

МИНПРОСВЕЩЕНИЯ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования  
"Тульский государственный педагогический университет им. Л.Н. Толстого"  
(ФГБОУ ВО "ТГПУ им. Л.Н. Толстого")

## Технологии и практикум веб-программирования

### рабочая программа дисциплины (модуля)

Закреплена за кафедрой	<b>информатики и информационных технологий</b>
ОПОП	<b>02.03.02 Фундаментальная информатика и информационные технологии направленность (профиль) Открытые информационные системы</b>
Квалификация	<b>Бакалавр</b>
Год начала подготовки	<b>2019</b>
Форма обучения	<b>очная</b>
Общая трудоемкость	<b>8 з.е.</b>

Виды контроля по семестрам:

экзамен 4

зачет 3

Семестр(Курс.Номер семестра на курсе)	3(2.1)		4(2.2)		Итого	
	УП	РПД	УП	РПД	УП	РПД
Лекции	18	18	22	22	40	40
Лабораторные	22	22	32	32	54	54
Итого ауд.	40	40	54	54	94	94
КСР	2	2	2	2	4	4
Контактная работа	42	42	56	56	98	98
Сам. работа	66	66	88	88	154	154
Часы на контроль	0	0	36	36	36	36
Итого трудоемкость в часах	108	108	180	180	288	288

Программу составил(и):

*к.т.н., доцент, Клепиков Алексей Константинович*

Рабочая программа дисциплины

**Технологии и практикум веб-программирования**

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 02.03.02  
Фундаментальная информатика и информационные технологии (уровень бакалавриата) (приказ Минобрнауки России от  
23.08.2017г. №808)

составлена на основании учебного плана:

02.03.02 Фундаментальная информатика и информационные технологии  
направленность (профиль) Открытые информационные системы  
утвержденного Учёным советом вуза от 30.05.2019 протокол № 6.

Рабочая программа одобрена на заседании кафедры

**информатики и информационных технологий**

Зав. кафедрой Богатырева Ю.И.

РПД утверждена Учёным советом университета  
протокол от 30.5.2019 г. № 6

## 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Достижение планируемых результатов обучения, соотнесенных с общими целями и задачами ОПОП

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.В
<b>2.1</b>	<b>Требования к предварительной подготовке обучающегося:</b>
1.	Вводный курс физики
2.	технологическая практика
<b>2.2</b>	<b>Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b>
1.	Алгоритмы и анализ сложности
2.	Основы медиапсихологии
3.	Правовое регулирование профессиональной деятельности
4.	Психологические основы программирования
5.	эксплуатационная практика
6.	Методы оптимизации и исследование операций
7.	Метрология и качество программного обеспечения
8.	научно-исследовательская работа
9.	Теория графов
10.	Технологии визуализации данных

## 3. СООТНЕСЕНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) С ИНДИКАТОРАМИ ДОСТИЖЕНИЯ КОМПЕТЕНЦИЙ

### 3.1 Компетенции обучающегося и индикаторы их достижения:

ПК-1: Способность приобретать и использовать организационно-управленческие навыки в конкретной профессиональной и социальной деятельности; разрабатывать, реализовывать и управлять процессами жизненного цикла программных продуктов

ПК-1.3	Имеет практический опыт управления процессами жизненного цикла программных продуктов имеет опыт поддержки информационных систем знает теоретические основы построения веб сайтов.
--------	---

ПК-2: Способность собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способность к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности

ПК-2.3	Имеет практический опыт составления технического задания на разработку информационной внедрять веб технологии и адаптировать веб-приложения умеет использовать современные среды разработки для реализации веб сайтов и сервисов.
--------	---

### 3.2 Результаты обучения по дисциплине:

**В результате освоения дисциплины обучающийся должен:**

	<b>Знать:</b>
З.1	теоретические основы построения веб сайтов.
	<b>Уметь:</b>
У.1	использовать современные среды разработки для реализации веб сайтов и сервисов.
	<b>Владеть:</b>
В.1	внедрять веб технологии и адаптировать веб-приложения;
В.2	имеет опыт поддержки информационных систем.

## 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	Содержание
	Тема 1. Введение в предмет.				

1.1	<p>Основные понятия и определения. Программа, программная система. Программный продукт. Программная система как технологический объект. Клиент-серверное взаимодействие. Протоколы передачи данных. Асинхронные и синхронные запросы к серверу. Среды разработки. /Лек/</p>	3	6	Л1.1 Л2.1 Л2.2 Л2.3 Л2.4	<p>Клиент-серверное взаимодействие один из главных моментов в изучении дисциплины веб-дизайн. Любой сайт, размещённый в сети Интернет, основывается на связке «клиент-сервер». И это не только электронные формы. Даже простое «перелистывание» страниц некоторого сайта в Интернете – пример клиент-серверного взаимодействия, ведь странички хранятся не на вашем личном компьютере, а подгружаются извне.</p> <p>Рассмотрим основные понятия, которые предполагает эта глава:</p> <p>Клиент — достаточно широкое понятие, начиная от некоторого физического лица и заканчивая некоторой программой на компьютере (например, почтовый клиент). В нашем случае это компьютер, оснащённый специальным программным обеспечением, которое позволяет пользователю задать запрос к другой машине и получить ответ. Код, выполняемый на стороне клиента, чаще всего называют клиентским кодом. Он обеспечивает создание пользовательского интерфейса. Здесь, когда речь заходит о браузере, важную роль играет JavaScript и библиотеки расширения (в нашем случае jQuery).</p> <p>Сервер — это компьютер, оснащённый специальным программным обеспечением, которое позволяет решить задачи предоставления пользователю доступа к некоторым услугам и ресурсам, которыми владеет и управляет данный сервер. Код, выполняемый на стороне сервера, чаще всего называют серверным кодом (серверным сценарием). Он обеспечивает обработку данных. Здесь важную роль играют серверные языки программирования. Примером таких языков являются PHP и Python.</p> <p>Хранилище данных – это система, предназначенная для хранения данных на сервере. Чаще всего под хранилищем данным подразумевают базу данных (например, MySQL). Также хранилищем может выступать файловая система сервера.</p> <p>Клиент-серверное взаимодействие — это обмен данными между клиентом и сервером.</p>
-----	---	---	---	--------------------------	---

1.2	<p>Основные понятия и определения. Программа, программная система. Программный продукт. Программная система как технологический объект. Клиентсерверное взаимодействие. Протоколы передачи данных. Асинхронные и синхронные запросы к серверу. Среды разработки. /Ср/</p>	3	22	Л1.1 Л2.1 Л2.2 Л2.3 Л2.4	<p>Браузер не единственное программное обеспечение, которое позволяет осуществлять клиент-серверное взаимодействие. Почтовые клиенты, программы обмена мгновенными сообщениями, игровые многопользовательские приложения – пример программного обеспечения, которое осуществляет клиент-серверное взаимодействие.</p> <p>Принципы взаимодействия для таких приложений во многом похожи. Схематически его можно описать следующим образом (Рис. 4.1)</p> <p>Рис. 4.1. Клиент-серверное взаимодействие</p> <p>Схема (Рис. 4.1.) предполагает обращение клиента к серверу и ответ на запрос, однако не учитывает специфику действий, производимые сервером (средства поддержки серверных сценариев).</p> <p>При работе, к примеру, с базой данных, хранящейся на сервере, подобная схема не подходит, так как необходимо произвести некоторые действия. Под действием понимается внутрисерверное взаимодействие (Рис. 4.2).</p> <p>Рис 4.2. Внутрисерверное взаимодействие</p> <p>Поскольку в данном случае речь идёт о взаимодействии с удалённой машиной (местоположение которой пользователю неизвестно) очень важно, чтобы запрос обрабатывался как можно быстрее, чтобы длительность отклика сервера была как можно меньше. Также очень важно, чтобы задач, которые выполняет сервер, было как можно меньше, а значит и количества запросов – эти моменты влияют на работоспособность сервера.</p>
-----	---	---	----	--------------------------	--

1.3	<p>Основные понятия и определения. Программа, программная система. Программный продукт. Программная система как технологический объект. Клиент-серверное взаимодействие. Протоколы передачи данных. Асинхронные и синхронные запросы к серверу. Среды разработки. /Лаб/</p>	3	6	Л1.1 Л2.1 Л2.2 Л2.3 Л2.4	<p>Теперь представьте такую ситуацию. Пользователь заполняет на некотором сайте форму регистрации. Нужно указать фамилию, имя, отчество, почту, пароль, телефон, возраст, пройти систему “анти-бот” (т.е. ввести значение, которое изображено на картинке) и др. данные. Полей очень много и все они по существу различны. Когда пользователь заканчивает ввод и нажимает на кнопку “оставить заявку” необходимо проверить правильность введенных данных (например, чтобы пароль был не меньше 10 символов, в телефоне – только цифры, провести проверку значений в поле “анти-бот” и др.). Если возложить всю эту работу на сервер, это во-первых сделает запрос большим по размерам, во-вторых усложнит операции, выполняемые на сервере, в-третьих увеличит время взаимодействия пользователя с системой ввода и сервером. К тому же, если допустить, что пользователь может вводить некорректные данные постоянно, количество запросов к серверу может резко возрасти, что ещё сильнее усугубит ситуацию.</p> <p>Чтобы сэкономить время и ресурсы, можно часть обработки возложить на клиентскую машину. Т.е. использовать JavaScript для сверки полей формы и только потом, когда все данные будут введены корректно, формировать запрос к серверу. Тогда на сервере будут выполняться только элементарные операции: запись, извлечение из хранилища необходимых данных, удаление и др.</p>
<b>Тема 2. Основы html.</b>					

2.1	Гипертекстовая разметка текста. Валидация разметки. Построение семантически верной разметки. Создание каркаса документа. /Лек/	3	6	Л1.1 Л2.1 Л2.2 Л2.3 Л2.4	<p>Чтобы сэкономить время и ресурсы, можно часть обработки возложить на клиентскую машину. Т.е. использовать JavaScript для сверки полей формы и только потом, когда все данные будут введены корректно, формировать запрос к серверу. Тогда на сервере будут выполняться только элементарные операции: запись, извлечение из хранилища необходимых данных, удаление и др.</p> <p>Такой подход хорош не только для форм. Например, JavaScript используется при создании Интернет-игр. Здесь прослеживается активное взаимодействие “пользователь-клиент-сервер” или “пользователь-сервер-пользователь”. Очевидно, что большую часть действий стоит перенести на клиентскую машину, а сервер использовать для связи, пересылки данных, хранения.</p> <p>Другими словами, очень важно разгружать сервер от лишней работы. И JavaScript является хорошим инструментом во многих случаях.</p> <p>В этой главе мы рассмотрим несколько примеров взаимодействия с формой: генерацию списков, проверку ввода, восстановление данных в поля формы. Часть примеров были рассмотрены в гл. 3.</p>
-----	--	---	---	-----------------------------	---

2.2	Гипертекстовая разметка текста. Валидация разметки. Построение семантически верной разметки. Создание каркаса документа. /Ср/	3	22	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	<p style="text-align: center;"><b>История HTML</b></p> <p>HTML был изобретён Тимом Бернерсом-Ли, физиком из исследовательского института ЦЕРН в Швейцарии. Он придумал идею интернет-гипертекстовой системы.</p> <p>Нурertext означает текст, содержащий ссылки на другие тексты, которые зрители могут получить немедленно. Он опубликовал первую версию HTML в 1991 году, состоящую из 18 тегов HTML. С тех пор каждая новая версия языка HTML появилась с разметкой новых тегов и атрибутов (модификаторов тегов).</p> <p>Согласно Справочнику HTML Element Reference от Mozilla Developer Network, в настоящее время существует 140 тегов HTML, хотя некоторые из них уже устарели (не поддерживаются современными браузерами).</p> <p>Из-за быстрого роста популярности HTML теперь считается официальным веб-стандартом. Спецификации HTML поддерживаются и разрабатываются консорциумом World Wide Web (W3C). Вы можете проверить последнее состояние языка в любое время на веб-сайте W3C (англ).</p> <p>Самым большим обновлением языка стало внедрение HTML5 в 2014 году. Было добавлено несколько новых семантических тегов к разметке, которые показывают смысл их собственного контента.</p>
2.3	Гипертекстовая разметка текста. Валидация разметки. Построение семантически верной разметки. Создание каркаса документа. /Лаб/	3	8	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	<p>Обзор наиболее используемых HTML-тегов</p> <p>HTML-теги имеют два основных типа: блок-уровень и встроенные теги.</p> <p>Элементы уровня блока занимают всё свободное пространство и всегда запускают новую строку в документе. Заголовки и параграфы — отличный пример блочных тегов.</p> <p>Встроенные элементы занимают столько места, сколько им нужно, и не запускают новую строку на странице. Они обычно служат для форматирования внутреннего содержимого элементов уровня блока. Ссылки и подчеркнутые строки — хорошие примеры встроенных тегов.</p>
<b>Тема 3. Основы css.</b>					



3.1	<p>Каскадная таблица стилей. Разработка стилевого файла.  Замещение стилей.  Переопределение стилей.  Назначение различных стилей различным классам элементов.  Доступ к элементам по классам, id, названию, местоположению. /Лек/</p>	3	6	<p>Л1.1 Л2.1 Л2.2  Л2.3 Л2.4</p>	<p>Всё, что ваш браузер может выводить на экран или запускать называется фронтендом, то есть это HTML, CSS и JavaScript. Сегодня разбираемся для чего нужна и как работает каскадная таблица стилей (CSS).</p> <p>CSS (Cascading Style Sheets, каскадные таблицы стилей) — язык описания внешнего вида HTML-документа. Это одна из базовых технологий в современном интернете. Практически ни один сайт не обходится без CSS, поэтому HTML и CSS действуют в единой связке.</p> <p>Каскадные таблицы стилей работают с HTML, но это совершенно другой язык. HTML структурирует документ и упорядочивает информацию, а CSS взаимодействует с браузером, чтобы придать документу оформление.</p> <p>На языке HTML мы создаем размеченный текст — документ с гиперссылками, таблицами, маркированными списками, разными начертаниями шрифтов, заголовками, подзаголовками и так далее. Получаем «простыню» текста с таблицами и иллюстрациями. Интернет изобрели ученые, и для них такое положение вещей было приемлемым. Но все изменилось, когда WWW пошел в массы и свои странички начали создавать простые пользователи, которые хотели индивидуальности и самовыражения, а также коммерческие компании со своими корпоративными стандартами оформления. В общем, веб-страницам понадобилось индивидуальное оформление: стиль.</p> <p>Поэтому берем HTML-основу — и подключаем к ней стиль CSS. С помощью CSS красиво оформляем существующий текст, то есть прописываем уникальные свойства элементам HTML.</p>
-----	--	---	---	--------------------------------------	--

3.2	<p>Каскадная таблица стилей. Разработка стилевого файла.  Замещение стилей.  Переопределение стилей.  Назначение различных стилей различным классам элементов.  Доступ к элементам по классам, id, названию, местоположению. /Ср/</p>	3	22	Л1.1 Л2.1 Л2.2 Л2.3 Л2.4	<p>Вспомним, как все начиналось. Первая версия стандарта CSS опубликована 17 декабря 1996 года. Основной посыл — добавить оформление документа без программирования или сложной логики. Оформление — это цвета, шрифты, расположение отдельных блоков на странице и т.д.</p> <p>В таблице стилей мы прописываем стили (правила) оформления. Каждое правило состоит из селектора и блока объявлений. Селектор определяет, на какие части документа распространяется правило. Блок объявлений помещается в фигурные скобки и состоит из одного или более объявлений, разделенных точкой с запятой. На самом деле это очень просто. Вот как выглядит простейшее правило с двумя свойствами для двух селекторов в таблице стилей:</p> <pre> селектор, селектор {     свойство: значение;     свойство: значение; } </pre> <p>Например, для назначения шрифта абзацу (HTML-элементу p) пишем такое правило:</p> <pre> p {     font-family: arial, helvetica, sans serif; } </pre> <p>Оно указывает использовать в абзацах данной веб-страницы шрифт Arial. Если он недоступен — использовать следующий по списку (Helvetica или Sans Serif).</p> <p>А вот правило CSS, которое создает скругленные углы для картинки headerTiny.png.</p>
-----	---	---	----	-----------------------------	---

3.3	Каскадная таблица стилей. Разработка стилевого файла. Замещение стилей. Переопределение стилей. Назначение различных стилей различным классам элементов. Доступ к элементам по классам, id, названию, местоположению. /Лаб/	3	8	Л1.1 Л2.1 Л2.2 Л2.3 Л2.4	<p>Разделение контента (HTML) и представления (CSS)</p> <p>Сама концепция CSS подталкивала к идее полностью разделить контент и оформление документа.</p> <p>В древние времена верстку документа выполняли HTML-таблицами. Например, если вы хотели боковое меню или сайт в три колонки, то создавали в HTML таблицу с колонками соответствующей ширины и прозрачными границами. Для «раздувания» отдельных ячеек таблицы вставлялись прозрачные картинки. В общем, было много интересных «костылей».</p> <p>Проблема в том, что при верстке таблицами оформление привязывалось к контенту, то есть при изменении контента часто приходилось менять оформление, переверстывать эти бесконечные таблицы. Это сводило с ума дизайнеров. Они с облегчением вздохнули, когда появился CSS.</p> <p>Выше приведен пример простого использования CSS. Важно заметить, что CSS изначально не был спроектирован для верстки и макетирования, а только для оформления страницы. Однако веб-дизайнеры и верстальщики уже не могли остановиться.</p>
<b>Тема 4. Основы php.</b>					

4.1	Разработка сценариев с использованием языка PHP. ООП в PHP. Модели обработки данных. MVC модель построения веб-приложения. Клиентские веб-приложения. Работа с javascript. /Лек/	4	10	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	<p>PHP (аббревиатура от Hypertext Preprocessor) является одним из языков сценариев, наряду с JavaScript и Python. Разница между этими языками заключается в том, что PHP в основном используется для коммуникации на стороне сервера, в то время как JavaScript может использоваться как для фронтенд, так и бэкенд-разработки. Python используется исключительно для написания бэкенда.</p> <p>Звучит слишком сложно? Вот почему, прежде чем углубляться в PHP, следует разобраться, что такое языки сценариев.</p> <p>Язык сценариев — это язык, который автоматизирует выполнение задач в специальной среде. Язык сценариев, как бы указывает статической странице (написанной на HTML и CSS) выполнять определённые действия по заданным вами правилам.</p> <p>Например, вы можете использовать скрипт валидации формы, чтобы убедиться, что все поля были заполнены перед отправкой формы на сервер. Сценарий запускается и проверяет все поля, как только пользователь нажимает кнопку “Отправить”.</p> <p>Если какое-то из обязательных полей не было заполнено, пользователь увидит соответствующее предупреждающее сообщение.</p> <p>Другие распространённые варианты использования языков сценариев включают отображение эффекта выпадающего меню при наведении курсора, создание динамических кнопок и анимации, открытие диалоговых окон и т. д.</p>
-----	--	---	----	-------------------------	--

4.2	Разработка сценариев с использованием языка PHP. ООП в PHP. Модели обработки данных. MVC модель построения веб-приложения. Клиентские веб-приложения. Работа с JavaScript. /Ср/	4	44	Л1.1 Л2.1 Л2.2 Л2.3 Л2.4	<p>PHP — не единственный язык сценариев для работы на стороне сервера. Тем не менее, если вы запускаете сайт на WordPress, у PHP здесь нет конкурентов.</p> <p>WordPress написан на PHP. Поэтому, если вы знаете этот язык, вы можете настроить ваш сайт как угодно. Во-первых, это даёт вам возможность редактировать существующие плагины (англ) и темы (англ), а во-вторых вы можете создать собственный плагин или тему с нуля. Если вы хотите стать WordPress-разработчиком, учите PHP.</p> <p>Если вы всё ещё сомневаетесь, вот ещё несколько аргументов в пользу этого языка:</p> <p>Невысокий порог вхождения. Язык легко учить, так как он имеет отличную документацию с примерами, описывающую функции.</p> <p>Широко используется для разработки совершенно разных платформ, например, интернет-магазинов, блогов, социальных сетей и так далее. Согласно статистике 79% всех сайтов (англ) используют PHP!</p> <p>Низкая стоимость. PHP — это проект с открытым исходным кодом, поэтому вы можете использовать его бесплатно.</p> <p>Большое сообщество. Если у вас возникнут какие-либо проблемы, вам не о чем беспокоиться, потому что в Интернете вы найдёте много профильных блогов, форумов и т.д.</p> <p>Интегрируется с базами данных — MySQL, Oracle, Sybase, DB2 и другими.</p>
-----	---	---	----	--------------------------	--

4.3	Разработка сценариев с использованием языка PHP. ООП в PHP. Модели обработки данных. MVC модель построения веб-приложения. Клиентские веб-приложения. Работа с JavaScript. /Лаб/	4	16	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	<p>Мы уже упоминали, что PHP имеет широкое применение. Тем не менее, наиболее часто он используется для создания динамического контента.</p> <p>Если вы знакомы с JavaScript, то знаете, что он также используется, чтобы превратить статическую страницу в динамичную и интерактивную. Следовательно, в этом плане языки имеют схожие функции.</p> <p>Почему нам нужны оба?</p> <p>На самом деле, чтобы создать динамический сайт, хватит одного JavaScript. Однако функциональность будет сильно отличаться.</p> <p>С помощью JavaScript вы можете создать динамический сайт с простыми эффектами, такими как ховер, автозамена, фотогалереи.</p> <p>Но у вас не получится написать сайт, позволяющий пользователям генерировать собственный контент, например, социальную сеть, интернет-магазин (создание заказа, регистрация вход и другие виды взаимодействий с сайтом). Без PHP не обойтись.</p> <p>Допустим, вы входите в социальную сеть. Чтобы отобразить информацию в профиле, PHP извлекает ваши данные из базы данных и отправляет результат в браузер.</p> <p>Всякий раз, когда вы вносите изменения в свой профиль, он будет сохранять новую информацию в базе данных. Весь этот процесс позволяет социальной сети показывать разный контент разным пользователям.</p> <p>Следовательно, JavaScript и PHP не являются конкурентами. Они дополняют друг друга.</p>
	<b>Тема 5. Концепция построения веб-приложений.</b>				
5.1	Правила построения приложений. Паттерны проектирования. SOAP приложения. API. Процесс модульного построения веб-приложений. /Лек/	4	12	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	<p>REST - это концепция (архитектура) для организации взаимодействия между независимыми объектами (приложениями) посредством протокола HTTP. Включает в себя набор принципов (рекомендаций) взаимодействия клиент-серверных приложений. Обычно он представлен в формате JSON.</p> <p>API - интерфейс взаимодействия с каким-либо объектом (программой, приложением), включающий в себя набор правил, которые позволяют одному приложению общаться с другим. Эти «правила» могут включать в себя операции создания, чтения, обновления и удаления. Примером API может служить всем известная библиотека jQuery.</p> <p>REST API позволяет вашему приложению взаимодействовать с одним или несколькими различными приложениями, используя концепции REST.</p>

5.2	Правила построения приложений. Паттерны проектирования. SOAP приложения. API. Процесс модульного построения веб-приложений. /Ср/	4	44	Л1.1 Л2.1 Л2.2 Л2.3 Л2.4	<p>Во многих приложениях REST API необходим, потому что это самый легкий способ создания, чтения, обновления или удаления информации между различными приложениями через Интернет или протокол HTTP. Эта информация представляется пользователю в одно мгновение, особенно если вы используете JavaScript для отображения данных на веб-странице.</p> <p>1.3 Где используется REST API?</p> <p>REST API может использоваться любым приложением, которое может подключаться к Интернету. Если данные из приложения могут быть созданы, прочитаны, обновлены или удалены с помощью другого приложения, это обычно означает, что используется REST API.</p>
5.3	Правила построения приложений. Паттерны проектирования. SOAP приложения. API. Процесс модульного построения веб-приложений. /Лаб/	4	16	Л1.1 Л2.1 Л2.2 Л2.3 Л2.4	<p>REST (RESTful) - это общие принципы организации взаимодействия приложения/сайта с сервером посредством протокола HTTP. Особенность REST в том, что сервер не запоминает состояние пользователя между запросами - в каждом запросе передается информация, идентифицирующая пользователя (например, token, полученный через OAuth-авторизацию) и все параметры, необходимые для выполнения операции.</p> <p>Всё взаимодействие с сервером сводится к 4 операциям (4 - это необходимый и достаточный минимум, в конкретной реализации типов операций может быть больше):</p> <ol style="list-style-type: none"> <li>1. получение данных с сервера (обычно в формате JSON, или XML)</li> <li>2. добавление новых данных на сервер</li> <li>3. модификация существующих данных на сервере</li> <li>4. удаление данных на сервере</li> </ol> <p>Операция получения данных не может приводить к изменению состояния сервера.</p> <p>Для каждого типа операции используется свой метод HTTP-запроса:</p> <ol style="list-style-type: none"> <li>1. получение - GET</li> <li>2. добавление - POST</li> <li>3. модификация - PUT</li> <li>4. удаление - DELETE</li> </ol>

## 5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

### 5.1. Типовые задания для проведения текущего контроля

#### Задание 1

Создание домашней страницы содержащей информацию о студенте. Обязательное содержание:

- Информация о том, чей сайт.
- Электронный почтовый адрес.
- Ссылка на рабочую страницу или сайт вуза студента.
- Мой ВУЗ - о ТПУ.
- Моя группа.
- Моя будущая профессия.
- Мои увлечения или хобби.
- Любая другая информация.

#### Задание 2

1. Добавьте в уже созданную страницу с личными данными и информацией о университете и группе свою фотографию в

- качестве изображения с заданными размерами, обтеканием по тексту и без рамки.
2. В ту же страницу добавьте фоновым изображением герб университета или факультета (при небольшом разрешении файла замостите им страницу).
  3. Создайте страницу-фотогалерею (фотоальбом) и выложите на ней от 4 различных изображений большого разрешения, но с заданными небольшими размерами.
  4. Создайте гипертекстовые переходы между всеми уже созданными страницами, так что бы на любую из страниц можно было попасть на любую другую страницу не более чем за 3 перехода (сделайте меню или карту сайта).
  5. На странице с личной информацией выделите пункты (о себе, о группе и т.п.), создайте оглавление и реализуйте переходы на пункты из оглавления.

### Задание 3

Поменяйте местами блоки theader, tbody иtfoot всеми возможными способами, проанализируйте и опишите результаты.

## 5.2. Типовые задания для проведения промежуточной аттестации

- 1. Разработка варианта игры крестики-нолики.
- 2. Калькулятор, выполняющий все арифметические действия и вычисляющий основные элементарные функции.
- 3. Калькулятор комплексных чисел выполняющий все арифметические действия.
- 4. Выполнение арифметических операций над полиномами. Сложение, вычитание, умножение полиномов.
- 5. Выполнение арифметических операций над полиномами. Деление с остатком.
- 6. Поиск k-го по величине элемента массива.
- 7. Определить положение точки относительно выпуклого многоугольника.
- 8. Площадь многоугольника.
- 9. Пересечение отрезков.
- 10. Алгоритмы теории чисел.
- 11. Диофантово уравнение.
- 12. Генерация перестановки, следующей за данной.
- 13. Некоторые алгоритмы на графах. Алгоритм Дейкстры.
- 14. Некоторые алгоритмы на графах. Алгоритм Флойда.
- 15. Некоторые алгоритмы на графах. Алгоритм Форда-Беллмана.
- 16. k несовпадений - алгоритм Ландау-Вишкина.
- 17. Длинные числа и операции с ними. Сложение и вычитание.
- 18. Длинные числа и операции с ними. Умножение.
- 19. Длинные числа и операции с ними. Деление.
- 20. Работа с датами.

## 5.3. Перечень видов оценочных средств

Экзамен  
Зачет  
Тест  
Выполнение лабораторных работ  
Индивидуальные проекты

## 5.4. Процедура применения оценочных материалов

Промежуточная аттестация может проводиться с применением электронного обучения и (или) дистанционных образовательных технологий в соответствии с "Порядком проведения промежуточной аттестации с применением электронного обучения и /или дистанционных образовательных технологий".

Описание балльно-рейтинговой системы по дисциплине.

Итоговая рейтинговая оценка по дисциплине складывается из следующих составляющих:

- 1) В течении семестра за выполнение заданий по курсу студент может максимально получить 80 баллов;
- 2) Обязательной формой текущей аттестации знаний является выполнение индивидуального проектного задания 20 баллов.
- 3) На зачёте ответ студента может быть максимально оценен в 20 баллов.

При этом, для получения положительной итоговой оценки на зачете необходимо получить не менее 60% по каждой составляющей и выполнить все задания для практических занятий.

Шкала перевода баллов в оценку: до 40 - «не зачтено»; 41 - 100 - «зачтено».

Для получения положительной итоговой оценки на экзамене необходимо получить не менее 50% по каждой составляющей и выполнить все практические задания.

Шкала перевода баллов в оценку: до 64 - «неудовлетворительно»; 65-75 - «удовлетворительно»; 76-85 - "хорошо"; 86-100 - "Отлично".

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 6.1. Рекомендуемая литература

#### 6.1.1. Основная литература

Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание



	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л1.1	Диков А. В.	Веб-технологии HTML и CSS: учебное пособие	, 2012	<a href="http://biblioclub.ru/index.php?page=book&amp;id=96968">http://biblioclub.ru/index.php?page=book&amp;id=96968</a>

### 6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л2.1	Романенко А. В.	Основы программирования для автоматизированных систем проектирования и управления инновациями: учебное пособие	, 2014	<a href="http://biblioclub.ru/index.php?page=book_view&amp;book_id=277966">http://biblioclub.ru/index.php?page=book_view&amp;book_id=277966</a>
Л2.2	Жданов С. А.	Информационные системы: учебник	, 2015	<a href="http://biblioclub.ru/index.php?page=book_view&amp;book_id=426722">http://biblioclub.ru/index.php?page=book_view&amp;book_id=426722</a>
Л2.3	Романенко А. В., Попов А. И.	Основы программирования для автоматизированных систем проектирования и управления инновациями: учебное пособие для студентов, обучающихся по направлению подготовки бакалавров "Инноватика"	Тамбов, 2014	<a href="http://biblioclub.ru/index.php?page=book&amp;id=277966">http://biblioclub.ru/index.php?page=book&amp;id=277966</a>
Л2.4	Жданов С. А., Соболева М. Л., Алфимова А. С.	Информационные системы: учебник	, 2015	<a href="http://biblioclub.ru/index.php?page=book&amp;id=426722">http://biblioclub.ru/index.php?page=book&amp;id=426722</a>

### 6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Университетская библиотека Online [Электронный ресурс] : электронная библиотечная система / ООО "Директ-Медиа" . - М. : [б. и.], 2001. - Загл. с титул. экрана. - Б. ц.
Э2	ИКТ [Электронный ресурс] : федеральный образовательный портал / ФГАУ ГНИИ ИТГ "Информика". - М. : [б. и.], 2003. - Загл. с титул. экрана. - Б. ц.
Э3	Math-Net.Ru [Электронный ресурс] : общероссийский математический портал / Математический институт им. В. А. Стеклова РАН ; Российская академия наук, Отделение математических наук. - М. : [б. и.], 2010. - Загл. с титул. экрана. - Б. ц.
Э4	Универсальные базы данных East View [Электронный ресурс] : информационный ресурс / East View Information Services. - М. : [б. и.], 2012. - Загл. с титул. экрана. - Б. ц.
Э5	Научная электронная библиотека eLIBRARY.RU [Электронный ресурс] : информационный портал / ООО "РУНЭБ" ; Санкт-Петербургский государственный университет. - М. : [б. и.], 2005. - Загл. с титул. экрана. - Б. ц.

### 6.3. Информационные технологии

#### 6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения

1.	Операционная система ROSA Enterprise Linux Desktop № RL00450-1-110518-01. RL00450-1-110518-17 от 11 мая 2018 г.
2.	Операционная система Microsoft Windows XP Professional Russian. Лицензия № 16698685 от 08.08.2003 г.
3.	Операционная система Microsoft Windows Professional 7 Russian. Лицензия №48497058 от 13.05.2011 г., договор № Пр/16/6 от 05 апреля 2016 г.
4.	Операционная система Microsoft Windows 10 Professional Russian. Контракт № ПР/ФЕН/15/18 от 23.10.2015 г., договор № Пр/16/6 от 05 апреля 2016 г.
5.	Программное обеспечение Microsoft Office Enterprise 2007 Russian. Лицензия №46138962 от 16.11.2009
6.	Программное обеспечение Microsoft Office 2013 Professional. Контракт № 405535 от 2 ноября 2015 года, контракт № ПР/ФЕН/15/18 от 23.10.2015 г.
7.	Комплексная система антивирусной защиты Kaspersky Endpoint Security для бизнеса – стандартный Russian Edition. 500-999 Node 2 year Educational Renewal License. Лицензия № 13C8-190514-084943-783-1256 от 15.05.2019
8.	Браузеры Google Chrome, Mozilla, Opera. Свободно распространяемое ПО
9.	Текстовый редактор NotePad++. Свободно распространяемое ПО
10.	Среда программирования и набор инструментов для программирования. MinGW 0.6.3 Свободно распространяемое ПО

#### 6.3.2 Перечень информационных справочных систем и профессиональных баз данных

1.	Портал Федеральных государственных образовательных стандартов высшего образования ( <a href="http://fgosvo.ru">http://fgosvo.ru</a> )
2.	Портал «Информационно-коммуникационные технологии в образовании» ( <a href="http://www.ict.edu.ru">http://www.ict.edu.ru</a> )

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)			
Ауд.	Назначение	Оборудование и технические средства обучения	Вид
2-16	Компьютерный класс	интерактивная доска, компьютеры, маркерная доска, принтер, сканер, стол преподавателя, столы учебные	Лаб
4-318	Компьютерный класс	компьютеры, маркерная доска, серверная стойка лаборатории МТС, стол преподавателя, столы компьютерные, столы учебный большой	Ср
4-307	Компьютерный класс	аудиоколонки, компьютеры, кондиционер, маркерная доска, столы компьютерные, столы учебные, телевизор	КСР
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лек
4-303	Помещение для самостоятельной работы	аудиоколонки, кондиционер, маркерная доска, столы компьютерные, столы учебные, компьютерная техника с возможностью подключения сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета	Ср
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Экзамен
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лаб

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)
<p>Приступая к изучению новой учебной дисциплины, студенты должны ознакомиться с рабочей программой, учебной, научной и методической литературой, имеющейся в библиотеке университета, встретиться с преподавателем, ведущим дисциплину, получить в библиотеке ре-комендованные учебники и учебно-методические пособия, осуществить запись на соответ-ствующий курс в среде электронного обучения университета.</p> <p>Глубина усвоения дисциплины зависит от активной и систематической работы студента на лекциях и практических занятиях, а также в ходе самостоятельной работы, по изучению ре-комендованной литературы.</p> <p>На лекциях важно сосредоточить внимание на ее содержании. Это поможет лучше вос-принимать учебный материал и уяснить взаимосвязь проблем по всей дисциплине. Основное содержание лекции целесообразнее записывать в тетради в виде ключевых фраз, понятий, тезисов, обобщений, схем, опорных выводов. Необходимо обращать внимание на термины, фор-мулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставлять в конспектах поля, на которых делать по-метки из рекомендованной литературы, дополняющей материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С целью уяснения теоретических положений, разрешения спорных ситуаций необходимо задавать преподавателю уточняющие вопросы. Для закрепления содержания лекции в памяти, необходимо во время самостоятельной работы внимательно прочесть свой конспект и дополнить его записями из учебников и рекомендованной литературы. Конспектирование читаемых лекций и их последую-щая доработка способствует более глубокому усвоению знаний, и поэтому являются важной формой учебной деятельности студентов.</p> <p>Прочное усвоение и долговременное закрепление учебного материала невозможно без продуманной самостоятельной работы. Такая работа требует от студента значительных усилий, творчества и высокой организованности. В ходе самостоятельной работы студенты выполняют следующие задачи: дорабатывают лекции, изучают рекомендованную литературу, готовятся к практическим занятиям, к коллоквиуму, контрольным работам по отдельным темам дисци-плины. При этом эффективность учебной деятельности студента во многом зависит от того, как он распорядился выделенным для самостоятельной работы бюджетом времени.</p> <p>Результатом самостоятельной работы является прочное усвоение материалов по пред-мету согласно программы дисциплины. В итоге этой работы формируются профессиональные умения и компетенции, развивается творческий подход к решению возникших в ходе учебной деятельности проблемных задач, появляется самостоятельности мышления. Целью практических занятий по данной дисциплине является закрепление теоретических знаний, полученных при изучении дисциплины.</p> <p>При подготовке к практическому занятию целесообразно выполнить следующие реко-мендации: изучить основную литературу; ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т. д.; при необходимости до-работать конспект лекций. При этом учесть рекомендации преподавателя и требования учебной программы.</p> <p>При выполнении практических занятий основным методом обучения является самостоятельная работа студента под управлением преподавателя. На них пополняются теоретические знания студентов, их умение творчески мыслить, анализировать, обобщать изученный материал, проверяется отношение студентов к будущей профессиональной деятельности.</p> <p>Оценка выполненной работы осуществляется преподавателем комплексно: по результатам выполнения заданий, устному сообщению и оформлению работы. После подведения итогов занятия студент обязан устранить недостатки, отмеченные преподавателем при оценке его работы.</p>