

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Воронежский государственный технический университет»**

**Кафедра высшей математики
и физико-математического моделирования**

**СОЗДАНИЕ ПРОСТЕЙШИХ
WEB-ДОКУМЕНТОВ**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

**к выполнению лабораторной работы по дисциплинам
«Информатика» и «Практикум по информационным
технологиям» для студентов направлений 14.03.01 «Ядерная
энергетика и теплофизика», 16.03.01 «Техническая физика»,
21.03.01 «Нефтегазовое дело», 22.03.02 «Металлургия»
и 28.03.01 «Нанотехнологии и микросистемная техника»
очной и очно-заочной форм обучения**

Воронеж 2021

УДК 004.738.52:681.3.06(07)
ББК 32.97я7

Составители:

С. А. Кострюков, В. В. Пешков, Г. Е. Шунин

Создание простейших Web-документов: методические указания к выполнению лабораторной работы по дисциплинам «Информатика» и «Практикум по информационным технологиям» для студентов направлений 14.03.01 «Ядерная энергетика и теплофизика», 16.03.01 «Техническая физика», 21.03.01 «Нефтегазовое дело», 22.03.02 «Металлургия» и 28.03.01 «Нанотехнологии и микросистемная техника» очной и очно-заочной форм обучения/ ФГБОУ ВО «Воронежский государственный технический университет»; сост.: С. А. Кострюков, В. В. Пешков, Г. Е. Шунин. – Воронеж: Изд-во ВГТУ, 2021. – 32 с.

В методических указаниях рассмотрены основные приемы и методы создания простейших Web-страниц, основные понятия языка HTML, приведен обзор возможностей HTML-редактора Microsoft Office SharePoint Designer.

Предназначены для студентов очной и очно-заочной форм обучения.

Методические указания подготовлены в электронном виде и содержатся в файле МУ-HTML.pdf.

Ил. 2. Библиогр.: 3 назв.

УДК 004.738.52:681.3.06(07)
ББК 32.97я7

Рецензент – И. М. Пашуева, канд. физ.-мат. наук, доц. кафедры высшей математики и физико-математического моделирования ВГТУ

*Издается по решению редакционно-издательского совета
Воронежского государственного технического университета*

ВВЕДЕНИЕ

Сегодня Интернет используется как источник разносторонней информации по различным областям знаний. Большинство документов, доступных на серверах Интернета, имеют гипертекстовый формат. Службу Интернета, управляющую передачей таких документов, называют Всемирной паутиной – World Wide Web (Web, WWW).

Среда WWW не имеет централизованной структуры. Она пополняется теми, кто желает разместить в Интернете свои материалы, и может рассматриваться как информационное пространство. Как правило, документы WWW хранятся на постоянно подключенных к Интернету компьютерах – web-серверах. Обычно на web-сервере размещают не отдельный документ, а группу взаимосвязанных документов – web-узел (чаще используется термин web-сайт или просто сайт). Размещение подготовленных материалов на сайте называется web-изданием или web-публикацией.

Отдельный документ WWW называют web-страницей. Обычно это комбинированный документ, который может содержать текст, графические иллюстрации, мультимедийные и другие вставные объекты. Для создания web-страниц используется язык HTML (HyperText Markup Language – язык разметки гипертекста), который при помощи вставленных в документ тегов описывает логическую структуру документа, управляет форматированием текста и размещением вставных объектов.

Гипертекст – это расширенный текст, включающий, помимо собственно текста, ещё и гиперссылки, иллюстрации, активные компоненты и другие дополнительные элементы. Разметка гипертекста означает использование специальных кодов, которые легко отделяются от смыслового содержания документа и используются только для его форматирования и оформления. Применение этих кодов подчиняется строгим правилам, определяемым спецификацией языка HTML. Последняя версия – HTML 5.2 – была представлена 14 декабря 2017 г. и может сейчас рассматриваться как окончательная редакция языка HTML, содержащая все необходимое для полноценной разметки документа.

Основные понятия языка HTML

Публикация в Интернете каких-либо материалов состоит из двух этапов: подготовки материалов и их опубликования. Подготовка состоит в создании документов, формат которых позволяет их размещение в Интернете, то есть web-страниц, написанных на языке HTML. Собственно публикация, то есть открытие общего доступа к материалам, осуществляется после решения организационных вопросов, связанных с получением дискового пространства на web-сервере для их размещения.

Как правило, для создания web-документов используют язык HTML (HyperText Markup Language – язык разметки гипертекста).

Управляющие конструкции HTML называются *тегами* (англ. *tag*) и вставляются непосредственно в текст документа. Все теги заключаются в угловые скобки <...>. Сразу после открывающей скобки помещается ключевое слово, определяющее тег, например <DIV>. При выводе документа HTML на экран сами теги не отображаются, но влияют на характер отображения документа. Теги HTML бывают парными и непарными. Непарные теги оказывают воздействие на весь документ или определяют разовый эффект в месте своего появления. Парные теги представляют собой открывающий и закрывающий теги, которые воздействуют на часть документа, заключенную между ними. Закрывающий тег отличается от открывающего символом «/» (косая черта) перед ключевым словом (например, </DIV>). Закрытие парных тегов выполняется так, чтобы соблюдались правила вложения.

<I>На этот текст воздействуют два тега</I>

Эффект от применения тега может изменяться путем добавления атрибутов. В случае парных тегов атрибуты присутствуют лишь в открывающем теге. Атрибуты – это дополнительные ключевые слова, отделяемые пробелом от ключевого слова, определяющего тег, а также от других атрибутов, и размещаемые до завершающего тег символа «>». Некоторые атрибуты требуют указания значения атрибута, которое отделя-

ется от ключевого слова атрибута символом «=» (знак равенства) и заключается в кавычки:

```
<H1 ALIGN="LEFT">
```

При создании документов HTML могут использоваться два способа их формирования. Первый заключается в разметке существующего (или создаваемого) документа вручную, при этом автор или редактор добавляет в документ теги разметки. Это можно сделать в текстовом редакторе или редакторе HTML, имеющем специальные элементы управления для упрощения ввода тегов. В обоих этих случаях работа ведется средствами HTML и требует знания языка и умения его применять.

Принципы второго подхода аналогичны принципам работы в текстовых процессорах, таких, как Microsoft Word. Сведения о форматировании частей документа можно рассматривать как «разметку», добавляемую в документ. Однако для использования текстового процессора не требуется знаний о формате документа и «языке разметки»: изменения, отображаемые на экране, вносятся в документ автоматически. Такой принцип соответствия экранного изображения реальному получил название WYSIWYG (от англ. «What You See Is What You Get» – «Что видите, то и получаете»).

В качестве редактора WYSIWYG для языка HTML можно использовать Microsoft Word или входящую в состав пакета Microsoft Office программу SharePoint Designer. Существует также достаточно много других программ того же назначения.

Работа вручную позволяет создавать более универсальные, более качественные и более разнообразные документы. Второй способ проще освоить, так как он не требует знания языка HTML. Однако в этом случае используются средства форматирования вместо средств описания, что может иногда приводить к нежелательным последствиям. При подготовке крупных web-узлов, содержащих десятки, а то и сотни web-страниц, та или иная автоматизация работы необходима. Как правило, в таких случаях используют комбинированный подход: «рядовые» страницы готовят с помощью автоматизированных средств, а в особо сложных или в особо важных случаях задают оформление вручную.

Структура документа HTML

Все документы HTML имеют одинаковую структуру, которая определяется фиксированным набором тегов структуры. Документ HTML всегда начинается с тега <HTML> и заканчивается соответствующим закрывающим тегом </HTML>. Внутри документа должны быть два основных раздела – раздел заголовков и тело документа, идущих именно в таком порядке. Раздел заголовков содержит данные, описывающие документ в целом, и ограничивается тегами <HEAD> и </HEAD>. В частности, он должен содержать общий заголовок документа, ограниченный парным тегом <TITLE>, </TITLE>.

Основное содержание помещается в теле документа, которое ограничивается тегами <BODY> и </BODY>. Вообще говоря, положение этих структурных тегов в документе нетрудно определить, даже если они опущены. Поэтому стандарт языка HTML требует только наличия тега <TITLE> (и, соответственно, </TITLE>). Тем не менее, при создании документа HTML опускать структурные теги не рекомендуется.

Простейший корректный документ HTML, содержащий все теги, определяющие структуру, выглядит следующим образом:

```
<HTML>
<HEAD><TITLE>Заголовок документа</TITLE></HEAD>
<BODY>
Это текст документа
</BODY>
</HTML>
```

Элементы HTML

В случае парных тегов их область действия – от открывающего до закрывающего тега. Эта часть документа рассматривается как элемент языка HTML. Например, можно говорить об «элементе BODY», включающем тег <BODY>, основное содержание документа и закрывающий тег </BODY>. Весь документ HTML формально можно рассматривать как «элемент HTML». В случае непарных тегов элемент совпадает с тегом, который его определяет.

Как правило, элемент языка HTML описывает часть документа и находится между тегами <BODY> и </BODY>. Такие элементы делятся на блочные и текстовые. Блочные элементы относятся к частям текста уровня абзаца. Текстовые элементы описывают свойства отдельных фраз и еще более мелких частей текста.

Правила вложения элементов:

- Элементы не должны пересекаться. То есть, если открывающий тег находится внутри элемента, то и соответствующий закрывающий тег должен стоять внутри того же элемента.

- Блочные элементы могут содержать вложенные блочные и текстовые элементы.

- Текстовые элементы могут содержать вложенные текстовые элементы, но не могут содержать вложенные блочные элементы.

Вообще говоря, все правила языка HTML можно рассматривать как «пожелания». Средство воспроизведения web-документа сделает всё возможное, чтобы истолковать разметку наиболее разумным образом. Тем не менее, гарантию правильного отображения документа даёт только четкое следование правилам языка.

Заголовки и абзацы

Основными функциональными элементами в большинстве web-страниц являются заголовки и абзацы. Язык HTML позволяет использовать заголовки шести уровней. Они задаются парными тегами от <H1> до <H6>. При выводе web-страницы на экран эти заголовки отображаются шрифтами разного размера.

Обычные абзацы задаются парным тегом <P> (при этом закрывающий тег </P> необязателен). При отображении на экране абзацы отделяются друг от друга пустой строкой. Например:

```
<H1>Заголовок первого уровня</H1>
```

```
<P>Первый абзац текста<P>Второй абзац текста
```

```
<H2>Заголовок второго уровня</H2>
```

```
<P>Третий абзац текста
```

Следует помнить, что из-за наличия специального тега, задающего абзац, обычного символа конца абзаца (клавиша

Enter) для создания нового абзаца недостаточно. Язык HTML рассматривает символы конца абзаца и пробелы особым образом. Любая последовательность, состоящая только из пробелов и символов конца абзаца, при отображении документа рассматривается как одиночный пробел. В частности, это означает, что символ конца абзаца даже не осуществляет перехода на новую строку (для этого используется непарный тег
).

В качестве разделителя абзацев может также использоваться горизонтальная линейка. Этот элемент задается непарным тегом <HR>. При отображении документа на экране линейка отделяет части текста друг от друга. Ее параметры задаются атрибутами тега. Например:

```
<HR ALIGN="RIGHT" SIZE="10" WIDTH="50%">
```

Этот тег создает горизонтальную линейку шириной в 10 пикселей, которая занимает половину ширины окна и располагается справа.

Гиперссылки

Гипертекстовая ссылка, или просто гиперссылка, является текстовым элементом, который задаётся парным тегом <A>. Этот тег должен содержать обязательный атрибут HREF=, в качестве значения которого используется адрес URL документа, на который указывает ссылка. Она может указывать на произвольную web-страницу на любом общедоступном сайте, либо другую часть этого же документа. Пример открывающего тега:

```
<A HREF="http://www.abcdef.com/index.htm">.
```

Адрес URL в абсолютной форме начинается с указания протокола и адреса сайта. Такая форма записи используется, если нужно перенаправить посетителя на другой сайт, и является внешней ссылкой. Если же используется относительный адрес, то в ссылке указывается относительный путь доступа для документа. При этом предполагается использование того же протокола и того же сайта, а ссылка рассматривается как внутренняя. Такая ссылка сохраняет работоспособность в случае изменения адреса сайта в целом, например, при переносе его на другой сервер.

Гиперссылки могут применяться для ссылки на файлы мультимедиа. Это удобно, так как при этом не приходится ждать загрузки мультимедийных файлов при работе с данной web-страницей. Если же нужно интегрирование объектов мультимедиа в web-страницу, то используется парный тег <OBJECT> или нестандартный непарный тег <EMBED>, который тоже поддерживается большинством браузеров.

Полный формат гиперссылки дает возможность ссылки на определенное место внутри той же страницы. Это можно сделать, пометив соответствующее место при помощи якоря. Якорь задается также парным тегом <A>, но здесь обязательным атрибутом является NAME=. Значение атрибута – произвольная последовательность символов (латинских букв и цифр, пробелы недопустимы), которая будет служить именем якоря. Для ссылки на якорь его имя указывается в конце адреса URL после символа «#»:

```
<A HREF="http://www.abcdef.com/index.htm#address">
```

Web-графика

В настоящее время для включения в web-страницы используются в основном три формата изображений – GIF, JPEG и PNG. Все браузеры способны распознавать и отображать файлы этих форматов.

Подготовку изображений можно производить в любом графическом редакторе, который способен сохранять файлы в указанных форматах, например, в программе Paint.

Рисунки в формате GIF (Graphic Interchange Format) содержат 256 цветов, заданных индексной палитрой. Файл архивирован и имеет намного меньший размер, чем соответствующий неупакованный растровый рисунок (например, в формате .BMP). Формат GIF поддерживает анимационные изображения.

Файлы с рисунками в формате JPEG (Joint Photographic Expert Group) имеют расширение .JPEG или .JPG. Этот формат служит для хранения фотографий, использующих 24-битный цвет. Алгоритм JPEG позволяет сжимать изображение как с

потерями, так и без потерь. Поддерживаются изображения с линейным размером не более 65535×65535 пикселей.

PNG (Portable Network Graphics) – растровый формат хранения графических данных, также использующий сжатие без потерь. PNG был создан как свободный формат для замены устаревшего формата GIF и позиционируется прежде всего для использования в Интернете и редактирования графики.

Рисунки хранятся на web-сайтах в отдельных файлах, но отображаются как элементы web-страниц. Для вставки изображения в web-страницу используется непарный тег . Этот тег должен включать обязательный атрибут SRC=, в котором задаётся адрес URL файла с изображением (относительный или абсолютный):

```
<IMG SRC="pic1.gif">
```

По умолчанию при выводе рисунка браузер будет использовать его реальные размеры. Если же рисунок нужно масштабировать, то используют атрибуты WIDTH= и HEIGHT=, где задаются соответственно ширина и высота рисунка в пикселах. Если эти параметры присутствуют, браузер может рассчитать требуемое место для отображения рисунка ещё до того, как рисунок будет загружен. Это немного ускорит вывод на экран данной web-страницы, поэтому рекомендуется всегда задавать эти атрибуты.

```
<IMG SRC="pic2.jpg" WIDTH="150" HEIGHT="50">
```

Расположение рисунка относительно текста и других элементов web-страницы может быть различным. Поскольку рисунок задается как элемент, находящийся внутри какого-то абзаца, то по умолчанию он рассматривается как встроенное изображение, включённое в строку текста. Для того, чтобы рисунок отображался независимо, его следует включить в отдельный абзац. Для рисунка, включённого в строку, можно задать режим взаимодействия с текстом при помощи атрибута ALIGN=.

```
<IMG SRG="pic3.gif" ALIGN="BOTTOM">
```

Этот атрибут может иметь следующие значения:

- ALIGN="BOTTOM" – выравнивается нижняя граница изображения и основание текстовой строки;

- ALIGN="MIDDLE" – выравнивается середина изображения и середина текстовой строки;

- ALIGN="TOP" – выравнивается верхняя граница изображения и верхний обрез текстовой строки.

Также возможно использование изображения, обтекаемого текстом, для чего также служит атрибут ALIGN=:

- ALIGN="LEFT" – рисунок размещается у левого края страницы, а текст абзаца – справа от него;

- ALIGN="RIGHT" – рисунок размещается у правого края страницы, а текст абзаца – слева от него.

В этом случае следует располагать тег в самом начале абзаца.

Рекомендуется, чтобы между рисунком и текстом присутствовал некоторый отступ, как по горизонтали, так и по вертикали. Его величину можно задать с помощью атрибутов HSPACE= (по горизонтали) и VSPACE= (по вертикали). Размеры отступов задаются в пикселах.

При создании web-страниц с рисунками следует иметь в виду, что ряд пользователей может отключить в своих браузерах функцию просмотра изображений (например, для ускорения загрузки страниц или уменьшения трафика). Таких пользователей можно ознакомить с содержанием рисунка при помощи альтернативного текста. Этот текст задается с помощью атрибута ALT= и выводится вместо рисунка, если он почему-то не может быть выведен.

Поскольку рисунок задается как текстовый элемент, то он может быть помещен внутрь другого текстового элемента, например, задающего ссылку (тег <A>). В этом случае рисунок становится изображением-ссылкой. При выводе документа на экран такой рисунок отличается синей рамкой и изменением формы указателя мыши при наведении на него.

Также изображение может быть использовано в качестве фонового рисунка web-страницы. При выводе такой страницы на экран рабочая область окна заполняется этим рисунком по-

следовательно (как паркетом), считая от верхнего левого угла окна. Фоновый рисунок можно задать при помощи атрибута BACKGROUND= в теге <BODY>. Его значение – абсолютный или относительный адрес URL файла с изображением.

<BODY BACKGROUND="sky.gif" TEXT="BROWN">

Форматирование текста

Управление форматированием текста не является основной задачей языка HTML, и поэтому теги, выполняющие эту задачу, начиная с версии 4.0 рассматриваются как устаревшие и их использование не рекомендуется, их функцию сейчас выполняют каскадные таблицы стилей – Cascading Style Sheets (CSS). Однако теги HTML по-прежнему предоставляют удобный способ управления видом документа на экране и поддерживаются браузерами.

Управлять параметрами шрифта позволяет парный тег . В него обязательно должен входить хотя бы один из атрибутов: COLOR=, FACE= или SIZE=.

Атрибут COLOR= определяет цвет текста, который обычно задается текстовым значением (например, COLOR="GREEN") или шестнадцатеричным кодом, в котором последовательные байты означают интенсивность красной, зеленой и синей компонент цвета. Например, COLOR="#00FF00" даст такой же результат, что и COLOR="GREEN". Атрибут FACE= определяет гарнитуру шрифта. Его значение сравнивается с именами всех шрифтов, установленных в системе. Атрибут SIZE= задаёт размер шрифта (в относительных единицах от 1 до 7). Для атрибута SIZE= можно задавать значение со знаком (плюс или минус), что будет означать увеличение или уменьшение размера шрифта относительно текущего значения.

В предыдущих версиях HTML параметры шрифта, используемые по умолчанию, задавались с помощью непарного тега <BASEFONT>, который помещался один раз внутри элемента BODY с теми же атрибутами, что и тег . В на-

стоящее время тег <BASEFONT> не поддерживается браузерами, кроме Internet Explorer.

Внешний вид символов задается при помощи парных тегов (полужирный), <I> (курсив), <U> (подчеркнутый), <S> (вычеркнутый). Их использование не рекомендуется. Вместо них следует применять теги, описывающие функциональные особенности текста, например, вместо

Внимание!

лучше написать

Внимание!

Так, для отображения цитат служит парный тег <CITE> (выводятся курсивом), аналогами курсивного и полужирного начертаний являются парные теги (выделение) и (сильное выделение). Также язык HTML содержит набор тегов для описания работы компьютерных программ. Для этого служат парные теги <CODE> (исходный текст программы), <KBD> (текст, вводимый с клавиатуры), <SAMP> (пример вывода программы) и <VAR> (программные переменные). Для вывода соответствующих элементов используется моноширинный шрифт. Кроме того, переменные выводятся курсивом, а клавиатурный ввод (в некоторых браузерах) – полужирным шрифтом.

Списки

Наиболее популярные виды списков, поддерживаемые языком HTML, это упорядоченные списки, неупорядоченные списки и списки определений. Все виды списков представляют собой блочные элементы.

Упорядоченные (нумерованные) списки создаются при помощи парных тегов и , а неупорядоченные (маркированные) – при помощи парных тегов и . Открывающие теги могут включать атрибуты, задающие вид маркера (в случае неупорядоченного списка) и вид и последовательность нумерации (для упорядоченного). Например, атрибут TYPE="I" задаёт римскую нумерацию элементов списка, а TYPE="1" – арабскую. Разрешается вложение списков друг в друга. Каждый

элемент списка задаётся при помощи парного тега , закрывающий тег здесь можно опускать.

Список определений создается с помощью парного тега <DL>. Список включает элементы двух типов: определяемые термины (задаются парным тегом <DT>) и определения (задаются парным тегом <DD>). Закрывающие теги (</DT> и </DD>) здесь можно опускать. Как правило, определяемые термины и определения чередуются, хотя жёстко это не зафиксировано. Определения выводятся на экран с отступом от левого края. Пример подобного списка:

```
<DL>
```

```
<DT>Поршень –
```

```
<DD>сплошной цилиндр или диск, который плотно входит  
внутрь полого цилиндра
```

```
</DL>
```

Создание таблиц

Для представления больших объемов структурированных данных наиболее удобны такие объекты, как таблицы. Также многие web-дизайнеры часто применяют их для точного размещения элементов страниц.

Таблица в HTML создается с помощью парного тега <TABLE>. В ней может присутствовать заголовок, определяемый парным тегом <CAPTION>. Строки таблицы задаются парными тегами <TR> (однако закрывающие теги </TR> можно опускать).

В каждой строке таблицы содержатся ячейки, которые могут быть двух разных типов: ячейки в заголовках столбцов и строк (задаются парным тегом <TH>), и обычные ячейки (задаются парным тегом <TD>). Как и для строк, закрывающие теги для ячеек (</TH> и </TD>) можно опускать. Например, пустая таблица из двух строк и двух столбцов задаётся следующим образом:

```
<TABLE>
```

```
<CAPTION>Пустая таблица</CAPTION>
```

```
<TR><TD><TD>
```

```
<TR><TD><TD>
```

```
</TABLE>
```

В каждой ячейке таблицы может находиться произвольный текст, включая любые теги HTML, допустимые в «теле» страницы. Например, ячейка может содержать вложенную таблицу или рисунок.

Когда таблица отображается на экране монитора, происходит ее автоматическое форматирование с подбором размеров ячеек в соответствии с объемом размещаемых данных и заданными атрибутами. Атрибуты элементов предоставляют широкий выбор разнообразных способов оформления таблиц. На рис. 1 приведен краткий перечень допустимых атрибутов.

Атрибут	Элемент	Назначение
ALIGN=	Таблица, заголовок, строка, ячейка	Выравнивание таблицы по горизонтали; выравнивание данных по горизонтали; размещение заголовка над или под таблицей
VALIGN=	Строка, ячейка	Выравнивание по вертикали
WIDTH=	Таблица, ячейка	Минимальная ширина
HEIGHT=	Ячейка	Минимальная высота
COLSPAN=	Ячейка	Протяженность в несколько столбцов
ROWSPAN=	Ячейка	Протяженность в несколько строк
BGCOLOR=	Таблица, ячейка	Цвет фона
CELLSPACING=	Таблица	Зазор между ячейками
CELLPADDING=	Таблица	Зазор между содержимым ячейки и ее границей
BORDER=	Таблица	Отображение границ ячеек и внешней рамки таблицы

Рис. 1

Вывод нескольких документов на одной странице

Язык HTML даёт возможность вывести в рамках одной web-страницы сразу несколько документов. Для этого web-страница разбивается на несколько частей, которые называются *фреймами*. Разбиение страницы описывается специальной конструкцией HTML, в которой вместо тела документа (элемент BODY) находится описание фреймов. Оно задаётся парным тегом <FRAMESET>. Элемент BODY в такой странице отсутствует, а при наличии – игнорируется браузером.

Открывающий тег <FRAMESET> обязательно содержит атрибут COLS= или ROWS=, который определяет способ разбиения окна. В первом случае окно разбивается вертикальными линиями (т.е. на колонки), во втором – горизонтальными.

Если же заданы оба атрибута, то создается сетка фреймов. Значение каждого из этих атрибутов – это перечисление через запятую размеров отдельных фреймов, например:

```
<FRAMESET COLS="60%,40%">
```

Эти значения могут задаваться в пикселах или в процентах от ширины окна. Для последней области в качестве размера можно использовать символ «*» (это значит, что ей будет выделено все оставшееся пространство):

```
<FRAMESET ROWS="40%,40%,*">
```

Между открывающим и закрывающим тегами `<FRAMESET>` и `</FRAMESET>` должно находиться ровно столько элементов, сколько областей было создано атрибутами `COLS=` и `ROWS=`. Можно при этом использовать дополнительные элементы `FRAMESET` (для описания дальнейшего разбиения на подобласти еще меньшего размера) или же непарные теги `<FRAME>`, которые задают способ использования области.

Тег `<FRAME>` обязательно должен включать атрибут `SRC=`, который указывает, какой именно документ загружается в соответствующую область при открытии страницы. Его значение – адрес URL нужного документа (абсолютный или относительный).

Атрибут `NAME=` позволяет задать имя для созданной области, например:

```
<FRAME SRC="text.htm" NAME="left">
```

Это имя может использоваться для загрузки новых документов в данную область. Для этого в тег `<A>`, задающий гиперссылку, нужно добавить атрибут `TARGET=`, значение которого – ранее заданное имя области. При переходе по данной ссылке новый документ загрузится именно в указанный фрейм.

Предположим, например, что начальная страница сайта включает два фрейма: слева находится панель навигации, а справа – сама текущая страница. Если правой области было присвоено имя, которое используется во всех ссылках в левой области, то клик на любой ссылке панели навигации будет приводить к обновлению содержимого правой области, а сама навигационная панель останется без изменений.

Работа в редакторе Microsoft Office SharePoint Designer 2007

Программа **SharePoint Designer 2007** входящая в состав пакета Microsoft Office 2007, рассчитана, в первую очередь, не на создание отдельных web-страниц, а на сопровождение полноценного web-узла. Она содержит средства контроля структуры узла, единства оформления, правильности внутренних и внешних гиперссылок. Средства создания страниц, рассматриваемые в данной работе, – это всего лишь небольшая и не основная часть возможностей этой программы.

Окно программы SharePoint Designer представляет собой сочетание окна редактора и окна браузера (рис. 2). При отображении редактором текста документа HTML отображаются даже обычно невидимые элементы (такие, как якоря). При этом текст можно редактировать такими же средствами, что и в обычном текстовом процессоре. Программа имеет три режима отображения документа – Конструктор, С разделением и Код.

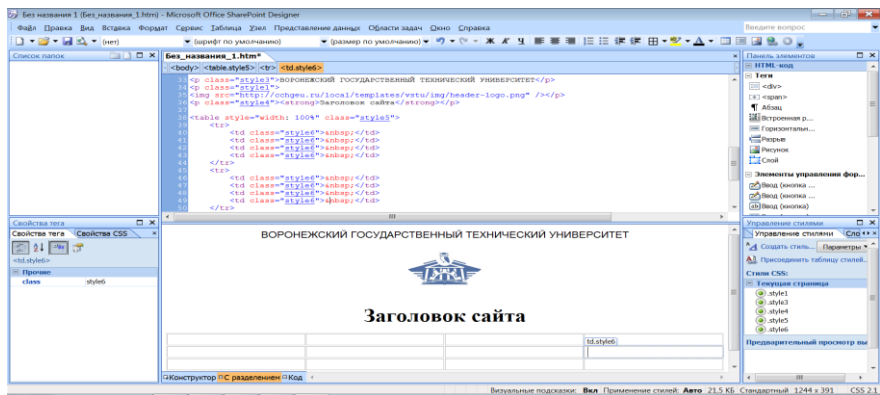


Рис. 2. Отображение web-документа в режиме С разделением

Основным средством форматирования является панель инструментов **Общие**, при этом почти все её инструменты присутствуют также на панели **Форматирование**. Эти панели включают раскрывающиеся списки **Стиль**, **Шрифт** и **Размер**, назначение которых то же, что и в программе Word, кнопки выбора начертания символов и выравнивания текста, кнопки создания

списков, как маркированных (неупорядоченных), так и нумерованных (упорядоченных), кнопки задания отступа текста, кнопки создания рамок и выбора цвета шрифта и фона, вставки таблиц, рисунков, гиперссылок и др.

Дополнительные инструменты форматирования, не вынесенные на панель в виде кнопок, содержатся в отдельных диалоговых окнах. Они доступны через верхнее меню, например, **Формат ► Шрифт**, **Формат ► Список**, и т.п.

Чтобы создать гиперссылку, нужно сначала выделить фрагмент текста, который будет использоваться как ссылка, и кликнуть на кнопке  **Вставить гиперссылку**, либо использовать команду меню **Вставка ► Гиперссылка**. В открывшемся диалоговом окне **Вставка гиперссылки** на панели **Связать с** можно выбрать тип гиперссылки. Это может быть ссылка на другой файл (кнопка «**файлом, веб-страницей**»), на другое место в этом же документе (кнопка «**местом в документе**»), новую страницу своего сайта (кнопка «**новым документом**»), при этом документ, на который будет указывать ссылка, создается немедленно), или адрес e-mail (кнопка «**электронной почтой**»).

Кнопки **Выбор рамки** и **Закладка** дают возможность указать имя фрейма, в котором будет открываться web-страница, и якорь, который необходимо использовать, соответственно.

Готовая гиперссылка будет отображаться в окне программы SharePoint Designer точно так же, как и в браузере – синим цветом и с подчеркиванием. Для проверки работоспособности гиперссылки нужно кликнуть на ней правой кнопкой мыши и выбрать в контекстном меню команду **По ссылке**.

Для создания таблицы служит кнопка  **Вставить таблицу**, а также команда меню **Таблица ► Вставить таблицу**. При использовании кнопки выводится шаблон, в котором задается число строк и столбцов таблицы. Далее таблица создается, используя заданные по умолчанию атрибуты. Изменить их можно с помощью команды контекстного меню **Свойства таблицы**. В открывшемся диалоговом окне можно увеличить размеры таблицы, а также задать дополнительные свойства, реализуемые как атрибуты соответствующих тегов – выравнива-

ние, поля, границы и т.д. Это же диалоговое окно выводится командой Таблица ► Вставить таблицу.

Чтобы ввести данные в таблицу, нужно установить курсор в выбранную ячейку и начать ввод.

Другие элементы, которые можно разместить на странице, можно найти в меню Вставка ► HTML-код или на расположенной справа Панели элементов (встроенная рамка, горизонтальная линейка, рисунок, слой и т.д.).

Для вставки формы и ее элементов также служит Панель элементов. В списке Элементы управления формы следует выбрать нужный элемент и перетащить его с помощью мыши на создаваемую страницу.

В HTML свойства каждого элемента определяются атрибутами тега. Редактор SharePoint Designer позволяет задавать их с помощью специальных диалоговых окон. Чтобы открыть такое окно, нужно кликнуть на нужном элементе правой кнопкой мыши. В нижней части контекстного меню находятся команды, относящиеся к элементам документа, рассматриваемым редактором как открытые. Порядок следования команд соответствует порядку вложения элементов. Выбрав одну из этих команд, Вы откроете диалоговое окно свойств соответствующего элемента. Элементы управления в этом окне соответствуют атрибутам открывающего тега для выбранного элемента.

Для вставки рисунка служит кнопка  Вставить рисунок из файла, а также команда меню Вставка ► Рисунок ► Из файла. При сохранении документа программа автоматически выполняет преобразование рисунка и его сохранение.

Чтобы вмешаться в процесс автоматического формирования web-страницы и внести изменения непосредственно в генерируемый код HTML, нужно перевести программу в режим Код или С разделением. При этом будет отображаться сгенерированный программой SharePoint Designer код HTML. Выделение цветом позволяет легко находить ключевые слова тегов и их атрибуты. Этот код можно редактировать вручную, однако при этом ответственность за правильность кода переносится с программы SharePoint Designer на создателя страницы.

Редактор SharePoint Designer также дает возможность просмотреть web-страницу в том виде, в каком ее представит браузер. Для этого служит кнопка  Просмотр в (конкретный браузер можно выбрать в раскрывающемся списке).

Публикация web-документов

Для опубликования сайта необходимо разместить документы HTML и все дополнительные файлы (рисунки, мультимедиа и др.) на web-сервере. Если не рассматривать организационные вопросы, такие, как получение дискового пространства на сервере, оплата и др., то остаются две основные проблемы.

– Как подготовить текст web-страниц сайта, чтобы при переносе на сервер не произошло нарушения структуры сайта?

– Как выполнить копирование Ваших файлов на сервер?

Первая задача решается организационными мерами в ходе работы над сайтом. При проектировании будущего сайта на своем компьютере нужно продумать структуру папок и сразу же помещать файлы в соответствующих им папках. Во внутренних гиперссылках необходимо пользоваться только относительными адресами. Тогда при копировании файлов и папок на сервер с сохранением структуры работоспособность гиперссылок сохранится, как и корректность подключения сторонних объектов (рисунков и объектов мультимедиа). При этом упрощается также обновление сайта в целом или его отдельных web-страниц.

Для переноса документов на сервер обычно используется прямое копирование данных через Интернет. Этот способ более надежен, чем передача данных на флеш-накопителе или другом носителе. Чтобы использовать его, нужно знать адреса, используемые для отправки файлов (как правило, для публикации web-документов применяется протокол FTP).

При использовании специализированных программ (например, редактора SharePoint Designer) необходимые инструменты для публикации, как правило, содержит сама программа. Рассмотрение этого вопроса, однако, не входит в круг задач данной лабораторной работы.

Лабораторная работа

Создание простейших web-документов

Упражнение 1

Создание простейшей web-страницы

1. Запустите программу Блокнот (Пуск ► Все программы ► Стандартные ► Блокнот).

2. Наберите текст:

```
<HTML>
<HEAD>
<TITLE>Заголовок документа</TITLE>
</HEAD>
<BODY>
Содержание
документа
</BODY>
</HTML>
```

3. Сохраните введенный текст в своей рабочей папке, задав имя файла first.htm. Проверьте, не добавил ли Блокнот при сохранении файла к его имени расширение .TXT.

4. Откройте файл first.htm в программе Internet Explorer, а затем в каком-либо другом браузере. Для этого откройте свою рабочую папку и кликните на имени файла правой кнопкой мыши. В открывшемся контекстном меню выберите команду Открыть с помощью ► Internet Explorer (либо запустите Internet Explorer из меню Пуск и дайте команду Файл ► Открыть, кликните на кнопке Обзор и найдите в своей рабочей папке файл first.htm).

5. Изучите вид созданной Вами простейшей web-страницы. Где отображается содержимое элементов TITLE и BODY? Обратите внимание на взаимное расположение слов «Содержание» и «документа», которые были введены в отдельных строках. Что произойдёт, если уменьшить ширину окна?

Упражнение 2

Изучение приемов форматирования абзацев

1. Откройте документ first.htm в программе Блокнот.
2. Удалите всё содержимое между тегами <BODY> и </BODY> и установите курсор между ними. Конкретное содержание заголовков и абзацев, которые будут вводиться в следующих пунктах упражнения, может быть произвольным.
3. Задайте заголовок первого уровня, разместив его между тегами <H1> и </H1>.
4. Задайте заголовок второго уровня, его следует разместить между тегами <H2> и </H2>.
5. Создайте первый абзац текста, набрав его после тега <P> (3-5 полных строк). Для этого можно, набрав небольшой фрагмент, скопировать его несколько раз. При этом внутри абзаца можно переходить на новую строку в любом месте.
6. После абзаца введите тег горизонтальной линейки <HR>.
7. Создайте еще один абзац, также начиная его с тега <P>.
8. Сохраните текст, задав имя файла paragraph.htm.
9. Откройте файл paragraph.htm в программе Internet Explorer или другом браузере.
10. Проверьте правильность отображения файла. Имеется ли связь между переходами на новую строку в исходном тексте и видом web-страницы в браузере?

Упражнение 3

Создание гиперссылок

1. Откройте файл first.htm в программе Блокнот.
2. Удалите всё содержимое между тегами <BODY> и </BODY> и установите курсор между ними.
3. Наберите текст: Текст до гиперссылки
4. Через пробел от него в той же строке задайте тег:
5. За ним без пробела в той же строке введите слово: гиперссылка
6. Далее без пробела в той же строке введите закрывающий тег .

7. После него через пробел в той же строке задайте фразу: текст после гиперссылки.

8. Сохраните текст, задав имя файла link.htm.

9. Откройте файл link.htm в программе Internet Explorer или другом браузере.

10. Проверьте правильность отображения гиперссылки – слово между тегами <A> и выделено как ссылка (подчеркиванием и цветом).

11. Кликните на ссылке и убедитесь, что при этом загружается файл first.htm, на который указывает ссылка.

12. Кликните на кнопке  Назад на панели инструментов, чтобы вернуться к предыдущей странице. Проверьте и убедитесь, что ссылка теперь отображается другим цветом, т.е. считается просмотренной.

Упражнение 4

Создание изображения и использование его на web-странице

1. Запустите программу Paint (Пуск ► Все программы ► Стандартные ► Paint). Задайте размеры нового рисунка, например, 50×50 пикселей (меню Paint ► Свойства).

2. Выберите для переднего плана красный цвет и для фона зеленый цвет. Выполните заливку рисунка фоновым цветом.

3. С помощью инструмента Кисть изобразите произвольный красный рисунок на имеющемся зеленом фоне.

4. Сохраните файл под именем pic1.gif (обратите внимание на тип файла в поле ввода под именем – он должен быть GIF, в противном случае вставка рисунка в web-страницу может не произойти).

5. Откройте файл first.htm в программе Блокнот.

6. Удалите всё содержимое между тегами <BODY> и </BODY> и установите курсор между ними.

7. Введите произвольный текст протяженностью 5-6 строк (для этого можно, набрав небольшой фрагмент, скопировать его несколько раз) и установите текстовый курсор в его начало.

8. Перед текстовым фрагментом введите тег

9. Сохраните текст, задав имя файла picture.htm.
10. Откройте файл picture.htm в программе Internet Explorer или другом браузере. Проанализируйте вид получившегося документа.
11. Вернитесь обратно в Блокнот и измените значение атрибута: ALIGN="TOP". Сохраните файл picture.htm.
12. Вернитесь в браузер и щелкните на кнопке Обновить. Как изменился вид страницы в результате изменения атрибутов?
13. Повторите шаги 11-12, изменяя значение атрибута на ALIGN="MIDDLE", ALIGN="BOTTOM", ALIGN="LEFT".
14. Вернитесь обратно в Блокнот и добавьте в конец тега атрибуты: HSPACE=40 VSPACE=20. Сохраните файл picture.htm.
15. Вернитесь в браузер и кликните на кнопке Обновить. Изучите результат применения указанных атрибутов при отображении страницы.

Упражнение 5

Приемы форматирования текста

1. Откройте документ first.htm в программе Блокнот.
2. Удалите всё содержимое между тегами <BODY> и </BODY> и установите курсор между ними. Конкретное содержание текста, который будет вводиться в следующих пунктах этого упражнения, может быть произвольным.
3. Для задания шрифта по умолчанию введите тег:
<BASEFONT SIZE="6" COLOR="BLUE">
Он определяет, что вывод текста по умолчанию будет производиться увеличенным шрифтом и синим цветом. Увидеть результат действия этого тега можно лишь в программе Internet Explorer, другие браузеры его не поддерживают.
4. Начиная с тега <P>, введите произвольный абзац текста, который будет выводиться заданным по умолчанию шрифтом.
5. Введите следующие теги:
<P>.
6. Введите очередной абзац текста, закончив его тегом .

7. В следующем абзаце произвольно используйте парные теги: (полужирный шрифт), <I> (курсив), <U> (подчеркивание), <S> (вычеркивание), <SUB> (нижний индекс), <SUP> (верхний индекс), в том числе, когда теги вложены друг в друга. С их помощью наберите формулы: H_2SO_4 , $x^2 + y^2 = r^2$.

8. В следующем абзаце текста произвольно используйте парные теги: (выделение), (сильное выделение), <CODE> (текст программы), <KBD> (ввод с клавиатуры), <SAMP> (пример вывода), <VAR> (переменная в программе).

9. Сохраните полученный текст под именем format.htm.

10. Откройте файл format.htm в программе Internet Explorer или другом браузере. Изучите, как использованные теги влияют на внешний вид текста.

11. Вернитесь обратно в Блокнот и удалите тег BASE-FONT. Вместо него между тегами <HEAD> и </HEAD> вставьте раздел стиля, например:

```
<HEAD>
  <TITLE>Заголовок документа</TITLE>
  <STYLE>
    BODY{color:blue;font-size:20px;font-family:Calibri;}
  </STYLE>
</HEAD>
```

12. Сохраните файл и обновите web-страницу в браузере. Проверьте правильность отображения текста.

Упражнение 6

Приемы создания списков

1. Откройте файл first.htm в программе Блокнот.
2. Удалите всё содержимое между тегами <BODY> и </BODY> и установите курсор между ними.
3. Введите заголовок первого уровня <H1>Тропические фрукты</H1>.
4. Задайте упорядоченный (нумерованный) список, вставив в текст тег <OL TYPE="I">, который начинает его.
5. Вставьте в текст элементы списка (виды фруктов), начиная каждый из них с тега .

6. Завершите список при помощи тега `</OL>`.
7. Сохраните документ, задав имя файла `list.htm`.
8. Откройте файл `list.htm` в программе Internet Explorer или другом браузере.
9. Проверьте правильность отображения нумерованного списка в браузере. Особое внимание обратите на способ нумерации, который задается атрибутом `TYPE=`. Измените нумерацию с римской на арабскую.
10. Вернитесь обратно в Блокнот и установите курсор после окончания списка. Добавьте тег горизонтальной линейки `<HR>`.
11. Введите заголовок `<H1>Птицы России</H1>`.
12. Задайте неупорядоченный (маркированный) список, введя тег `<UL TYPE="SQUARE">`, который начинает его.
13. Вставьте в текст элементы списка (виды птиц), начиная каждый из них с тега `<LI>`.
14. При помощи тега `</UL>` завершите список. Сохраните документ `list.htm`.
15. Вернитесь в браузер и щелкните на кнопке Обновить. Проверьте правильность отображения списка.
16. Измените вид маркера, заданный при помощи атрибута `TYPE=`, например, `TYPE="CIRCLE"`.
17. Вернитесь обратно в Блокнот и установите курсор после окончания списка. Добавьте тег горизонтальной линейки `<HR>`.
18. Вставьте в текст тег `<DL>`, начинающий список определений.
19. Вставьте в список 3-4 определяемых слова, ставя перед каждым из них тег `<DT>` (например, Вектор, Матрица, Производная, и т.д.). После каждого из слов можно поставить тире.
20. Вставьте после каждого из слов соответствующие определения, начиная их тегом `<DD>`.
21. Завершите список определений тегом `</DL>`. Сохраните документ.

22. Вернитесь в браузер и кликните на кнопке Обновить. Проверьте правильность отображения списка определений на web-странице.

Упражнение 7

Создание таблиц

1. Запустите программу Блокнот и откройте в ней файл first.htm.

2. Удалите всё содержимое между тегами <BODY> и </BODY>, затем установите курсор между ними.

3. Введите тег

```
<TABLE BORDER="10" WIDTH="100%">
```

4. Затем введите строку:

```
<CAPTION ALIGN="TOP">Список телефонов</CAPTION>
```

5. В первой строке таблицы обычно располагаются заголовки столбцов. Задайте её, например, так:

```
<TR BGCOLOR="YELLOW" ALIGN="CENTER">
```

```
<TH>Фамилия<TH>Номер телефона
```

6. Задайте последующие строки таблицы. Каждая из них начинается тегом <TR> и состоит из нескольких ячеек. Каждая ячейка, в свою очередь, начинается тегом <TD>:

```
<TR><TD>Иванов И.И.<TD>212-34-56
```

```
<TR><TD>Петров П.П.<TD>256-34-12
```

7. Аналогично задайте ещё 2-3 строки таблицы.

8. Завершающую строку таблицы можно задать в виде:

```
<TR><TD ALIGN="CENTER" COLSPAN="2">На первом этаже здания имеется бесплатный телефон-автомат.
```

9. Таблица завершается тегом </TABLE>. Затем командой Сохранить как сохраните документ под именем table.htm.

10. Откройте файл table.htm в программе Internet Explorer или другом браузере. Изучите, как в браузере отображается созданная таблица. Особое внимание обратите на влияние её атрибутов на внешний вид таблицы.

11. Измените ширину окна браузера. Как при этом меняется внешний вид таблицы?

Упражнение 8

Использование фреймов

1. Запустите программу Блокнот.

2. Введите следующий текст:

```
<HTML>
<HEAD>
  <TITLE>Описание фреймов</TITLE>
</HEAD>
<FRAMESET ROWS="60%,*">
  <FRAME SRC="table.htm">
  <FRAMESET COLS="35%,65%" NORESIZE>
    <FRAME SRC="first.htm">
    <FRAME SRC="link.htm">
  </FRAMESET>
</FRAMESET>
</HTML>
```

3. Сохраните документ под именем frames.htm.

4. Откройте файл frames.htm в программе Internet Explorer или другом браузере. Как выглядят созданные ранее документы в отдельных фреймах?

5. Изучите, как ведет себя отображаемая браузером web-страница, если изменить ширину окна браузера.

6. Методом перетаскивания мышью определите, можно ли изменять положение границ фреймов.

7. Кликните на ссылке, которая содержится в одном из фреймов, и проверьте, будет ли в нём отображён новый документ.

8. Кликните на кнопке **Назад** на панели инструментов. Убедитесь, что при возвращении к предыдущему документу структура фреймов не нарушается.

9. Вернитесь обратно в программу Блокнот и измените параметры и структуру фреймов, изменив, например, их количество и относительные размеры. Сохраните документ.

10. Вернитесь в браузер и кликните на кнопке **Обновить**. Проверьте, соответствует ли текущий вид web-страницы Вашему замыслу. Если нет, перейдите обратно в Блокнот и исправьте ошибки.

Упражнение 9

Создание web-документа с помощью редактора Microsoft Office SharePoint Designer 2007

1. Запустите программу SharePoint Designer.
2. В автоматически созданной пустой web-странице наберите произвольный текст. Он должен включать заголовок, 2-3 абзаца с отступами первой строки, список (маркированный или нумерованный), гиперссылки, таблицу и иллюстрации.
3. С помощью кнопок панели инструментов отформатируйте текст. Используйте полужирный, курсив, подчёркнутый и другие атрибуты, а также различные шрифты, цвет и размер символов.
4. Чтобы создать таблицу, кликните на кнопке Вставить таблицу .
5. Чтобы добавить иллюстрацию, используйте кнопку  Вставить рисунок из файла. Иллюстрации возьмите по Вашему выбору.
6. Сохраните файл под именем wysiwyg.htm. Подтвердите, если нужно, сохранение изображений, для которых требуется преобразование формата.
7. Откройте файл wysiwyg.htm в программе Internet Explorer или другом браузере. Обратите внимание на различия во внешнем виде документа при просмотре в браузере и в программе SharePoint Designer.
8. С помощью мыши измените ширину окна браузера. Как при этом изменяется вид созданного Вами документа?
9. Перейдите обратно в программу SharePoint Designer и кликните на кнопке HTML-код.
10. Изучив код HTML, автоматически сгенерированный программой, установите, как использованные команды форматирования реализованы с помощью тегов HTML.
11. Закройте программу SharePoint Designer.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Информатика. Базовый курс: Учеб. пособие для втузов / Под ред. С.В. Симоновича. – 3-е изд. Стандарт третьего поколения. – СПб.: Питер, 2016. – 637 с
2. Бурковский А. В. Информатика. Практикум: Учеб. пособие / А. В. Бурковский. – Воронеж: ГОУВПО «Воронежский государственный технический университет», 2006. – 133 с.
3. Ковентри П. Microsoft Office SharePoint Designer 2007 / П. Ковентри. – М.: Эком, 2008. – 544 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
Основные понятия языка HTML	4
Структура документа HTML.....	6
Элементы HTML.....	6
Заголовки и абзацы	7
Гиперссылки	8
Web-графика.....	9
Форматирование текста	12
Списки.....	13
Создание таблиц.....	14
Вывод нескольких документов на одной странице.....	15
Работа в редакторе Microsoft Office SharePoint Designer 2007 ..	17
Публикация web-документов.....	20
Лабораторная работа	21
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	30

СОЗДАНИЕ ПРОСТЕЙШИХ WEB-ДОКУМЕНТОВ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению лабораторной работы по дисциплинам
«Информатика» и «Практикум по информационным
технологиям» для студентов направлений 14.03.01 «Ядерная
энергетика и теплофизика», 16.03.01 «Техническая физика»,
21.03.01 «Нефтегазовое дело», 22.03.02 «Металлургия»
и 28.03.01 «Нанотехнологии и микросистемная техника»
очной и очно-заочной форм обучения

Составители:
Кострюков Сергей Александрович,
Пешков Вадим Вячеславович,
Шунин Геннадий Евгеньевич

Отпечатано в авторской редакции

Подписано к изданию 30.11.2021.

Уч.-изд. л. 2,0

ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский проспект, 14