


Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сахалинский государственный университет»

Кафедра информатики

УТВЕРЖДАЮ

Руководитель основной профессиональной
образовательной программы

 _____ Осипов Г.С.

«22» мая 2025 г.

РАБОЧАЯ ПРОГРАММА

Дисциплины

Б1.В.06 Web-технологии, языки и средства создания web-приложений

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

09.03.03 Прикладная информатика

профиль

Автоматизированные системы обработки информации и управления

Квалификация

бакалавр

Форма обучения

очная

РПД адаптирована для лиц с ограниченными возможностями здоровья и инвалидов

Южно-Сахалинск

2025 г.

Рабочая программа дисциплины Web-технологии, языки и средства создания web-приложений составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.03 Прикладная информатика

Программу составил(и):

Соболев Е.И.,

заместитель директора по информационным технологиям Территориального фонда обязательного медицинского страхования Сахалинской области

Рабочая программа дисциплины Web-технологии утверждена на заседании кафедры информатики, протокол № 9 от 22 мая 2025 г.

Исполняющий обязанности
заведующего кафедрой информатики

 Осипов Г.С.

1. Цель и задачи дисциплины

Цель дисциплины

Целью освоения дисциплины Web-технологии, языки и средства создания web-приложений является ознакомление студентов с базовыми концепциями и приемами web-программирования, формирование представления о современных web-технологиях, о проблемах, тенденциях и развитии web-конструирования и web-программирования.

Задачи дисциплины

Основными задачами изучения дисциплины являются:

- изучение основных принципов web-программирования;
- ознакомление с техническими, алгоритмическими, программными и технологическими решениями, используемыми в данной области;
- выработка практических навыков аналитического и экспериментального исследования основных методов и средств, используемых в области, изучаемой в рамках данной дисциплины.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Web-технологии, языки и средства создания web-приложений» относится к базовой части, формируемой участниками образовательного процесса, Блока 1 Дисциплины (модули) подготовки студентов по направлению подготовки бакалавров 09.03.03 Прикладная информатика

Пререквизиты дисциплины: изучение данной дисциплины базируется на знаниях, полученных при изучении следующих дисциплин: Теоретические основы информатики; Операционные системы, Языки и методы программирования.

Постреквизиты дисциплины: основные положения данной дисциплины выступают опорой для изучения дисциплины: Объектно-ориентированное программирование, Сети и системы передачи информации, Безопасность Web-приложений, подготовить к прохождению учебной и производственной практик, к выполнению выпускной квалификационной работы.

3. Формируемые компетенции и индикаторы их достижения по дисциплине

Код компетенции	Содержание компетенции	Код и наименование индикатора достижения компетенции
ПКС-1	Способен решать задачи профессиональной деятельности с учетом текущего состояния и тенденций развития информационных технологий	ПКС-1.1 - Знает виды моделей бизнес-процессов, требования к информационной системе, виды архитектур ИС; технологии программирования, тестирования и внедрения ИС; ПКС-1.2 - Умеет разрабатывать модели бизнес-процессов, требования к информационной системе, архитектуру ИС, применять технологии программирования, тестирования и внедрения ИС; ПКС-1.3 – Владеет методами разработки модели бизнес-процессов, требований к информационной системе, архитектур ИС, технологиями программирования, тестирования и внедрения ИС

4. Структура и содержание дисциплины (модуля)

4.1. Структура дисциплины (модуля)

Общая трудоемкость дисциплины составляет 8 зачетных единиц (288 академических часа).

Вид работы	Трудоемкость, акад. часов		
	семестр		всего
	4	5	
Общая трудоемкость	180	108	288
Контактная работа:	78	56	134
Лекции (Лек)	36	16	52
Лабораторные работы (Лаб)	36	34	70
Контактная работа в период теоретического обучения (КонтТО) (<i>Проведение текущих консультаций и индивидуальная работа со студентами</i>)	5	5	10
Контактная работа в период промежуточной аттестации (КонтПА)	1	1	2
Промежуточная аттестация - экзамен	26	26	52
Самостоятельная работа:	76	26	102
- самостоятельное изучение разделов (перечислить);	4	2	6
- самоподготовка (проработка и повторение лекционного материала, материала учебников и учебных пособий);	28	4	32
- подготовка к лабораторным занятиям;	36	16	52
- подготовка к промежуточной аттестации и т.п.)	8	4	12

4.2. Распределение видов работы и их трудоемкости по разделам дисциплины

№ п/п	Раздел дисциплины/ темы	семестр	Виды учебной работы (в часах)				Формы текущего контроля успеваемости, промежуточной аттестации
			контактная			Самостоятельная работа	
			Лекции	Практические занятия	Лабораторные занятия		
3 семестр							
1.	Тема 1. Принципы разработки Web-документов	4	4	0	6	14	контрольные вопросы по теме лекции, задания к лабораторным работам, домашнее задание
2.	Тема 2. Язык гипертекстовой разметки страниц HTML		12	0	14	26	контрольные вопросы по теме лекции, задания к лабораторным работам, домашнее задание, тест

3.	Тема 3. Использование стиля при оформлении сайта. Спецификации CSS		20	0	16	28	контрольные вопросы по теме лекции, задания к лабораторным работам, домашнее задание, тест
	Экзамен					8	Устный экзамен (по билетам)
	итого:	104	36	0	36	76	
5 семестр							
1.	Тема 4. Язык создания динамических страниц JavaScript	5	6	0	12	6	контрольные вопросы по теме лекции, задания к лабораторным работам, домашнее задание, тест
2.	Тема 5. Программирование на стороне сервера с использованием языка PHP		6	0	10	8	контрольные вопросы по теме лекции, задания к лабораторным работам, домашнее задание, тест
3.	Тема 6. Базы данных MySQL		4	0	12	8	контрольные вопросы по теме лекции, задания к лабораторным работам, домашнее задание, тест
	экзамен					4	Устный экзамен (по билетам)
	итого:	103	16	0	34	26	

4.3. Содержание разделов дисциплины

4 семестр

Тема 1. Принципы разработки Web-документов

Организация Web-сайта (Web-документов). Современные технологии разработки Web-документов. Язык HTML как средство создания Web-страниц. Структура документа. Основные элементы языка. Дизайн в Web-сайтах.

Тема 2. Язык гипертекстовой разметки страниц HTML

Язык гипертекстовой разметки страниц HTML: общая структура документа, абзацы, цвета, ссылки, списки, графика (графические форматы, графический объект как ссылка), таблицы. Фреймы. Общие подходы к дизайну сайта. Разработка макета страницы. Формы.

Тема 3. Использование стиля при оформлении сайта. Спецификации CSS

Использование стиля при оформлении сайта. Возможности CSS. Свойства текста. Свойства цвета и фона. Свойства шрифта. Свойства блоков. Свойства списков. Классы. Псевдоклассы.

5 семестр

Тема 4. Язык создания динамических страниц JavaScript

Преимущества и ограничения программ, работающих на стороне клиента. Понятие динамических страниц. Язык JavaScript: основы синтаксиса. Объектная модель HTML страницы. Событийная модель DHTML: связывание событий с кодом, всплытие событий, объект Event. Применение DHTML.

Тема 5. Программирование на стороне сервера с использованием языка PHP

Введение в программирование на стороне сервера на примере PHP. Принцип работы. Синтаксис языка программирования PHP. Переменные. Константы. Операторы в PHP.

Циклы. Массивы. Работа со строками. Функции в PHP. Встроенные функции. Работа с датой и временем в PHP. Связь PHP и HTML. Создание HTML-страниц средствами PHP.

Тема 6. Базы данных MySQL

Варианты хранения информации в сети Internet. Принципы хранения информации в базах данных MySQL. Архитектура базы данных MySQL (таблицы, связи, триггеры). Проектирование баз данных. Нормализация таблиц. Синтаксис запросов к базе данных. Механизм работы с базами данных. Управление форматами даты и времени. Подключение к базе данных из PHP файла. Вывод данных на PHP-страницу, попавших в выборку по SQL запросу. Передача параметров в запрос.

4.4 Темы и планы лабораторных занятий

4 семестр

Лабораторное занятие №1 (6 ч.)

Тема: Принципы разработки Web-документов

Вопросы для обсуждения:

1. Организация Web-сайта (Web-документов).
2. Современные технологии разработки Web-документов.
3. Язык HTML как средство создания Web-страниц.
4. Структура документа.
5. Основные элементы языка. Дизайн в Web-сайтах.

Примерные задания:

1. Создать html-файл. Добавить в него изображение. Нарисовать с использованием произвольного графического редактора схему места обучения (рекламный план положения СахГУ с указанием улиц, адреса, способа приезда на общественном транспорте, входа). Вставить схему в html-файл.

Лабораторное занятие №2 (14 ч.)

Тема Язык гипертекстовой разметки страниц HTML. Расширенный синтаксис HTML

Вопросы для обсуждения:

1. Язык гипертекстовой разметки страниц HTML: общая структура документа, абзацы, цвета, ссылки, списки, графика (графические форматы, графический объект как ссылка), таблицы.
2. Фреймы.
3. Общие подходы к дизайну сайта.
4. Разработка макета страницы.
5. Формы.

Примерные задания:

1. Спроектировать структуру веб-сайта по любой теме.
2. Разработать эскиз оформления веб-сайта (использовать любой графический редактор).

Лабораторное занятие №3 (16 ч.)

Тема Использование стиля при оформлении сайта. Спецификации CSS. Базовый синтаксис CSS3.

Вопросы для обсуждения:

1. Использование стиля при оформлении сайта.

2. Возможности CSS.
3. Свойства текста. Свойства цвета и фона. Свойства шрифта.
4. Свойства блоков.
5. Свойства списков.
6. Классы.
7. Псевдоклассы *Примерные задания:*

1. Создайте несколько фрагментов текста и, применив к ним классы CSS, получите, например, такой результат:



2. Оформите 3 таблицы с помощью стилей CSS:

Оформление таблицы с помощью стилей CSS		
1	2	3
4	5	6

Оформление таблицы с помощью стилей CSS		
1	2	3
4	5	6

CSS таблица

Оформление таблицы с помощью стилей CSS			
	книги	журналы	газеты
Цена	20	15	3
Количество	100	100	1000
Всего	2000	1500	3000

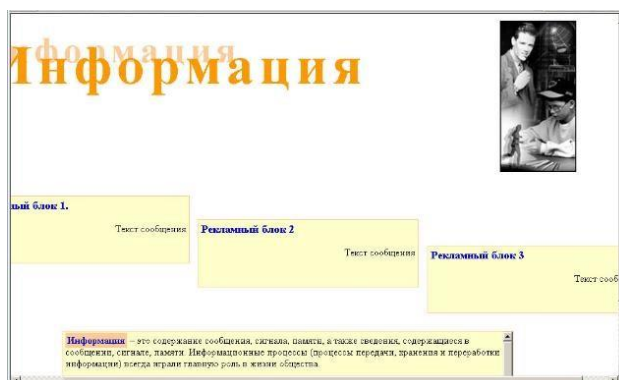
Использование стиля при оформлении сайта. Спецификации CSS. Блочные и строчные элементы. Примерные задания:

1. Выполнить верстку макета страницы с блочной структурой по разработанному эскизу для вашего сайта (см. задание к лабораторной работе № 2).
2. Создать внешние таблицы стилей (раздельные для устройств screen, print и handheld) для вашего сайта (см. задание к лабораторной работе № 2).
3. Подключить созданные таблицы к макету страницы.
4. Проверить правильность отображения веб-страниц в различных браузерах.

Использование стиля при оформлении сайта. Спецификации CSS. Верстка сайта.

Примерные задания:

1. Создайте собственную раскладку страницы с помощью блоков. Страница должна быть масштабируемой. В качестве образца можете использовать один из приведенных вариантов (см. рис.).



5 семестр

Лабораторное занятие №1 (12 ч.)

Тема: Язык создания динамических страниц JavaScript

Вопросы для обсуждения:

1. Преимущества и ограничения программ, работающих на стороне клиента.
2. Понятие динамических страниц. Язык JavaScript: основы синтаксиса.
3. Объектная модель HTML страницы.
4. Событийная модель DHTML: связывание событий с кодом, всплытие событий, объект Event.
5. Применение DHTML.

Примерные задания:

1. Написать скрипт «Tip of the Day» (Совет дня). Скрипт должен выводить случайную строку («совет») из заданного массива строк. Скрипт разместить во внешнем файле, подключить его на все страницы вашего сайта (см. задание к лабораторной работе № 2).
2. Создать простой HTML-документ и выполнить для этого документа следующие задания:
 - Добавить два абзаца с произвольным текстом.
 - Организовать между двумя абзацами вывод приветственного сообщения в диалоговом окне, задав необходимые команды внутри тэга <script>.
 - Добавить команду вывода аналогичного приветственного сообщения в окно браузера после закрытия диалогового окна.
 - Сохранить документ с именем Ex1.html в рабочей папке.
3. Создать простой HTML-документ и выполнить для этого документа следующие задания:
 - Добавить два абзаца с произвольным текстом.

- Организовать между двумя абзацами вывод приветственного сообщения в диалоговом окне, задав необходимые команды JavaScript во внешнем файле. Для этого:
 - создать новый текстовый файл,
 - поместить в него код JavaScript,
 - сохранить файл с именем main.js следующим образом: указать тип файла «Все файлы», кодировку «UTF-8».
 - Добавить ссылку на внешний скриптовый файл из рабочего HTML-документа. □ Сохранить документ с именем Ex2.html в рабочей папке.
4. Создать простой HTML-документ и выполнить для этого документа следующие задания:
- Сохранить документ с именем Ex3.html в рабочей папке.
- Добавить в документ код JavaScript так, чтобы в диалоговом окне появлялось поле с надписью: «Введите сюда своё имя» и со значением по умолчанию в поле «Введите имя». Для этого используйте метод `prompt(...)` объекта `window`. Для хранения введенного значения заведите новую переменную.
 - Организовать вывод введенного значения имени в окно браузера в виде: «Ваше имя <.....>».
 - Дополнить код, чтобы в новом диалоговом окне появилась надпись «Начать заново?». При положительном ответе появлялось диалоговое окно: «Не надоело?», при отказе – «Ну, и правильно!». Использовать для написания методы `alert(...)` и `confirm(...)` объекта `window`.

Язык создания динамических страниц JavaScript. Модель DOM

Примерные задания:

1. Написать скрипт, проверяющий код защиты от автоматического постинга и вырезающий ссылки из формы ввода комментария (на странице отзывов и комментариев).
- Пояснение:* В ходе выполнения задания требуется написать клиентскую часть на JavaScript, которая генерирует арифметический пример, ответ на который должен дать пользователь. Другой вариант — генерация произвольной строки, которую должен воспроизвести пользователь. После того, как пользователь ввел ответ программа должна проверить его правильность.
2. Установить веб-сервер Apache.
 3. Создать виртуальный хост <ваш-номер-зачетной-книжки>.local.
 4. Разместить на созданном сайте все веб-документы (гипертекстовые страницы, файлы стилей, клиентские скрипты на JavaScript), созданные в предыдущих лабораторных работах.

Лабораторное занятие №2 (10 ч.)

Тема Программирование на стороне сервера с использованием языка PHP

Вопросы для обсуждения:

1. Введение в программирование на стороне сервера на примере PHP.
2. Принцип работы. Синтаксис языка программирования PHP.
3. Переменные. Константы. Операторы в PHP. Циклы.
4. Массивы.
5. Работа со строками.
6. Функции в PHP. Встроенные функции. Работа с датой и временем в PHP.

7. Связь PHP и HTML.
8. Создание HTML-страниц средствами PHP.

Программирование на стороне сервера с использованием языка PHP. Разработка серверных скриптов.

Примерные задания:

1. Написать скрипт, учитывающий количество кликов по ссылкам на скачивание файлов, записывающий эти данные в файл, а затем выполняющий редирект на скачиваемый файл. Формат записи может быть, например таким: имя_файла;дата/время;ip-адрес клиента;
2. Средствами PHP создайте генератор web-страниц. Подготовьте форму, в которую бы вводился заголовок страницы, цвет фона, цвет шрифта и текст страницы. Создайте HTML-страницу с помощью этой формы.
3. Средствами PHP напишите классическую игру «Угадай число». Пусть компьютер создает случайное число и дает пользователю возможность его угадать. Сообщайте пользователю, является ли его ответ слишком большим, слишком маленьким или правильным. После того, как пользователь угадает значение, сообщите ему, сколько попыток для этого потребовалось.
4. Создайте средствами PHP простую гостевую книгу. Позвольте пользователю вводить информацию в форму, и когда он нажимает кнопку Submit, добавьте его комментарий в конец страницы.

Программирование на стороне сервера с использованием языка PHP. Функциональное программирование на стороне сервера.

Примерные задания:

1. Написать скрипт, читающий файл статистики скачиваний (см. ЛР 8) и подсчитывающий количество скачиваний по каждому файлу. Полученную информацию внедрить в страницы вашего сайта возле ссылок на соответствующие файлы (программа, руководство пользователя и прочие файлы в форматах pdf, doc, zip и т.п.).
2. Создать форму регистрации пользователя, позволяющую передать на сервер (метод GET) следующие данные:
 - Имя пользователя (обязат.)
 - Логин (обязат.)
 - Пароль и подтверждение пароля (обязат.)
 - Адрес электронной почты (обязат.)
 - Пол (не обязат.)
 - Дата рождения (не обязат.) □ Страна и город (не обязат.)
3. Написать серверный php-скрипт, принимающий регистрационные данные и отображающий их на веб-странице.
4. Изменить метод отправки данных из формы на POST, внести соответствующие изменения в скрипты и сравнить результаты выполнения.
5. Создать форму загрузки от 1-го до 5-ти файлов на сервер и написать php-скрипт, принимающий эти файлы. Скрипт должен выводить сообщение об успешной загрузке или об ошибке в случае неудачи.

Лабораторное занятие №3 (12 ч.)

Тема Базы данных MySQL

Вопросы для обсуждения:

1. Варианты хранения информации в сети Internet.
2. Принципы хранения информации в базах данных MySQL.
3. Архитектура базы данных MySQL (таблицы, связи, триггеры).
4. Проектирование баз данных. Нормализация таблиц.
5. Синтаксис запросов к базе данных.
6. Механизм работы с базами данных.
7. Управление форматами даты и времени.
8. Подключение к базе данных из PHP файла.
9. Вывод данных на PHP-страницу, попавших в выборку по SQL запросу.
10. Передача параметров в запрос.

Примерные задания:

1. Создать базу данных mysite (использовать СУБД MySQL), содержащую следующие таблицы:
 - myusers — для хранения данных о пользователях сайта (структура таблицы соответствует данным из регистрационной формы);
 - mycontent — для хранения текстовой информации (контента), отображаемой на страницах сайта. Примерная структура таблицы: pageid, userid, lastmod, title, description, content.
 2. Написать скрипт register.php, получающий данные о пользователе со страницы регистрации и записывающий эти данные в таблицу myusers.
 3. Создать форму авторизации (логин и пароль) и написать скрипт login.php, сравнивающий эти данные с записями из таблицы myusers. Если пара "логин + пароль" найдена в БД, то пользователь считается авторизованным, в ином случае выполняется перенаправление на страницу регистрации.
 4. Создать форму добавления контента на сайт и скрипт addpage.php, сохраняющий полученную информацию в таблице mycontent.
- Примечание:* Поскольку использование сессий будет рассмотрено позже, то значение userid в этом задании следует задать явно (например, в переменной).
5. Написать скрипт page.php, выбирающий данные из таблицы mycontent (по pageid) и отображающий их в виде веб-страницы.

Взаимодействие сервера с базами данных.

Примерные задания:

1. Изменить скрипт авторизации login.php (см. ЛР 10) таким образом, чтобы для авторизованного пользователя открывалась новая сессия, в которой сохраняются логин и ID пользователя.
2. Переписать скрипт добавления материалов на сайт с учетом сохранения значения userid (полученного из переменной сессии) в таблице mycontent.
3. Написать скрипт logout.php, выполняющий закрытие сессии для авторизованного пользователя.

5. Темы дисциплины (модуля) для самостоятельного изучения

4 семестр

№	Название темы	Количество часов
1.	Схема передачи и приема данных внутри протокола HTTP	4

Вопросы для самоконтроля:

1. Перечислите основные способы передачи информации внутри протокола HTTP.
2. Опишите заголовок пакета HTTP.
3. Перечислите базовые уязвимости заголовка пакета HTTP.

5 семестр

№	Название темы	Количество часов
1.	Способы хранения информации о пользователе и его настройках	4

Вопросы для самоконтроля.

1. Дайте определение понятия «сессия».
2. Перечислите основные проблемы хранения данных на стороне пользователя.
3. Перечислите способы защиты данных пользователя.

6. Образовательные технологии

№ п/п	Наименование раздела	Виды учебных занятий	Образовательные технологии
1.	Тема 1. Принципы разработки Web-документов	Лекция	Традиционная лекция в аудитории с мультимедиа проектором
		Лабораторное занятие	Лабораторное занятие в компьютерном классе
		Самостоятельная работа	Изучение материала по теме лекции, подготовка домашнего задания
2.	Тема 2. Язык гипертекстовой разметки страниц HTML	Лекция	Традиционная лекция в аудитории с мультимедиа проектором
		Лабораторное занятие	Лабораторное занятие в компьютерном классе
		Самостоятельная работа	Изучение материала по теме лекции, подготовка домашнего задания
3.	Тема 3. Использование стиля при оформлении сайта. Спецификации CSS	Лекция	Традиционная лекция в аудитории с мультимедиа проектором
		Лабораторное занятие	Лабораторное занятие в компьютерном классе
		Самостоятельная работа	Изучение материала по теме лекции, подготовка домашнего задания
4.	Тема 4. Язык создания динамических страниц JavaScript	Лекция	Традиционная лекция в аудитории с мультимедиа проектором
		Лабораторное занятие	Лабораторное занятие в компьютерном классе

		Самостоятельная работа	Изучение материала по теме лекции, подготовка домашнего задания
5.	Тема 5. Программирование на стороне сервера с использованием языка PHP	Лекция	Традиционная лекция в аудитории с мультимедиа проектором
		Лабораторное занятие	Лабораторное занятие в компьютерном классе
		Самостоятельная работа	Изучение материала по теме лекции, подготовка домашнего задания
6.	Тема 6. Базы данных MySQL	Лекция	Традиционная лекция в аудитории с мультимедиа проектором
		Лабораторное занятие	Лабораторное занятие в компьютерном классе
		Самостоятельная работа	Изучение материала по теме лекции, подготовка домашнего задания

7. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине (модулю)

Примерный вариант теста по теме «Язык гипертекстовой разметки страниц HTML»

- Требуется написать химическую формулу Сульфида мышьяка (As_2S_3). Каким образом это можно сделать?
 - `As2S3`
 - `As2S3`
 - `As2S3`
 - `As2S3`
- В каких случаях атрибут выравнивания align имеет более высокий приоритет? а) `<TH align="left">`
 - `<COL align="left">`
 - `<TABLE align="left">`
 - `<OL align="left">`
 - `<UL align="left">`
- Какой вариант написан с ошибкой?
 - `<ol type="N">`
 - `<ol type="I">`
 - `<ol type="i">`
 - `<ol type="A">`
- Необходимо сделать ширину таблицы на всю страницу (либо родительского контейнера). Как надо написать тег `<table>`?
 - `<table>`
 - `<table width="auto">`
 - `<table width="100%">`
 - `<table width="100">`
- Какой из приведенных фрагментов кода создает переключатель?
 - `<input Type="checkbox" nAME="a1" vAlue="1"><input TYPE="checkbox" nAME="a1" vAlue="2"><input TYpe="text" nAME="a1" vAlue="2">`

- b) `<input TYPE="radiobutton" nAME="a1" vAlue="1"><input TYPE="radiobutton" nAME="a1" vAlue="2">`
 - c) **`<input TYPE="radio" nAME="a1" vAlue="1"><input TYPE="radio" nAME="a1" vAlue="2">`**
 - d) `<input Type="checkbox" nAME="a1" vAlue="1"><input TYPE="checkbox" nAME="a1" vAlue="2"><input TYpe="text" nAME="a1" vAlue="2">`
 - e) `<input TYPE="radiobutton" nAME="a1" vAlue="1"><input TYPE="radiobutton" nAME="a1" vAlue="2">`
6. На сайте внутри папки pages находится файл page.html. А внутри папки images находится файл foto.jpg. Причём папки images и pages лежат в корне сайта. Как правильно написать путь к foto.jpg из файла page.html:
- a) **`../images/foto.jpg`**
 - b) `../images/pages/foto.jpg`
 - c) `pages/images/foto.jpg`
 - d) `images/foto.jpg`
7. Какого тега НЕ существует:
- a) `<pre>`
 - b) ``
 - c) **`<adress>`**
 - d) ``
8. Какой атрибут элемента FORM определяет список кодировок для вводимых данных? а) alt
- b) **`accept-charset`**
 - c) `enctype-charset`
 - d) `act-charset`
 - e) `enct-charset`
9. Какой из вариантов содержит ошибку:
- a) `Ссылка`
 - b) `Ссылка`
 - c) **`Ссылка`**
 - d) `Ссылка`
10. Размер окна браузера 1000 пикселей. На страницу добавили блок с шириной 40%. Затем в этот блок добавили таблицу с шириной 50%. Какова будет ширина таблица в пикселях? а) 250 пикселей.
- b) Недостаточно данных.
 - c) **200 пикселей.**
 - d) 500 пикселей.

Примерный вариант теста по теме «Использование стиля при оформлении сайта. Спецификации CSS»

1. Как изменить цвет фона для всех элементов h1 на странице?
- a) **`h1 {background-color: #ccc;}`**
 - b) `h1[all] {background-color: #ccc;}`
 - c) `h1.all {background-color: #ccc;}`
 - d) `h1:all {background-color: #ccc;}`

2. Написан HTML-код: `<div id="myid">Содержимое</div>`. Как задать стиль для тега `<div>`?
- a) `div.myid {margin: 1px;}`
 - b) `.myid {margin: 1px;}`
 - c) **`div#myid {margin: 1px;}`**
 - d) `div[myid] {margin: 1px;}`
3. Какой вариант задания цвета НЕ сработает?
- a) `color: #aaa;`
 - b) `color: #aaaaaa;`
 - c) `color: #000;`
 - d) **`color: #hhh;`**
4. Как правильно вставляются комментарии в CSS-код?
- a) **`/* Мой комментарий */`**
 - b) `# Мой комментарий #`
 - c) `# Мой комментарий`
 - d) `// Мой комментарий`
5. Написан HTML-код: `<p>Синий не синий</p>`. Какой CSS-код внутри тега `` сделает синий цвет?
- a) `span {color: blue;}`
 - b) `p span {color: blue;}`
 - c) **Все представленные варианты подойдут**
 - d) `.blue {color: blue;}`
6. Шрифт задан следующим образом: `font-family: Arial, Times New Roman, Helvetica, sansserif;`. Что здесь не так?
- a) **Нужно поставить кавычки у Times New Roman.**
 - b) Шрифта Helvetica не существует.
 - c) Нельзя указывать больше 3-х различных шрифтов.
 - d) Вместо свойства `font-family` нужно использовать свойство `font`.
7. Какой селектор написан с ошибкой?
- a) **`input["text"]`**
 - b) `input[type="text"]`
 - c) `input.text`
 - d) `div p`
8. Какое принципиальное отличие селектора ID от селектора CLASS?
- a) Никакого отличия между ними нет.
 - b) **ID должен быть уникальным на странице, а одинаковый CLASS может быть у нескольких элементов.**
 - c) Разница только в наборе свойств, который может быть использован для этих селекторов. Например, для ID нельзя задавать свойство `border`, а для CLASS можно.
 - d) CLASS должен быть уникальным на странице, а одинаковый ID может быть у нескольких элементов.
9. Есть такой CSS-код во внешнем файле: `p {color: blue;}`. На странице написан такой HTML-код: `<p style="color: red;">текст</p>`. Какого цвета будет "текст"? а) Синего.
- b) Чёрного.
 - c) В браузере IE8 синего, а в других красного.
 - d) **Красного.**

10. Необходимо, чтобы при наведении указателя мыши на ссылку, она меняла цвет. Какой псевдоэлемент нужно использовать?

- a) active
- b) onmouseover
- c) link
- d) **hover**

Примерный вариант теста по теме «Язык создания динамических страниц JavaScript»

1. Написан следующий код:

```
<script type="text/javascript">
  var a = 1;
  function func() {
    a = 10;
  }
  func();
  alert(a);
</script>
```

Что будет выведено в появившемся окне?

- a) 1
 - b) undefined
 - c) **10**
 - d) Ошибка, поскольку в коде идёт попытка 2 раза объявить переменную с одинаковым именем.
2. Какая разница между равенством и эквивалентностью?
- a) Знак эквивалентности проверяет лишь значения операндов, а знак равенства значения и их типы.
 - b) Никакой разницы нет.
 - c) Знак эквивалентности работает лишь для строк, а знак равенства применим к любым типам.
 - d) **Знак равенства проверяет лишь значения операндов, а знак эквивалентности значения и их типы.**
3. Написан следующий код:

```
<script type="text/javascript">
  var a = 1;
  var b = a++ + 1;
  alert(b);
</script>
```

Что будет выведено в появившемся окне?

- a) Ничего не будет, поскольку инкремент нельзя соединять со знаком сложения.
 - b) 1
 - c) 3
 - d) **2**
4. Что делает функция printMessage("Text")?
- a) Ошибка, поскольку в функции printMessage() 2 параметра.
 - b) Выводит окно с текстом Text.
 - c) **Функция printMessage() вообще не существует.**
 - d) Выводит окно с текстом "Text".
5. Написан следующий код:

```
<script type="text/javascript">
  var a = 1;
  function func() {
    var a = 10;
  }
  func();
  alert(a);
</script>
```

Что будет выведено в появившемся окне?

- a) 1
 - b) Ошибка, поскольку в коде идёт попытка 2 раза объявить переменную с одинаковым именем.
 - c) undefined
 - d) 10
6. Какая функция вызывает окно с текстовым полем, в которое пользователь может ввести строку?
- a) alerts()
 - b) **prompt()**
 - c) promt()
 - d) alert()
7. Написан следующий код:

```
<script type="text/javascript">
  var arr = new Array();
  arr[arr.length] = 0;
  arr[arr.length] = 1;
  alert(arr.length);
</script>
```

Что будет выведено в появившемся окне?

- a) 0
 - b) 2
 - c) 1
 - d) Код нерабочий, поскольку писать arr[arr.length] можно только при длине массива отличной от 0.
8. Есть код:

```
<script type="text/javascript">
  var a = 1;
  function func() {
    alert(a);
  }
  func();
</script>
```

Что будет написано в появившемся окне?

- a) Ничего не будет, поскольку возникнет ошибка "переменная не определена". b) undefined
 - c) 0
 - d) 1
9. Вызваны две функции: parseFloat("\$101.2") и parseFloat("101.2 USD"). Какая из них вернёт число?
- a) parseFloat("\$101.2")
 - b) **parseFloat("101.2 USD")**
 - c) Обе функции.
 - d) Ни одна из них.

10. Есть код:

```
<script type="text/javascript">
  var a = "5";
  var b = 2;
  var c = a + b;
  alert(c);
</script>
```

Что будет написано в появившемся окне?

- a) NaN
- b) 7
- c) Ошибка выполнения скрипта.
- d) 52

Примерный вариант теста по теме «Программирование на стороне сервера с использованием языка PHP»

1. Есть строка: "my string". Есть код:

```
<?php
  if (strpos("mystring", "m") == false) echo "Символа m в строке mystring нет";
?>
```

В чём заключается ошибка в коде?

- a) Здесь нет ошибок, так как strpos("mystring", "m") не равен false, и строки "Символа m в строке mystring нет" не появляется.
 - b) Функция strpos() неправильно использована. Сначала должна идти искомая строка, а уже потом та строка, в которой происходит поиск.
 - c) **Функция strpos() вернула 0, который в PHP равен false. Чтобы не было ошибки надо вместо знака равенства (==) использовать знак эквивалентности (===).**
 - d) Нет фигурных скобок у блока операторов при срабатывании условия.
2. Есть массив \$arr. Как вывести все элементы массива \$arr?
- a) echo \$arr;
 - b) Это возможно сделать только через цикл.
 - c) print \$arr;
 - d) **print_r(\$arr);**
3. Каков результат выполнения у данного скрипта:

```
<?php
  $x = 5;
  echo "Переменная $x = 5";
?>
```

- a) Ошибка, поскольку строки с переменными должны быть всегда в одинарных кавычках.
 - b) **Переменная 5 = 5**
 - c) Переменная \$x = 5
 - d) Переменная x = 5
4. Какая разница между равенством и эквивалентностью?
- a) Знак эквивалентности работает лишь для строк, а знак равенства применим к любым типам.
 - b) **Знак равенства проверяет лишь значения операндов, а знак эквивалентности значения и их типы.**
 - c) Знак эквивалентности проверяет лишь значения операндов, а знак равенства значения и их типы.

d) Никакой разницы нет.

5. Каков результат выполнения у данного скрипта:

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

a) 013

b) 024

c) 24

d) **13**

6. Какого оператора в PHP не существует?

a) **! = =**

b) != =

c) +=

d) >=

7. Что выведет такой код:

```
<?php
$x = 10;
$x += --$x;
echo $x;
?>
```

a) 19

b) 20

c) **18**

d) 21

8. Каков будет результат выполнения скрипта:

```
<?php
function myfunc($x = 0) {
    echo $x;
}
myfunc();
myfunc(5);
?>
```

a) Ошибка, поскольку параметрам в функции нельзя присваивать значения. b) 0

c) 5

d) **05**

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

a) location.href = "http://google.ru";

b) header("Redirect: http://google.ru");

c) **header("Location: http://google.ru");**

d) document.location = "http://google.ru";

10. Какого типа данных в PHP нет?

a) Resource.

b) null.

c) Объект.

d) **Указатель.**

Примерный вариант теста по теме «Базы данных MySQL» 1.

Какое из утверждений о первичном ключе НЕ верно?

- a) **Первичный ключ может содержать NULL значения.**
 - b) Каждая таблица имеет первичный ключ.
 - c) Первичный ключ содержит только уникальные значения.
 - d) Первичный ключ НЕ может содержать NULL значений.
2. В некоторой таблице есть поле "field". Как правильно вывести все записи, где значение поля "field" содержит строку (либо подстроку) "string":
- a) WHERE `field` LIKE "string"
 - b) WHERE `field` == "string"
 - c) **WHERE `field` LIKE "%string%"**
 - d) WHERE `field` = "string"
3. Как удалить таблицу "my_table":
- a) DELETE FROM "my_table"
 - b) DELETE TABLE "my_table"
 - c) **DROP TABLE "my_table"**
 - d) DELETE "my_table"
4. Выберите верное утверждение относительно индекса:
- a) Индекс ускоряет добавление записей в таблицу.
 - b) **Индекс позволяет ускорить выборку с тем полем, для которого он сделан.**
 - c) Индекс ускоряет абсолютно любые запросы с таблицей.
 - d) Индекс позволяет сэкономить место, занимаемое таблицей.
5. Для чего используется команда GRANT?
- a) Для очистки таблицы.
 - b) **Для управления пользователями.**
 - c) Для перезапуска сервера MySQL.
 - d) Для очистки базы данных.
6. Написан такой запрос: SELECT text FROM articles WHERE id='15' Однако он не срабатывает. Почему?
- a) Таблицу articles надо заключить в обратные кавычки, поскольку это зарезервированное слово.
 - b) **Поле text надо заключить в обратные кавычки, поскольку это зарезервированное слово.**
 - c) Поле id надо заключить в обратные кавычки, поскольку это зарезервированное слово.
 - d) Надо писать не id='15', а id="15".
7. Для чего используется ключевое слово DISTINCT?
- a) Для снижения нагрузки на сервер с потерей производительности выполнения запроса.
 - b) **Для выборки только уникальных записей по конкретному полю.**
 - c) Для ускорения выборки по конкретному полю.
 - d) Такого ключевого слова не существует.
8. Необходимо извлечь все записи, где значение поля "field" начинается с подстроки "abc". Какой запрос для этого надо использовать?
- a) SELECT * FROM `my_table` WHERE `field` LIKE '%abc%'
 - b) SELECT * FROM `my_table` WHERE `field` <> 'abc'
 - c) SELECT * FROM `my_table` WHERE `field` STARTSWITH 'abc'
 - d) **SELECT * FROM `my_table` WHERE `field` LIKE 'abc%'**

9. Что делает конструкция: ORDER BY ASC
- Сортирует записи по убыванию.
 - ASC вообще не существует.
 - ASC - это функция, которая здесь некорректно использована.
 - Сортирует записи по возрастанию.**
10. Как обновить запись сразу в двух колонках:
- UPDATE `my_table` SET `field_1` = 'value_1', `field_2` = 'value_2'**
 - UPDATE `my_table` SET `field_1` = 'value_1' SET `field_2` = 'value_2'
 - UPDATE `my_table` SET `field_1` = 'value_1', SET `field_2` = 'value_2'
 - UPDATE `my_table` SET `field_1` = 'value_1' AND `field_2` = 'value_2'

Примерные вопросы к экзамену (4 семестр)

- Глобальные компьютерные сети: основные понятия, принципы функционирования.
- Каталоги ресурсов. Поисковые системы.
- Язык гипертекстовой разметки страниц HTML: общая структура документа, абзацы, цвета, ссылки.
- Язык гипертекстовой разметки страниц HTML: списки, графика (графические форматы, графический объект как ссылка).
- Язык гипертекстовой разметки страниц HTML: таблицы.
- Язык гипертекстовой разметки страниц HTML: формы.
- Фреймы.
- Общие подходы к дизайну сайта. Разработка макета страницы 9. Использование стиля при оформлении сайта. Возможности CSS.
- CSS. Свойства текста. Свойства цвета и фона.
- CSS. Свойства шрифта. Свойства блоков.
- CSS. Свойства списков. Классы. Псевдоклассы.

Примерные вопросы к экзамену (5 семестр)

- Хостинг. Бесплатный хостинг. FTP. Размещение Интернет-ресурса на сервере провайдера.
- Регистрация Интернет-ресурса в каталогах и поисковых системах.
- Преимущества и ограничения программ, работающих на стороне клиента. Язык JavaScript: основы синтаксиса.
- Введение в программирование на стороне сервера на примере PHP. Принцип работы.
- Синтаксис языка программирования PHP.
- Переменные. Константы. Операторы в PHP. Циклы. Массивы. Работа со строками.
- Функции в PHP. Встроенные функции.
- Работа с датой и временем в PHP.
- Методы передачи параметров между страницами (GET, POST). Обработка действий пользователя при помощи форм.
- Принципы хранения информации в базах данных MySQL. Архитектура базы данных MySQL (таблицы, связи, триггеры).
- Подключение к базе данных из PHP файла. Вывод данных на PHP-страницу, попавших в выборку по SQL запросу. Передача параметров в запрос.
- Принципы проектирования страниц. Разделение информации по таблицам в базе данных.
Вывод группы данных, сортировка данных.
- Создание HTML-страниц средствами PHP.

8. Система оценивания планируемых результатов обучения

Критерии оценивания

Критерии оценивания (экзамен)

Оценка «отлично» выставляется студенту, глубоко и прочно усвоившему программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагающему, в ответе которого увязывается теория с практикой, он показывает знакомство с литературой, правильно обосновывает и использует рациональные и современные средства решения поставленной проблемы.

Оценка «хорошо» выставляется студенту, твердо знающему программный материал, грамотно и по существу излагающему его, который не допускает существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении поставленной задачи.

Оценка «удовлетворительно» выставляется студенту, который знает только основной программный материал, но не усвоил особенностей, допускает в ответе неточности, некорректно формулирует основные законы и правила, затрудняется в выполнении практических задач.

Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает в ответе существенные ошибки, с затруднениями выполняет практические задания.

Форма контроля	За одну работу		Всего	
	Мин. баллов	Макс. баллов	Мин. баллов	Макс. баллов
Текущий контроль:				
Активная работа на занятии	0,25	0,5	15	18
Выполнение домашнего задания	0,13	0,25	4,5	9
Выполнение заданий лабораторной работы	0,25	0,5	15	18
Выполнение заданий самостоятельной работы	1	3	1	3
Выполнение теста	0,3	0,6	1,5	3
Ответ на коллоквиуме	3	6	3	6
Промежуточная аттестация (экзамен)			20	43
Итого за семестр /экзамен			60	100

9. Учебно-методическое и информационное обеспечение дисциплины

9.1. Основная литература

- Маркин, А. В. Web-программирование : учебник / А. В. Маркин. — 2-е изд. — Москва : Ай Пи Ар Медиа, 2024. — 286 с. — ISBN 978-5-4497-3244-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/141273.html>
- Беликова С.А. Основы HTML и CSS: проектирование и дизайн веб-сайтов: учебное пособие по курсу «Web-разработка» / С.А. Беликова, А.Н. Беликов. — Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2020. — 174 с. — ISBN 978-5-9275-3435-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/100186.html>
- Введение в СУБД MySQL: учебное пособие. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 228 с. — ISBN 978-

- 5-4497-0912-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/102004.html>
4. Брылёва А.А. Программные средства создания интернет-приложений: учебное пособие / А.А. Брылёва. — Минск: Республиканский институт профессионального образования (РИПО), 2019. — 380 с. — ISBN 978-985-503-934-2. — Текст: электронный // Электроннобиблиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/94327.html>
5. Титов В.А. Разработка WEB-сайта средствами языка HTML: учебное пособие / В.А. Титов, Г.И. Пещеров. — Москва: Институт мировых цивилизаций, 2018. — 184 с. — ISBN 978-5-9500469-3-3. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/80643.html>

9.2. Дополнительная литература

1. Диков А.В. Клиентские технологии веб-дизайна. HTML5 и CSS3: учебное пособие / А.В. Диков. — СПб: Лань, 2019. — 188 с.
2. Шабашов В.Я. Организация доступа к данным из PHP приложений для различных СУБД: учебное пособие / В.Я. Шабашов. — Москва, Берлин: Директ-Медиа, 2019. — 122 с.
3. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS / Р. Никсон. — СПб: Питер, 2013.
4. Разработка Web-приложений на PHP и MySQL: Пер. с англ. / Л. Томсон, Л. Веллинг. — К.: Издательство «ДиаСофт», 2012. — 672 с.
5. Коржинский С.Н. Настольная книга Web-мастера: эффективное применение HTML, CSS и JavaScript. — М.: Издательский дом «КноРус», 2011. — 320 с.
6. Кириченко А.В. Динамические сайты на HTML, CSS, Javascript И Bootstrap. Практика, практика и только практика / А.В. Кириченко, Е.В. Дубовик. — Санкт-Петербург: Наука и Техника, 2018. — 272 с. — ISBN 978-5-94387-763-6. — Текст: электронный // Электроннобиблиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/77578.html>
7. HTML, CSS, SCRATCH, PYTHON.

9.3. Программное обеспечение

1. Microsoft Office 2010 Russian Academic OPEN 1 License (бессрочная), (лицензия 49512935);
2. Microsoft Sys Ctr Standard Sngl License/Software Assurance Pack Academic License 2 PROC (бессрочная), (лицензия 60465661)
3. Microsoft Win Home Basic 7 Russian Academic OPEN (бессрочная), (лицензия 61031351),
4. Microsoft Office 2010 Russian Academic OPEN, (бессрочная) (лицензия 61031351),
5. Microsoft Windows Proffesional 8 Russian Upgrade Academic OPEN (бессрочная), (лицензия 61031351),
6. Microsoft Internet Security&Accel Server Standart Ed 2006 English Academic OPEN, (бессрочная), (лицензия 41684549),
7. Microsoft Office Professional Plus 2010 Russian Academic OPEN, (бессрочная), (лицензия 60939880),
8. Microsoft Windows Server CAL 2008 Russian Academic OPEN, (бессрочная), (лицензия 60939880),
9. Microsoft Windows 10 Pro, 64 bit, Rus, OEM, Операционная система
10. Неисключительное право на использование ПО Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition.
11. Неисключительное право на использование ПО Kaspersky Security для виртуальных и облачных сред, Server, VirtSvr, License, Education Renewal
12. ABBYYFineReader 11 Professional Edition, (бессрочная), (лицензия AF11-2S1P01-102/AD),

13. Microsoft Volume Licensing Service, (бессрочная), (лицензия 62824441),
14. Microsoft Windows Pro 64bit OEM, (бессрочная), контракт № 6-ОАЭФ2014 от 05.08.2014
15. Visual Studio Professional
16. «Антиплагиат. ВУЗ». Лицензионный договор № 5044 от 14.05. 2022 года (ежегодное продление).

9.4. Профессиональные базы данных и информационные справочные системы современных информационных технологий

1. Информационная система «Единое окно доступа к образовательным ресурсам. Раздел Информатика и информационные технологии» (<https://habr.com/>)
2. Крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки- (<https://github.com/>)
3. База книг и публикаций Электронной библиотеки "Наука и Техника" (<http://www.n-t.ru>)
4. Электронная библиотечная система ZNANIUM.COM (<http://znanium.com/>)
5. Электронная библиотечная система «BOOK.ru» издательства «КноРус медиа» (<https://www.book.ru/>)
6. Журнал «КомпьютерПресс» (www.compress.ru)
7. Издательство «Открытые системы» (www.osp.ru)
8. Издание о высоких технологиях (www.cnews.ru)
9. Справочно-правовая система «Консультант Плюс» (<http://www.consultant.ru>)
10. Сайт о программировании (<https://metanit.com/>)
11. Электронная библиотечная система eLIBRARY.RU (<http://www.elibrary.ru>)
12. Электронная библиотечная система IPRbooks (<http://www.iprbookshop.ru>)
13. Электронная библиотечная система Юрайт (<http://www.biblio-online.ru>)

10. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

Учебные и учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации: *Для слепых и слабовидящих:*

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
- для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
- письменные задания оформляются увеличенным шрифтом;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере. *Для глухих и слабослышащих:*

- лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования;
- письменные задания выполняются на компьютере в письменной форме;
- экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

Для лиц с нарушениями опорно-двигательного аппарата:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации: ***Для слепых и слабовидящих:***

- в печатной форме увеличенным шрифтом; - в форме электронного документа; - в форме аудиофайла. ***Для глухих и слабослышащих:***
- в печатной форме;
- в форме электронного документа. ***Для обучающихся с нарушениями опорно-двигательного аппарата:***
- в печатной форме;
- в форме электронного документа; - в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения: ***для слепых и слабовидящих:***

- автоматизированным рабочим местом для людей с нарушением зрения;
- при необходимости обучающимся предоставляется увеличивающее устройство, допускается использование увеличивающих устройств, имеющихся у обучающихся; ***для глухих и слабослышащих:***
- автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих;
- акустический усилитель и колонки; ***для обучающихся с нарушениями опорно-двигательного аппарата:***
- передвижными, регулируемые эргономическими партами СИ-1;
- компьютерной техникой со специальным программным обеспечением.

11. Материально-техническое обеспечение дисциплины (модуля)

Для преподавания и изучения дисциплины используется лекционная аудитория, обеспеченная мультимедиа проектором и сопутствующим оборудованием, интерактивной доской. Используются УМК дисциплины (на бумажном и электронном носителях), фонд

научной библиотеки университета, методические и учебно-методические материалы кафедры информатики. ***К рабочей программе прилагаются:***

Приложение 1 – Фонд оценочных средств для проведения аттестации уровня сформированности компетенций обучающихся по дисциплине (модулю);

Приложение 2 – Методические указания для обучающихся по освоению дисциплины (модуля).