могут содержать теги.
- 4. Поясните основные команды <HTML>, <HEAD>, <TITLE>, <BODY>.
- 5. Что такое парные не парные метки.
- 6. Как программировать цвета на страницах,
- 7. Как форматировать текст.

3. ЛАБОРАТОРНАЯ РАБОТА № 3

Название: Размещение на веб-странице графики

Цель работы: Научиться внедрять в html-документ графические изображения.

Теоретическая часть: Одна из наиболее привлекательных черт Web — возможность включения ссылок на графические и иные типы данных в HTML-документ. Делается это при помощи тега . Использование этого тега позволяет значительно улучшить внешний вид и функциональность документов.

Существует два способа использования графики в HTMLдокументах. Первый — это внедрение графических образов в документ, что позволяет пользователю видеть изображения непосредственно в контексте других элементов документа. Это наиболее используемая техника при проектировании документов, называемая иногда "inline image". Синтаксис тега:

Элементы синтаксиса тега:

URL

Обязательный параметр, имеющий такой же синтаксис, как и стандартный URL. Данный URL указывает броузеру где находится рисунок. Рисунок должен храниться в графическом формате, поддерживаемом броузером. На сегодняшний день форматы GIF и JPG поддерживаются большинством броузеров.

ALT="text"

Этот необязательный элемент задает текст, который будет отображен броузером, не поддерживающим отображение графики или с отключенной подкачкой изображений. Обычно, это короткое описание изображения, которое пользователь мог бы или сможет увидеть на экране. Если данный параметр отсутствует, то на месте рисунка большинство броузеров выводит пиктограмму (иконку), активизировав которую, пользователь может увидеть изображение. тег ALT рекомендуется, если ваши пользователи используют броузер, не поддерживающий графический режим, например Lynx.

HEIGTH=n1

Этот необязательный параметр используется для указания высоты рисунка в пикселах. Если данный параметр не указан, то используется оригинальная высота рисунка. Это параметр позволяет сжимать или растягивать изображения по вертикали, что позволяет более четко определять внешний вид документа. Однако, некоторые броузеры не под-

держивают данный параметр. С другой стороны, экранное разрешение у вашего клиента может отличаться от вашего, поэтому будьте внимательны при задании абсолютной величины графического объекта.

WIDTH=n2

Параметр также необязателен, как и предыдущий. Позволяет задать абсолютную ширину рисунка в пикселах.

ALIGN

Этот параметр используется, чтобы сообщить броузеру, куда поместить следующий блок текста. Это позволяет более строго задать расположение элементов на экране. Если данный параметр не используется, то большинство броузеров располагает изображение в левой части экрана, а текст справа от него.

С версии HTML 2.0 у тега появились дополнительные параметры:

Новые параметры:

BORDER

Этот параметр позволяет автору определить ширину рамки вокруг рисунка.

VSPACE

Позволяет установить размер в пикселах пустого пространства над и под рисунком, чтобы текст не наезжал на рисунок. Особенно это важно для динамически формируемых изображений, когда нельзя заранее увидеть документ.

HSPACE

То же самое, что и VSPACE, но только по горизонтали.

<u>Фоновые рисунки</u>

Большинство броузеров позволяет включать в документ фоновый рисунок, который будет отображаться на фоне всего документа.Описание фонового рисунка включается в тег ВОDY и выглядит следующим образом:

<BODY BACKGROUND="picture.gif">

Ход работы:

<u>Размещение графики на Web-странице.</u>

Ter является одиночным, т.е. закрывающий тег не применяется.

Графика в Web, как правило, распространяется в трех форматах: GIF, JPG, PNG. Перед выполнением упражнения поместите выбранный файл

в ту же папку, которая будет использована для хранения создаваемой Web-страницы.

1. Внесите изменения в файл *schedule.html* (если файл утрачен, создайте новый):

<HTML>

<HEAD>

<TITLE> Учебный файл HTML </TITLE>

</HEAD>

<BODY BGCOLOR="#FFFFFF" TEXT="#330066">

<P ALIGN=CENTER>

<FONT

COLOR="#008080"

SIZE="7">Paсписание


```
<FONT SIZE="6"><I>занятий на вторник</I></FONT><BR><BR>
```


</P>

</BODY>

</HTML>

3. Самостоятельно внесите изменения в файл *schedule.html*, опробовав использование таких атрибутов графики как ALT, BORDER, HEIGHT, WIDTH (Табл.2.)

<u>Фоновое отображение графики на Web-странице</u>

1. Поместите файл еще один графический файл в ту же папку, что и *schedule.html*.

2. Внесите изменения в файл schedule.html:

<HTML>

<HEAD>

<TITLE> Учебный файл HTML </TITLE>

</HEAD>

```
<BODY BACKGROUND="back.jpg" TEXT="#330066">
```

<P ALIGN=CENTER>

<FONT

COLOR="#008080"

```
SIZE="7"><B>Paсписание</B></FONT><BR>
```

 <I>занятий на вторник</I>

</P>

</BODY>

</HTML>

На экране вы увидите:



Таблица 2

Атрибуты графики			
Атрибут	Формат	Описание	
ALT	</img 	Надпись "картинка" выводится на экран при подведении указателя мыши к изо- бражению.	
BORDER	</img 	Задает рамку вокруг изображения тол- щиной 3 пикселя.	
ALIGN	</img 	Выравнивает изображение относительно текста по верхней границе текста.	
HEIGHT	</img 	Вертикальный размер изображения принудительно устанавливается в 111 пикселей.	
WIDTH	</img 	Горизонтальный размер изображения принудительно устанавливается в 220 пикселей.	
VSPACE	</img 	Атрибут добавляет верхнее и нижнее пустые поля высотой 8 пикселей.	
HSPACE	</img 	Добавляет левое и правое пустые поля шириной 8 пикселей.	

В действительности же графический файл back.jpg выглядит так:



Графические элементы возможно применять в качестве различного вида "украшений". Если Вы придумаете что-нибудь оригинальное, ваша веб-страница будет смотреться необычно и запомнится посетителю. Вот несколько ставших уже традиционными вариантов такого применения графики.

Во-первых, это различные графические разделители, обычно горизонтальные, применяемые вместо горизонтальной черты (<HR>).

Вот так он выглядит в окне браузера:

Хотя тег <HR> и поддается настройке, графический разделитель вместо <HR> часто выглядит лучше:

Во-вторых, можно применить графический элемент в качестве буквицы. Встроить буквицу в текст можно следующим образом:



измот пример встроенной в текст буквицы, она добавлена в начало параграфа. Для экономии места приводится текст только того места странички где вставлена буквица. На всякий случай в качестве альтернативного текста дается буква "В", чтобы пользователю с отключенной графикой не приходилось строить догадки относительно первой буквы.

- 1. Разместите в Вашем документе *schedule.html* разделитель в формате .gif.
- 2. Задайте этому изображению высоту, равную 2 пикселям.
- 3. Сместите анимированный разделитель в центр документа.
- 4. В слове Расписание замените букву Р изображением, буквицы.
- 5. Изменяя значение атрибута ALIGN, добейтесь наилучшего расположения буквы на экране.
- 6. Используя любой графический редактор, создайте свое, альтернативное, изображение буквы *P* и встройте его в документ. <u>Графические маркеры.</u>

Язык HTML позволяет создавать маркированные и нумерованные списки.

Пример маркированного списка: элемент списка элемент списка

•••

элемент списка

В терминах языка HTML это выглядит так: >элемент списка >элемент списка <1i>.... >элемент списка Пример нумерованного списка: 5. элемент списка 6. элемент списка 7. ... 8. элемент списка В терминах языка HTML это выглядит так: < 0 >>элемент списка >элемент списка <1i>.... >элемент списка

Одно дело, когда маркерами списка являются стандартные кружочки, и совсем другое — когда каждый сам имеет возможность создать маркер. Маркером может быть все, что угодно — от просто цветных и немного выпуклых кружков и квадратов до изощренных миниатюрных художественных работ. Чтобы проиллюстрировать возможность вставки в список графических маркеров, заменим кружки в маркированном списке на красные треугольники.

Сначала надо создать такой треугольник в любой графической программе. Этот файл нужно поместить рядом с уже созданным файломschedule.html.

Теперь введем в тег атрибут STYLE= (этот атрибут подробнее мы рассмотрим при изучении CSS — Cascading Style Sheets):

<UL STYLE="list-style-image: url('marker.gif');">

Ваш список должен выглядеть так:

элемент списка

элемент списка

•••

элемент списка

Обратите внимание на то, что при указании имени файла мы заключили его не в обычные двойные кавычки, а в одинарные. Это сделано потому, что все значение атрибута STYLE= заключено в двойные кавычки. Оформление кнопок.

Еще одно частое применение графических элементов — это оформление кнопок. Вообще говоря, кнопка — это элемент взаимодействия с пользователем, так как предполагается, что когда он на ней нажмет, то что-нибудь произойдет. Сейчас мы не будем рассматривать методы такого взаимодействия. Пока мы только научимся создавать кнопки и использовать в них элементы графики.

Чтобы создать кнопку, поместите в Ваш html-документ следующий код:

<BUTTON>ЭТО КНОПКА</BUTTON>

В браузере она должна выглядеть так: ЭТО КНОПКА

Между тегами <BUTTON>...</BUTTON> можно поместить не только текст, но и изображение. Если мы поместим туда тег , то получим кнопку с графическим изображением.

Новое выбранное изображение поместите в ту же папку, что и файл *schedule.html* под именем *home-button.gif*.

Замените текст ЭТО КНОПКА на описание тега , указывающего на файл home-button.gif.

В тег <BUTTON> введите атрибут onClick — этот атрибут позволяет описать действия, которые необходимо отработать браузеру при шелчке мыши по объекту. Обновленный тег должен вылядеть так:

<BUTTON onClick="location.href='http://www.tpu.ru'">

Контрольные вопросы

- 1. Назовите тег включения рисунков на страницы, и его атрибуты.
- 2. Поясните –.
- 3. Объясните параметры BORDER, VSPACE, HSPACE.
- 4. Поясните понятие графических маркеров.
- 5. Поясните <UL STYLE="list-style-image: url('marker.gif');">.

4. ЛАБОРАТОРНАЯ РАБОТА № 4

Название: Размещение на веб-странице таблиц, форматирование текста.

Цель работы: Научиться создавать и редактировать таблицы в терминах языка HTML.

Теоретическая часть: Таблицы представляют собой особую часть HTML-документа. Данные в ней организованы в виде прямоугольной сетки, состоящей из вертикальных столбцов и горизонтальных рядов. Каждая клетка таблицы является ячейкой. Ячейки могут содержать в себе текст, графику или другую таблицу. Таблица состоит из трех основных частей:

- название таблицы,
- заголовки столбцов,
- ячейки.

Таблица заполняется горизонтальными рядами ячейка за ячейкой слева направо. Заполнение начинается с левого верхнего угла и заканчивается правым нижним. Каждая ячейка должна быть заполнена. Для создания пустых ячеек используются пробелы.

Таблица 3

Тег	Форма записи	Примечание	
TA BLE	<table>текст<th>Объявление таблиц.</th></table>	Объявление таблиц.	
TR	<tr>текст</tr>	Объявление строки.	
TD	<td>текст</td>	текст	Объявление ячейки в строке.

Теги оформления таблиц

Таблица 4

Атрибуты тега <TABLE>

Атрибут	Форма записи	Примечание
BORDER	<table border="X"></table>	Задает рамку вокруг таблицы.
WIDTH	<table WIDTH=XX%></table 	Задает ширину таблицы как XX%
BGCOLOR	<table BGCOLOR="#RRGGBB"></table 	Задает цвет фона таблицы.

Таблица 5

Атрибуты тегов <TD> и <TR>

Атри- бут	Форма записи	Примечание	
ALIG N	<td align="X"></td> <td>Устанавливает выравнивание по горизонтали (Right, Left, Center)</td>		Устанавливает выравнивание по горизонтали (Right, Left, Center)
GN VALI	<td VALIGN=X></td 	Устанавливает выравнивание по вертикали (Top, Middle, Bottom, Baseline)	
BGC OLOR	<td BGCOLOR= "#RRGGBB"></td 	Задает цвет фона ячейки.	

Еще один тег для оформления ячеек таблиц — тег <TH>...</TH> — нужен для задания заголовочных ячеек. Он во всем совпадает с тегом <TD>, но в отличие от него, содержимое выдается жирным шрифтом и центрируется.

Если нужно задать заголовок всей таблицы, используйте тег <CAPTION параметры>...</CAPTION>. Он должен быть внутри тега <TABLE>, но вне описания ячеек. Тег имеет один параметр: ALIGN - указывает положение заголовка: он может быть в верхней (TOP) или нижней (BOTTOM) части таблицы.

Ход работы:

Создание простой таблицы.

Запустите стандартную программу Блокнот (Notepad) и наберите следующий текст с элементами форматирования:

<html>

<body>

Mama

Папа

Cын

</body>

</html>

Для просмотра созданной Web-страницы используйте броузер Microsoft Internet Explorer.

Вставьте в тег следующие атрибуты: width=50% align="center" bgcolor="yellow" bordercolor="blue". Просмотрите обновленный документ в браузере.

Добавьте в тег атрибут: cellspacing=5. Просмотрите обновленный документ в браузере.

Измените значение атрибута cellspacing на 10, 30 и посмотрите что произойдет с таблицей.

Добавьте между второй парой тегов : теги дядя дедушка Просмотрите полученный документ в окне браузера.

Добавьте между третьей парой тегов : теги дочь

Добавьте в первой паре тегов colspan=3. Просмотрите полученный документ в окне браузера.

Добавьте в первой паре тегов dign="center" Просмотрите полученный документ в окне браузера.

Добавьте во второй паре тегов arpuбутrowspan=2 Просмотрите полученный документ в окне браузера.

Удалите из второй пары тегов tr>:тегtr>:смотрите полученный документ в окне браузера.

Задание на самостоятельное выполнение.

Создайте страницу, содержащую расписание Вашей учебной группы. HTML-код должен выглядеть примерно так:

<HTML>

<HEAD>

<TITLE> Учебный файл HTML </TITLE>

</HEAD>

<TITLE> Расписание занятий V курса</TITLE>

</HEAD>

<BODY BGCOLOR="FFFFFF">

<P ALIGN=CENTER>

 V курс

</P>

<FONT COLOR="BLUE" SIZE="4"

FACE="COURIER"> Понедельник

<TABLE BORDER="1" WIDTH=100% BGCOLOR="99CCCC">

<TR BGCOLOR="CCCCFF" ALIGN=CENTER>

<TD>Пара</TD> <TD>3-8581</TD> <TD>3-8582</TD> <TD>3-8583</TD>

</TR>

<TR>

<TD>1</TD> <TD>Мировые инф. рес-сы</TD> <TD>Проектирование Интернет-приложений</TD> <TD>Txн. про-ект-ия СП</TD>

</TR>

<TR>

<TD>2</TD> <TD>Проектирование Интернет-приложений</TD> <TD>Мировые инф. рес-сы</TD> <TD>Проектирование Интернет-приложений</TD>

</TR>

 $\langle TR \rangle$

<TD>3</TD> <TD>Txн. проект-ия СП</TD> <TD>Txн. проект-ия СП</TD> <TD>Mupoвые инф. рес-сы</TD>

</TR>

</TABLE>

</BODY>

</HTML>

Сохраните файл под именем schedule_monday.html.

Используя различные варианты оформления таблиц, создайте файлы с именами *schedule_tuesday.html* и *schedule_wednesday.html*, содержащие расписание на вторник и среду, соответственно.

В файле *schedule_friday.html* создайте таблицу такого вида:

военка		

Для созданной таблицы установите следующие параметры: расположение на экране — центральное, ширина 550 пикселей. Цвет вертикальной общей ячейки — голубой. Цвет всей таблицы — зеленый. Расположение текста в ячейках — по центру.

В файле *schedule_sunday.html* создайте таблицу такого вида:

выходной	

Для созданной таблицы установите следующие параметры: расположение на экране — левостороннее, ширина 550 пикселей. Цвет общей горизонтальной ячейки — красный. Цвет всей таблицы — желтый.

В файле schedule_thursday.html создайте таблицу такого вида:

Проектирование Интернет-приложений	

Для созданной таблицы установите следующие параметры: расположение на экране — правостороннее, ширина 550 пикселей. Цвет общей ячейки — оранжевый. Цвет всей таблицы — серый.

Контрольные вопросы

- 1. Назовите теги оформления таблиц
- 2. Назовите атрибуты тега *<TABLE>*
- 3. Назовите атрибуты тега $\langle TD \rangle u \langle TR \rangle$
- 4. Поясните отличие простой таблицы от форматированной
- 5. Поясните <UL STYLE="list-style-image: url('marker.gif');">
- Как оформляется кнопка <BUTTON onClick="location.href='http://www.tpu.ru'">.
- 7. Что такое маркированный и нумерованный списки?

5. ЛАБОРАТОРНАЯ РАБОТА № 5

Название: Размещение на веб-странице внешних и внутренних гиперссылок.

Цель работы: Научиться формировать гиперссылки на смежные документы Internet и на метки в текущем документе.

Теоретическая часть: Важнейшим свойством языка HTML является возможность размещения на странице ссылок на другие документы. Возможны ссылки: на удаленный HTML файл, на некоторую точку в текущем HTML-документе, на любой файл, не являющийся HTMLдокументом.

В качестве ссылки можно использовать текст или графику.

Ссылки в пределах одного документаФ

Такие ссылки требуют наличие двух частей: метки и самой ссылки. Метка определяет точку, к которой происходит переход по ссылке. Ссылка использует имя метки. Ссылки выделяются цветом или подчеркиванием, в зависимости от того, как настроен броузер.

Ссылка должна выглядеть примерно так:

Понедельник</А>

Перед именем метки *ПН*, указывающей куда производится ссылка, ставится символ #. Между символами > и < располагается текст *Понедельник*, на котором производится щелчок для перехода по ссылке.

Метка должна выглядеть примерно так:

Понедельник</А>

<u>Ссылки на другой HTML-документ</u>

Ссылки позволяют щелчком по выделенному слову или фразе перейти к другому файлу.

Ссылка должна выглядеть примерно так:

<*A HREF="example.html">Пример*

После имени файла *example.html*, указывается между символами > и < текст *Пример*, по которому производится щелчок для перехода к этому файлу.

Ход работы:

Создание ссылок в пределах одного документа.

Скачайте и сохраните файл yoga.html.

Откройте файл с помощью текстового редактора "Блокнот" (Notepad) и поместите в начало страницы следующий код:

<TABLE WIDTH=100%>

<TR>

<TD>Понедельник</TD>

<TD>Вторник</TD>

```
<TD>Cpeдa</TD>
<TD>Четверг</TD>
<TD>Пятница</TD>
<TD>Суббота</TD>
</TR>
</TR>
</TABLE>
<BR>
Определите для слова Суббота в расписании метку:
<A NAME="CБ">Суббота</A>
Определите ссылку для выбранной метки:
...
<TABLE WIDTH=100%>
<TR>
<TD>Понедельник</TD>
```

```
<TD>Cpeдa</TD>
```

<TD>Суббота</TD>

•••

Сохраните файл.

Для просмотра созданной Web-страницы используйте броузер Microsoft Internet Explorer.

Вы должны увидеть Web-страницу с определенной в ее верхней части ссылкой "Суббота", щелчок мыши по ссылке должен перемещать видимую часть страницы, фокусируя внимание пользователя на расписании на выбранный день.

Определите ссылки и соответствующие метки для остальных дней недели.

Продемонстрируйте полученный документ преподавателю.

Создание ссылок на другой HTML-документ.

Создайте с помошью текстового редактора "Блокнот" (Notepad) файл *my_schedule.html* и поместите в начало страницы следующий код:

<TABLE WIDTH=100%>

```
\langle TR \rangle
```

```
<TD>Понедельник</TD>
```

```
<TD>Вторник</TD>
```

```
<TD>Cpeдa</TD>
```

```
<TD>Четверг</TD>
```

```
<TD>Пятница</TD>
```

```
<TD>Суббота</TD>
```

```
</TR>
```

</TABLE>

Сохраните созданный файл в ту же папку, что и файлы schedule_monday.html, schedule_tuesday.html, ... schedule_sunday.html из лабораторной работы N4.

Определите для слова *Понедельник* ссылку на внешний документ: Понедельник</А>

Сохраните файл.

Для просмотра Web-страницы используйте броузер Microsoft Internet Explorer.

Убедитесь, что после щелчка указателем мыши на ссылку Понедельник в окне браузера загружается страница *schedule_monday.html*.

Определите ссылки для остальных дней недели.

Создание графической ссылки.

Сохраните графический файл в ту же папку, что и файл *my_schedule.html*.

Внесите изменения в файл *my_schedule.html* так, чтобы в конце страницы была ссылка на страницу *yoga.html*. В качестве ссылки используется графический файл *yoga.gif*:

<CENTER>

</CENTER>

Сохраните файл.

Для просмотра созданной Web-страницы используйте броузер Microsoft Internet Explorer.

Измените созданную ссылку таким образом, чтобы по щелчку мыши браузер загружал страницу *yoga.html* и фокусировался на метке *Суббота*.

Продемонстрируйте полученный документ преподавателю.

Задание карты ссылок.

Элемент *IMG* позволяет использовать изображения, отдельные части которых связаны со ссылками и позволяют выполнять переходы. Такие изображения называются картами (map). Для определения полей карты используется атрибут

usemap="#Имя"

Это имя ставится в соответствие со значением соответствующих атрибутов элементов AREA и MAP (см. ниже), которые определяют конфигурацию карты. Задание атрибутов usemap придает элементу IMG свойства, характерные для элемента A, то есть возможность осуществления перехода.
Для определения карты необходимо обязательное совместное использование сразу трёх элементов: AREA, IMG и MAP.

Элемент *МАР* необходим для общего определения карты. Внутри него определяются области карты при помощи элементов AREA и задается имя карты при помощи атрибута:

пате="Имя"

Для каждой области карты должен быть создан свой элемент *AREA*. Он не имеет конечного тега. Этот элемент должен включать атрибут, определяющий ссылку:

href="http://Adpec"

Атрибут для задания текста, заменяющего изображение карты, не является обязательным:

alt="Текст подсказки"

Атрибуты, определяющие форму области на карте, являются обязательными. Существует три стандартных вида областей: круг (circle), прямоугольник (rect) и многоугольник произвольной формы (polygon).

Для круга необходимо задать координаты центра и радиус (r), выраженные в пикселах. Координаты центра отсчитываются от левого края (x) и верхнего края (y) рисунка. Шаблон для задания круговой области таков:

shape="circle", coords=x, y, r

Для определения области произвольной конфигурации задают координаты (x, y) каждого из углов многоугольника, который точно или приблизительно соответствует по форме этой области:

shape="poly" coords=x1, y1, x2, y2, x3, y3 ...

При определении прямоугольной области задают координаты верхнего левого и правого нижнего углов прямоугольника:

shape="rect" coords=x1, y1, x2, y2

Сохраните графический файл в ту же папку, что и файл *my_schedule.html*.

В начало файла *my_schedule.html* вставьте следующий код: <map name="schedule_map">

<area alt="Понедельник" shape="rect" coords="2, 0, 312, 31" href="schedule_monday.html">

<area alt="Пятница" shape="rect" coords="2, 118, 312, 148" href="schedule_friday.html">

</map>

Сохраните файл.

6. ЛАБОРАТОРНАЯ РАБОТА № 6

Название: Изучение возможностей языка JavaScript для проведения анкетирования.

Цель работы: Изучить основы языка программирования JavaScript на стороне клиента.

Теоретическая часть:

JavaScript — это язык программирования, предназначенный прежде всего для создания интерактивных HTML-страниц. Первая версия языка была разработана фирмой Netscape Communications в 1995 году. Обратите внимание, с языком Java, созданным фирмой Sun Microsystems, у JavaScript мало общего.

Программу на языке JavaScript обычно встраивают прямо в HTMLфайл. Чтобы программа была выполнена, HTML-файл должен быть открыт в браузере пользователя.

JavaScript — интерпретируемый язык. Это означает, что для исполнения программы не требуется предварительная компиляция (преобразование исходного текста программы в системо-зависимый машинный код). Текст программы интерпретируется, то есть анализируется и сразу же исполняется.

Приведем простейший пример HTML-страницы с программой на JavaScript:

<html></html>				
<head></head>				
<title>3</title>	аголовок страницы<			
<script 1<="" td=""><td>anguage="JavaScript</td><td>."></td><td></td><td></td></tr><tr><td>function cli</td><td>ck_func(){</td><td></td><td></td><td></td></tr><tr><td>alert('Hello</td><td>, world!');</td><td></td><td></td><td></td></tr><tr><td>}</td><td></td><td></td><td></td><td></td></tr><tr><td></script>	>			
<body></body>				
<input< td=""><td>type="button"</td><td>value="Click</td><td>me!"</td><td>on-</td></input<>	type="button"	value="Click	me!"	on-
Click="javascrip	ot:click_func();">			
Текст прог	раммы заключен ме	ежду тегами <i><scri< i=""></scri<></i>	<i>РТ></i> и <td>RIPT>.</td>	RIPT>.

Программы заключен между тегами <SCRPT> и </SCRPT>. Программа состоит из одной функции *click_func()*, которая, в свою очередь вызывает стандартную функцию *alert* с параметром "*Hello*, *world*!". Функция активизируется по событию *onClick* — нажатие на кнопку. Результатом выполнения программы является вывод на экран окна предупреждения с заданным текстом.

Язык *JavaScript* может не только выводить на экран вновь создаваемы окна, но и динамически изменять уже загруженную браузером страницу. Перепишем как указано ниже:

<HTML> <HEAD> <TITLE>Заголовок страницы</TITLE> <SCRIPT language="JavaScript"> function click_func(){ document.write('Hello, world!'); } </SCRIPT> $\langle HEAD \rangle$ $\langle BODY \rangle$ type="button" value="Click me!" <input on-Click="javascript:click_func();"> </BODY>

</HTML>

Вместо вызова функции *alert* производится вызов функции *write* объекта *document*. Эта функция выводит на экран текст, передаваемый в качестве параметра и стирает все, что содержалось на странице до ее вызова.

Объект document создается браузером в момент загрузки страницы. Необходимо отметить, что в JavaScript определена целая иерархия объектов, которые должен создавать браузер всякий раз при загрузке страницы или при открытии окна. Эти объекты доступны для использования разработчиками. Их использование сводится к вызову методов объекта или чтения/переопределения его свойств. Методы объекта вызываются так:

объект.метод([параметры])

Вспомните, так было для объекта *document*, когда мы вызывали его метод *write*.

Доступ к свойствам осуществляется по той же схеме:

объект.свойство.

Например, *document.location* будет содержать строку с адресом страницы.

Разберем, как пользоваться подобными свойствами в программе. Рассмотрим пример:

<title>Tecтируем JavaScript</title> <SCRIPT LANGUAGE="JavaScript">

var bgc = document.bgColor; var fgc = document.fgColor: var lc = document.linkColor; var al = document.alinkColor; var vlc = document.vlinkColor; var url = document.location; var ref = document.referrer; var t = document.title; var lm = document.lastModified; document.write("Цвет фона этой страницы: " + bgc); document.write("
Uвет текста этой страницы: " + fgc); document.write("
Цвет ссылок этой страницы: " + lc); document.write("
Цвет активных ссылок: " + al); document.write("
UBET посещенных ссылок: " + vlc); document.write("
URL этой страницы: " + url); document.write("
До этого вы были на странице: " + ref); document.write("
Заголовок этой страницы: " + t); document.write("

hocледние изменения внесены: " + lm) </SCRIPT>

Иерархию объектов языка JavaScript можно сопоставить с иерархией объектов в реальном мире. Колеса, руль и педали — это все объекты, принадлежащие велосипеду. Велосипед представляет собой объект, принадлежащий к средствам наземного транспорта, которые, в свою очередь, принадлежат к более общей категории средств передвижения.

В терминах объектов JavaScript эти взаимоотношения могли бы быть выражены следующим образом:

средстваПередвижения.наземныйТранспорт.велосипед.педаль

Наиболее общий объект высшего уровня находится слева в выражении, и слева направо происходит переход к более частным объектам, являющимся при этом наследниками высших в иерархии объектов. Таким образом, чтобы обратиться к определенному свойству этих объектов, вы должны определить имя объекта и всех его предков.

Проиллюстрируем иерархию объектов:

navigator	
window	
+ <u>location</u>	
+ <u>history</u>	
+ <u>document</u>	
+ <u>forms</u>	
elements (text fields, textarea, checkbox, pa	ssword
radio, select, button, submit, reset)	

+--links

+--<u>images</u>

+--<u>anchors</u>

Используя иерархию объектов, попробуем вывести в статус-панель адрес первой ссылки на текущей странице. Статус-панель находиться в самом низу вашего обозревателя:

Если эта панель скрыта, вы можете подключить ее: меню View > Status Bar.

При нажатии на в статус-панели должен появиться адрес первой ссылки на текущей странице.

Исходный текст выглядит так:

<input type=button value="эту кнопку" onClick="javascript:window.status=window.document.links[0].href">

Обратите внимание, что первую ссылку на странице мы нашли в массиве *links* под номером 0. К самому массиву доступ был получен перечислением всех объектов иерархии текущего документа window.document.links. Адрес, относящийся к ссылке, хранится в свойстве *href* объекта *link*, в нашем случае это *links[0]*.

Ход работы:

Объявление функций.

С помощью текстового редактора "Блокнот" (Notepad) создайте страницу *testjs.html* и поместите в начало страницы следующий код:

<html>

<head>

<title>Script test</title>

</head>

<body>

<script language="JavaScript">

document.write("Добро пожаловать на мою страницу!
"); document.write("Это JavaScript!
");

document.write("Добро пожаловать на мою страницу!
"); document.write("Это JavaScript!
");

document.write("Добро пожаловать на мою страницу!
"); document.write("Это JavaScript!
");

</script>

</body>

</html>

Сохраните страницу и просмотрите ее с помощью браузера.

Такой скрипт выведет на экран следующий текст:

Добро пожаловать на мою страницу!

Это JavaScript!

три раза. Наверняка вы заметили, что для получения необходимого результата определенная часть его кода была повторена три раза. Это неэффективно, поэтому попробуем оптимизировать код.

Определите функцию *welcome_func()*, которая будет выводить на экран строки:

Добро пожаловать на мою страницу!

Это JavaScript!

Затем вызовите функцию несколько раз. Для этого вы можете использовать цикл:

```
for(i=0; i<3; i++){
  welcome_func();
}</pre>
```

}

Функции могут также использоваться в совместно с процедурами обработки событий. Рассмотрим пример. Создайте страницу *testjs2.html*, введите следующий код:

```
<html>
<head>
<script language="JavaScript">
function calculation() {
var x = 12;
var y = 5;
var result = x + y;
window.status = result;
}
</script>
</head>
<body>
<form>
<input type="edit" onFocus="calculation()">
</form>
</body>
</html>
```

В приведенном примере интрес представляют следующие нововведения:

var x = 12; — так в JavaScript объявляются переменные, тип переменной не дифференцируется — одинаково объявляются переменные типа double, integer, String и прочие; *window.status* = *result*; — свойству window.status присваивается новое значение, в нашем случае это строка 17, эта строка будет отображена в статус-панели браузера. *onFocus* — это команда, которая вызывает действие, когда пользователь "фокусируется" на элементе страницы, в нашем случае — помещает курсор в текстовое поле.

Сохраните файл.

Для просмотра созданной Web-страницы используйте браузер Microsoft Internet Explorer.

Вы должны увидеть Web-страницу с определенным в ее верхней части окном редактирования, установка курсора в этом окне приведет к появлению строки 17 в статус-панели браузера.

На этой же странице создайте функцию, которая будет вызываться при потере фокуса полем редактирования и помещать в статус-панель результат, полученный от разности чисел x и y. Обработчику события "потеря фокуса" в JavaScript соответствует параметр *onBlur*.

Манипуляция объектами.

Сохраните на жестком диске своего компьютера файлы <u>aircraft1.jpg</u> и <u>aircraft2.jpg</u>.

Создайте с помошью текстового редактора "Блокнот" (Notepad) файл *testjs3.html* и поместите в начало страницы следующий код:

В своем скрипте вы можете назначать этому объекту новое изображение:

document.images["aircraft"].src='new_image.jpg'

Обратите внимание, что к интересующему нас объекту *Image* в массиве *images* мы обращаемся не по порядковому номеру, а по имени элемента *images["aircraft"]*. Более того, если имя объекта на странице оригинально, мы можем обратиться к нему напрямую, минуя упоминание массива:

document.aircraft.src='new_image.jpg'

Попробуем заменить изображение динамически, используя JavaScript. Модифицируйте свою страницу. Исходный текст должен выглядеть так:

<script language="JavaScript">

document.aircraft.src='aircraft2.jpg';

</script>

Попробуйте просмотреть страницу в браузере — сразу же после загрузки документа картинка *aircraft1.jpg* должна замениться на *aircraft2.jpg*.

Эффект происходит очень быстро и заметить его сложно. Модифицируем страницу таким образом, чтобы рисунки замещались при наведении курсора мыши на рисунок:

<script language="JavaScript">

function onOver(){

document.images["aircraft"].src='aircraft2.jpg';

}

</script>

Попробуйте просмотреть страницу в браузере — картинка *aircraft1.jpg* заменяется на *aircraft2.jpg* при наведении указателя мыши на изображение.

Добавьте обработчик события *onMouseOut* — удаление указателя мыши с поверхности изображения. Пусть по этому событию активным становится pucyнok *aircraft1.jpg*.

Скачайте и сохраните на своей машине файл <u>aircraft3.jpg</u>.

Добавьте обработчик события *onMouseDown* — нажатие мыши. Пусть при клике на изображении активным рисунком становится *aircraft3.jpg*

Сохраните файл.

Управление объектами формы.

Форма (form) — это контейнер html-страницы, в котором могут быть размещены элементы, позволяющие пользователю вводить данные для отправки на сервер. К таким элементам можно отнести тектстовые поля, списки, кнопки выбора и т.п.

С помощью текстового редактора "Блокнот" (Notepad) создайте страницу *testform.html* и поместите на странице следующий код:

```
<head>
<title>Управление объектами формы</title>
</head>
<body bgcolor=#ffffff>
<img src="aircraft1.jpg" name="pic1">
\langle p \rangle
<form name="myForm">
Ваше имя:
\langle br \rangle
<input type="text" name="name" value=""><br>
e-mail:
<br>
<input type="text" name="email" value=""><br><br>
<br>
<input type="button" value="Haжмитe!" name="myButton">
</form>
```

22.

```
<img src="aircraft2.jpg" name="pic2">
```

25.

Томский Политехнический Университет

28.

</body>

</html>

Вот так должна выглядеть Ваша страница:

На странице размещено два рисунка, одна ссылка и форма с двумя полями для ввода текста и одной кнопкой. Как уже отмечалось выше, с точки зрения языка JavaScript, загруженная в настоящий момент htmlстраница, представляет собой объект *document*. Через этот объект можно получить доступ ко всем элементам, размещенным на странице. На следующем рисунке иллюстрируется иерархия объектов, создаемая HTML-страницей, приведенной в примере:

Создадим скрипт, который позволит заполнять текстовые поля формы предопределенными значениями.

B paзделе <head> ... </head> paзместите следующий код: <script> function fillIn(){ document.forms[0].elements[0].value = "Евгений"; document.forms[0].elements[1].value = "vest@list.ru"; } </script>

Привяжите запуск функции к событию onClick кнопки "Нажмите!".

Сохраните страницу и просмотрите ее в окне браузера. При правильно работе скрипта по нажатию кнопки "Нажмите!" текстовые поля должны заполняться значениями.

Так как все элементы на странице имеют свои уникальные имена (обратите внимание на параметр *name* в каждом элементе), в скрипте их можно вызывать не по индексам, а по именам в соответствующих массивах:

```
<script>
function fillIn(){
document.forms['myForm'].elements['name'].value = "Евгений";
document.forms['myForm'].elements['email'].value = "vest@list.ru";
}
</script>
```

```
или даже так:
<script>
function fillIn(){
document.myForm.name.value = "Евгений";
document.myForm.email.value = "vest@list.ru";
}
</script>
```

Видоизмените скрипт таким образом, чтобы в поле "Ваше имя" по нажатию кнопки мыши выводилась ширина и высота (в формате "ширина X высота") первой картинки, в а поле "E-mail" — второй. Для этого воспользуйтесь свойствами объекта *Image width* и *height*.

Видоизмените скрипт таким образом, чтобы название кнопки "Нажмите!" при получении кнопкой фокуса (событие *onFocus*) менялось на "Фокус получен", при потере фокуса (событие *onBlur*) — опять появлялось название "Нажмите!". Для этого вам придется изменять свойство *value* кнопки.

Видоизмените скрипт таким образом, чтобы при наведении кнопкой мыши на первый рисунок, изображения менялись местами, при наведении кнопки мыши на второй рисунок — возвращались на свои места.

Видоизмените свой скрипт таким образом, чтобы при нажатии на ссылку "Томский Политехнический Университет" сначала выводилось окно предупреждения с указанием адреса страницы, на который осуществляется переход. Для этого добавьте в описание ссылки событие *onClick*, которое будет инициировать запуск функции *alert()*, в качестве параметра этой функции передавайте свойство *href* объекта *Link*.

Видоизмените скрипт таким образом, чтобы при на ведении указателя мыши на ссылку в статус-панели браузера появлялась надпись "Томский Политехнический Университет". Для этого добавьте в описание ссылки обработчик события *onMouseOver*, который будет изменять значение свойства *status* объекта *window*.

Контрольные вопросы

- 1. Особенности программирования на JavaScript.
- 2. Поясните назначение функций click_func(), alert
- 3. Что такое интерпретируемый язык?
- 4. Поясните следущую команду
- 5.
- 6. <script language="JavaScript">
- 7. document.aircraft.src='aircraft2.jpg';
- 8. </script>
- 9. Поясните свойства объекта Image width и height

7. ЛАБОРАТОРНАЯ РАБОТА № 7

Название Изучение основ языка серверного скриптования РНР.

Цель работы: Изучить основы языка серверного скриптования РНР.

Теоретическая часть:

PHP (официальное название "PHP: Hypertext Preprocessor") — язык скриптования на стороне сервера, встраиваемый в HTML-документ.

Для того, чтобы понять, что это означает, разберем пример простого скрипта:

```
<html>
<head>
<title>Пример</title>
</head>
<body>
<?php
```

```
есно "Привет, это РНР-скрипт!"; ?>
```

```
</body>
```

</html>

Обратите внимание, не программный код выводит на экран теги html-документа, сам гипертекстовый документ содержит небольшой встроенный программный код. Иными словами, вебмастер по-прежнему может создавать гипертекстовые страницы с помошью привычного ему редактора, а затем, при необходимости, встраивать элементы динамической обработки — небольшие скрипты. В рассмотренном примере вся программа состоит из одной строки:

есно "Привет, это PHP-скрипт!";

Сервер "узнает" о том, что это исполняемая программа по специальным тегам:

<?...?>

Отличие языка *PHP* от языков клиентского программирования (таких, как *JavaScript*), состоит в том, что программный код исполняется на стороне сервера, т.е. еще до того, как страницу получит браузер пользователя.

Все страницы, содержащие программный код, должны иметь расширение .*phtml* или .*php*, однако, если вы имеете доступ к админимстрированию сервера, вы можете настроить препроцессор PHP таким образом, чтобы на содержание скрипта проверялись все страницы, включая .*html*.

Ход работы:

Вывод на экран и переменные в РНР.

С помощью текстового редактора "Блокнот" (Notepad) создайте файл *testphp1.phtml* и поместите в начало страницы следующий код:

<html>

<head>

<title>Вывод на экран и переменные в PHP</title>

<?php

есно "Привет, мир!";

?>

</head>

<body>

</body>

</html>

Запустите *Far manager*, в открывшемся окне нажмите клавиши Alt+F2, в появившемся меню выберите *FTP* и нажмите *Enter*.

Создайте учетную запись для подключения к удаленному серверу. Для этого нажмите сочетание клавиш Shift+F4, в появившемся окне в поле адреса введите *osu.cctpu.edu.ru*, поставьте галочку напротив фразы *Ask for password every time* и нажмите кнопку сохранить.

Теперь попробуйте перейти на удаленный сервер по протоколу *FTP*, для этого позиционируйте указатель на вновь созданную учетную запись и нажмите *Enter*. Появится диалоговое окно, в котором нужно указать имя пользователя и пароль, их вам предоставит преподаватель.

Как только вы получите доступ к серверу *OSU* по протоколу *FTP*, создайте на удаленной машине свою папку.

Для этого нажмите клавишу *F7* и введите имя папки, в качестве имени папки рекомендую использовать вашу фамилию.

Теперь вы можете скопировать созданный вами файл в свою папку, для этого перейдите клавишей *Tab* на диск *C*:, найдите на нем свой файл и скопируйте его, используя клавишу *F5*.

Теперь вы можете запустить свою страницу, введя в адресную строку браузера адрес: http://osu.cctpu.edu.ru/students/Baua Фамилия/testphp1.phtml

Этот скрипт не несет ничего полезного, так как только выводит на экран фразу "Привет, мир!". Для того, чтобы придать скрипту полезные функции, познакомимся с переменными. Переменная характеризуется именем, типом и значением. Имя может быть любым и включать в себя

цифры, буквы английского алфавита и разрешенные символы (например, символ подчеркивания или тире). По типу переменные делятся на целые, с плавающей запятой и символьные. Попробуем ввести в скрипт переменные каждого из перечисленных типов:

```
<html>
<html>
<title>Вывод на экран и переменные в PHP</title>
</head>
<body>
<?php
$i = 6; // целое
$d = 4.89; // дробное
$str = "PHP для всех!"; // строка
echo ($i + $d);
echo "<br>Привет, мир! ".$str;
?>
</body>
</html>
```

Обратите внимание, что все переменные в РНР должны начинаться с символа dollar (\$), тип переменной не задается в явном виде, он вычисляется препроцессором РНР в зависимости от контекста. В результате работы скрипта на экран первой строкой будет выведена сумма значений двух переменных \$i и \$d и результат конкатенации двух строк "
br>Привет, mup!" и значения переменной \$str.

Поддерживаются все арифметические операции и функции, многоуровневые скобки, логические операции, операции увеличения или уменьшения на единицу и многое другое. Кроме того, очень просто и естественно организуется сравнение *если — то — иначе*. Для этого в РНР применяется конструкция:

```
if (условие)
{
код для выполнения N1;
}
else
{
код для выполнения N2;
}
Рассмотрим несложный пример:
<html>
<head>
<title>Вывод на экран и переменные в PHP</title>
```

```
</head>
<body>
<?php
$i = 6; // целое
v = 7;
$d = 4.89; // дробное
$str = "PHP для всех!"; // строка
echo (\$i + \$d);
echo "<br>http://str;
if(\$i == \$v)
echo i + v;
}
else
echo $i.$v;
}
?>
</body>
</html>
```

При сравнении на истину применяется два знака равенства для того, чтобы интерпретатор мог без труда отличить сравнение от присваивания. Результат работы скрипта — 67, т.к \$*i* не равно \$*v*, а команда *echo* \$*i*.\$*v*; объединяет две строки: "6" и "7". Неравенство (ложь) обозначается символами !=, допустимы все остальные арифметические и логические символы и операторы (например, or, and, >, <= и т.д.).

В РНР есть средства быстрого изменения переменной на единицу в сторону увеличения или уменьшения. Для этого нужно указать имя переменной и за ней, без знака равенства, — подряд два плюса или минуса соответственно. Например, a++; — переменная a будет увеличена на единицу.

Поддерживается одновременное присваивание одного значения нескольким переменным — a = b = 4;. Введите на странице *testphp1.phtml* следующий код:

<html> <head> <title>Вывод на экран и переменные в PHP</title> </head> <body> <?php \$b = \$a = 5; echo "
переменная a=\$a, b=\$b"; \$c = \$a++; echo "
переменная a=\$a, c=\$c"; \$e = \$d = ++\$b; echo "
переменная e=\$e, d=\$d, b=\$b"; \$g = 10; \$h = \$g += 10; echo "
переменная g=\$g, h=\$h"; ?> </body> </html>

Обратите внимание, что операция присваивания тоже дает свой результат, так операция a = 5 дала результат 5, именно поэтому переменная *b* была проинициализирована в 5.

В операции \$c = \$a++; сначала было произведено присваивание, а затем переменная \$a была увеличена на 1.

В операции d = ++b; сначала была увеличена на 1 переменная b, а уже затем присвоено получившееся значение.

Измените страницу таким образом, чтобы на экран выводился результат возведения числа Πu в степень 101,2.

<u>Передача параметров по ссылке, передача параметров из формы</u> (GET и POST -- запросы).

С помощью текстового редактора "Блокнот" (Notepad) создайте файл *testphp2.phtml* и поместите в начало страницы следующий код:

<html>

<head>

<title>Вывод на экран входящих параметров</title>

</head>

<body> <?php echo \$message.", ".\$name;

?>

</body>

</html>

Скопируйте файл на сервер *OSU* в свою папку, затем запустите его: http://osu.cctpu.edu.ru/students/Baua_Фамилия/testphp2.phtml?message=h ello&name=world

Обратите внимание, что переменные *message* и *name* передаются в адресной строке, отделяются от адреса страницы знаком ?, между собой

разделены амперсантом (&). Вы можете изменить значения переменных прямо в адресной строке, нажать *Enter* и получить другой результат работы скрипта.

Создайте страницу testphp2_form.html. Задайте на ней форму: <form name="myform" method="get" action="testphp2.phtml">

Cooбщение:

<input type="text" name="message">

//мя:

<input type="text" name="name">

<input type="submit" value="Обработать данные">

</form>

Скопируйте страницу на сервер, затем запустите её и проверьте работоспособность скрипта.

Обратите внимание, что на странице с формой элементам назначены имена, соответствующие именам переменных в принимающем скрипте.

Попробуйте изменить метод *GET* на метод *POST*, теперь значения переменных не должны передаваться в открытом виде.

Вместо элемента <input type="text" name="name"> определите на странице выпадающее меню, такое, как приведено ниже:

Измените работу скрипта таким образом, чтобы пользователь мог выбрать имя из списка, ввести приветствие, нажать кнопку *"Обработать данные"* и получить приветствие для выбранного имени.

Измените страницу с формой так, чтобы пользователь мог выбрать приветствие, нажав кнопку зависимой фиксации (*radiobutton*):

Привет

• Здравствуйте

[©] Рад Вас снова видеть

• Добрый день

Измените страницу с формой так, чтобы пользователь мог выбрать фон страницы (средство выбора — на Ваше усмотрение).

Динамическое формирование страницы.

Создайте 3 файла с именами: *testphp3_inc1.html*, *testphp3_inc2.html* и *testphp3_inc3.html*. Поместите небольшое стихотворение в каждый.

Создайте файл testphp3.phtml, поместите в него следующий код:

<html>

<head>

<title>Динамическое формирование страниц</title> </head>

```
<body bgcolor=silver>
<?php
$file = "";
if ($link == 1) { $file = "testphp3_inc1.html"; }
if ($link == 2) { $file = "testphp3_inc2.html"; }
if ($link == 3) { $file = "testphp3_inc3.html"; }
if ($file == "") { ?>
<h3>Пожалуйста, выберите стихотворение:</h3>
<a href="testphp3.phtml?link=1">стихотворение N1</a><br>
<a href="testphp3.phtml?link=2">стихотворение N2</a><br>
<a href="testphp3.phtml?link=3">стихотворение N3</a>
<? } else {
include($file);
}
?>
</body>
</html>
```

Сохраните созданные файлы и скопируйте на сервер, затем запустите файл *testphp3.phtml*.

При выборе ссылки скрипту передается в качестве параметра идентификатор страницы, содержание которой должно быть включено в результирующий файл.

Измените скрипт таким образом, чтобы стихи включались в таблицу с желтым фоном; ссылка на страницу, которая импортирована в файл в настоящий момент, была неактивна.

Создайте массив *\$titles*, который будет содержать названия стихов. Из этого массива должно формироваться содержимое тега <title>...</title> в зависимости от импортируемой страницы.

Внизу страницы поместите ссылки Предыдущее, К оглавлению и Следующее. Ссылка Предыдущее должна вести к предыдущему (по порядковому номеру) стихотворению, ссылка Следующее должна вести к следующему (по порядковому номеру) стихотворению. Ссылки должны исчезать со страницы, если предыдущие/следующие стихотворения исчерпаны. Ссылка К оглавлению должна приводить страницу в первоначальный вид и исчезать на странице оглавления.

Перенесите файлы *testphp3_inc1.html*, *testphp3_inc2.html* и *testphp3_inc3.html* в папку *includes* и внесите соответствующие изменения в скрипт.

Работа с файлами.

Создайте файл *testphp4_form.html*, поместите в него форму, которая будет содержать текстовое поле *<textarea* name=message cols=10

rows=4></textarea>, поле редактирования <input type=text name=person> и кнопку для отправки данных.

Настройте форму таким образом, чтобы данные направлялись скрипту *testphp4.phtml*.

Создайте файл *testphp4.phtml*, поместите в него следующий код: <html> <head> <title>Гостевая книга</title> </head><body bgcolor=silver> <?php \$filename = "messages.txt"; \$fp = fopen(\$filename,"a"); if(\$fp) fputs(\$fp, \$message." ".\$person); fclose(\$fp); } include(\$filename); ?> </body> </html>

Создайте пустой файл *messages.txt* — в него будут сохраняться все сообщения пользователей.

Скопируйте все полученные файлы на сервер. Для того, чтобы ваш скрипт смог открывать файл *messages.txt* и сохранять в него текст, необходимо установить соответствующие права для этого файла в файловой системе сервера. Для этого позиционируйтесь на файл *messages.txt*, нажмите сочетание клавиш Ctrl+A, появится окно настроек *Set attributes*, поставьте крестики под буквами W(write) и нажмите OK.

Запустите файл *testphp4_form.html*, убедитесь в работоспособности скрипта.

Должно происходить следующее: переменные *\$message* и *\$person* принимаются скриптом *testphp4.phtml*, открывается файл *messages.txt* (делается это методом *fopen*), помещает в его конец строку, состоящую из значений переданных переменных, разделенных пробелом (делается это методом *fputs*).

Измените работу скрипта таким образом, чтобы сообщения и имена пользователей выводились в отформатированном виде, например, поместите их в ячейки таблицы. Измените работу скрипта таким образом, чтобы форма находилась в файле *testphp4.phtml*, иными словами, удалите файл *testphp4_form.html*.

Измените работу скрипта таким образом, чтобы под каждым сообщением пользователя выводилась дата его размещения. Для этого используйте функцию.

В начало страницы поместите картинку.

Атрибуты тега * width* и *height* задайте автоматически, для этого используйте функцию GetImageSize.

Игра "Угадай число".

Создайте файл *testphp5_form.html*, поместите в него форму, которая будет содержать поле редактирования и кнопку для отправки данных.

Настройте форму таким образом, чтобы данные направлялись скрипту *testphp5.phtml*.

Скрипт *testphp5.phtml* должен работать по следующей схеме: от формы на странице *testphp5_form.html* принимается целое число, методом *rand* генерируется случайное целое число, если предположение пользователя и выбор сервера совпадают, должно выводится поздравительное сообщение, иначе должно выводиться сообщение с предложение нием повторить игру.

Предложите методы проверки входящих параметров — необходимо исключить возможность ввода пользователем текстовых сообщений.

Контрольные вопросы

Что такое язык скриптов?

- 1. Какое расширение должны иметь страницы, содержащие программный код?
- 2. Как происходит вывод на экран страниц РНР?
- 3. Какие переменные РНР вы знаете?
- 4. Как происходит передача параметров по ссылке, передача параметров из формы (GET и POST -- запросы)?
- 5. Что такое динамическое формирование страницы?

8. ПРАКТИЧЕСКИЯ РАБОТА №1

Название: Изучение основных этапов разработки, внедрения и раскрутки веб-представительства

Теоретическая часть

Планирование.

Для многих обычных людей слово "сайт" - это однозначное понятие: "несколько страничек в интернете, на которых можно что-то почитать, что-то посмотреть, что-то скачать и т.д.". Что же касается вебмастера, то когда он слышит от кого-то "мне нужен сайт" - это выражение как правило практически ни о чем ему не говорит.

И все-таки, первое что нужно понять услышав заветные "мне нужен сайт" - это, что же именно требуется сделать. И отсюда вытекает первая и главная составляющая сайтостроительства - помочь заказчику понять чего он хочет сам. Решить же эту задачу нам помогут ответы на следующие вопросы:

- 1. Будет ли сайт коммерческим
- 2. Кто будет посещать будущий сайт? (потенциальная аудитория)
- 3. Что на сайте увидят посетители? (контент, функционал)
- 4. Чем сайт будет удобен посетителям? (юзабилити, дизайн, функционал)
- 5. Чем сайт будет уникален? (или он ничем не будет уникален)
- 6. Как сайт будет приносить прибыль? (монетизация, моральное удовлетворение, и т.д.)

Ответы на эти, казалось бы простые, но на самом деле весьма и весьма сложные и объемные по информации вопросы дадут нам представление о том, чего же хочет заказчик. И, конечно же, из ответов на данные вопросы будет строиться дальнейшая стратегия всего сайта в целом, и такие параметры как функционал, дизайн контент и пр. будут полностью зависеть от этих ответов.

Отсюда вытекает первый этап работы над сайтом:

Первостепенная задача при создании сайта - понять чего конкретно хочет заказчик.

При изучении, можно сказать философского вопроса необходимости заказчика в сайте, вполне может выясниться и что, как таковой, сайт заказчику и совершенно не нужен, а также возможны варианты когда заказчик будет не готов к возможным затратам на создание сайта или недооценил уровня текущей конкуренции среди уже имеющихся сайтов. Короче, заказчик может перегореть желанием создать свой сайт. Если же этого не произошло, то после понимания сути вопроса можно действовать по стандартной уже обкатанной схеме создания сайта.

Следующим этапом создания сайта будет построение структуры сайта. Фактически на данном этапе происходит построение карты сайта (меню сайта), определяется назначение тех или иных пунктов меню и компонентов, составляющих систему сайта, выясняются группы и роли пользователь, которые будут участвовать в дальнейшей жизни сайта, определяются параметры безопасности.

По завершении данного этапа наступает следующий этап.

Проработка системы управления сайтов и подбор необходимых составляющих компонентов для реализации всего закладываемого в будущий сайт функционала.

На данном этапе оцениваются все возможности имеющихся в распоряжении CMS и делается выбор в пользу одной, наиболее подходящей для реализации будущего функционала сайта. Также разработанная структура разбивается на отдельные функциональные модули с проработкой стратегии наиболее быстрого и качественного внедрения каждого из функциональных модулей. На этом этапе также происходит распределение ролей участников процесса создания сайта, оговариваются временные рамки.

Это последний этап планирования сайта, после которого начинается понятие разработка.

Разработка.

Разработку рекомендуется вести на отдельном тестовом хостинге с возможностями доступа заказчика к нему. Разработка может как совмещаться с внедрением уже готовых модулей сайта, так и идти отдельным этапом, по окнчании которого внедряется весь сайт полностью готовый к эксплуатации.

Первый этап разработки сайта - отрисовка шаблонов макетов дизайна.

На данном этапе реализуются спланированные заранее идеи относительно дизайна сайта. Чем сложнее сайт, тем большее количество функциональных модулей он будет иметь, и тем большее количество шаблонов дизайна понадобится отрисовывать. Окончанием работ по данному этапу будет считаться готовый к верстке psd-макет с шаблоном дизайнов модулей. При этом psd-макет сайта должен обладать рядом отличительных свойств, позволяющих html-верстальщику быстро и без особых проблем сверстать его, а также учитывать специфики создания веб-сайтов, обладать достаточной вместимостью и не иметь сложных к реализации с точки зрения веб-дизайна элементов. Все psd-макеты сайта передаются в собственность заказчика и могут быть использованы им в дальнейшем по своему собственному усмотрению. Фактически, передача макетов заказчику является признаком окончания всех работ по дизайну.

Второй этап разработки сайта - верстка отрисованных шаблонов макетов дизайна.

По результатам работы на данном этапе заказчик должен получить на руки сверстанные макеты веб-сайта. Верстка макетов должна соответствовать требованиям валидности (не должна содержать ошибок), кроссбраузерности (должна смотреться одинаково хорошо в различных браузерах), а также смотреться одинаково хорошо в различных разрешениях экрана и соответствовать некоторым другим правилам html-верстки. На этапе верстки также происходит отладка и программирование Java Script спецэффектов сайта.

Третий этап разработки сайта - развертывание CMS будущего сайта.

Сюда входят настройка системы управления сайтом на хостинге, создание баз данных, создание групп пользователей и разграничение прав доступа, различные настройки системы.

Четвертый этап разработки сайта - настройка всех компонентов и шаблонов дизайна всех модулей системы и текстовых страниц.

В данный этап входит настройка всех функциональных модулей системы, а также настройка общих шаблонов дизайна будущего сайта и настройка шаблонов дизайна каждого из модулей с доработкой функционала. По завершению данного этапа заказчик получает возможность увидеть в целом картину будущего сайта и протестировать его в браузре.

Внедрение.

После того, как сайт прошел этапы планирования и разработки - он готов к внедрению. В понятие внедрения входят наполнение сайта контентом, тестирование функционала и перенос сайта и баз данных на рабочий хостинг.

Готовый разработанный сайт тестируется на предмет возникновения различного рода ошибок, связанных с перегруженностью программного кода, нагруженностью баз данных, большим количеством посетителей и пр. факторов. Все выявленные ошибки протоколируются и исправляются.

Одновременно с тестированием сайта ведется его наполнение. Возможны варианты как ручного так и скриптового автоматического наполнения данными. По окончании всех работ сайт вместе с базами данных переносится на рабочий хостинг, где начинает работать уже в полную меру. Дальнейшее сотрудничество с заказчиком возможно в плане регулярного наполнения сайта, раскрутки, рекламы и продвижения в интернет, либо саппорта, а также в плане разработки нового функционала, в случае необходимости.

В целом же все цели создания сайтов можно разбить на следующие группы:

- Онлайн-представительство компании/бренда (сайт-визитка/промосайт). Возможность дать потенциальному клиенту, привлеченному оффлайн посмотреть на то, чем занимается компания, ознакомится с продукцией, узнать где располагаются офисы и т.д. Вообщем обеспечить посетителя информацией.
- 2. Сайт под продвижение в интернете. Основная задача сайта привлекать новых клиентов в сети интернет. Как правило данная цель ставится по причине относительно низкой стоимости продвижения в сети интернет по отношению к другим источникам рекламы, учитывая достигаемый эффект охвата требуемых целевых аудиторий.
- 3. Заработок на сайте (Moneymaking). Главная задача сайта получение максимальной прибыли различными способами.
- 4. Функциональные системы. Построение системы функциональных возможностей, требуемых для обеспечения координации лиц, задействованных в рабочих и бизнес-процессах.

Выбор приоритета в поставленных целях, которым будет служить сайт, характеризует степень смещения акцента в области планирования достижения будущих показателей дизайна, функциональности, наполнения, юзабилити, роста SEO и др. параметров создаваемого сайта. Конечно хорошо когда все данные показатели планируются изначально на максимально высоком уровне. Однако в условиях ограниченного бюджета создания сайта сделать все по высшему разряду не получится.

Так например, качество дизайна будущего сайта играют большую роль как фактор успешности будущего проекта для 1-ой и 3-ей группы целей. А для 4-ой группы целей создания сайтов иногда можно обойтись и вообще без дизайна, здесь главный параметр это юзабилити, производительность и оперативность работы создаваемой системы, а также отказоустойчивость и защита от взлома и, например, кражи данных.

Что касается 3-ей группы целей - здесь самое главное, это привлечь и удержать на сайте как можно большее количество посетителей и превратить их в постоянных пользователей сайта.

Ход работы

- 1. Обсудить в группах этапы внедрения сайтов. Сделать выводы и законспектировать в отчет по практическим работа.
- 2. Дать оценку целям создания сайта. Законспектировать цели в отчет, предложить свои варианты.
- 3. Обосновать собственную необходимость в создании сайта (блог, визитка, форум и пр.)

9. ПРАКТИЧЕСКАЯ РАБОТА №2

Название: Разработка технического задания на создание Интернет сайта

Теоретическая часть ОБРАЗЕЦ ТЕХНИЧЕСКОГО ЗАДАНИЯ НА РАЗРАБОТКУ WEB-САЙТА

Техническое задание на разработку Веб-сайта

- 1. Имя сайта (название домена). <u>www._ru</u> . (Если домен www. .ru будет занят, возможна замена имени)
- 2. Название сайта. Сайт ООО "_____". Далее Фирма.
- 3. Назначение сайта (цель создания сайта).
- 4. Представление Фирмы в Интернет: информация о Фирме, история Фирмы, партнёры Фирмы, Заказчики Фирмы, цены на оказываемые услуги, справочная техническая и юридическая информация, советы клиентам, сопроводительные графические рисунки, юридический адрес, почтовый адрес, схема проезда, контактная информация, банковские реквизиты, сведения об имеющихся вакансиях. Сайт должен способствовать привлечению клиентов и нахождению деловых партнеров.
- 5. Язык сайта. Русский. Основные ключевые слова, по которым сайт должны находить по запросам в поисковых системах и Интернет каталогах.

Согласно материалам Заказчика.

Примечание:

Перечень ключевых слов для Веб-дизайнера сайта носит справочный характер и не входит в число обязательных параметров, подлежащих проверке при приемке сайта. Занимаемые сайтом позиции в рейтингах, каталогах и поисковых системах не оговариваются.

- 6. Объём и состав текстовой и графической информации в электронном виде. Согласно материалам Заказчика.
- 7. Предполагаемая возрастная аудитория сайта. От 30 лет и старше.
- 7.1. Предполагаемое возрастное ядро аудитории от 35 до 50 лет.

- 7.2. Данная информация носит рекомендательный характер. Цифровые показатели контролю и проверке при приёмке сайта не подлежат.
- Количество страниц сайта.
 Сайт должен содержать следующие html страницы: 1 Главная (домашняя) страница; 2 - Прайс-лист; 3 - Фото (каталог) товаров; 4,5,6,7,8,9,10 - Справочная информация; 11 - О Фирме; 12 - Офис; 13 - Партнёры; 14 - Вакансии; 15 - Потребности; 16 - Сервисы.
- Количество html страниц сайта определяется Веб-дизайнером самостоятельно, исходя из объёма предоставленных материалов Заказчика.

Кнопки управления (навигация сайта).

Определяются Веб-дизайнером самостоятельно.

С каждой страницы сайта должен быть обеспечен переход (установлена гиперссылка) на главную страницу сайта. Сайт должен содержать страницу "Содержание" (карта сайта).

- Блок схема сайта.
 Определяется Веб-дизайнером самостоятельно.
 Головная (начальная) страница сайта должна содержать гиперссылки, обеспечивающие переход с нее на не менее чем 95% страниц сайта, но не более чем 160 гиперссылок.
- 11. Объём сайта, Мб. Не оговаривается.
- 12. Оформление рисунков.

Все рисунки объемом более 1 Кб должны быть выполнены с замещающим текстом. Рисунки размером более 15 Кб должны быть выполнены с предпросмотром. Формат всех рисунков gif или jpg (jpeg).

13. Пропускная способность линии связи.

Среднее время загрузки страниц не должно превышать 28 секунд при скорости соединения 28.8 Кбит/сек. Допускается увеличение времени загрузки отдельных страниц до 36 секунд, но не более чем на 30% числа страниц сайта. Головная (начальная) страница должна иметь время загрузки не более 40 секунд.

Примечание: Во всех случаях не учитывается время загрузки подгружаемых элементов (счетчики, баннеры, информеры и т.д.).

- 14. Основной диапазон разрешения мониторов, на которых будет просматриваться сайт. От 600х800 до 1240х1024 пикселей. Основное разрешение, на которое оптимизируется сайт: 1024х768 пикселей.
- 15. Минимальное разрешение монитора, на котором будет просматриваться сайт 600 х 800 пикселей. При указанном разрешении возможность просмотра страниц сайта без горизонтальной прокрутки браузера не предусматривается.

- 16. Основной браузер, которым будет просматриваться сайт, и его минимальная версия. IE 5.5 и выше.
- 17. Цветовая палитра.

Основной режим мониторов, на которых будет просматриваться сайт: 15 разрядов цветов и выше (число цветов 65536 и выше). При разработке сайта должен быть обеспечена возможность его просмотра при использовании безопасной цветовой палитры (разрядность цветов 8). Изменения оттенков цветов, при просмотре сайта с использованием безопасной цветовой палитры, не оговариваются.

18. Общий фон сайта.

Общий фон сайта светлый (белый). Допускается использование светлого фонового рисунка.

- 19. Размер и вид шрифта сайта. Размер шрифта сайта должен быть в пределах 10-12 для оформления текста. Размер шрифта для оформления заголовков, названия страниц и т.д. не оговаривается. Вид (название) шрифта не оговаривается.
- 20. Регистрация сайта в каталогах, рейтингах, топах и пр. Оговаривается дополнительно.
- 21. Проведение рекламной кампании по раскрутке сайта. Раскрутка сайта определяется отдельным техническим заданием. В настоящем техническом задании раскрутка сайта не оговаривается и не входит в состав выполняемых работ (услуг).
- 22. Срок разработки сайта.
- 23. Три недели со дня зачисления 70% предоплаты на расчётный счёт Веб-студии.

Порядок передачи сайта.

Веб-дизайнер передает сайт на CD ROM, а также логин, пароль и название (код передачи данных) по протоколу ftp.

Заказчик обязан проверить наличие грамматических и орфографических ошибок на сайте в течение трех рабочих дней. Обнаруженные ошибки Веб-дизайнер обязан устранить в течение трех рабочих дней.

24. Сопровождение сайта.

Сопровождение сайта определяется отдельным техническим заданием. В настоящем техническом задании сопровождение сайта не оговаривается и не входит в состав выполняемых работ (услуг).

 Дополнительные условия. Каждая страница сайта должна содержать логотип и название Фирмы. Внизу на каждой странице сайта должна быть указана контактная информация.

Сайт должен содержать не менее двух счетчиков подсчета посетителей.

Материалы предоставляемые Заказчиком:

Текстовая (формат Word) и графическая информация (формат jpeg и gif), представленные на CD ROM.

Примечание:

- Задание на сайт может быть изменено с учетом конкретных требований.
- Задание на сайт предназначено для русскоязычных сайтов, объемом не более 50 html страниц. Если сайт имеет версию на иностранном языке или версию для просмотра на мобильных устройствах, задание на сайт должно быть дополнено соответствующими пунктами.
- Веб-дизайнер не несет ответственности за несоответствие сайта эстетическим ожиданиям Заказчика при условии выполнения технического задания на сайт.

Подписи Сторон: Веб-студия "Лада"

Заказчик

Ход работы

1. Используя теоретическую часть составить техническое задание на разработку сайта, обоснование которого проведено в практической работе №1.

10. ПРАКТИЧЕСКАЯ РАБОТА №3

Название: Изучение работы международных и российских организаций в области стандартизации информационных систем

Теоретический материал

Международная организация по стандартизации, ИСО (ISO), содействует развитию стандартизации и активизации роли стандартов во всем мире. Ее основной задачей является развитие сотрудничества и международный обмен в интеллектуальной, научной, технической и экономической сферах деятельности. Эта неправительственная организация, учрежденная в 1947 г., в настоящее время объединяет представителей из 140 стран. Результатом деятельности ИСО является публикация согласованных международных стандартов во всех направлениях жизнедеятельности.

Деятельность ИСО осуществляется следующим образом. Представители различных государств самостоятельно выбирают для себя форму участия в работе организации и могут являться:

- действительными членами;
- членами с совещательным голосом;
- наблюдателями.

Международная стандартизация — это совокупность организаций по стандартизации и продуктов их деятельности: стандартов, рекомендаций, технических отчетов и другой научно-технической продукции.

На этой страничке дана информация об основных международных (ИСО, МЭК, МСЭ, ЕОК) и региональных (СЕН, СЕНЭЛЕК, ЕТСИ) организациях по стандартизации.

Международная организация по стандартизации ИСО (ISO) создана в 1946 году.Имеет неправительственный характер. При создании организации и выборе ее названия учитывалась необходимость того, чтобы аббревиатура наименования звучала одинаково на всех языках. Для этого было решено использовать греческое слово isos - равный. Вот почему на всех языках мира Международная организация по стандартизации имеет краткое название ISO (ИСО).

Главная цель ИСО - содействие стандартизации в мировом масштабе для облегчения международного товарообмена и взаимопомощи, а также для расширения сотрудничества в области интеллектуальной, научной, технической и экономической деятельности путем разработки**международных стандартов**. Членами ИСО являются не правительства, а **национальные организации по стандартизации**. Главными структурными подразделениями ИСО являются технические комитеты, подкомитеты и рабочие группы, выполняющие основной вид деятельнсти - разработку международных стандартов.

Международная электротехническая комиссия МЭК (IEC) создана в 1906 году. Цель деятельности - содействие международному сотрудничеству по вопросам стандартизации в области электротехники, радиоэлектротехники и связи. В отличие от ИСО, МЭК состоит из национальных комитетов, которые представляют интересы всех отраслей промышленности. В качестве таких национальных комитетов выступают национальные организации по стандартизации. Между ИСО и МЭК заключено соглашение по разграничению сфер деятельности. Бюджет ИСО и МЭК складывается из взносов стран и от продажи международных стандартов. Международный союз электросвязи МСЭ (ITU) — это международная организация, координирующая деятельность государственных организаций и коммерческих компаний по развитию сетей и услуг электросвязи в мире. Корни МСЭ уходят в 60-е гг. XIX в., когда была подписана первая Международная телеграфная конвенция (1865 г.). Большим достижением МСЭ является принятие в 1999 г. Рекомендаций по системе телевидения высокой четкости. В ней зафиксированы базовые параметры (число строк разложения, формат кадра, система развертки) телевидения XXI века. Парк стандартов МСЭ составляет 1500 единиц.

Международная электротехническая комиссия — IEC (МЭК) разрабатывает стандарты в области электротехники, радиоэлектроники, связи. Она была создана в 1906 г., т.е. задолго до образования ИСО. Разновременность образования и разная направленность МЭК и ИСО определили факт параллельного существования двух крупных международных организаций. С учетом общности задач ИСО и МЭК, а также возможности дублирования деятельности отдельных технических органов между этими организациями заключено соглашение, которое направлено, с одной стороны, на разграничение сферы деятельности, а с другой — на координацию технической деятельности.

Число членов МЭК (62 страны по состоянию на 2006 год) меньше, чем членов ИСО. Это обусловлено тем, что многие развивающиеся страны практически не имеют или имеют слаборазвитую электротехнику, электронику и связь. Россия является членом МЭК с 1911 года. Высший руководящий орган МЭК — Совет, в котором представлены все национальные комитеты. Бюджет МЭК, как и бюджет ИСО, складывается из взносов стран - членов этой организации и поступлений от продажи международных стандартов. Структура технических органов МЭК такая же, как и ИСО: технические комитеты, подкомитеты и рабочие группы. В МЭК функционируют 174 комитета и подкомитета, часть которых (как и в ИСО) разрабатывает международные стандарты (MC) общетехнического и межотраслевого характера, а другая — МС на конкретные виды продукции (бытовая радиоэлектронная аппаратура, трансформаторы, изделия электронной техники).

В настоящее время разработано свыше 5200 стандартов, технических отчетов, рекомендаций. Следует отметить важность проводимых МЭК работ по установлению требований безопасности для бытовых электроприборов и машин. В связи с различным подходом к обеспечению безопасности в разных странах ТК 61 «Безопасность бытовых электроприборов» выпущено более 40 МС, устанавливающих требования практически ко всем электробытовым приборам и машинам. Разработка МС в этой области имеет особенно важное значение в связи с созданием в МЭК системы сертификации электробытовых приборов и машин на соответствие их МС МЭК.

К настоящему времени на международном уровне сформировалась мощная корпорация организаций, разрабатывающих стандарты в области информатизации. Это ИСО, МЭК, МСЭ-Т, СЕН, СЕНЕЛЕК и др.

В России подобными разработками занимаются Госстандарт России, Минсвязи России и РАН.

В настоящее время из 1500 международных стандартов в области информатизации в России разработано порядка 350 стандартов (ГОСТ, ГОСТ Р, РД, Р, Пр), что составляет 28%.

С целью обеспечения совместимости сетей и систем федерального, регионального, корпоративного и других уровней Минсвязи России разработал Государственный профиль взаимосвязи открытых систем России, который был принят и введен Постановлением Госстандарта России в виде Рекомендаций Р50.1.022-2000.

Анализ состояния работ по стандартизации в области информатизации показывает, что наиболее значительное отставание в разработке стандартов происходит в таких важных тематических областях, как терминология - 48% (должно быть 100%), программная инженерия (12,5%), взаимосвязь открытых систем (35%), функциональные стандарты (29%), защита информации (24%). Полностью отсутствуют стандарты, носящие общетехнический и межотраслевой характер, стандарты, определяющие классификацию, требования к информационным ресурсам и услугам и т.д..

Выбор основных направлений развития работ по стандартизации в сфере информатизации, финансируемых за счет ограниченных бюджетных средств, должен быть направлен на разработку нормативных документов общесистемного и межотраслевого характера и на определение методологии построения информационных систем, образующих в конечном итоге национальную информационную инфраструктуру (НИИ).

Данный подход следует рассматривать в трех направлениях:

- реализация госполитики;
- разработка проблемно-ориентированных функциональных стандартов;
- разработка первоочередных взаимоувязанных стандартов.

Из достаточно большого количества стандартов следует сделать оптимально-минимальную выборку только тех стандартов, которые определяют общие требования, основные положения, эталонные модели и

т.д.. Разработка таких стандартов должна проводиться в рамках бюджетных средств.

На основании вышеизложенного следует, что работы по стандартизации в сфере информатизации должны проводиться по следующим четырем этапам:

Этап 1. Создание системы стандартизации в сфере информатизации (2000 г.).

Этап 2. Разработка и сопровождение нормативных документов общефункционального назначения (2000-2001 гг.)

Этап 3. Разработка проблемно-ориентированных функциональных стандартов (профилей) на договорной основе.

Этап 4. Разработка и сопровождение основополагающих (базовых) нормативных документов по 5-ти наиболее важным направлениям в сфере информатизации

Ход работы:

1.Используя теоретический материал и интернет-ресурсы, выявить цели и задачи международных и российских органов по стандартизации в области информационных ресурсов.

11. ПРАКТИЧЕСКАЯ РАБОТА 4

Название: Общие сведения о жизненном цикле вебстраниц ASP.NET

Теоретический материал

ASP.NET надстраивает над протоколом НТТР еще один уровень абстракции, с помощью которого разработчики создают Web-сайты. Благодаря ASP.NET программисты могут применять такие высокоуровневые сущности, как классы и компоненты в рамках объектноориентированной парадигмы. А специальный инструментарий делает работу в инфраструктуре ASP.NET настолько целостной и быстрой, насколько это вообще возможно. В конечном счете, именно инструменты разработки отвечают за создание и развертывание приложения. Они выдвигают парадигму программирования и заставляют разработчиков играть по ее правилам.

Ключевой инструмент разработки приложений ASP.NET 2.0 — Microsoft Visual Studio 2005 — преемница Visual Studio .NET 2003. Она содержит массу новых возможностей и усовершенствований, разработанных специально для того, чтобы позволить Web-программистам пре-

одолеть некоторые ограничения, возникавшие при использовании Visual Studio .NET 2003.

Visual Studio 2005 — это среда-контейнер, объединяющая функциональность нескольких визуальных дизайнеров. Один дизайнер служит для создания приложений на основе Windows Forms, один — для создания ASP.NET-сайтов, еще один — для Web-сервисов и т. д. Visual Studio .NET 2003 использовала единственную модель для разработки приложений — проектно-ориентированный подход. Однако в реальности такой подход оказался не лучшим вариантом — во всяком случае, для ASP.NET и Web-приложений.

Создание Web-сайта

Давайте создадим демонстрационный Web-сайт с помощью Visual Studio 2005. Новый Web-сайт можно создать, выбрав пункт меню File | New, а затем Web Site. Появившееся диалоговое окно предлагает выбрать тип сайта, который вы хотите создать. Обратите внимание на два похожих варианта: ASP.NET Web Site и ASP.NET Internet Site.

Если вы выберете пункт Web Site, Visual Studio сгенерирует минимальное число файлов, необходимых для создания Web-сайта. В сущности, будут созданы страница default.aspx и пустой каталог Data. Если же вы укажете вариант Internet Site, то будет задействован стартовый набор ASP.NET — образец персонального Web-сайта с некоторыми встроенными стандартными возможностями. Давайте выберем вариант под названием Web Site.

Для редактирования Web-страницы можно выбрать одно из трех представлений — Design, Source и Server Code. Режим Design показывает HTML-страницу, позволяя выбирать и редактировать управляющие и статические элементы, а также предоставляет возможность предварительного графического просмотра страницы. Режим Source показывает разметку HTML вместе со встроенным кодом, используя при этом такие богатые возможности, как подсветку синтаксиса, IntelliSense, подсказки и автозавершение. Режим Server Code показывает только встроенный код, если он есть. Хорошая новость: теперь Visual Studio поддерживает подсветку синтаксиса и встроенного кода.

Ход работы

Создание Web-сайта

Visual Studio 2005 создает файл проекта, но не использует его для отслеживания всех файлов, входящих в приложение. Корневой каталог сайта сам неявно определяет Web-проект. Для добавления к проекту нового файла его надо просто скопировать или создать в этом каталоге, и

он тут же появится в проекте. Если этого не происходит, щелкните правой кнопкой в Solution Explorer и выберите Refresh Folder.

Добавим HTML-код для создания страницы Hello, World. Как минимум, нам нужно текстовое поле для ввода сообщения и кнопка, чтобы послать его всему миру. Следующий HTML-код формирует страницу:

```
<% @Page language="C#" %>
<script runat="server">
void Send_Click(object sender, EventArgs e)
ł
  MsgSent.Text = Msg.Text;
</script>
<html>
<head runat="server">
  <title>Hello, World</title>
</head>
<body>
  <form runat="server" id="MainForm">
    <h1>Send a message to the world</h1>
    <asp:textbox runat="server" id="Msg"
      text="Hello, ASP.NET 2.0" />
    <asp:button runat="server" id="Send" text="Send"
           onclick="Send_Click" />
    <hr />
    <b>Last message sent: </b>
    <asp:label runat="server" id="MsgSent" Font-Italic="True" />
    <hr />
  </form>
</body>
</html>
```

Добавим HTML-код для создания страницы Hello, World. Как минимум, нам нужно текстовое поле для ввода сообщения и кнопка, чтобы послать его всему миру. Следующий HTML-код формирует страницу:

```
<% @Page language="C#" %>
<script runat="server">
void Send_Click(object sender, EventArgs e)
{
MsgSent.Text = Msg.Text;
```

```
}
</script>
<html>
<head runat="server">
  <title>Hello. World</title>
</head>
<body>
  <form runat="server" id="MainForm">
    <h1>Send a message to the world</h1>
    <asp:textbox runat="server" id="Msg"
      text="Hello, ASP.NET 2.0" />
    <asp:button runat="server" id="Send" text="Send"
           onclick="Send_Click" />
    <hr />
    <b>Last message sent: </b>
    <asp:label runat="server" id="MsgSent" Font-Italic="True" />
    <hr />
  </form>
</body>
</html>
```

Компиляция по требованию для всех ресурсов

ASP.NET 1.х поддерживает динамическую компиляцию файлов нескольких типов: страниц ASP.NET (.aspx), Web-сервисов (.asmx), пользовательских элементов управления (.ascx), обработчиков HTTP (.ashx) и файлов global.asax. Перечисленные файлы автоматически компилируются по требованию при первом обращении к ним со стороны Webприложения. Скомпилированная копия становится недействительной, как только в соответствующем исходном файле будут обнаружены изменения. Такая система позволяет програм-мистам быстро и с минимальными издержками разрабатывать приложения — вы просто нажимаете Save, и все готово.

В ASP.NET 2.0 возможность компиляции по требованию предусмотрена и для других типов файлов: обычных файлов с классами (.vb и .cs), ресурсных файлов (.resx), файлов обнаружения Web-сервисов (.wsdl) и файлов схем типизированных наборов данных (.xsd). Поскольку исполняющая среда ASP.NET гарантирует, что все изменения в указанных файлах отслеживаются незамедлительно, необходимость в явной компиляции отпала. Изменения в ASP.NET сделали возможным пересмотр всего механизма отделенного кода. В то же время система компоновки ASP.NET 2.0 совместима со схемой отделенного кода в приложениях, созданных в более старых версиях ASP.NET.

Жизненный цикл страницы

Страница ASP.NET начинает свой жизненный путь, когда исполняющая среда ASP.NET вызывает метод *IHttpHandler::ProcessRequest* для динамического класса, созданного в качестве оболочки исходного кода запрашиваемого ресурса .aspx. В этом методе заполняются встроенные элементы страницы, такие как контекст HTTP и объекты *Request* и *Response*. Затем метод строит дерево управляющих элементов, объявленных в коде .aspx, и выясняет, не выполняется ли страница в режиме возврата формы, обратного вызова или возврата формы для другой страницы. (О режиме обратного вызова я подробно расскажу позднее.) Первую возможность вступить в игру разработчики получают, когда происходит событие *PreInit*.

• **Предынициализация страницы** Когда обработчик события *PreInit* возвращает управление, страница загружает информацию персонализации и инициализирует темы. Это подходящий момент для вмешательства, чтобы установить тему страницы программным путем.

• Инициализация страницы Каждому управляющему элементу страницы рекурсивно предоставляется возможность инициализации и настройки состояния. Дочерние элементы генерируют собственные события *Init* до того, как будет сгенерировано событие *Init* формы-контейнера.

• Завершение инициализации страницы Событие *InitComplete* возникает в конце цикла инициализации всех дочерних управляющих элементов. Между *Init* и *InitComplete* выполняется только одно действие — происходит настройка отслеживания состояния отображения управляющих элементов.

• У каждого элемента управления вызывается метод *TrackViewState*. По умолчанию этот метод устанавливает флаг, заставляющий элемент отслеживать изменения в его состоянии отображения.

• Загрузка состояния элемента управления и состояния отображения В ASP.NET 2.0 состояние элемента управления — это некое защищенное состояние отображения, которое обязан поддерживать каждый управляющий элемент. В отличие от состояния отображения состояние управляющего элемента нельзя модифицировать программно или отключить.

Состояние управляющего элемента загружается перед состоянием отображения. Оба этих шага не генерируют событий, но могут быть изменены путем переопределения пары защищенных методов.

• Загрузка возвращенных данных Возвращенные формой значения обрабатываются, и их ID сопоставляются с ID объявленных управляющих элементов. Если совпадение обнаружено, возвращенное значение обновляет состояние элемента управления. (Как это происходит у конкретных элементов, зависит от их реализации интерфейса *IPostBackDataHandler*.)

• Предзагрузка страницы Событие *PreLoad* происходит, когда страница заканчивает обработку всех значений, возвращенных из формы. Вы можете обработать это событие, если хотите выполнить какие-то действия перед началом загрузки страницы.

Загрузка страницы Событие Load генерируется рекурсивно для всех элементов, содержащихся на странице, а затем и для самой страницы. Перед тем, как стадия загрузки завершится, выполняется несколько интересных действий. Страница делает вторую попытку найти соответствие между возвращенными из формы значениями и управляющими элементами. Эта попытка нужна для загрузки возможного состояния динамических элементов управления, созданных или восстановленных в событии Load. Если страница содержит обработчик обратного вызова по событию, это событие будет сгенерировано в данный момент. И, наконец, если возвращенные формой данные модифицируют такие свойства дочернего элемента, которые требуют отправки уведомления об изменении, то генерируются соответствующие уведомления. Так, если возвращенные значения изменяют свойство Text элемента TextBox, генерируется серверное событие TextChanged.

• События возврата формы Страница выполняет серверный код, связанный с событием на клиентской стороне (например, щелчком кнопкой мыши), которое привело к возврату формы. Этот шаг — основа модели событий в ASP.NET.

• Завершение загрузки страницы Возникновение события *LoadComplete* отмечает окончание предварительных действий. С этого момента страница переходит в фазу рендеринга.
• **Предрендеринг** Перед генерацией события *PreRender* страница проверяет, что все нужные дочерние элементы созданы. Событие *PreRender* сначала генерируется для самой страницы, а затем рекурсивно для всех дочерних элементов. В конце этого цикла страница вызывает событие *PreRenderComplete* и сохраняет данные персонализации, состояния управляющих элементов и представлений.

• Рендеринг На этом этапе создается разметка, отправляемая клиенту. Реализацию по умолчанию можно изменить, переопределив метод *Render*. С этой фазой не связаны никакие пользовательские события.

• Выгрузка страницы В этот момент выполняется очистка ресурсов перед освобождением объекта страницы. Событие, которое вам нужно перехватить, называется Unload. Дочерние элементы выгружаются перед выгрузкой страницы. За событием Unload следует событие Dispose, сигнализирующее об уничтожении объекта страницы.

Программирование страницы

По сравнению с реализацией в ASP.NET 1.х класс *Page* в ASP.NET 2.0 предоставляет несколько новых возможностей, таких как возврат формы для другой страницы, разбиение содержимого страницы на части и персонализация. В этом разделе мы подробнее познакомимся с первыми двумя пунктами, отложив персонализацию до главы 4. Возврат формы для другой страницы (cross-page posting) — это функциональность, давно и настоятельно требуемая сообществом разработчиков ASP.NET. Разбиение страницы на части — результат плотной интеграции между ASP.NET и Mobile ASP.NET. Но начнем мы наш беглый обзор с другой долгожданной функциональности — возможности программирования тэга *<head>* Web-страницы.

Элемент управления HtmlHead

Управляющий элемент *HtmlHead* входит в пространство имен *System.Web.UI.HtmlControls*. Экземпляр этого элемента создается автоматически, если страница содержит тэг *<head>* с атрибутом *runat=server*. Обратите внимание: этот параметр задается по умолчанию, если вы добавляете новую страницу к Web-проекту в Visual Studio 2005, как продемонстрировано в следующем фрагменте:

<head runat="server">

<title>Untitled Page</title>

</head>

Заголовок страницы можно получить с помощью нового свойства *Header* класса *Page*. Свойство возвращает *null*, если тэг *<head>* отсутствует, или не содержит атрибута *runat*.

Элемент управления реализует интерфейс *IPageHeader*, состоящий из трех свойств-наборов — *Metadata*, *LinkedStylesheet*, *Stylesheet* — и строкового свойства *Title*. Свойство *Metadata* — это словарь, содержащий все дочерние тэги <*meta*> заголовка:

Header.Metadata.Add("CODE-LANGUAGE", "C#");

Этот код приводит к генерации следующей разметки:

<meta name="CODE-LANGUAGE" content="C#" />

Чтобы создать другие виды метаданных, например *Http-Equiv*, можно прибегнуть к добавлению литеральных управляющих элементов к набору *Controls* заголовка. Обратите внимание на необходимость явного приведения объекта *Header* к типу *Control*, поскольку свойство *Header* объявлено с типом *IPageHeader*, у которого нет свойства *Controls*:

LiteralControl equiv;

equiv = new LiteralControl ("<meta httpequiv='refresh' content='3' />")

((Control) Header).Controls.Add(equiv);

Для добавления файла таблиц стилей можно использовать следующий код:

Header.LinkedStyleSheets.Add("MyStyles.css");

И, наконец, свойство *Title* элемента *HtmlHead* позволяет получить и установить заголовок страницы:

Header.Title = "This is the title";

Учтите, что это свойство возвращает правильный заголовок страницы только в том случае, если тэг $\langle title \rangle$ корректно размещен в пределах тэга $\langle head \rangle$. Некоторые браузеры слишком снисходительны в этом вопросе и позволяют разработчикам определять этот тэг вне заголовка.

Возврат формы для другой страницы

Для реализации такого возврата формы нужно выполнить всего пару шагов. Во-первых, необходимо указать свойство *PostBackUrl* для тех кнопок и серверных элементов, которые могут вызвать этот возврат. Если свойство *PostBackUrl* установлено, исполняющая среда ASP.NET связывает соответствующий HTML-элемент с новой функцией JavaScript. Вместо старой знакомой *__doPostback*, она использует новую функцию *WebForm_DoPostBackWithOptions*:

<form runat="server">

```
<asp:textbox runat="server" id="Data" />
<asp:button runat="server" id="buttonPost"
Text="Click"
PostBackUrl="~/target.aspx" />
</form>
Объявление кнопки визуализируется в следующую разметку:
<input type="submit" name="buttonPost" id="buttonPost"
value="Click"
onclick="javascript:WebForm_DoPostBackWithOptions(
new WebForm_PostBackOptions("buttonPost",
```

"", false, "", "target.aspx", false, false,))" />

В результате, когда пользователь щелкает кнопку, текущая форма отсылает свое содержимое указанной странице-адресату. А что происходит с состоянием отображения? Если страница содержит элемент, выполняющий возврат формы для другой страницы, создается новое скрытое поле — *PREVIOUSPAGE*. В нем находится информация о состоянии отображения, применяемом для обслуживания запроса. Эта информация используется вместо первоначального состояния отображения страницы-адресата.

Для обращения к странице-отправителю и всем ее элементам нужно использовать свойство *PreviousPage*. Приведем пример страницы, способной обращаться к содержимому текстового поля, определенного в только что показанной серверной форме:

```
<% @ page language="C#" %>
<script runat="server">
void Page_Load(object sender, EventArgs e)
{
    if (PreviousPage == null)
    {
        Response.Write("Invoke me only through cross-
        page posting.");
        Response.End();
        return;
    }
```

```
// Извлекаем данные из текстового поля.

TextBox txt = (TextBox) PreviousPage.FindControl("Data");

Response.Write("<b>You passed:</b> " + txt.Text);

}

</script>

<html>

<head runat="server">

<title>Target page</title>

</head>

<body>

</body>

</html>
```

Разбиение страницы на части

Если вам приходилось работать с мобильными элементами управления ASP.NET, вам должно быть известно о разбиении вывода на части. Разбиение страницы на части — это возможность автоматического вывода содержимого формы по частям, в соответствии с характеристиками целевого мобильного устройства. Каждая из этих частей получается путем разбиения разметки целой формы на блоки примерно одинакового размера.

В ASP.NET 2.0 появился новый элемент управления страницами под названием *Pager*, который добавляет похожую функциональность всем серверным управляющим элементам, в том числе и мобильным. Этот элемент позволяет разбивать на части вывод всех контейнерных управляющих элементов, основываясь на их «весе» или числе дочерних элементов результирующей страницы. *Pager* также имеет свой собственный пользовательский интерфейс, позволяющий перемещаться по страницам.

Далее приведен исходный код страницы:

```
<% @ Page Language="C#" %>
```

```
<html>
<script runat="server">
void Page_Load (Object sender, EventArgs e)
{
for (int i=0; i<50; i++) {
TextBox txt = new TextBox ();
txt.Text = i.ToString();
panelSample.Controls.Add (txt);
panelSample.Controls.Add (new LiteralControl("<br>));
```

```
}
</script>
</script>
</body>
<form runat="server">
<h1>Pager in action</h1>
<asp:contentpager runat="server" id="pagerPanel"
Mode="NextPreviousFirstLast"
ControlToPaginate="panelSample"
ItemsPerPage="9" />
<asp:panel runat="server" id="panelSample" borderwidth="1" />
</form>
</body>
</html>
```

Элемент ContentPager разбивает содержимое контейнера, основываясь на одном из двух алгоритмов — в зависимости от «веса» или количества элементов. Выбор алгоритма определяется значением свойства ItemsPerPage. Если это свойство задано, то используется алгоритм вывода на основе количества элементов, в противном случае используется весовой алгоритм. Оба алгоритма работают сходно, однако весовой алгоритм предоставляет более точный контроль за выводом элементов на каждой странице. В сущности, алгоритм, основанный на количестве элементов, пытается поместить на страницу ровно столько элементов, сколько задано в свойстве *ItemsPerPage*. Алгоритм на основе веса присваивает каждому контейнеру максимальный вес, который при разбивке на страницы не должен быть превышен. Например, максимальный вес класса *Page* равен 4000. Каждому элементу управления также присваивается вес по умолчанию (100). Как видите, поведение обоих алгоритмов по умолчанию практически одинаково, но вес каждого элемента может быть переопределен, что позволит изменить стандартную форму разбивки.

Объектная модель сценариев страницы

Отличная новость, касающаяся объектной модели сценариев в ASP.NET 2.0, состоит в том, что теперь они могут вызывать серверные события из клиентского кода, не отправляя данные страницы на сервер и не обновляя изображение. Такая разновидность удаленных сценариев реализована посредством механизма обратных вызовов, что дает разработчикам значительные преимущества. При использовании обратных вызовов в сценариях, результаты выполнения методов на серверной стороне передаются напрямую функции JavaScript, после чего она может обновить пользовательский интерфейс с помощью Dynamic HTML. Обращение к серверу происходит, но страница обновляется не полностью.

Сценарии с обратными вызовами — не единственная хорошая новость. Модель сценариев значительно улучшена. В результате, установка фокуса ввода на определенный элемент управления теперь сводится к вызову метода *SetFocus*.

Сценарии с обратными вызовами

Использование обратных вызовов в сценариях позволяет выполнять внеплановые обращения к серверу. Эти обращения — специальный вид возврата формы, так что полный цикл обмена данными все равно всегда присутствует, но, в отличие от классического возврата формы, обратные вызовы в сценариях не перерисовывают всю страницу и поддерживают у пользователя иллюзию, что все выполняется на стороне клиента. Обратные вызовы можно применять для обновления отдельных элементов страницы, например диаграмм или панелей, реализуя различные представления одних и тех же данных. Вы также можете загружать дополнительную информацию по требованию, автоматически заполнять одно или несколько полей и т. п. В частности, в ASP.NET 2.0 элемент управления TreeView интенсивно использует обратные вызовы для реализации разворачивания и сворачивания ветвей. А в элементе GridView (см. главу 7) обратные вызовы задействованы в функциях разбиения на страницы и сортировки без использования явных возвратов формы.

Обратный вызов в сценарии начинается с события на клиентской стороне (обычно щелчка мыши), которое запускает встроенную функцию JavaScript под названием *WebForm_DoCallback*. Если для срабатывания этой функции необходимо нажимать на кнопку, не используйте тэг *<asp:button>*. Этот тэг преобразуется в кнопку отправки, что в любом случае приведет к обновлению всей страницы. Используйте кнопку-ссылку или клиентскую кнопку, к примеру, тэг HTML 4.0 *<button>*. При вызове функции *WebForm_DoCallback* необходимо указать несколько аргументов, как показано ниже:

WebForm_DoCallback(

раgeID, // ID страницы, выполняющей вызов argument, // строковый аргумент для передачи // серверному коду returnCallback, // код JavaScript, инициируемый // по завершении обратного вызова context, // данные, передаваемые вызову returnCallback

errorCallback); //код JavaScript, вызываемый при появлении ошибок

Первый аргумент содержит ID страницы и является обязательным. Когда происходит возврат из серверного кода, инфраструктура ASP.NET 2.0 вызывает функцию, указанную как returnCallback, и передает ей возвращенное сервером значение (всегда строковое) и любую контекстную информацию. Функция returnCallback несет ответственность за обработку возвращенного значения и обновление пользовательского интерфейса с помощью Dynamic HTML.

Рассмотрим пример. Страница, поддерживающая внеплановые обреализовывать серверу, должна интерфейс ращения К ICallbackEventHandler. Для этого необходимо директиву указать @Implements:

<% @ Implements Interface="System.Web.UI.ICallbackEventHandler " %>

Интерфейс ICallbackEventHandler содержит только один метод — *RaiseCallbackEvent*:

public virtual string RaiseCallbackEvent(string eventArgument) {

// eventArgument содержит переданный(е) параметр(ы).

// TODO: Извлечь из параметра значения, переданные сервером,

и запаковать все в возвращаемую строку.

Следующая страница демонстрирует пример применения обратных вызовов в сценариях от начала до конца. На ней показан раскрывающийся список с именами работников, взятыми из базы данных Northwind. По кнопке More Info запускается код JavaScript, передающий серверу ID выбранного в списке работника:

<% @ page language="C#" %>

<% @ import namespace="System.Data" %>

<% @ implements interface="System.Web.UI.ICallbackEventHandler"

%>

}

<script language="javascript">

function UpdateEmployeeViewHandler(result, context) {

// Результат представлен в виде строки значений,

// разделенных запятыми.

var obj = result.split(",");

 $e_{ID.innerHTML} = obj[0];$

e_FName.innerHTML = obj[1];

e_LName.innerHTML = obj[2];

```
e_Title.innerHTML = obj[3];
    e_Country.innerHTML = obj[4];
    e_Notes.innerHTML = obj[5];
  }
</script>
<script runat="server">
  public virtual string RaiseCallbackEvent(string eventArgument) {
    // Получаем дополнительную информацию
    // об указанном работнике.
    int empID = Convert.ToInt32 (eventArgument);
    EmployeesManager empMan = new EmployeesManager();
    EmployeeInfo emp = empMan.GetEmployeeDetails(empID);
    // Упаковываем данные в строку.
    string[] buf = new string[6];
    buf[0] = emp.ID.ToString();
    buf[1] = emp.FirstName;
    buf[2] = emp.LastName;
    buf[3] = emp.Title;
    buf[4] = emp.Country;
    buf[5] = emp.Notes;
    return String.Join(",", buf);
  }
  void Page_Load (Object sender, EventArgs e) {
    // Заполняем раскрывающийся список.
    EmployeesManager empMan = new EmployeesManager();
    DataTable dt = empMan.GetListOfNames();
    cboEmployees.DataSource = dt;
    cboEmployees.DataTextField = "lastname";
    cboEmployees.DataValueField = "employeeid";
    cboEmployees.DataBind();
    // Подготавливаем вызов функции Javascript.
    string callbackRef = GetCallbackEventReference(this,
         "document.all['cboEmployees'].value",
         "UpdateEmployeeViewHandler", "null", "null");
    // Привязываем функцию обратного вызова
    // к клиентской кнопке.
    buttonTrigger.Attributes["onclick"] =
         String.Format("javascript:{0}", callbackRef);
</script>
```

| <html></html> |
|---|
| <hr/> |
| <pre><body></body></pre> |
| <10rm runal= server > |
| <asp:dropdownlist id="cboEmployees" runat="server"></asp:dropdownlist> |
|
button runat="server" id="buttonTrigger">More Info |
| |
| |
| <tb>ID</tb> |
| <td< td=""></td<> |
| /> |
| Last Namespan id="e_LName" /> |
| |
| <td< td=""></td<> |
| |
| <pre>Country</pre> |
| <pre></pre> |
| $$ |
| $<$ tr> Notes <1> 1 |
| |
| |
| |
| |
| |

Страница отправляет вызов на сторону сервера, где начинается обычный процесс ее инициализации. Генерируются события *Init и Load*, а свойство *IsCallback* устанавливается в *true*. Вызов *RaiseCallbackEvent* происходит после окончания фазы загрузки. После этого запрос завершается, не переходя в фазу рендеринга. Специальный системный компонент — диспетчер обратных вызовов, — создаваемый на клиентской стороне, управляет процессом взаимодействия между страницей и сервером. Диспетчер обратных вызовов — это библиотека сценариев, передаваемая клиенту ASP.NET. Он отвечает за создание запросов к серверу и генерирует соответствующее событие на серверной стороне. Он также разбирает получаемый ответ и передает данные в указанную функцию JavaScript, вызываемую после завершения обратного вызова.

12. ПРАКТИЧЕСКАЯ РАБОТА 5

Название: Сравнительный анализ и Оценка эффективности применения технологий web2.0 в учебном процессе

Теоретический материал

Разработка и эксплуатация информационных образовательных ресурсов неразрывно связана с оценкой эффективности их применения в образовательном процессе. Образовательная среда как педагогическая система, в которой реализуется учебно-воспитательный процесс, в общем случае состоит из таких элементов, как цели образования, содержание образования, преподаватель, студенты, технологическая подсистема. Сетевые информационные образовательные ресурсы (далее - сетевой ресурс) - это дидактический, программный и технический комплекс, предназначенный для обучения с преимущественным использованием среды Интернет/Интранет независимо от расположения обучающих и обучающихся в пространстве и во времени. Обучение с помощью сетевых ресурсов может рассматриваться как целенаправленный, организованный процесс взаимодействия обучающихся (студентов) с обучающими (преподавателями), между собой и со средствами обучения, причем он некритичен к их расположению в пространстве и во времени.

Сетевой ресурс может использоваться в учебном процессе в различных пропорциях в очной, заочной, вечерней формах получения образования. Таким образом, сетевой ресурс - это учебно-методический интерактивный комплекс, использование которого позволяет реализовать полный дидактический цикл обучения по дисциплине учебного плана.

Дидактические свойства сетевого ресурса в процессе обучения позволяют реализовать:

- представление на экранах мониторов персональных компьютеров преподавателей и студентов учебно-методической информации, а также возможность получения твердых копий целенаправленно выбираемой части информации, содержащейся в сетевом ресурсе;
- диалоговый обмен между участниками образовательного процесса в реальном (on-line) и отложенном (off-line) режиме учебной, методической, научно-образовательной и функциональнопроизводственной информацией и другой, отражающей учебнометодическую и административную информацию (в текстовом, речевом, графическом и мультимедийном вариантах) независимо от

расположения участников образовательного процесса в пространстве и во времени;

- обработка передаваемой и получаемой информации (хранение, распечатка, воспроизведение, редактирование) в реальном и отложенном времени;
- доступ к различным источникам информации (порталам, электронным библиотекам, базам данных, ресурсам Интернет т.п.);
- доступ к удаленным вычислительным ресурсам, лабораторным практикумам, учебным курсам и контролирующим материалам;
- организация коллективных форм общения преподавателя со студентами и студентов между собой посредством теле и видеоконференций;
- обмен определенной заранее заданной части информации в конфиденциальной форме и регламентированный доступ;
- техническая консультационная поддержка функционирования сетевого ресурса.

С целью разработки методов и алгоритмов использования сетевых ресурсов в реальном учебном процессе, а также обеспечения подготовки преподавателей для самостоятельной разработки учебнометодических материалов и организации внедрения этих разработок в учебный процесс, необходимо наличие:

- электронных учебно-методических комплексов по каждой учебной дисциплине;
- документооборота;
- преподавателей, специально подготовленных для работы в новой образовательной среде;
- электронных форм проверки знаний обучающихся.

Конечной целью становится создание технологической платформы, которая позволит в полной мере использовать ресурс в реальном учебном процессе вуза.

Основными элементами (блоками) сетевого ресурса являются:

- 1. инструктивный блок: учебные планы и программы, учебные и учебно-методические материалы;
- 2. информационный блок: научные и официальные материалы, курсы лекций, учебная, научная и специальная литература, иллюстративный материал, законы и нормативные акты, ссылки на библиотеки, периодические электронные издания;
- 3. коммуникативный блок; и контрольный блок: студенческие работы, диссертации (авторефераты, аннотации), образовательные и научные центры, государственные и муниципальные органы власти,

научные, научно-практические конференции, семинары и симпозиумы, ссылки на форумы, интернет - конференции, чаты, профессиональные союзы, ассоциации, общества, структура и наполнение блоков сетевого ресурса.

Для создания сетевых ресурсов с помощью готовых инструментальных средств разработки и использования сетевых курсов преподавателю желательно подготовить учебный материал форме, описанной ниже.

1. Структура и содержание инструктивного блока.

Объем блока около 10-15 стр.

Содержание блока включает в себя:

1. Учебную программу;

2. Руководство по изучению дисциплины.

2. Информационный блок (Е-лекция)

Объем блока составляет 15-20 стр. для социально-гуманитарных дисциплин. Объем информации, которое включается в блок, равно объему содержания традиционной лекции, а количество блоков равно количеству, указанному в тематическом плане.

Структура содержания каждого блока включает в себя:

2.1 введение к Е-лекции (перечислить, что будет знать и уметь студент после ее изучения).

2.2 учебную информацию, которая представляется в рукописном варианте в виде текста, если необходимо с рисунками, графиками, таблицами (желательно напечатанной в редакторе WinWord).

В случае психолого-педагогической необходимости по желанию автора могут быть включены:

- гиперссылки (ссылки отмечаются номерами в тексте, а содержание их печатается отдельно, при ссылках на ресурсы Интернет указывается только адрес сайта);
- аудио, видео фрагменты (изготовление и сценарии их использования согласовываются отдельно);
- готовые компьютерные обучающие программы по тематике дисциплины, встраиваемые в блок (сценарии их применения согласовываются отдельно);

2.3 выводы.

3. Коммуникативный блок (Е-семинар)

Количество блоков равно числу семинаров, указанному в тематическом плане, а его объем составляет ориентировочно 1-2 стр.

Каждый блок имеет следующую структуру.

3.1 тема семинара;

3.2 учебные вопросы;

3.3 литература;

4. Контрольный блок (Е-экзамен)

Объем 4-5 стр. Содержание блока включает в себя:

4.1 Количество и содержание тестов идентично тестам, подготавливаемым для тестов ЕГЭ;

4.2 Вопросы к экзамену не менее 30. Половина вопросов может состоять из типовых задач.

Методические рекомендации по работе преподавателей и студентов с учебными сетевыми ресурсами

1. Для преподавателей.

Работа преподавателя с учебными сетевыми ресурсами портала должна начинаться с анализа результатов работы по изучению курсов студентами.

С этой целью:

обращается к блоку "сведения о студентах", а также к блоку " контроля за качеством изучения студентов учебного материала";

определяет уровень усвоения каждым студентом содержания учебного материала из разделов и тем информационного блока;

разрабатывает для каждого студента частные задания на подготовку к работе с коммуникативным блоком ;

определяет время на выполнение заданий студентами и начало их работы с коммуникативным блоком;

в зависимости от уровня подготовленности студентов к работе с коммуникативным блоком преподаватель выбирает методику проведения практического занятия: семинара, игры, тренинга и т.д.;

в ходе работы с коммуникативным блоком преподаватель должен постоянно проводить сравнение целей (цели изучения формулируются преподавателем в инструктивном блоке) и результатов изучения студентами учебного материала;

2. Для студентов.

Работа студента с учебным сетевым курсом должна начинаться с обращения к инструктивному блоку в сетевом курсе. В ходе работы студент обязан:

а. Методика работы студента с инструктивным блоком:

изучить учебную программу и понять требования руководства по изучению содержания учебной дисциплины;

отметить и вывесить на доску объявлений для преподавателя неясные вопросы, получить на них ответы;

б. Методика работы студента с информационным блоком:

изучить и понять содержание информационного блока;

пройти тестирование по всем запланированным преподавателем теоретическим темам;

подготовиться к работе с коммуникативным блоком, выполнив частные задания преподавателя;

дать заявку преподавателю на участие в практическом занятии(семинаре, игре, тренинге) в реальном или отсроченном времени;

в. Методика работы студента с коммуникативным блоком:

принять участие во всех практических занятиях, запланированных преподавателем, и получить положительную оценку за результаты своей работы;

г. Методика работы студента с контрольным блоком:

пройти контрольное тестирование за весь курс и получить положительную оценку.

Критерии структуризации информационных ресурсов

По виду функционирования ресурса можно выделить: динамический ресурс, содержание которого периодически обновляется и статический ресурс, содержание которого не обновляется.

По профилям обучения можно выделить: ресурс, содержащий информацию гуманитарно-филологического направления; ресурс, содержащий информацию социально-экономического направления; ресурс, содержащий информацию химико-биологического направления; ресурс, содержащий информацию технико-технического направления; ресурс, содержащий информацию физико-математического направления и др.

По доступу к ресурсу можно выделить: сайты, предоставляющие открытый доступ к ресурсу (не требует регистрации); сайты, предоставляющие доступ к ресурсу через регистрацию; сайты, предоставляющие частичный доступ к ресурсу (отдельные компоненты ресурса предоставляются пользователю без регистрации на сайте, а другие компоненты – после регистрации в конкретном разделе сайта).

По типам образовательных web-сайтов можно выделить:

- ресурс, доступный на сайтах дистанционного обучения. К такому типу относятся сайты, содержащие электронные курсы лекций, лабораторные практикумы, электронные задачники, электронные учебники; сайты, предназначенные для тестирования, оценки знаний; сайты центров дистанционного обучения и центров тестирования.
- ресурс, доступный на сайтах исследовательской деятельности. К такому типу относятся сайты, содержащие исследовательские работы студентов, студентов и др.; научные лаборатории; творческие мастерские; сайты научно-исследовательских и учебных центров.

- 3. ресурс, доступный на сайтах консультативного назначения. К такому типу относятся консультативные сайты по общеобразовательным предметам; сайты для консультационной помощи начнометодическим центрам.
- 4. ресурс, доступный на сайтах "виртуальных" методических объединений нений. К такому типу относятся сайты методических объединений преподавателей по дисциплинам дистанционного обучения; сайты, предназначенные для тематических телеконференций и чатов по вопросам образования; сайты, предназначенные для творческого взаимодействия преподавателей, преподавателей в сети; сайты, предназначенные для повышения квалификации педагогических кадров.
- 5. Ресурс, доступный на сайтах образовательных Internet-проектов. К такому типу относятся сайты, проводящие телекоммуникационные олимпиады и викторины; предназначенные для проведения образовательных конкурсов; сайты, осуществляющие информационноразвлекательные проекты образовательной тематики; сайты, предназначенные для формирования связей между участниками образовательного процесса.
- 6. ресурс, доступный на сайтах учебных заведений. К такому типу относятся: сайты вузов, сайты факультетов вузов, сайты высших учебных заведений.
- ресурс, доступный на сайтах культурной и образовательной информации. К такому типу относятся: виртуальные библиотеки, сайты библиотек, виртуальные журналы и газеты образовательной тематики, сайты образовательных газет и журналов, виртуальные музеи, сайты музеев, виртуальные клубы (кафе).
- ресурс, доступный на сайтах справочного характера категории "Образование". К такому типу относятся: электронные энциклопедии, сайты-словари, электронные справочники, сайты-каталоги, базы данных, сайты, содержащие справочную информацию о проводимых конференциях и т.п.

Определим учебное информационное взаимодействие (УИВ), реализованное на базе ресурсов порталов как информационное взаимодействие, направленное на обеспечение деятельности студентов по:

- 1. освоению, закреплению и применению знаний, умений и навыков в условиях осуществления информационной коммуникации;
- 2. самостимулированию к поиску учебной информации;
- 3. решению учебных задач на базе использования информационного ресурса Интернет;
- 4. самооценке учебных достижений;

5. осознанию социальной значимости культурных ценностей и человеческого опыта, процессов и явлений, рассматриваемых в условиях использования информационного ресурса Интернет.

2. Дидактические цели использования информационных ресурсов портала

Дидактические цели, преимущества и основные методы использования сетевых информационных образовательных ресурсов для участников образовательного процесса: преподавателей, администраторов, менеджеров, абитуриентов, обучаемых(учащихся или студентов), родителей.

Целями использования информационных ресурсов портала для *преподавателей* являются:

• совершенствование традиционной методики обучения;

• использование инновационных форм и методов обучения;

• осуществление культурно-просветительской деятельности.

Целью использования информационных ресурсов портала для *администраторов, менеджеров* является поиск нормативной и методической документации.

Целями использования информационных ресурсов портала для абитуриентов, обучаемых (учащихся и студентов) являются:

• подготовка к занятию;

• подготовка в ВУЗ;

• подготовка к внеучебным мероприятиям;

• использование ресурса для выполнения учебных заданий на занятии;

• досуговая деятельность.

Целями использования информационных ресурсов портала для *родителей* являются:

• подготовка к внеучебным мероприятиям;

• поиск информации по подготовке в ВУЗ;

• досуговая деятельность.

Блок тестирования. С его помощью может быть реализована функциональность широкой системы компьютерных тестов, используемых в локальных программных продуктах, на базе интернет-технологий. Могут быть реализованы четыре базовых класса тестовых заданий:

• с выбором ответа из предложенных вариантов - закрытый тип;

• с вводом произвольного фрагмента текста - открытый тип.

• с установлением взаимосвязей между предложенными двумя группами объектов;

• с классификацией предложенного набора объектов по нескольким предложенным категориям;

• с установлением логической последовательности.

Необходимо предусмотреть интерфейсы связи данного блока с базами данных студентов, преподавателей, и с блоком электронного деканата

База данных студентов. Эта подсистема необходима для полноценного функционирования технологической платформы дистанционного и открытого образования.

Основные требования:

• обеспечение хранения, ввода, редактирования, вывода информации об студентов;

• поддержка набора данных об студентов;

• поддержка данных ролевой авторизации студентов в общей системе;

• наличие внутренних открытых интерфейсов к модулям электронного деканата и к модулям информационно-аналитических сервисов;

• возможность расширения набора данных об учащемся (преемственность снизу вверх).

База данных преподавателей.

Основные требования:

• обеспечение хранения, ввода, редактирования, вывода информации о преподавателях;

• поддержка набора данных о преподавателях;

• поддержка данных ролевой авторизации преподавателей в общей системе;

• наличие внутренних открытых интерфейсов к модулям электронного деканата и к модулям информационно-аналитических сервисов;

• возможность расширения набора данных о преподавателях.

Электронный деканат.

Данная система, должна являться центральным блоком всей технологической платформы системы дистанционного обучения. Она является наиболее технологически сложным, ответственным, и наукоемким блоком любой технологической платформы образования в целом.

Первыми блоками данной системы являются следующие:

• блок зачисления по результатам экзаменов (как средство пополнения базы данных студентов);

• информационный блок по направлениям и специальностям (дисциплины, сроки обучения, учебные планы);

• информационный блок по преподавателям (как средство интеграции с базой данных преподавателей, а также с преподаваемыми дисциплинами); • блок аттестации, включающий интегрирующий модуль с блоком тестирования, а также с полученными off-line результатами контрольных работ, зачетов и экзаменов);

• интерфейсы к блоку информационно-аналитических сервисов.

Информационно-аналитические сервисы.

Блок информационно-аналитических сервисов предназначен для формирования информационных документов для руководителей и участников учебного процесса. Данный блок должен быть реализован максимально гибко в отношении возможности дополнения новыми сервисами и формами отчетности, методами обработки информации. Он должен взаимодействовать с блоком функций электронного деканата посредством внутреннего точно специфицированного и свободно расширяемого открытого интерфейса.

Поиск информации.

Данный отдельный модуль должен представлять программное средство, которое может быть встроено, в частности, как в общую интернет-систему. Он должен обеспечивать полнотекстовый поиск по:

• текстам учебных материалов, размещенных на портале;

• по глоссариям используемых терминов, причем как по левым частям (терминам), так и по правым (текстам глоссарных определений);

• по авторам материалов.

Методы осуществления учебного информационного

взаимодействия на базе информационных ресурсов портала

В разделе рассмотрены психолого-педагогические и организационно-методические условия осуществления учебного информационного взаимодействия (УИВ); методы использования информационных ресурсов и осуществления УИВ в зависимости от типа учебного проекта.

Условия осуществления учебного информационного взаимодействия на базе информационных ресурсов порталов

Так как информационное взаимодействие с использованием ресурса портала носит образовательный характер, целесообразно определить следующий спектр условий его осуществления: психологопедагогических, организационно-методических, эргономических, технологических и технических.

В настоящих методических рекомендациях представлены психолого-педагогические и организационно-методические условия осуществления УИВ на базе информационных ресурсов портала. Знание этих условий необходимо преподавателям, организаторам информатизации образования, для решения задач, связанных: с функционированием "виртуальных" открытых образовательных систем телекоммуникационного доступа на базе распределенного информационного ресурса, обеспечивающих социальную адаптацию к жизнедеятельности в информационном обществе; с обеспечением педагогически целесообразного использования потенциала ресурса и организацией УИВ на базе локальных и глобальной сетей.

В качестве психолого-педагогических условий осуществления УИВ на базе ресурса портала можно выделить: использование педагогически значимой учебной информации, доступной из ресурса; взаимодействие с ресурсом в рамках учебного телекоммуникационного проекта; использование телеконференций образовательного назначения для осуществления УИВ в Интернет. Опишем эти условия.

Использование педагогически значимой учебной информации, доступной из ресурса портала.

В условиях информатизации образования значимость учебной литературы приходится все время пересматривать. Коммерциализация Интернет способствует появлению на рынке печатной продукции, литературы низкого качества, зачастую ненаучной. Выходит большое количество "учебных" печатных изданий, не имеющих грифа Министерства образования Российской Федерации, достаточно высокой стоимости. Это положение ставит преподавателя перед проблемой выбора учебной литературы. Проблема решается (хотя и частично), если использовать в учебном процессе педагогически значимую учебную информацию, доступную из ресурса портала.

Для этого, преподавателю предварительно необходимо решить задачу подбора нужной учебной информации из ресурса портала и методики ее использования обучаемым. Далее необходимо остановиться на вопросе использования информации, доступной из ресурса, и способах ее применения преподавателем. Преподаватель может использовать ресурс портала при подготовке следующих материалов: конспектов лекций, лабораторных и практических работ, тестовых и контрольных заданий, сценариев внеклассных мероприятий и т.п.

Выделим характерные особенности осуществления УИВ на базе ресурса портала:

• доступность (доступ к ресурсу возможен с любого ученического компьютера, имеющего выход в Интернет, что делает этот метод доступным для любого преподавателя, работающего с таким обучаемым);

• открытость (гипертекстовая система подачи информации позволяет обучаемому находить собственную "траекторию" изучения учебного материала, углублять и расширять знания по своему усмотрению и возможностям);

• визуализация учебного материала (с помощью средств технологии мультимедиа, использующихся в WWW-сервисе Интернет, можно при

необходимости встраивать в учебный материал иллюстрации, анимацию, видеоролики, озвучивать информацию, встраивать элементы теста);

• интерактивность ресурса (интерактивные формы информационного взаимодействия, размещенные на Web-сайтах портала, позволяют студенту выбирать условия поиска, отбора и загрузки учебной информации).

Взаимодействие с ресурсом портала в рамках учебного телекоммуникационного проекта.

Интернет является "информационно-коммуникационной средой" (Роберт И.В.) УИВ студента с ресурсом, с другим студенту, с ученическим коллективом, а также с преподавателями. Это является самым существенным при использовании технологии Интернет в учебных целях. Следовательно, использование Интернет в образовательных целях должно предусматривать не только формирование специальных знаний, умений и навыков работы с прикладными программами, владение приемами Web-дизайна, быть направлено на творческое развитие личности, но и способствовать формированию умений анализировать собственную интеллектуальную деятельность, то есть обучаться рефлексии через УИВ с ресурсом портала.

Определим особенности использования ресурса и осуществления УИВ в зависимости от типа проекта, основываясь на типизации проектов по доминирующему методу или по доминирующей в проекте деятельности.

Информационные проекты предполагают осуществление деятельности по сбору информации, представленной в ресурс, по узкой тематике, её анализу, дальнейшей систематизации и представлению в виде аннотации к найденному ресурсу и, ссылки на этот ресурс. Эти аннотации могут быть использованы студентами при самостоятельном изучении отдельных тем или расширении знаний в соответствующих темах. Следовательно, аннотации к ресурсу должны быть точными и достаточно подробными.

Практико-ориентированные проекты подразумевают практический результат проектной деятельности, четко ориентированный на социальные интересы. В ходе проекта должны быть получены не только значимые результаты, но и выявлены способы внедрения их в практику. В задачи такого проекта может входить инсталляция необходимого клиентского программного обеспечения, полученного путем поиска на портале, выявление адресов необходимых серверов, оценка качества ресурса этих серверов, разработка практических рекомендаций к использованию этих ресурсов в образовательных целях и т.п. Таким образом, в результате проекта, должно быть создано либо практическое пособие, либо необходимые рекомендации.

Исследовательские проекты предназначены для стимулирования познавательной деятельности студентов и их самостоятельной работы по сбору, обработке, анализу полученных результатов и требуют четко обозначенных целей, актуальности проекта, его социальной значимости.

Выполнение проектного задания может способствовать осознанию учащимися важности осуществления УИВ в телекоммуникационной среде; формированию системы базовых знаний и навыков и дальнейшему их пополнению и развитию; выработке устойчивой мотивации и ощущения потребности в приобретении новых знаний, необходимых в работе над проектом; активизации познавательной деятельности студентов, особенно при выполнении ими проектно-коммуникационных исследований; развитию творческих способностей, позволяющих реализовать проектную задачу в соответствии с собственным видением; воспитанию инициативности в получении новых знаний и самостоятельности в расширении сфер их применения; осознанию учащимися себя творцами собственных знаний.

Использование телеконференций образовательного назначения (ТКОН) для осуществления УИВ в Интернет.

В основе использования образовательных телеконференций лежат различные коммуникационные технологии: списки рассылки, системы электронной почты (electronic mail), электронные доски объявлений (BBS-Bulletin Board System), on-line телеконференции на основе технологии IRS (Интернет Relay Chat), так называемых "чатов".

Современные технологии Интернет позволяют основывать телеконференции как на письменной или текстовой коммуникации в синхронном или асинхронном режиме, так и на устном общении между участниками.

Опишем существующие типы ТКОН. Все существующие ТКОН в сети можно условно разделить на три группы, в зависимости от: уровня доступа к информации телеконференции; управления телеконференцией; способа обмена информацией в телеконференции.

В первой группе закрытые ТКОН отличаются от открытых тем, что они доступны только для ограниченного числа зарегистрированных пользователей.

Во второй группе модерируемые ТКОН характеризуются наличием ведущего – модератора, отвечающего за организацию и проведение телеконференции. В премодерируемых конференциях сообщения также обрабатываются модератором, но уже после того, как они опубликованы в телеконференции. При этом модератор может удалить либо оставить сообщение или внести комментарий. Немодерируемые ТКОН характеризуются отсутствием ведущего, участники сами в ходе обмена сообщениями направляют развитие дискуссии.

К третьей группе относятся конференции в отсроченном режиме и в режиме реального времени. Например, в Интернет конференции в отсроченном режиме организуются в виде дискуссионных групп (Newsgroups), списков рассылки (Maillists), электронной почты (E-mail) и т. п. Здесь общение между участниками происходит за счет обмена сообщениями по электронной почте. Конференции в режиме реального времени включают в себя IRC (Internet Relay Chat), ICQ (I seek you), разговорные серверы с WWW-интерфейсом, видео телеконференции. В этом случае общение между участниками осуществляется с помощью специальных программных средств и оборудования.

Наиболее просты с точки зрения организации телеконференции в отсроченном режиме. Однако, если количество участников невелико, и они не находятся в удаленных часовых поясах, имеет смысл проводить ТКОН в режиме реального времени или видео телеконференции.

Конференции в отсроченном режиме, проводимые с помощью электронной почты с использованием списка рассылки, имеют свои преимущества. Данный вид конференции более предпочтителен для организации образовательных телекоммуникаций среди студентов, так как большинство российских вузов не имеют прямого доступа в сеть Интернет. Создание образовательного списка рассылки в разных его вариантах – свободном, модерируемом, имеющем ограничения к доступу и др. – позволяет организовать работу студентов в сети в соответствии с решаемыми образовательными задачами. В зависимости от выбранного варианта работы списка рассылки строится и обучающая структура электронной конференции. Существенное преимущество таких конференций перед беседой в реальном времени – больший период времени, отводимый на обдумывание своих вопросов, ответов и суждений, что позволяет участнику подготовить свой ответ, привести его в соответствие с заданными требованиями. В чат-конференции из-за ограниченного времени трудно избежать малосодержательных реплик. Кроме того, по электронной почте участники могут высылать иллюстративный материал, мультимедийную информацию, которую в обычных чат-программах представить пока невозможно. Наиболее распространенными из почтовых программ являются TheBat, Internet Mail, Outlook Express, Eudora, Exchange Mail, Beauty Mail и др.

Другой разновидностью телеконференций, проводимых в отсроченном режиме времени являются *телеконференции, проводимые на* специальном сервере новостей. На них также надо подписываться. На деле это означает определение пользователем списка адресов телеконференций, которые будут прочитываться при каждом сеансе подключения к сети (загрузке и выгрузке почты). Самой распространенной сетью, организованной по такому принципу, является Usenet – международная сеть телеконференций. Дискуссионные группы в ней называют группами новостей (newsgroups) или телеконференциями. Материалом для телеконференций служат присылаемые пользователями новые статьи и ответы на опубликованные ранее. Для работы с данными телеконференциями в Outlook Express необходимо настроить программу чтения новостей. Недостатком является трудность в выделении главного сообщения и поступивших на него откликов в случае большого количества участников, переписывающихся друг с другом в одной телеконференции по различным темам. В списке Outlook можно читать только главные сообщения или только одно сообщение и все отклики на него.

Интернет - конференции портала являются электронной проекцией обычных конференций. Единственным отличием является широта аудитории, место проведения и технические ограничения.

Конференция проходит в 4 этапа:

1. Определяется тема конференции, ее ведущий и сроки проведения.

2. Объявляется сбор докладов и публикуется пресс-релиз конференции

3. Проводится обсуждение докладов.

4. Конференция закрывается. Публикуется результирующая сводка.

Правила проведения Интернет - конференций на базе портала "Социально-гуманитарное и политологическое образование":

1. Ведущим конференции может стать любое лицо

о Преподаватель;

• Студент;

• Менеджер;

• Исследователь;

• Абитуриент.

1. Ведущий определяет тему конференции в рамках множества тем "Социально-гуманитарное и политологическое образование".

2. Тема, сроки проведения, методики извещения аудитории согласовываются с представителями редакционного совета портала путем электронной переписки, телефонного разговора, либо лично при встрече.

3. Ведущий конференции назначается таковым после утверждений, указанных в п.3. 4. Ведущий получает доступ к он-лайн заполнению информации о конференции и базы докладов после регистрации на Портале, что подразумевает наличие у ведущего личного адреса электронной почты. Если адрес отсутствует - Портал берет на себя обязательство завести адрес для ведущего.

5. Ведущий обязан представить к обсуждению более одного доклада от разных авторов.

6. Конференция автоматически открывается для обсуждения в 00.00 указываемой даты начала и закрывается в 23.59 указанной даты конца конференции.

7. В процессе обсуждения действует цензура соответствия текста общепринятым нормам человеческого общения

8. Итогом конференции является заполняемый ведущим постскриптум

Методы использования различных средств обучения для осуществления УИВ с применением информационных ресурсов портала. Уровни использования аудиовизуальной информации при работе с порталом.

Использование в учебном процессе Web-технологий для осуществления УИВ на базе ресурса. "Web-технология – это технология, интегрирующая и унифицирующая несколько решений задач в области сетевых баз данных, задач на уровне клиента и сервера и мультимедийных задач". Она способствовала интенсивному развитию телекоммуникационных технологий. Одной из важных возможностей Web-технологии является возможность работы на базе Гипертекста и Гипермедиа.

Учитывая эти возможности, можно составлять образовательные курсы с применением аудиовизуальной информации, т.е. ресурсы, основанные на работе в глобальной сети, тем самым кардинальным способом меняя привычное информационное взаимодействие типа "преподаватель - слушатель".

Применение Интранет-технологий для организации и осуществления УИВ. Интранет-технология представляет собой технологию, использующую протоколы Интернет и интерфейс web-технологии не только в локальных сетях, но и на отдельных компьютерах. Webброузеры выполняют новый уровень воспроизведения мультимедиаинформации.

Можно выделить следующие уровни использования аудиовизуальной информации броузерами:

• уровень встроенных возможностей данного конкретного броузера;

• уровень внешних или вспомогательных программ – вспомогательных программ просмотра; • уровень технологии подключаемых модулей, при которой в окне броузера (а не из внешней программы) будут доступны инструменты по воспроизведению звука, анимации и видео, будет возможно производить все необходимые действия, реализовывать возможности меню.

Благодаря первым двум уровням броузер превращается в открытую платформу с широкими возможностями, когда для документа любого типа, получаемого броузером, имеется возможность открыть соответствующую ему программу. Авторские ресурсы позволяет реализовать технология третьего уровня.

Вышеперечисленные возможности броузеров позволяют сделать УИВ более эффективным за счет предоставления возможности работы с мультимедийными приложениями, имеющимися на портале. Участники такого взаимодействия в Интернет получают возможность не только просматривать текст, графику, видеоклипы, получать ответы на свои запросы, просматривать каталоги и т.п., но и создавать авторские ресурсы. То есть эти возможности Интернет/Интранет делают Интернет интерактивным средством УИВ.

Методы организации учебной деятельности, реализуемые на базе информационных ресурсов портала

Под учебной деятельностью, реализующейся в информационнокоммуникационной предметной среде, будем понимать динамическую систему, обеспечивающую условия взаимодействия между обучающимся (обучающимися), обучаемым и средствами информационных и коммуникационных технологий, направленную на достижение учебных целей. Встраиваемость возможностей информационных и коммуникационных технологий в обучающие средства и системы, доступные на портале, моделирующие и имитирующие на экране учебные сюжеты, объекты, процессы, явления, обеспечивает реализацию новых видов учебной деятельности как по форме, так и по методам представления и извлечения знания.

Перечислим эти виды учебной деятельности.

Регистрация, сбор, накопление, хранение, обработка информации об изучаемых объектах, явлениях, процессах, в том числе реально протекающих, и передача достаточно больших объемов информации, представленной в различной форме.

Например, каждый преподаватель, имеющий доступ к порталу может, в свою очередь, формировать распределенный информационный образовательный ресурс, создавая каталоги сайтов по направлениям деятельности, которые его интересуют. С помощью такого ресурса он может создавать собственную электронную библиотеку в виде базы данных либо просто текстового файла. Организовать поиск необходимых документов можно: по ключевому слову, по создателю (алфавитный поиск), по временному периоду, по географическому региону.

Преподаватель может создавать ресурс самостоятельно, вместе со своими учащимися, обмениваться информацией с коллегами. В процессе этой деятельности, которую можно считать информационной деятельностью преподавателя по поиску, сбору, структурированию и продуцированию информации происходит процесс непрерывной переподготовки преподавателя на основе распределенного информационного ресурса портала. Дать возможность преподавателям и студенту совместно работать со своими коллегами как в стране, так и на международном уровне, разделяя новые идеи и инициативы он-лайнового сотрудничества для увеличения (углубления) их преподавательских и ученических возможностей с целью увеличить здоровье и благополучие планеты.

Интерактивный диалог - взаимодействие пользователя с компонентами учебного взаимодействия, в том числе и со средством обучения, функционирующим на базе информационных и (или) коммуникационных технологий, характеризующееся реализацией развитых средств ведения диалога (возможность задавать вопросы в произвольной форме, с использованием так называемого "ключевого" слова или с определенным набором таких "ключевых" слов или символов), при обеспечении возможности выбора вариантов содержания учебного материала или режима работы в соответствии с личными предпочтениями пользователя (педагогов, администраторов, менеджеров, студентов, абитуриентов, родителей).

Управление отображением на экране моделей различных объектов, явлений, процессов как виртуальных, так и реальных.

Автоматизация процессов контроля (в том числе самоконтроля) за результатами учебной деятельности с последующей коррекцией по результатам контроля (в том числе "встраивание" тестирования), автоматизация процессов тренировки учебных умений и навыков.

Портал может содержать ссылки на сайты тестирования, позволяющие реализовывать контроль и самоконтроль учебной деятельности студентов.

<u>Интерактивные тесты</u> при работе с такими сайтами преподавателю следует обращать внимание на следующие возможности, которые должен предоставлять сайт: • каждый преподаватель может зарегистрироваться на сайте и обеспечить доступ к нему своих студентов по соответствующему выбранному им паролю;

• для каждой группы своих студентов он может отобрать необходимый для очередного занятия список тестов, причем этот список может корректироваться по его желанию;

• преподаватель может распечатать материал тестов вместе с решениями;

• преподаватель может корректировать содержание каждого теста и составлять свой собственный тест (в том числе на своем родном языке); при этом у него есть возможность хранить этот тест в своей собственной подбазе либо придать ему статус разделяемого ресурса, и тогда тест автоматически помещается в соответствующий раздел общей базы данных тестов;

• студент имеет доступ к отобранным тестам по паролю, сообщенному ему преподавателем;

• после решения студентом очередной выделенной группы тестов преподаватель получает электронное письмо с подробной статистикой результатов по каждому студенту, тесту и вопросу;

• допускается создание тестов следующих типов: множественный выбор, ДА/НЕТ, установление соответствия, выстраивание объектов в определенной последовательности,

ввод в качестве ответа ключевого слова.

Кроме того, зарегистрированные преподаватели автоматически становятся участниками телеконференции в режиме электронной почты, регулярно информирующей о новых сетевых взаимодействиях, организованных на этом сайте.

Продуцирование информации - деятельность по созданию информационного продукта, отличающегося определенными существенными признаками, характеризующими его качество или принадлежность к определенной сфере использования.

Огромное значение в создании и работе над web-pecypcaми играет роль преподавателя, который может заинтересовать студентов той или иной проблемой и решить ее с помощью портала. Предполагается, что работа с web-pecypcaми должна быть понятной и доступной, отвечающей всем требованиям интерактивности. В качестве web-pecypcoв могут выступать сайты, которые подготавливаются студентами и служат для решения конкретных образовательных задач. Web-pecypcы разрабатываются для максимальной интеграции информации, доступной из ресурсов портала в различные учебные предметы на разных уровнях обучения в учебном процессе. Особенностью ресурсов является то, что материал для работы студентов может находиться на различных сайтах, ссылки на которые размещены в каталоге портала.

Образовательные ресурсы можно разделить по основным направлениям:

- углубленное изучение учебных дисциплин;
- региональные проблемы,
- проектная деятельность.

Одной из разновидностей учебной деятельности по продуцированию информации являются учебные проекты.

Под учебным проектом на базе Интернет-ресурса понимается результат самостоятельной творческой деятельности студентов по созданию и размещению в Интернет образовательного ресурса, организованной на основе использования информационных и коммуникационных технологий Интернет, имеющей социально значимую и образовательную цель осуществления учебного информационного взаимодействия в Интернет.

Так как на портале имеются ссылки на сайты соревновательных и информационных Интернет-проектов, то преподаватель может использовать материал этих ресурсов, учитывая следующие методические рекомендации.

1. Для организации выполнения проекта целесообразно выделить группу технического обеспечения из наиболее подготовленных студентов и закрепить их за участниками соответствующих проектов.

2. Так как проекты обычно выполняются с использованием иностранных языков, где иностранный язык является лишь инструментом общения, а не главной целью, следует создать такую же группу языкового обеспечения, наличие которой будет способствовать расширению круга участников проекта. Кроме того, стремление использовать разнообразные информационные ресурсы, несомненно, будет способствовать усилению роли иностранных языков в образовании.

3. Проекты можно реализовывать как со студентами, так и с учащимися среднего и старшего звена школы.

4. Целями проектов могут быть, например: языковые (практическое овладение языком; изучение языковых особенностей; изучение языковых реалий), в области ИТ (закрепление навыков использования электронной почты; закрепление навыков создания текстовой информации; ознакомление с разными способами оцифровки изображений), культурологические/воспитательные (организация межкультурного общения как формы образования и воспитания; знакомство с семейными традициями другой культуры; формирование глобального мышления будущего члена информационного общества) и т.п. 5. В проектах может быть реализована интеграция различных дисциплин, входящих в систему дистанционного обучения.

6. Если у преподавателя не опыта проведения телекоммуникационных проектов, то можно начинать с проведения проектов по типу "Друзья по переписке". Цели проекта при этом могут быть совершенно различными: от изучения возможностей электронной почты до решения глобальных культурологических проблем. Они могут решать образовательные задачи как для студентов, так и для преподавателей. По характеру обмена письмами они могут быть: взаимно однозначный обмен письмами между студентами двух групп; взаимно однозначный обмен письмами типа "группа – группа"; обмен письмами типа "группа – несколько групп".

7. Преподаватель при подготовке проекта может опираться на следующую классификацию проектов:

по методу исследования (например, творческий);

по содержательному аспекту (например, языковой, обучающий);

по характеру координации

Результаты работы по проекту могут быть оформлены в виде: Web-сайта по материалам переписки на сервере, отчета, презентации, доклада и т.п.

8. Проект может выполняться как в учебное, так и во внеурочное время. На занятиях в течение небольшого времени обсуждается содержательная часть общего характера, записывается необходимый теоретический материал. Студенты дома готовят материал, набирают тексты на компьютере в вузе (при необходимости) в отведенное для этого время, копируют в специально созданные папки переписки и отправляют письма с вложениями нужной информации. Порядок получения писем может быть следующим:

все письма приходят на один фиксированный адрес;

письма принимаются дважды в день и перемещаются в специальную папку;

координатор проекта (преподаватель) – переписывает письма в текстовом формате в специальный (почтовый) раздел файл-сервера;

студенты – координаторы ежедневно просматривают почтовый раздел файл-сервера, печатают тексты писем, удаляют распечатанную информацию из раздела файл-сервера и передают тексты писем соответствующим адресатам; ответы набираются студентами в отведенное после занятий время и отправляются самостоятельно.

Все вышеперечисленные ресурсы предоставляют преподавателю возможности организации исследовательской деятельности студентов, возможности проведения экспериментов, обеспечивая при этом про-

зрачность и дружественность интерфейса, удобство навигации и направлены на усиление мотивационного и развитие содержательного и операционального компонентов деятельности студентов.

Кроме того, можно рассмотреть виды деятельности, которые целесообразно реализовывать, используя образовательный потенциал распределенных информационно-образовательных ресурсов, ссылки на которые имеются на портале "Социально-гуманитарное и политологическое образование", в зависимости от уровня подготовки преподавателя к использованию ресурса (в порядке возрастания квалификации педагога).

Организация деятельности обучаемых по поиску актуальной информации. Методы использования информационных ресурсов портала в процессе консультирования и самостоятельной деятельности обучаемых.

Этот вид деятельности предполагает просмотр образовательных и других ресурсов, двигаясь по темам и разделам сайтов, специальным образом структурированных в виде иерархических каталогов общего назначения, предметного, классифицированным по тематическим разделам портала.

Данный вид деятельности рекомендуется начинающим пользователям, от которых требуется лишь знание адресов и умение пользоваться броузером, чтобы попасть на нужный сайт, а также папкой "Избранное" для использования готовых ссылок и накапливания своих собственных. Преимущество этого вида деятельности заключается в том, что сайты с обобщенной иерархической структурой содержат большое количество ссылок на различные ресурсы, содержащих избыточную для преподавателя информацию. К недостаткам этого вида деятельности можно отнести обилие информации, несущественной с точки зрения конкретных учебных задач и отвлекающих студентов от цели работы. С другой стороны, избыточность распределенного информационного ресурса требует умения отличать главное от второстепенного и воспитывают целеустремленность.

Поисковая деятельность с помощью поисковых машин.

Следующим уровнем поиска (а также уровнем сложности использования) является организация поиска по ключевому слову, выражению, а также с использованием средств расширенного поиска. Здесь от преподавателей и студентов требуется умение составить запрос на поиск, отобрать нужную информацию, количество и качество которой существенно зависит от локализации запроса. Использование средств расширенного поиска, описанных на каждой поисковой машине, позволяет снизить избыточность информации, полученной в результате поиска. Тем не менее, этот метод достаточно сложен, и в процессе занятия можно столкнуться с ситуацией, когда не все студенты справляются с задачей отбора необходимой информации.

Использование потенциала распределенных информационных ресурсов для организации исследовательской деятельности студентов в процессе занятия.

По мере освоения (или использования) потенциала распределенных информационных ресурсов преподаватели могут использовать его непосредственно для исследовательских задач своей предметной области. Включая выполнение студентами исследовательских проектов с помощью ресурсов Интернет как в off-line, так и в online-режиме на занятиях и в качестве домашних заданий, можно адресовать их к определенным ресурсам для проведения исследования, изучения методики проведения эксперимента, просмотра опыта и т.д.

Такие проекты можно достаточно просто и эффективно осуществить, используя возможности мультимедиа, которые предоставляют информационные ресурсы. Собранные в процессе работы данные рекомендуется использовать для пополнения информационных ресурсов портала

Сложность организации данного вида деятельности заключается в том, что использование некоторых описанных ресурсов требует от преподавателя специальной научной подготовки, а также адаптации ресурсов портала для студентов либо создания специального пояснительного комментария.

Проведение совместных научных исследований с другими партнерами на базе распределенных информационных ресурсов портала.

Этот вид деятельности является модификацией предыдущего, поскольку в процессе исследовательской работы студенты с помощью преподавателя устанавливают контакты с партнерами, работающими над той же проблемой, предоставляют партнерам ресурсы своего сайта, работают с ресурсами в виде распределенных баз данных, обмениваются с партнерами способами обработки и интерпретации информации, участвуют в совместных дискуссиях.

Соответственно от преподавателей, кроме работы с ресурсом Интернет предметной области, требуется умение найти партнеров по проекту, спланировать и организовать совместную работу студентов, провести совместную дискуссию, оформить полученные совместные результаты.

Такие ресурсы предоставляют возможность выбора из существующих направлений исследований либо предложить свою идею и найти партнеров, желающих принять участие в ее осуществлении. К достоинствам этого вида деятельности можно отнести синергизм воспитательного воздействия на основе работы с представителями разных культурных особенностей, религиозных убеждений, языков. Студенты могут увидеть сходство и различие своего процесса обучения и процесса обучения своих партнеров, что усиливает взаимопонимание и обеспечивает эффективность совместной деятельности. а также воспитывает чувство принадлежности (причастности) к глобальному информационному обществу.

Недостатком такого рода деятельности является снижение контроля за результатом по сравнению с организацией такого же проекта внутри одной группы. Кроме того, совместная работа осложняется особенностями учебных процессов участников, разницей часовых поясов для организации online-взаимодействий, языковыми различиями и т.д.

Создание собственного распределенного информационного ресурса в виде тематического каталога предметной области либо персонального web-сайта.

По уровню сложности этот вид деятельности не сложнее предыдущего, однако он требует от преподавателя дополнительных знаний и умений создавать такие страницы. Тематические каталоги должны быть структурированы по темам соответствующей предметной области, снабжены кратким комментарием к содержанию и дополнены примерами использования. Поскольку образовательный портал, как определено выше сформулированными требованиями и принципами его создания, является открытым ресурсом, встраивание таких страниц не представляет особого труда.

Дополнительным преимуществом является возможность сократить время поиска студентами необходимых для изучения темы ресурсов и, следовательно, оптимизировать время занятия. Кроме того, как было отмечено выше, это позволяет преподавателю повысить уровень своей квалификации и обеспечить студентов важной для изучения данной темы информацией.

Обучение и помощь в создании студентами своих собственных ресурсов Интернет.

Одной из таких форм деятельности является участие в международных проектах, которые провозглашают своей целью создание ресурсов Интернет интернациональными группами студентов, представляющих национальные особенности образования каждого участника группы. Необходимым условием участия в деятельности этой организации является наличие интернациональной команды студентов и преподавателей-тренеров. Далее команда с помощью средств телекоммуникаций выбирает тему исследования, связанную с углубленным изучением какого-нибудь раздела предметной области (или нескольких областей), проектирует свой ресурс Интернет и выставляет его на сайте или портале. Все этапы создания ресурса, все коммуникации участников осуществляются через Интернет.

Такой вид деятельности позволяет организовать более тесные связи между преподавателями и студентами в процессе обучения, имеющего своей целью создание ресурса по результатам изучения выбранной темы. При этом изменяется роль преподавателя, который выступает в качестве консультанта и организатора исследовательской деятельности студента, кроме того, преподаватели-тренеры обмениваются идеями совершенствования своей стратегической деятельности для достижения целей и получения необходимых результатов. Параллельно с этим студенты, которые в большей степени осваивают технологию проектирования Веб-страниц, в свою очередь становятся консультантами для преподавателей, что повышает их значимость и усиливает чувство собственного достоинства.

Создание ресурсов Интернет требует от участников (в том числе и от преподавателей) больших временных затрат, в том числе, в основном, во внеурочное время, но отдача от работы, как правило, окупает затраченное дополнительное время, поскольку студенты глубоко погружаются в изучаемый предмет, а преподаватели более глубоко осваивают технологию создания Веб-страниц.

Методы экспертной оценки качества и областей применения образовательных информационных ресурсов, входящих в портал.

Еще одним видом информационной деятельности, которая может осуществляться над ресурсами портала, является его экспертная оценка, осуществляемая педагогом с целью выявления педагогической целесообразности их использования в учебном процессе. При осуществлении экспертной оценки психолого-педагогического и программнотехнического качества образовательных ресурсов целесообразно использовать оценочные тесты или оценочные листы, в которых определены направления экспертизы.

Направлениями экспертной оценки качества содержания, структуры и формы представления учебного материала, доступного из ресурсов портала могут выступать основные, характерные для информационного ресурса, содержащегося в учебно-методической литературе и дополнительные, присущие только для информационного ресурса Интернет.

К основным направлениям экспертной оценки качества содержания, структуры и формы представления учебного материала, доступного

из ресурсов портала можно отнести: научность; доступность; наглядность; логика; полнота.

Обеспечение *научности* содержания предполагает глубину и корректность изложения учебного материла, доступного из ресурсов портала, с учетом последних научных достижений. При этом учебный материал должен содержать достоверные сведения и базироваться на методах изучаемой науки.

Обеспечение *доступности* содержания предполагает реализацию необходимой степени теоретической сложности сообразно возрастным и индивидуальным особенностям обучающихся. При этом в отличие от электронных изданий образовательного назначения, ресурс Интернет может содержать усложненный и избыточный материал, дифференцированный по уровням сложности, который может использоваться как педагогом, так и учащимися. Поэтому педагог самостоятельно определяет степень доступности материала.

Обеспечение *наглядности* содержания предполагает реализацию возможностей современных средств визуализации (например, изображение объемных фигур, схем, таблиц, графиков и т.п. средствами компьютерной графики) объектов, процессов, явлений и их моделей. Наглядность способствует повышению интереса обучающихся к представленной в Интернет информации и способствует эффективной интериоризации знаний.

Обеспечение *логики* изложения учебного материала в ресурсах портал предполагает четко определенную последовательность изложения разделов определенной системы знаний в изучаемой предметной области, как правило, выполненных в виде гиперссылок. При этом знания, умения и навыки обучающихся должны формироваться в определенном порядке, устанавливаемом обучающим или самим обучающимся: каждый раздел учебного материала логически связывается с другими и может изучаться в произвольном порядке, что отличает ресурс Интернет от учебников, учебных пособий и т.п.

Обеспечение полноты содержания предполагает наличие всех тем раздела учебного материала изучаемой предметной области.

Перечислим *дополнительные* направления экспертной оценки качества содержания, структуры и формы представления учебного материала, доступного из ресурсов Интернет: достоверность; интерактивность; гипермедийность; наличие различных режимов работы с ресурсами Интернет.

Обеспечение достоверности изложения учебного материала в ресурсах Интернет предполагает авторитетность авторов, которая определяется в Интернет количеством ссылок на них.

Обеспечение интерактивности изложения учебного материала в ресурсах Интернет предполагает реализацию учебного интерактивного диалога обучающегося с ресурсом Интернет.

Экспертная оценка интерактивности ресурсов Интернет предполагает выявление следующих параметров.

1. Наличие диалоговой формы для заполнения обучающимся/обучающим. Диалог с ресурсом Интернет представляет собой варьирование либо последовательности, либо объема выдаваемой информации и реализует обмен сообщениями на специально разработанном языке общения или на естественном языке. При этом взаимодействие между ресурсом Интернет и обучающимся моделирует, как правило, отдельные аспекты реального общения между преподавателем и обучающимся.

При экспертной оценки следует учитывать, что по параметру активности существует несколько видов диалога: ресурс Интернет управляет диалогом и задает пользователю варианты ответов; ресурс Интернет управляет диалогом, но ввод данных свободный; пользователь управляет диалогом, но варианты обращений ему заданы; пользователь управляет диалогом при свободном вводе данных.

Кроме того, эксперту для учеты индивидуальных особенностей обучающихся следует обратить внимание на существующие формы организации диалога в ресурсе Интернет.

Диалог, организованный в форме графических образов (графический диалог), позволяет манипулировать не только значениями, но и произвольно построенными динамическими образами ситуаций и процессов, развивает не только воображение, но и мышление обучающихся, поскольку приучает соотносить знаковые и образные представления ситуаций и формирует обобщения более высокого порядка.

Диалог, организованный в вопросно-ответной форме должен отвечать требованиям естественности, последовательности, краткости, гибкости и поддержки пользователя. Естественность означает, что при взаимодействии с ресурсом Интернет обучающийся не должен существенно изменять имеющиеся традиционные способы решения задач. Стиль ведения диалога должен быть разговорным, а не письменным, фразы, по возможности, не должны требовать дополнительных пояснений, а лексикон соответствовать возрастным особенностям обучающегося. Требование последовательности, предполагает, что после освоения одной части ресурса Интернет не возникает затруднений при работе с другой его частью. Последовательность относится также к единообразному размещению данных на экране, единообразным используемым форматом ввода сообщений. Требование краткости предполагает ввод обучающимся минимального размера сообщений, что обеспечивает более быстрое взаимодействие с ресурсом Интернет и облегчает контроль правильности ответов обучающегося. Поддержка пользователя осуществляется в виде подсказок, справочной информации или обратной связи. Подсказка и справочная информация могут опережать действия обучающегося, обратная связь осуществляется после ввода им сообщения. Требование гибкости диалога подразумевает, насколько хорошо он соответствует различным уровням подготовки обучающегося. Индивидуализация обучения на базе ресурса Интернет зависит от гибкости используемого диалога.

Диалог, организованный в форме "меню" предоставляет обучающемуся в различных формах возможные варианты данных для ввода, и он может либо скопировать один из вариантов посимвольным вводом с клавиатуры, либо выбрать его по номеру в списке вариантов, либо выделить "мышью". Меню может быть организовано в виде блока, в виде строки данных, в виде пиктограмм, в виде списка с пронумерованными вариантами. Меню может быть использовано в ресурсе Интернет как для ввода управляющих сообщений, так и для выбора ответа обучающимися.

2. Возможность моделирования объекта или процесса изучения. Высококачественное визуальное моделирование называют также замещением. Замещение возможно благодаря использованию образов, хранящихся на специальных серверах баз данных. Образы, в свою очередь, могут быть структурированы и организованы для дальнейшего использования в ресурсе Интернет в режиме реального времени. Они появляются в обучающих диалогах согласно ситуационному сценарию в зависимости от условий, задаваемых обучающимся или системой.

3. Наличие контроля обучения. При экспертизе обучающих ресурсов Интернет важным является наличие возможности запоминания значений контроля обучающегося. Это значит, что обучаемый должен чувствовать, что во время сеанса обучения система постоянно контролирует его действия. Модули системы должны также предоставлять возможность обучаемому самому выбирать и контролировать следующее: содержание обучения, шаг обучения, направление обучения, используемые в системе стили и методики.

При экспертной оценке следует учитывать время ответа, отводимое обучающемуся на тестовое задание в ресурсе Интернет. Как правило, обучающийся получает разную информацию в зависимости от того, уложился или не уложился его ответ в отведенное для этого время.

Обеспечение гипермедийности, включающее два параметра: наличие визуальных объектов навигации (рисунки, анимация и т.п.), воз-
можность вызова прикладной программы изложения учебного материала в ресурсе Интернет.

Обеспечение наличия различных режимов работы с ресурсом Интернет: режим обучающегося, режим обучающего.

При принятии решения об использовании того или иного ресурса портала в образовательных целях, следует, как уже говорилось ранее, провести его оценку (предлагается еще один из подходов к этому виду деятельности), руководствуясь следующими наиболее значимыми требования к образовательным Интернет-ресурсам.

Ход работы:

1. По предложенным параметрам провести анализ образовательно гесурса, сделать выводы и предложения. Результат обсудить в группах.

13. ПРАКТИЧЕСКАЯ РАБОТА 6

Название: Диагностика IP-протокола

Цель работы: Научиться проверять работоспособность сетевого подключения.

Теоретическая часть: Существуют различные утилиты, позволяющие быстро продиагностировать IP-подключение. Однако большинство операций легко может быть выполнено с использованием команд самой операционной системы.

Пользователи Windows XP для диагностики сетевого подключения могут воспользоваться специальным мастером. Эта программа вызывается из меню задачи Сведения о системе (Пуск > Все программы > Стандартные > Служебные > Сведения о системе > меню Сервис > Диагностика сети):

В ходе выполнения лабораторной работы Вы познакомитесь с утилитами, запускаемыми из командной строки, позволяющими детально продиагностировать работоспособность подключения Вашего компьютера к сети.

Ход работы:

Ipconfig

Для отображения параметров IP-протокола используются утилиты *ipconfig* (Windows NT/2000/XP) и *winipcfg* (Windows 9x). Эта утилита выводит на экран основные параметры настройки протокола TCP/IP: значения адреса, маски, шлюза.

Нажмите кнопку *Пуск*, выберите строку меню *Выполнить*, наберите символы cmd и нажмите клавишу *Enter* на клавиатуре.

В открывшемся окне наберите *ipconfig /all*. При нормальной работе компьютера на экран должен вывестись примерно такой листинг:

Windows IP Configuration Host Name : vest Primary Dns Suffix: tvs.tomsk.ru Node Type : Hybrid IP Routing Enabled. : No WINS Proxy Enabled. : No DNS Suffix Search List. : tvs.tomsk.ru tomsk.ru Ethernet adapter Local Area Connection: Connection-specific DNS Suffix .: tvs.tomsk.ru Description : Intel(R) PRO/100 S Desktop Adapter Physical Address. : 00-02-B3-8D-44-53 Dhcp Enabled. Yes Autoconfiguration Enabled : Yes Subnet Mask : 255.255.255.0 Default Gateway: 83.172.10.254 DHCP Server : 83.172.10.2 DNS Servers : 192.168.0.1 83.172.10.2 Primary WINS Server : 83.172.10.2 Secondary WINS Server : 213.183.109.3 Lease Obtained. : 24 августа 2004 г. 9:40:41 Lease Expires : 27 октября 2004 г. 9:40:41 Отключите сетевое подключение, повторите команду. При отсутствующем соединении на экран выводится примерно такой листинг: Windows IP Configuration Host Name : vest Primary Dns Suffix: tvs.tomsk.ru Node Type : Hybrid IP Routing Enabled. : No WINS Proxy Enabled. : No DNS Suffix Search List. : tvs.tomsk.ru tomsk.ru Ethernet adapter Local Area Connection: Media State: Media disconnected Description : Intel(R) PRO/100 S Desktop Adapter Physical Address. : 00-02-B3-8D-44-53

Обратите внимание, что программа вывела на экран только данные о "физических" параметрах сетевой карты и указала, что отсутствует подключение сетевого кабеля (*Media disconnected*).

В составе Windows 95/98/МЕ существует программа, которая в графическом режиме показывает основные настройки протокола TCP/IP:

Это winipcfg. Для вызова программы следует набрать ее имя в командной строке и нажать клавишу *Enter* (это возможно только в том случае, если Вы работаете с OC Windows 95/98/ME).

Ping

Команда используется для проверки протокола TCP/IP и достижимости удаленного компьютера. Она выводит на экран время, за которое пакеты данных достигают заданного в ее параметрах компьютера.

Проверка правильности установки протокола TCP/IP. Откройте командную строку и выполните команду:

ping 127.0.0.1

Адрес 127.0.0.1 — это личный адрес любого компьютера. Таким образом, эта команда проверяет прохождение сигнала "на самого себя". Она может быть выполнена без наличия какого-либо сетевого подключения. Вы должны увидеть приблизительно следующие строки:

Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

По умолчанию команда посылает пакет 32 байта. Размер пакета может быть увеличен до 65 кбайт. Так можно обнаружить ошибки при пересылке пакетов больших размеров. За размером тестового пакета отображается время отклика удаленной системы (в нашем случае — меньше 1 миллисекунды). Потом показывается еще один параметр протокола — значение TTL. TTL — "время жизни" пакета. На практике это число маршрутизаторов, через которые может пройти пакет. каждый маршрутизатор уменьшает значение TTL на единицу. При достижении нулевого значения пакет уничтожается. Такой механизм введен для исключения случаев зацикливания пакетов.

Если будет показано сообщение о недостижимости адресата, то это означает ошибку установки протокола IP. В этом случае целесообразно

удалить протокол из системы, перезагрузить компьютер и вновь установить поддержку протокола TCP/IP.

Проверка видимости локального компьютера и ближайшего компьютера сети. Выполните команду

ping 212.73.124.100

На экран должны быть выведены примерно такие строки:

Pinging 212.73.124.100 with 32 bytes of data:

Reply from 212.73.124.100: bytes=32 time=5ms TTL=60

Reply from 212.73.124.100: bytes=32 time=5ms TTL=60

Reply from 212.73.124.100: bytes=32 time=4ms TTL=60

Reply from 212.73.124.100: bytes=32 time=4ms TTL=60

Ping statistics for 212.73.124.100:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 4ms, Maximum = 5ms, Average = 4ms

Наличие отклика свидетельствует о том, что канал связи установлен и работает.

Проверка работоспособности сервера имен Internet. Выполните команду

ping www.tomsk.ru

Если система сможет различить IP-адрес этого хоста, то система распознавания имен работоспособна. На экран должен быть выведен примерно такой листинг:

Pinging www.tomsk.ru [212.73.124.100] with 32 bytes of data:

Reply from 212.73.124.100: bytes=32 time=3ms TTL=60

Reply from 212.73.124.100: bytes=32 time=1ms TTL=60

Reply from 212.73.124.100: bytes=32 time=7ms TTL=60

Reply from 212.73.124.100: bytes=32 time=15ms TTL=60

Ping statistics for 212.73.124.100:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 15ms, Average = 6ms

Если не будет ответа на ввод команды с именем существующего хоста, то это может свидетельствовать либо об ошибке в задании DNSсерверов, либо об их неработоспособности.

Tracert

При работе в Сети одни информационные серверы откликаются быстрее, другие медленнее, бывают случаи недостижимости желаемого хоста. Для выяснения причин подобных ситуаций можно использовать специальные утилиты. Например, команда *tracert*, которая обычно используется для показа пути прохождения сигнала до желаемого хоста. Зачастую это позволяет выяснить причины плохой работоспособности канала. Точка, после которой время отклика резко увеличено, свидетельствует о наличии в этом месте "узкого горлышка", не справляющегося с нагрузкой.

В командной строке введите команду:

tracert www.lenta.ru

Вы должны увидеть примерно такой листинг:

Tracing route to www.lenta.ru [81.19.69.28]

over a maximum of 30 hops:

1 <1 ms <1 ms<1 ms 83.172.10.1 * 2 * * Request timed out. * * 3 * Request timed out. 4 81 ms 234 ms 176 ms s2-0-edge52.tomsknet.ru [217.18.135.237] 5 135 ms 221 ms 36 ms fe2-0-border.tomsknet.ru [217.18.130.229] 6 205 ms 249 ms 262 ms tomsk-car0-ge0-2.rt-comm.ru [213.59.1.121] 7 123 ms 241 ms 276 ms kochenevo-dsr0-ml3.rt-comm.ru [217.106.15.1] 8 309 ms 273 ms 241 ms kochenevo-bbn0-po3-7.rt-comm.ru [217.106.6.153] 89 ms 109 ms 143 ms tschelkun-bbn0-po1-1.rt-comm.ru 9 [217.106.6.129] 10 150 ms 121 ms 177 ms shigony-bbn0-po2-2.rt-comm.ru [217.106.6.121] 11 77 ms 81 ms 94 ms msk-bbn0-po1-6.rt-comm.ru [217.106.6.65] 12 95 ms 116 ms 85 ms msk-dsr2-ge4-0-0.rt-comm.ru [217.106.7.229] 13 108 ms 104 ms 107 ms 213.59.1.250 14 125 ms 94 ms 94 ms NP-6009.core.rambler.ru [217.73.194.6] 94 ms lenta12.cust.ramtel.ru [81.19.69.28] 15 137 ms 80 ms Trace complete.

<u>Nslookup</u>

Утилита используется для получения информации от DNS-сервера. По умолчанию (после запуска без указания параметров) осуществляется подключение к указанному в настройках протокола серверу DNS. Набирая необходимые имена в качестве запроса, вы можете получить информацию о данных DNS по этому имени, найти почтовый сервер, обслуживающий домен, уточнить данные регистрации и т.п.

Выполните команду nslookup.

Наберите server ns.cctpu.edu.ru и нажмите Enter — этой командой мы указываем, какой DNS-сервер мы хотим использовать для получения интересующих нас данных.

Наберите *set type=all* и нажмите *Enter* — этой командой мы указали, что нас будут интересовать все данные касательно задаваемого нами домена.

Наберите *tpu.ru* и нажмите *Enter* — этой командой мы запрашиваем данные по домену *tpu.ru*

```
На экране вы должны получить примерно такой листинг:
      nslookup
Default Server: don.tvs.tomsk.ru
Address: 83.172.10.2
      server don.tvs.tomsk.ru
Default Server: don.tvs.tomsk.ru
Address: 83.172.10.2
      set type=all
      tpu.ru
Server: don.tvs.tomsk.ru
Address: 83.172.10.2
       internet address = 213.183.112.98
tpu.ru
tpu.ru
        nameserver = ns.tpu.ru
tpu.ru
        nameserver = ns.tsu.ru
tpu.ru
        nameserver = ns.tomsk.su
tpu.ru
        nameserver = ns2.tpu.ru
tpu.ru
        nameserver = ns2.tomsk.su
tpu.ru
primary name server = ns.tpu.ru
responsible mail addr = hostmaster.tpu.ru
serial = 2004082601
refresh = 86400 (1 day)
retry = 3600 (1 \text{ hour})
expire = 3600000 (41 days 16 hours)
default TTL = 43200 (12 hours)
tpu.ru
        MX preference = 5, mail exchanger = relay1.tpu.ru
        MX preference = 10, mail exchanger = relay.tpu.ru
tpu.ru
        MX preference = 20, mail exchanger = info.tsu.ru
tpu.ru
ns.tpu.ru internet address = 195.208.172.1
ns.tsu.ru internet address = 212.192.100.1
ns.tomsk.su
                internet address = 212.73.124.2
ns2.tpu.ru
                internet address = 213.183.112.102
ns2.tomsk.su
                internet address = 212.73.97.13
```

| relay1.tpu.ru | internet address = 213.183.112.100 |
|---------------|------------------------------------|
| relay.tpu.ru | internet address = 195.208.172.15 |

ЛИТЕРАТУРА

- Тузовский А. Ф. Проектирование Интернет-приложений [Электронный ресурс]: учебно-методическое пособие / А. Ф. Тузовский; Национальный исследовательский Томский политехнический университет (ТПУ). — 1 компьютерный файл (pdf; 4.5 MB). — Томск: Изд-во ТПУ, 2011. Схема доступа: <u>http://www.lib.tpu.ru/fulltext2/m/2011/m427.pdf</u>
- Шерстнёв В. С. Программное обеспечение интернет-серверов [Электронный ресурс] : учебное пособие / В. С. Шерстнёв; Национальный исследовательский Томский политехнический университет (ТПУ). — 1 компьютерный файл (pdf; 1.8 MB). — Томск: Изд-во ТПУ, 2010.
- 3. Схема доступа: <u>http://http://www.lib.tpu.ru/fulltext2/m/2011/m259.pdf</u>
- Корчуганова М.А. Мировые информационные ресурсы: учебное пособие [Электронный ресурс].-:, 2009. - 1 электрон. опт. диск (CD-ROM) с.
- 5. Балабанов И.Т. Электронная коммерция. Учебник для ВУ-Зов:Питер, СПб, 2011. – 200 С.
- Корчуганова М.А. Мировые информационные ресурсы и стандарты информатизации: лабораторный практикум. [Электронный ресурс].- 2013. - 1 компьютерный файл (pdf; 1.8 MB). – Схема доступа: <u>http://moodle.uti.tpu.ru:8080/mod/resource/view.php?id=3692</u>
- 7. <u>http://lphp.ru/</u> Сайт «PHP уроки»
- 8. <u>http://phpclub.ru/</u> Сайт «PHP клуб»
- 9. <u>https://www.coursera.org/</u> Интернет-портал онлайн курсов «Coursera»
- 10. Международные информационные ресурсы и стандарты информатизации // Moodle UTI TPU: [Электронный ресурс] Режим достуna: http://moodle.uti.tpu.ru:8080/course/view.php?id=155
- 11. Рощин С.М. Как быстро найти нужную информацию в Интернете. – 2010. - 1 компьютерный файл. – Схема доступа:
- 12. http://ezproxy.ha.tpu.ru:2117/view/book/1137/

ОГЛАВЛЕНИЕ

| 1. ЛАБОРАТОРНАЯ РАБОТА № 1 | |
|-----------------------------|-----|
| 2. ЛАБОРАТОРНАЯ РАБОТА № 2 | |
| 3. ЛАБОРАТОРНАЯ РАБОТА № 3 | |
| 4. ЛАБОРАТОРНАЯ РАБОТА № 4 | |
| 5. ЛАБОРАТОРНАЯ РАБОТА № 5 | |
| 6. ЛАБОРАТОРНАЯ РАБОТА № 6 | |
| 7. ЛАБОРАТОРНАЯ РАБОТА № 7 | |
| 8. ПРАКТИЧЕСКИЯ РАБОТА № 1 | |
| 9. ПРАКТИЧЕСКАЯ РАБОТА № 2 | 60 |
| 10. ПРАКТИЧЕСКАЯ РАБОТА № 3 | |
| 11. ПРАКТИЧЕСКАЯ РАБОТА № 4 | 67 |
| 12. ПРАКТИЧЕСКАЯ РАБОТА № 5 | |
| 13. ПРАКТИЧЕСКАЯ РАБОТА № 6 | 109 |

Учебное издание

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ

Методические указания к выполнению лабораторных работ по курсу «Международные информационные ресурсы и стандарты информатизации» для магистров II курса, обучающихся по направления 230700 «Прикладная информатика в аналитической экономике»

Составитель КОРЧУГАНОВА Марина Анатольевна

Печатается в редакции составителя

Отпечатано в Издательстве ЮТИ ТПУ в полном соответствии с качеством предоставленного оригинал-макета

Подписано к печати 14.02.2016 г. Формат 60х84/23 Бумага офсетная. Плоская печать. Усл. печ. л. 6,52. Уч-изд. л. 5,81. Тираж 20 экз. Заказ 1620. Цена свободная. ИПЛ ЮТИ ТПУ. Ризограф ЮТИ ТПУ. 652000, г. Юрга, ул. Московская, 17.