


**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Воронежский государственный технический университет»
в городе Борисоглебске



УТВЕРЖДАЮ

Директор филиала

 /В.В. Григораш/
31 августа 2021 г.

**РАБОЧАЯ ПРОГРАММА
дисциплины**

«Web программирование»

Направление подготовки 09.03.02 Информационные системы и технологии

Профиль Информационные системы и технологии цифровизации

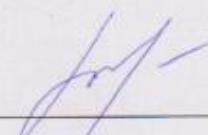
Квалификация выпускника бакалавр

Нормативный период обучения 4 г

Форма обучения очная

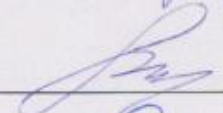
Год начала подготовки 2021

Автор программы



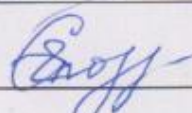
Е.Н. Королев

Заведующий кафедрой
естественнонаучных дисциплин



Л.И. Матвеева

Руководитель ОПОП



Е.А. Позднова

Борисоглебск 2021

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Целью дисциплины является изучение основных принципов и технологий web программирования. Изучение принципов и особенностей построения серверных и клиентских приложений, двух и трехзвенных архитектур построения сетевых приложений процессами. Кроме того, задачей курса является изучение идеологии и архитектуры современных распределенных систем, а также получения навыков написания клиент-серверных приложений с различной архитектурой и с использованием различных современных протоколов.

1.2. Задачи освоения дисциплины

- изучение принципов web программирования с использованием протоколов TCP и UDP;
- приобретение навыков работы в современных средах разработки web приложений;
- приобретение навыков разработки сетевых приложений, как клиентской, так и серверной части;
- изучение языка программирования PHP;
- изучение JavaScript.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Web программирование» относится к дисциплинам части, формируемой участниками образовательных отношений блока Б.1 учебного плана.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Web программирование» направлен на формирование следующих компетенций:

ПК-1 - Способен выполнять работы по созданию (модификации) и сопровождению информационных систем и ресурсов для различных прикладных областей

ПК-2 - Способен выполнять проектирование информационных систем и ресурсов для различных прикладных областей

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-1	знать синтаксис и основные возможности языка php и JavaScript
	уметь разрабатывать приложения с использованием php и JavaScript при создании (модификации) и сопровождении систем, автоматизирующих задачи организационного управления и бизнес-процессы

	владеть способностью сопровождения систем, автоматизирующих задачи организационного управления и бизнес-процессы реализованных с использованием html, css, javascript, php
ПК-2	знать методы проектирования информационных систем среднего и большого уровня сложности
	уметь выполнять проектирование автоматизированных систем среднего и большого уровня сложности
	владеть инструментальными средствами, методами и навыками проектирования и разработки web приложений с использованием возможностей javaScript и php

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Web программирование» составляет 5 зачетных единиц.

Распределение трудоемкости дисциплины по видам занятий

Очная форма обучения

Вид учебной работы	Всего часов	Семестры
		6
Аудиторные занятия (всего)	90	90
В том числе:		
Лекции	36	36
Практические занятия (ПЗ)	-	-
Лабораторные работы (ЛР), в том числе в форме практической подготовки	54 16	54 16
Самостоятельная работа	54	54
Курсовой проект (работа)	+	+
Контрольная работа	-	-
Вид промежуточной аттестации (Экзамен)	36	36
Общая трудоемкость	час зач. ед.	180 5
		180 5

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	HTML	Основные теги. HTML5	6	10	8	24
2	CSS	Свойства CSS, классы, универсальные, теговые и псевдоклассы. CSS3	6	6	8	24
		Практическая подготовка обучающихся		4		
3	JavaScript	Основы JavaScript. Включения, команды, комментарии, переменные. Операторы, выражения конструкции. Объекты в JavaScript, свойства объектов, методы объектов.	6	6	8	24
		Практическая подготовка обучающихся		4		
4	Язык PHP	Возможности и особенности PHP, синтаксис PHP, управляющие конструкции	6	8	10	24
5	Объектно-ориентированное программирование в PHP	Работа с классами и объектами в PHP	6	4	10	24
		Практическая подготовка обучающихся		4		
6	PHP, работа с MySQL	Функции соединения с сервером MySQL, обработка запросов, обработка ошибок	6	4	10	24
		Практическая подготовка обучающихся		4		
Итого			36	54	54	144

Практическая подготовка при освоении дисциплины (модуля) проводится путем непосредственного выполнения обучающимися отдельных элементов работ, связанных с будущей профессиональной деятельностью, способствующих формированию, закреплению и развитию практических навыков и компетенций по профилю соответствующей образовательной программы на практических занятиях и (или) лабораторных работах:

№ п/п	Перечень выполняемых обучающимися отдельных элементов работ, связанных с будущей профессиональной деятельностью	Формируемые профессиональные компетенции
1	разработка приложения с использованием php и JavaScript при создании (модификации) и сопровождении систем по варианту, выданного преподавателем	ПК-1
2	выполнять проектирование автоматизированной системы среднего и большого уровня сложности с использованием инструментальных средств, и навыков проектирования и разработки web приложений с использованием возможностей javascript и php	ПК-2

5.1 Перечень лабораторных работ

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.

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

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 6 семестре для очной формы обучения.

Примерная тематика курсового проекта: «Разработка Web-ориентированной информационной системы обработки заказов с использованием CSS, JavaScript, PHP и MySQL»

Задачи, решаемые при выполнении курсового проекта:

- Анализ предметной области.
- Проектирование информационной системы и ресурсов для выбранной предметной области.
- Разработка компонент информационной системы.

Курсовой проект включает в себя графическую часть и расчетно-пояснительную записку.

Учебным планом по дисциплине «Web программирование» не предусмотрено выполнение контрольной работы (контрольных работ) в 6 семестре.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-1	знать синтаксис и основные возможности языка php и JavaScript	Знание основных конструкций языка php и javascript	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	уметь разрабатывать приложения с использованием php и JavaScript при создании (модификации) и сопровождении систем, автоматизирующих задачи организационного управления и бизнес-процессы	Умение пользоваться средой и возможностями языка при разработке web приложений	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть способностью сопровождения систем, автоматизирующих задачи организационного управления и бизнес-процессы реализованных с использованием html, css, javascript, php	Владение способностью настроить среду программирования для решения поставленной задачи	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-2	знать методы проектирования автоматизированных систем среднего и большого уровня сложности	Знание методов проектирования web приложения	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь выполнять проектирование автоматизированных систем среднего и большого уровня сложности	Умение пользоваться средой и возможностями языка при решении конкретной задачи выбранной предметной области	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть инструментальными средствами, методами и навыками проектирования и разработки web приложений с использованием возможностей javascript и php	Владение инструментальными средствами, методами и навыками разработки web приложений с использованием возможностей javascript и php	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 6 семестре для очной формы обучения по четырехбалльной системе:

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ПК-1	знать синтаксис и основные возможности языка php и JavaScript	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов

	уметь разрабатывать приложения с использованием php и JavaScript при создании (модификации) и сопровождении систем, автоматизирующих задачи организационного управления и бизнес-процессы	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть способностью сопровождения систем, автоматизирующих задачи организационного управления и бизнес-процессы реализованных с использованием html, css, javascript, php	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-2	знать методы проектирования автоматизированных систем среднего и большого уровня сложности	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь выполнять проектирование автоматизированных систем среднего и большого уровня сложности	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть инструментальными средствами, методами и навыками проектирования и разработки web приложений с использованием возможностей javascript и php	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Вложенный список в HTML

1 вариант

```
<ul>
  <li>1
</ul>
```

```
<li>1.1</li>
<li>1.2</li>
</ul>
</li>
<li>2</li>
</ul>
```

2 вариант

```
<ul>
<li>1</li>
<ul>
<li>1.1</li>
<li>1.2</li>
</ul>
<li>2</li>
</ul>
```

2. Задаёт области картинки, которые могут стать ссылками. Описывается внутри тега `<map>`. Картинки с такими ссылками называются «картой изображений».

1. `<area>`
2. `<map>`
3. `<form>`
4. `<head>`

3. Элемент описывает заголовок первого уровня

1. `<h1>`
2. `<isindex>`
3. ``
4. `<h2>`

4. Элемент ... служит для отображения информации об авторе. Он может включать в себя любые теги HTML. По умолчанию текст внутри этого элемента отображается курсивным начертанием.

1. `<address>`
2. `<canvas>`
3. `</address>`
4. `<dd>`

5. Определяет тип страницы. Такой элемент необходимо задавать самым первым в начале файла, чтобы браузер знал, как правильно отображать данные (в соответствии с какими стандартами).

1. `<!DOCTYPE>`
2. `<article>`
3. `<aside>`
4. `<!-->`

6. С помощью тега ... можно создавать ссылки и указывать места на странице, куда именно необходимо осуществить переход (якорь).

1. **<a>**
2. **
**
3. **<div>**
4. **<colgroup>**

7. Этот элемент указывает «подвал» сайта. Зачастую используется с тегами **<header>** и **<article>**.

1. **<footer>**
2. **<header>**
3. **<article>**
4. **<form>**

8. Позволяет группировать заголовки страницы или раздела

1. **<hgroup>**
2. **<marquee>**
3. ****
4. **<pre>**

9. Тег ... задает связи страницы с другими файлами. Размещается внутри контейнера **<head>**.

1. **<link>**
2. **<thead>**
3. **<tfoot>**
4. **<style>**

10. Определяет отдельный элемент в списке

1. ****
2. **<style>**
3. **<th>**
4. **<nav>**

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Какова длина `a.length` массива `a` ?

```
var a = []  
a[1] = 5  
a[5] = 55
```

1. **6**
2. **1**
3. **3**
4. **Другое**

2. Из родительского окна во фрейм child> поставили переменную:
frames.child.users = ["Маша", "Паша", "Даша"]

1. **if (users instanceof parent.Array)**
2. `if (users instanceof Array)`
3. `if (users.constructor.name == "Array")`
4. `if(typeof users == "array")`

3. Что делает оператор ===?

1. **Сравнивает без приведения типа**
2. Сравнивает по ссылке, а не по значению
3. Нет такого оператора

4. Каких бинарных операторов НЕТ в javascript?

1. #
2. !
3. *
4. ^
5. %
6. &
7. >>
8. >>>

5. Какая арифметическая операция приводит к ошибке в javascript?

1. **Никакая из вышеперечисленных**
2. Деление на ноль
3. Умножение числа на строку
4. Корень из отрицательного числа

6. Чему равно a+b+c?

```
var a = 1  
var b = { toString:function() {return '1'} }  
var c = Object(1)
```

1. **111**
2. 11[object Object]
3. 2[object Object]
4. 55

7. Чему равна переменная day?

```
var day = parseInt("08.01.2008".split('.')[0])
```

- 1.0
- 2.2
- 3.4
- 4.6

8. Какое из этих ключевых слов ООП не используется в javascript (1.5) ?

1. **super**
2. **this**
3. **instanceOf**
4. **new**
5. Все есть

9. Как правильно вывести приветствие через 5 секунд после запуска скрипта?

1. **setTimeout('alert("Привет!")', 5000);**
2. **setTimeout(function() {alert("Привет!")}, 5000);**
3. **sleep(5); alert("Привет!");**
4. **sleep(5000); alert("Привет!");**
5. **setTimeout(function() {alert("Привет!")}, 5);** 10. Какие из этих вызовов корректно бросят исключение ?

1. **throw "Ошибка"**
2. **throw new Error("Ошибка")**
3. **throw { message: "Ошибка" }**
4. **throw Error("Ошибка")**
5. Ни один

7.2.3 Примерный перечень заданий для решения прикладных задач

1. При попытке выполнения кода возникла ошибка. С чем она связана?

```
<?php  
echo "Привет";  
session_start();  
?>
```

1. **Нельзя выводить информацию до начала сессии.**
2. Функции `session_start()` не существует, правильная функция – `start_session()`.
3. После начала сессии необходимо её использовать, а в коде она никак не используется.
4. В этом коде нет ошибок.

2. Какая разница между равенством и эквивалентностью в php?

1. **Знак равенства проверяет лишь значения операндов, а знак эквивалентности значения и их типы.**
2. **Никакой разницы нет.**

3. Знак эквивалентности проверяет лишь значения операндов, а знак равенства значения и их типы.

4. Знак эквивалентности работает лишь для строк, а знак равенства применим к любым типам.

3. Какой результат работы кода:

```
<?php
for ($i = 0; $i < 5; $i++) {
if ($i % 2 == 0) continue;
echo $i;
}
?>
```

1. 13

2. 024

3. 013

4. 24

4. Какая функция в PHP позволяет найти вхождение подстроки?

1. **strpos()**

2. str_repeat()

3. strtok()

4. substr()

5. Какой результат работы кода:

```
<?php
function myfunc(&$a) {
$a++;
}
$b = 5;
myfunc($b);
echo $b;
?>
```

1. 6

2. Ошибку из-за лишнего символа “&” перед именем переменной в функции.

3. 56

4. 5

6. Как сделать редирект (например, на google.ru) на PHP?

1. **header(“Location: http://google.ru”);**

2. document.location = “http://google.ru”;

3. location.href = “http://google.ru”;

4. header(“Redirect: http://google.ru”);

7. Что будет выведено в результате работы скрипта?

```
interface foo {}
class_alias('foo', 'bar');
echo interface_exists('bar') ? 'yes' : 'no';
```

1. yes
2. NULL
3. no
4. Ошибка

8. Что будет выведено в результате работы скрипта?

```
class number
{
    public $a= 10;
    public $b=20;
    private $c=30;
}
$numbers = new number();

foreach($numbers as $var => $value)
{
    echo "$value ";
}
```

1. 10 20
2. Ошибка
3. 10 20 0
4. 10 20 30

9. Что будет выведено в результате работы скрипта?

```
class A {}
class B1 extends A {};
class_alias('A', 'B2');
$b1 = new B1; echo get_class($b1);
$b2 = new B2; echo get_class($b2);
```

1. B1A
2. B2A
3. B1B2
4. Ошибка

10. Что будет выведено в результате работы скрипта?

```
class Object
{
    function Object( $entity )
    {
        $entity->name="John";
    }
}
```

```

}
class Entity
{
    var $name = "Maria";
}
$entity = new Entity();
$obj = new Object( $entity );
print $entity->name ;

```

1. John

2. JohnMaria

3. Maria

4. Ошибка

7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом

7.2.5 Примерный перечень вопросов для подготовки к экзамену

1. HTML, структура. Теги head, body, b, d, u, hr, font. Описание, синтаксис, атрибуты.
2. HTML, структура. Теги a, ul, ol, input. Описание, синтаксис, атрибуты.
3. HTML, структура. Теги img, button, form, textarea, table. Описание, синтаксис, атрибуты.
4. HTML, структура. Теги head, body, b, d, u, hr, font. Описание, синтаксис, атрибуты.
5. Особенности HTML5. Проверка форм.
6. CSS, назначение. Способы применения.
7. Использование универсальных, теговых и псевдоклассов в CSS/
8. Основы JavaScript. Включения, команды, комментарии, переменные.
9. Операторы, выражения, конструкции JavaScript.
10. Функции, циклы, события JavaScript.
11. Объекты в JavaScript, свойства объектов, методы объектов.
12. Организация массивов в JavaScript. Создание массивов, доступ к переменным. Свойства объекта Array, методы Array. Работа с объектами в JavaScript.
13. Основы PHP, основные возможности, особенности, преимущества.
14. Типы данных в PHP. Выражения в PHP. Операторы PHP. Управляющие конструкции языка PHP. Управляющие конструкции языка PHP. Встроенные (стандартные) функции PHP
15. Стандартные функции языка PHP. Функции обработки строк. Функции для работы с массивами. Функции для работы с каталогами. Функции для работы с файловой системой. Математические функции. Функции протокола HTTP. Функции URL. Функции сокетов. Сетевые функции.
16. Типы данных PHP. Скалярные типы данных. Смешанные типы данных. Специальные типы данных. Псевдотипы данных. Манипуляции с типами данных.
17. Классы и объекты в PHP. Наследование классов. Полиморфизм классов. Работа с объектами. Стандартные функции языка PHP.

18. Пользовательские функции языка PHP. Функции языка PHP для работы с MySQL.

19. Функции языка PHP для работы с MySQL. Соединение с сервером, выбор базы данных, выполнение запросов.

20. Функции языка PHP для работы с MySQL. Функции обработки запросов.

21. Работа с сессиями в php.

22. Обработка данных формы в php.

23. Загрузка файлов на сервер в php.

24. Основы технологии AJAX. Основы технологии AJAX, преимущества, недостатки. Вызов AJAX.

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 6 до 10 баллов

3. Оценка «Хорошо» ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» ставится, если студент набрал от 16 до 20 баллов.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	HTML	ПК-1, ПК-2	Тест, защита лабораторных работ, требования к курсовому проекту.
2	CSS	ПК-1, ПК-2	Тест, защита лабораторных работ, требования к курсовому проекту.
3	JavaScript	ПК-1, ПК-2	Тест, защита лабораторных работ, требования к курсовому проекту.
4	Язык PHP	ПК-1, ПК-2	Тест, защита лабораторных работ, требования к курсовому проекту.
5	Объектно-ориентированное программирование в PHP	ПК-1, ПК-2	Тест, защита лабораторных работ, требования к курсовому проекту.

6	PHP, работа с MySQL	ПК-1, ПК-2	Тест, защита лабораторных работ, требования к курсовому проекту.
---	---------------------	------------	--

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методике выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Зудилова, Т. В. Web-программирование HTML / Т. В. Зудилова, М. Л. Буркова. — Санкт-Петербург : Университет ИТМО, 2012. — 70 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/65748.html> — Режим доступа: для авторизир. пользователей

2. Кириченко А.В. Динамические сайты на HTML, CSS, Javascript И Bootstrap. Практика, практика и только практика [Электронный ресурс]/ Кириченко А.В., Дубовик Е.В.— Электрон. текстовые данные.— СПб.: Наука и Техника, 2018.— 272 с.— Режим доступа:<http://www.iprbookshop.ru/77578.html>.— ЭБС «IPRbooks». 2018

3. Основы JavaScript : учебное пособие. — 2-е изд. — Москва : ИНТУИТ, 2016. — 184 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100300> . — Режим доступа: для авториз. пользователей.

4. Джо, Б. Практикум по программированию на JavaScript : учебное пособие / Б. Джо. — 2-е изд. — Москва : ИНТУИТ, 2016. — 160 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100380> . — Режим доступа: для авториз. пользователей.

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

Перечень ПО, включая перечень лицензионного программного обеспечения:

Microsoft Office 64-bit;
ОС Windows 7 Pro;
Mozilla Firefox 81.0 (x64 ru);
Google Chrome;
Microsoft .NET Framework;
Notepad++;
Android Studio 3.5

Ресурсы информационно-телекоммуникационной сети «Интернет»:

<http://window.edu.ru>

<http://www.edu.ru/>

Образовательный портал ВГТУ

Современные профессиональные базы данных и информационные справочные системы:

<http://htmlbook.ru/html/img>

<https://habr.com/ru/post/247495/>

<https://msiter.ru/articles/sayt-na-php-eto-prosto>

<http://uchitel-program.ru/web/testy/php/>

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Для проведения обучения по дисциплине используется компьютерный класс.

Компьютерный класс оснащен:

- персональные компьютеры с установленным ПО, подключенные к сети Интернет (11 шт.);
- принтер цветной лазерный;
- доска магнитно-маркерная поворотная.

Помещение для самостоятельной работы. Читальный зал с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

По дисциплине «Web программирование» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, защитой курсового проекта.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: <ul style="list-style-type: none">- работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций;- выполнение домашних заданий и расчетов;- работа над темами для самостоятельного изучения;- участие в работе студенческих научных конференций, олимпиад;- подготовка к промежуточной аттестации.

Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.
---------------------------------------	---

