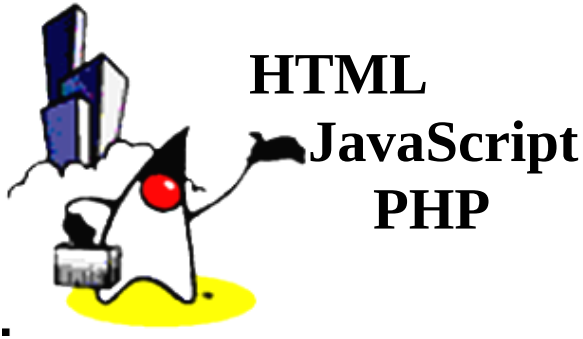
ФГБОУ ВПО “Воронежский государственный технический университет”

Кафедра систем автоматизированного проектирования и информационных систем

WEB ПРОГРАММИРОВАНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам
для студентов направления 09.03.02 «Информационные системы и технологии» (профиль «Информационные системы и технологии цифровизации»)



Воронеж 2019

Составитель канд. техн. наук Е.Н. Королев
УДК 681.3

Web программирование: методические указания к лабораторным работам для студентов направления 09.03.02 «Информационные системы и технологии» (профиль «Информационные системы и технологии цифровизации») сост. Е.Н. Королев. Воронеж, 2019. 20с.

Методические указания посвящены рассмотрению вопросов web программирования с использованием html, javaScript и php.

Библиогр. 2 назв.

Рецензент д-р техн. наук, проф. К.А. Разинкин

Ответственный за выпуск зав. кафедрой д-р техн. наук, проф. Я.Е. Львович

Печатается по решению редакционно-издательского совета Воронежского государственного технического университета

1. ВВЕДЕНИЕ В ЯЗЫК HTML

Документ HTML представляет собой файл, содержащий обыкновенный текст со специальными командами. Такой файл может быть подготовлен в произвольном текстовом редакторе. Хотя существуют специальные программы-конверторы и HTML-редакторы.

Документ HTML состоит из содержимого, то есть собственно полезной информации, и команд, задающих структуру. Каждая команда (управляющая конструкция) HTML документа, называемая «тег», должна заключаться в угловые скобки следующим образом: <тег>. Чаще всего в документе встречаются парные теги (открывающий и соответствующий ему закрывающий), так как браузеру необходимо знать область действия тега. Существуют и одиночные теги, однако, используются они только там, где область действия очевидна и дополнительной информации не требуется (ясно, например, что если мы встретили тег "начало абзаца" (<P>), то предыдущий абзац уже закончился). В сомнительном же случае лучше перестраховаться и поставить закрывающий парный тег, иначе документ может оказаться нечитаемым. Открывающий и закрывающий теги называются одинаково и отличаются друг от друга только символом "наклонная черта" или "слэш" - "/", который ставится сразу после открывающей угловой скобки закрывающего тега. Закрытие парных тегов выполняется так, чтобы соблюдались правила вложения.

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

Кроме того, тег может включать атрибут, дающий дополнительную информацию браузеру. Например, при помощи атрибута можно попросить браузер изменить величину шрифта, ориентацию изображения по отношению к строке следующего за ним текста, поменять цвет фона документа и т. д. В парных тегах атрибуты добавляются только к открывающему тегу.

Атрибуты представляют собой дополнительные ключевые слова, отделяемые от ключевого слова, определяющего тег, и от других атрибутов пробелами и размещаемые до завершающего тег символа ">". Способ применения некоторых атрибутов требует указания значения атрибута. Значение атрибута отделяется от ключевого слова атрибута символом "=" (знак равенства) и заключается в кавычки.

<H1 ALIGN="LEFT">

Язык HTML в большинстве случаев совершено равнодушен к регистру, в котором набираются теги. Скажем, браузеру совершенно все равно, наберете вы тег, служащий для рисования горизонтальной линии, как <HR> или <hr> - эффект будет один и тот же. HTML не признает никакого дополнительного форматирования текста, кроме как с помощью тегов. В результате текст, превосходно смотрящийся в текстовом редакторе, в окне браузера сольется в единую нечитаемую массу. Так, на месте нескольких пробелов будет лишь один пробел. Исчезнут все заголовки, пустые строки, деление текста на абзацы. Без тегов HTML браузер просто игнорирует все элементы форматирования.

Определение HTML как языка разметки основывается на том, что при удалении из документа всех тегов получается текстовый документ, совершенно эквивалентный по содержанию исходному гипертекстовому документу. Таким образом, при отображении документа HTML сами теги не отображаются, но влияют на способ отображения остальной части документа.

Структура HTML-документа

```
<HTML>
<HEAD>
<TITLE> Моя домашняя страница </TITLE>
</HEAD>
<BODY>
</BODY>
```

</HTML>

CSS это язык стилей, определяющий отображение HTML-документов. Например, CSS работает с шрифтами, цветом, полями, строками, высотой, шириной, фоновыми изображениями, позиционированием элементов и многими атрибутами.

HTML используется для структурирования содержимого страницы. CSS используется для форматирования, этого структурированного содержимого.

Конкретные преимущества CSS:

– управление отображением множества документов с помощью одной таблицы стилей;

– более точный контроль над внешним видом страниц;

2. ОСНОВЫ JAVASCRIPT

JavaScript это язык программирования для создания интерактивных веб-страниц. JavaScript встраивается прямо в веб-страницы и исполняется браузером во время их загрузки. JavaScript добавляется на веб-страницы с помощью тэга <script>.

Пример:

<script type="text/javascript">

document.write("Данный текст выведен на страницу с помощью JavaScript.");

</script>

Атрибут type тэга <script> указывает браузеру какой скриптовый язык мы встраиваем. В случае если это JavaScript атрибут type должен иметь значение text/javascript;

Разместив document.write() между тэгами <script> и </script> мы сообщаем браузеру обрабатывать ее как команду JavaScript, поэтому после загрузки страницы браузер выведет: "Данный текст выведен на страницу с помощью JavaScript.".

Чтобы JavaScript код не смешивался с HTML разметкой необходимо размещать его в секции head.

```
<html>
<head>
<script type='text/javascript'>
function example(){
  alert('Если Вы видите это сообщение, значит страница была полностью загружена.');
}
</script>
</head>
<body onload='example()'>
</body>
</html>
```

Если размещать JavaScript в самом конце секции body, то скрипт не начнет выполняться до полной загрузки документа и это не приведет к возможным ошибкам.

JavaScript код необязательно должен непосредственно содержаться в HTML документе, он также может храниться во внешнем текстовом файле с расширением **.js**.

Использовать внешние файлы скриптов удобно в случаях, когда необходимо определять код, который будет работать на нескольких страницах веб-сайта.

Внешние скрипты, также, как и обычные подключаются к страницам с помощью тэга <script> однако в этом случае содержимое тэга должно оставаться пустым и к нему должен быть добавлен атрибут src содержащий адрес внешнего .js файла.

В JavaScript возможно использование разных типов окон.

Окна оповещения используются в случаях, когда необходимо, чтобы пользователь обязательно обратил внимание на определенную информацию.

Когда окно оповещения будет вызвано пользователь должен будет нажать кнопку "OK" для, того чтобы продолжить просмотр страницы.

```
alert("Текст окна оповещения");
```
Пример:
```
<html>
<head>
<script type='text/javascript'>
function example(){
 alert('Если Вы видите это сообщение, значит страница была полностью загружена.');
}
</script>
</head>
<body onload='example()'>
</body>
</html>
```
Окна подтверждения используются в случаях, когда необходимо, чтобы пользователь подтвердил или отклонил что-либо.Когда окно подтверждения будет вызвано пользователь должен будет нажать либо "OK", либо "Отмена", чтобы продолжить. Если пользователь нажмет "OK" вернется true (истина), если пользователь нажмет "Отмена" вернется false (ложь).
```
var x=confirm("Текст окна подтверждения");
```
Пример:
```
<html>
<head>
<script type='text/javascript'>
function popBox(){
  x=confirm("Нажмите на любую кнопку");
  if (x==true){
  document.write('Вы нажали OK');
  }
  else {
    document.write('Вы нажали Отмена.');
  }
}
```

```
</script>
</head>
<body onload='popBox()'>
</body>
</html>
```

Окна запроса используются в случаях, когда от пользователя необходимо получить определенную информацию. Когда окно запроса будет вызвано пользователь должен будет ввести определенные данные и нажать на "OK". Если пользователь не хочет вводить данные он может нажать "Отмена" и окно сразу будет закрыто. Если пользователь введет что-либо в окно и нажмет "OK" будет возвращено введенное пользователем значение, если пользователь нажмет "Отмена", то будет возвращено null.

var x=prompt("Текст окна запроса", "Заполнитель поля ввода");

Пример:

```
<html>
<head>
<script type='text/javascript'>
//Функция будет вызвана после загрузки страницы
function popBox1()
{
  ex1=prompt("Введите Ваше имя:", "Дмитрий");
  document.write('Ваше имя: '+ex1);
}
</script>
</head>
<body onload='popBox1()'>
</body>
</html>
```

3. ЯЗЫК PHP

PHP это язык программирования, специально разработанный для написания web-приложений, то есть сценариев, исполняющихся на Web-сервере.

Аббревиатура PHP означает “Hypertext Preprocessor". Синтаксис языка берет начало из C, Java и Perl. PHP достаточно прост для изучения. Преимуществом PHP является предоставление web-разработчикам возможности быстрого создания динамически генерируемых web-страниц.

Для работы PHP необходимо иметь web-сервер, интерпретатор PHP, субд (mysql), mysqlphpmyadmin. Можно все это отдельно скачать и настроить друг под друга, но проще воспользоваться готовыми сборками такими как XAMPP или Denver.

XAMPP и Denver это пакеты бесплатных кроссплатформенных приложений, который содержит в себе Apache, PHP, MySQL, phpMyAdmin и много других полезных дополнительных утилит.

Приведем пример простейшего скрипта на PHP:

```
<html>
   <head>
      <title>Пример</title>
   </head>
   <body>
      <?
      echo "Привет, я - скрипт PHP!"
        ?>
   </body>
</html>
```

Код сценария начинается после открывающего тэга “<?” и заканчивается закрывающим “?>”. Между этими двумя тэгами текст интерпретируется как программа, и в HTML-документ не попадает. Если же программе нужно что-то вывести, она должна воспользоваться оператором echo. PHP

устроен так, что любой текст, который расположен вне программных блоков, ограниченных **<?** и **?>**, выводится в браузер непосредственно. Каждое выражение заканчивается точкой с запятой.

Имена переменных обозначаются знаком $. То же самое "Привет, я - скрипт PHP! " можно получить следующим образом:

```
<?php
$message = "Привет, я - скрипт PHP!";
echo $message;
?>
```

Имена переменных чувствительны к регистру букв: например, $var — не то же самое, что $Var или $VAR:

Отличительным преимуществом PHP является то, что в PHP не нужно ни описывать переменные явно, ни указывать их тип. Интерпретатор все это делает сам. Однако иногда он может ошибаться (например, если в текстовой строке на самом деле задано десятичное число), поэтому изредка возникает необходимость явно указывать, какой же тип имеет то или иное выражение.

Чуть чаще возникает потребность узнать тип переменной (например, переданной в параметрах функции) прямо во время выполнения программы.

PHP поддерживает восемь простых типов данных:

Четыре скалярных типа:

- boolean (двоичные данные)
- integer (целые числа)
- float (числа с плавающей точкой или 'double' 8 байт) double(8 байт), float(4 байта)
- string (строки)

Два смешанных типа:

- array (массивы)
- object (объекты)

И два специальных типа:

resource (ресурсы)
NULL ("пустые")

Рассмотрим основные функции PHP, применяемые для работы с MySQL сервером.Основной функцией для соединения с сервером MySQL является mysql_connect(), которая подключает скрипт к серверу баз данных MySQL и выполяет авторизацию пользователя базой данных. Синтаксис у данной функции такой:

mysql_connect ([string $hostname] [, string $user] [, sting $password]);

Все параметры данной функции являются необязательными, поскольку значения по умолчанию можно прописать в конфигурационном файле php.ini. При желании можно указать другие имя MySQL-хоста, пользователя и пароль. Параметр $hostname может быть указан в виде: хост:порт.

Функция возвращает идентификатор (типа int) соединения, и вся дальнейшая работа осуществляется только через этот идентификатор. При следующем вызове функции mysql_connect() с теми же параметрами новое соединение не будет открыто, а функция возвратит идентификатор существующего соединения.

Для закрытия соединения предназначена функция mysql_close(int $connection_id).

Вообще, соединение можно и не закрывать - оно будет закрыто автоматически при завершении работы PHP скрипта. Если используется более одного соединения, при вызове mysql_close() нужно указать идентификатор соединения, которое вы хотите закрыть. Вообще не закрывать соединения - плохой стиль, лучше закрывать соединения с MySQL самостоятельно, а не надеясь на автоматизм PHP.

Если используется только одно соединение с базой данных MySQL за все время работы сценария, то можно не сохранять его идентификатор и не указывать идентификатор при вызове остальных функций.

Функция mysql_connect() устанавливает обыкновенное соединение с MySQL. Однако, PHP поддерживает постоянные соединения - для этого используется функция mysql_pconnect(). Аргументы этой функции такие же, как и у mysql_connect().

В чем разница между постоянным соединением и обыкновенным соединением с MySQL? Постоянное соединение не закрывается после завершения работы скрипта, даже если скрипт вызвал функцию mysql_close(). Соединение закрывается лишь тогда, когда удаляется процесс-владелец (например, при завершении работы или перезагрузке веб-сервера Apache).

PHP работает с постоянными соединениями следующим образом: при вызове функции mysql_pconnect() PHP проверяет, было ли ранее установлено соединение. Если да, то возвращается его идентификатор, а если нет, то открывается новое соединение и возвращается идентификатор.

Постоянные соединения позволяют значительно снизить нагрузку на сервер, а также повысить скорость работы PHP скриптов, использующих базы данных.

При работе с постоянными соединениями нужно следить, чтобы максимальное число клиентов Apache не преывшало максимального числа клиентов MySQL, то есть параметр MaxClient (в конфигурационном файле Apache - httpd.conf) должен быть меньше или равен параметру max_user_connection (параметр MySQL).

Функция выбора базы данных

Функция mysql_select_db (string $db [, int $id]) выбирает базу данных, с которой будет работать PHP скрипт. Если открыто не более одного соединения, можно не указывать параметр $id.

Все запросы к текущей базе данных отправляются функцией mysql_query(). Этой функции нужно передать один параметр - текст запроса. Текст запроса может содержать пробельные символы и символы новой строки (\n). Текст

должен быть составлен по правилам синтаксиса SQL. Пример запроса:

```
$q = mysql_query("SELECT * FROM mytable");
```

Приведенный запрос должен вернуть содержимое таблицы mytable. Результат запроса присваивается переменной $q. Результат - это набор данных, который после выполнения запроса нужно обработать определенным образом.

Есть возможность узнать значение каждого поля. Это можно сделать с помощью следующей функции:

```
mysql_result (int $result, int $row, mixed $field);
```

Параметр функции $row задает номер записи, а параметр $field - имя или порядковый номер поля.

4. ЗАДАНИЯ НА ЛАБОРАТОРНЫЕ РАБОТЫ

1. Лабораторная работа №1. Разработка личной страницы с использованием HTML5.
2. Лабораторная работа №2. Разработка личной страницы с использованием HTML5 и CSS.
3. Лабораторная работа №3. Разработка личной страницы с использованием HTML5 и CSS и JavaScript.
4. Лабораторная работа №4. Разработка личной страницы с использованием PHP.
5. Лабораторная работа №5. Разработка личной страницы с использованием ООП в PHP.

Лабораторная работа №6. Разработка личной страницы с использованием ООП в PHP и хранением и обработкой данных в MySql.

Задание на лабораторную работу № 1

Цель: освоить главные понятия и средства языка HTML.

Задание на лабораторную работу: Создать документ HTML, представляющий собой резюме.

Требования к выполнению лабораторной работы № 1

1. Используйте тег <TITLE> для описания заголовка.
2. В тег <BODY> вставить атрибуты TEXT и BGCOLOR.
3. Используйте несколько разделов. Например, разделами могут быть: учеба, увлечения, адрес. Разделы отделить друг от друга тегом разрыва строки и горизонтальной линией. Название разделов выделить жирным шрифтом.
4. Оформите линию нестандартным способом. Для этого следует внести в тег <HR> атрибуты длины, толщины, цвета и выравнивания линии. Например, задайте длину 50 % от ширины окна, выравнивание по центру, толщину 5 пикселей, зеленым цветом.
5. Использовать заголовки разного уровня.
6. Использовать форматирование шрифта. Измените вид шрифтов в разных разделах резюме. Для этого используйте тег <FONT> с атрибутами SIZE, COLOR, FACE. Используйте теги <b>, <i>, <s>, <u>.
7. Использовать тег абзаца <p>
8. Оформить часть текста в виде списка. Для этого следует контейнером <OL> выделить блок (например, об увлечениях), разбить его на пункты тегами <LI>. Аналогично контейнером <UL> и тегами <LI> оформить в виде маркированного списка другой блок (например, перечень друзей).
9. Информацию о родственниках оформить в виде таблицы. Количество столбцов в таблице должно быть не менее 5. Количество строк не менее 3. Использовать объединение ячеек.

Для задания названий столбцам используйте теги <TH> … </TH>

Пример:

<TR>
<TH> Фамилия </TH>
<TH> Имя </TH>
<TH> Отчество </TH> <TH> Возраст </TH>
</TR>

Для заполнения содержания таблицы используйте теги <TD> … </TD>. Чтобы объединить ячейки, добавьте атрибуты ROWSPAN. Пример:

```
<TR>
<TD ROWSPAN=3> Иванов </TD>
<TD ROWSPAN=3> Иван</TD>
<TD> Иванович </TD> <TD> 50 </TD>
</TR>
```

10. Добавьте в тег <TABLE> атрибуты BORDER, WIDTH, ALIGN, BGCOLOR, BORDERCOLOR, CELLSPACING.

11. В открывающий тег какой-нибудь ячейки таблицы вставить атрибут BACKGROUND="имя файла". Пример:

```
<TD BACKGROUND="море.jpg">
```

12. Вставьте в резюме свою фотографию с помощью тега <IMG SRC="…"> (вместо многоточия укажите URL файла с фотографией)

13. Добавить гиперссылку на свой почтовый адрес.

Пример: <A HREF="mailto://adm@mail.ru"> e-mail </A>

14. Добавить ссылки на 2 Html файла с дополнительной информацией. При этом необходимо использовать тег <a> с параметрами href и name.

15. Разделить окно на 3 фрейма.

Пример:

```
<HTML>
<HEAD>
<TITLE> Начальный документ сайта… (Ваша фамилия)
</TITLE>
</HEAD>
<FRAMESET ROWS=20%,*>
<FRAME SRC="Заголовок.html">
<FRAMESET COLS=30%,*>
<FRAME SRC="Резюме.html">
<FRAME SRC="Приветствие.html">
</FRAMESET>
</FRAMESET>
```

</HTML>

Для создания файла Html-файла в среде NetBeans необходимо выбрать элемент меню File, затем New File. В появившемся окне, во вкладке Categories выбрать элемент Other, после чего во вкладке File Types выбрать тип файла: Html File и нажать кнопку Next.

Задание на лабораторную работу № 2

Цель: ознакомление с базовым синтаксисом и основными элементами CSS–документа, научиться создавать и применять таблицы стилей для управления представлением содержимого web-страниц.

Задание на лабораторную работу: Использовать применение CSS к созданным Вами HTML-документам.

Требования к выполнению лабораторной работы № 2

1. Для 3 разных html-файлов использовать разные методы применения SCC-стилей (атрибут style, тэг style, ссылку на таблицу стилей);
2. Настройки стилей для основного html-файла должны размещаться в отдельном файле (например, styles.css). В коде этой страницы необходимо исключить теги <b>, <i>, <s>, <u>, <font>. Соответствующие стили должны быть реализованы через CSS; Для чего необходимо создать CSS файл и подключить его ко всем страницам сайта с помощью тега <LINK>.
3. Задайте по умолчанию следующие параметры для страницы переопределив, тег <body>):

- цвет фона;
- размер шрифта;
- цвет шрифта;
- семейство шрифта (например, Arial).

4. Попробуйте использование универсального и тегового класса. Используйте псевдоклассы. Задайте по умолчанию следующие свойства ссылок для всех страниц:

- цвет и оформление ссылки;
- цвет и оформление посещенной ссылки;
- цвет и оформление активной ссылки;
- цвет и оформление ссылки, в момент нахождения курсора мыши над ней.

В комментариях (в файле css) поясните эти параметры.

Для создания файла CSS в среде NetBeans необходимо выбрать элемент меню File, затем New File. В появившемся окне, во вкладке Categories выбрать элемент Other, после чего во вкладке File Types выбрать тип файла: Cascading Style Sheet и нажать кнопку Next.

Задание на лабораторную работу № 3

Цель: ознакомиться с базовым синтаксисом и основными возможностями управления содержимым web-страницы на стороне клиента с помощью JavaScript;

Задание на лабораторную работу:

Создать документ HTML, с обработкой данных формы с помощью JavaScript.

Требования к выполнению лабораторной работы № 3

– создать форму для регистрации пользователя;

– для регистрации необходимо ввести фамилию, имя, отчество, выбрать пол, ввести логин и пароль, email;

– обеспечить проверку правильности заполнения полей формы: логин должен быть не менее 7 символов и включать как буквы, так и цифры; email должен содержать символ @. пароль должен быть не менее 8 символов и должен включать как буквы, так и цифры;

– выводить информацию о разработчике в отдельном окне по нажатию на соответствующую кнопку. В этом окне создать кнопку закрывающую данное окно. При этом использовать объект window.

Задание на лабораторную работу № 4

Цель: изучить работу с окнами в web приложении с помощью JavaScript и изучить обработку событий;

Задание на лабораторную работу:

Создать документ HTML, с реализацией окон и обработкой событий с помощью JavaScript.

Требования к выполнению лабораторной работы № 4

– реализовать окна оповещения, подтверждения и запроса;

– реализовать обработку событий, при этом создать обработчики не менее 3 разных событий;

- после успешной регистрации в отдельном окне запускать видео с кнопкой приостановки/возобновления;
- JavaScript вынести во внешний текстовый файл с расширением **.js**.

Задание на лабораторную работу № 5

Цель: освоить главные понятия и средства языка PHP.

Задание на лабораторную работу: Установить Denver. Создать простейшие программы на PHP.

Требования к выполнению лабораторной работы № 5

– создайте PHP-страницу приветствия от Вашего имени.

– создайте счетчик посещаемости на вашей PHP-странице, значение которого должно храниться в файле и отображаться на странице в специально отведенной области.

– для каждого десятого посещения на странице должна выводиться соответствующая надпись.

– отображать дату и время загрузки страницы.

Задание на лабораторную работу № 6

Цель: освоить обработку данных формы.

Задание на лабораторную работу: Написать программу обработки данных анкеты.

Требования к выполнению лабораторной работы № 6

– Создать два файла. Первый (main.html) предоставляет возможность заполнения анкеты, а второй (visual.php) формирование и отображение ее на экране.

– Html файл должен содержать основные элементы html формы: однострочные поля ввода, поле ввода пароля, зависимые и независимые переключатели, кнопку сброса параметров, многострочное текстовое поле, список выбора.

– Обеспечить возможность прикрепления к анкете файла резюме и отправки его на сервер.

Задание на лабораторную работу № 7

Цель: освоить работу с MySql в PHP.

Задание на лабораторную работу: Обеспечить сохранение данных анкетирования полученных в лабораторной работе №6 в базе данных. Предварительно спроектировать базу данных.

Требования к выполнению лабораторной работы № 7

– После отображения содержимого анкеты файлом visual.php, предложить возможность сохранения анкеты в базе данных.

– Обеспечить возможность отображения ранее сохраненной в базе данных анкеты.

Задание на лабораторную работу № 8

Цель: освоить отображение таблицы из базы данных с помощью PHP на Web-странице.

Задание на лабораторную работу: Отобразить содержимое таблицы базы данных в Web-странице не зная структуры таблицы.

Требования к выполнению лабораторной работы № 8

– Обеспечить ввод имени таблицы из базы данных.
– Обеспечить вывод содержимого выбранной таблицы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Орлов, А.В. PHP. Полезные приемы [Текст] / А.А. Орлов. - М. Горячая линия – Телеком, 2013. – 272 с.

2. Котеров, Д. PHP 5 [Текст] / Д. Котеров, А. Костарев. – Спб.: БХВ- Петербург, 2014. – 1104 с.

СОДЕРЖАНИЕ

WEB ПРОГРАММИРОВАНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторным работам № 1-8 по курсам
“Web-дизайн” и “Web-ориентированное программирование”
для студентов направлений 230400.62 «Информационные системы и технологии» (профиль «Информационные системы и технологии») и 230100.62 «Информатика и вычислительная техника» (профиль «Системы автоматизированного проектирования») очной формы обучения

Составитель
Королев Евгений Николаевич

В авторской редакции

Подписано в печать 18.11.2014.

20

Формат 60х84/16. Бумага для множительных аппаратов.
Усл. печ. л. 1,3. Уч-изд. л. 1,1 Тираж 30 экз.
“С” Заказ №

ФГБОУ ВПО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14