

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Лиодинович
Должность: Ректор
Дата подписания: 25.03.2026 16:00:29
Уникальный программный ключ:
5cf0d6f89e80f49a334f6a4ba58e91f3326b9926



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

Институт кибербезопасности и цифровых технологий

Региональный партнёр

ФГБОУ ВО

«Дагестанский государственный технический университет»



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Б1.В.12 Web-программирование

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Направленность (профиль подготовки): «Прикладной искусственный интеллект»

Квалификация выпускника бакалавр

Форма обучения очная

Махачкала 2023

ПАСПОРТ

фонда оценочных средств по дисциплине Б1.В.12 Web-программирование

1. Результаты обучения по дисциплине:

Код	Наименование компетенции	Индикатор достижения компетенции (закрепленный за дисциплиной)	В результате освоения дисциплины обучающийся должен:	Другая дисциплина (дисциплины)/практика, участвующая в формировании компетенции
ПК-1	Способен проектировать интеллектуальное программное обеспечение для решения практических задач	ПК-1.1. Осуществляет проектирование компонентов программного обеспечения с элементами искусственного интеллекта	Знать основы проектирования компонентов программного обеспечения. Владеть средствами проектирования ПО с элементами искусственного интеллекта.	Проектирование встраиваемых систем ИИ Технологии программирования Безопасность современного программного обеспечения/ Методы и средства защиты компьютерной информации Технологии семантического Web Базы данных и знаний программно-аппаратных комплексов вычислительной техники Разработка кроссплатформенных приложений Разработка мобильных приложений Высокопроизводительные вычисления/ Параллельное программирование Учебная (ознакомительная) практика, Производственная (проектно-технологическая)
		ПК-1.2. Создаёт варианты реализации компонент ПО на основе анализа предъявляемых требований	Знать возможности современных и перспективных средств разработки программных продуктов, технических средств. Уметь вырабатывать варианты реализации требований.	

2. Программа оценивания контролируемой компетенции:

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции / индикатора	Наименование оценочного средства
1	Тема 1-5	ПК-1	Защита лабораторных работ Защита курсовой работы Вопросы экзамена
Форма промежуточной аттестации в 7 семестре – (экзамен)			

Вопросы (задания) для экзамена

по дисциплине: Б1.В.12 Web-программирование

Вопросы:

1. Сервер, сайт, домашняя страничка
2. Элементы web-страницы
3. Верстка сайта. Виды подходов к верстке, преимущества, недостатки.
4. HTML-ссылки, таблицы, рисунки
5. HTML-формы
6. Протокол HTTP
7. Get, Post запросы в HTTP
8. Структура URL
9. Типы данных в JavaScript
10. Методы и свойства объекта Array (JavaScript)
11. Методы и свойства объекта String (JavaScript)
12. Методы объекта Data (JavaScript)
13. Методы объекта Math (JavaScript)
14. BOM и DOM – объектная модель браузера и документа
15. Каскадные таблицы стилей. Принцип наследования
16. Каскадные таблицы стилей. Свойства шрифта
17. Каскадные таблицы стилей. Свойства текста
18. Каскадные таблицы стилей. Свойства цвета и фона
19. Каскадные таблицы стилей. Группировка селекторов
20. Каскадные таблицы стилей. Идентификаторы
21. Каскадные таблицы стилей. Классы
22. Каскадные таблицы стилей. Виды селекторов и приоритеты.
23. Каскадные таблицы стилей. Свойства списков
24. Каскадные таблицы стилей. Селекторы
25. JSX, для чего он нужен?
26. В чем разница VDOM и DOM?
27. Функциональные и классовые компоненты React
28. React hooks
29. Создание HTTP-сервера на Node.js

Задания:**Основы HTML****1. Форматирование символов**

- Создайте HTML-документ, содержащий текст, заданный преподавателем
- Сделайте несколько копий текста
- Определите для трех копий шрифты: Arial, Verdana, Time Roman
- Для следующих трех копий задайте разные цвета шрифта
- Для следующих копий задайте разный размер шрифта
- К каждому из экземпляров текста сделайте заголовок, используя соответствующий тег

2. Форматирование фрагментов текста

- Создайте HTML-документ, содержащий текст, заданный преподавателем
- Сделайте несколько копий текста
- Для первой копии задайте начертание курсив
- Вторая копия полужирным шрифтом
- Третья копия подчеркнута
- Для следующих копий задайте разный размер шрифта
- Используйте подстрочные и надстрочные символы для задания простых формул

3. Форматирование текста

- Создайте HTML-документ, содержащий текст, заданный преподавателем
- Сделайте несколько копий текста
- Для каждого из фрагментов текста выберите различные способы выравнивания

4. Форматирование символов. Навигация между документами

- Создайте документ, содержащий информацию о себе
- Создайте документ, содержащий теоретический материал по одному из разделов курса "Информатика и программирование" либо курса "Высокоуровневые методы информатики" (Основы HTML). Раздел должен быть согласован с преподавателем
- Создайте документ, иллюстрирующий теоретический материал, используйте графический редактор
- Создайте документ, содержащий либо программу, либо код HTML. Программа должна быть с отступами и текст представлен моноширинным начертанием
- Свяжите документы между собой с помощью гиперссылок по схеме, предложенной преподавателем

5. Форматирование документа. Навигация в одном документе

- Подготовьте HTML- документ, представляющий фрагмент записной книжки. В этом документе для каждого студента группы должен соответствовать один абзац (тег P), начинающийся с фамилии, имени и отчества. Далее в том же абзаце поместите электронный адрес, и, возможно, другие сведения
- Фамилии в документе должны располагаться в алфавитном порядке
- Вставьте в тех местах, где сменяется первая буква, горизонтальную линию (тег HR), расположенную по центру
- Вставьте перед горизонтальной линией гиперссылку на начало документа
- Перед первой фамилией, начинающейся на новую букву, вставьте заголовок, состоящий из одной этой буквы
- В начале документа создайте текстовое меню, каждый пункт которого состоит из одной буквы, при щелчке по гиперссылке обеспечивается переход к первой фамилии, начинающейся на соответствующую букву
- Проверьте, как работают гиперссылки в HTML- документе

6. Использование горизонтальных линий
 - Создайте HTML-документ, содержащий текст, заданный преподавателем
 - Сделайте несколько копий текста
 - Для разделения получившегося текста используете горизонтальные линии, как рельефные, так и вдавленные
 - Линии должны быть разной длины, толщины, цвета
 - Используйте разные способы выравнивания текста, разный цвет шрифта и линий
7. Организация ссылок на другие документы
 - Создайте несколько HTML-документов раскрывающих тему, определенную преподавателем
 - Организуйте связи между документами с помощью ссылок
 - Продемонстрируйте созданный электронный архив
8. Организация ссылок внутри документа
 - Создайте HTML-документ, содержащий текст, заданный преподавателем
 - В начале документа сформируйте оглавление
 - Организуйте ссылки внутри документа таким образом, что в конце каждого раздела можно вернуться к оглавлению, либо перейти к предыдущему разделу, либо к следующему разделу
9. Формирование нумерованных списков
 - Создайте список, соответствующий оглавлению рассмотренной главы. Список должен быть трехуровневым
 - Создайте список, являющийся оглавлением документа, предложенного преподавателем
 - Создайте список определений по теме, предложенной преподавателем
10. Создание маркированных списков
 - Создайте список, соответствующий оглавлению рассмотренной главы. Список должен быть маркированным
 - Создайте многоуровневый маркированный список
11. Создание списков-определений
 - Создайте список определений по заданной теме
12. Создание списков в HTML
 - Создайте нумерованный список для рекомендуемой литературы
 - Создайте маркированный список для перечисленных разделов учебного материала
 - Создайте список студентов вашей подгруппы, используя в качестве маркера списка графическое изображение
 - Создайте список определений для указанных преподавателем терминов
 - Создайте оглавление в начале документа по разделам, соответствующим заданиям. Используйте список в качестве оглавления, организуйте навигацию по разделам документа. В конце каждого раздела предусмотрите возврат к оглавлению
13. Создание простых таблиц
 - Создайте таблицу расписания экзаменов. В таблице указывается название предмета, фамилия преподавателя, дата и время экзамена, и номер аудитории
 - Подготовьте несколько вариантов расписания, варьируя разную толщину рамки вокруг таблицы, расстояние между рамками смежных ячеек, расстояние между рамкой ячейки и данными внутри ячейки
 - Для расписания используйте разные способы расположения заголовка таблицы
14. Разноцветные таблицы
 - Создайте таблицу, определите различный цвет для ячеек таблицы
 - Выберите фоновое изображение для всей таблицы, строк и ячеек
 - Поместите изображение в таблицу

- Создайте таблицу, в которой вставлено изображение и текст
 - Создайте доску с расставленными шашками
15. Использование таблиц при размещении разных элементов Web-страницы
- Создайте таблицу, представляющую шахматную доску
 - Создайте таблицу для представления шахматной доски с расставленными фигурами
 - Создайте документ, в котором информация располагается определенным образом
 - Создайте таблицу, в левой части которой располагается оглавление, в правой части - текст занятия, сверху заголовков
 - Используйте таблицу при создании титульной страницы курса лекций.
16. Сложные таблицы
- Создайте таблицу, в которой объединены строки
 - Создайте таблицу с объединенными ячейками
17. Таблицы в HTML
- Создайте таблицу, каждая строка которой содержит номер книги, фамилию автора, название книги, название издательства, год издания. В таблице книги должны быть упорядочены по годам издания
 - Создайте таблицу, в которой хранится изображение. Предусмотрите разный цвет фона для документа, таблицы, ячейки. Таблица должна иметь рамку толщиной в 20 пиксель. Предусмотрите отступ содержимого ячейки от границы сетки
 - Создайте таблицу, представляющую шахматную доску с расставленными шашками. Таблица должна быть снабжена нумерацией, принятой в шахматах, т.е. нумерация A1 соответствует черной клетке таблицы
 - Создайте таблицу для построения страницы, содержащей одно и тоже изображение в верхней части экрана, в центре и в нижней части экрана
 - Создайте таблицу, которая позволяет представить текст в две колонки. Текст внутри прижат к верхней границе ячейки
 - Создайте таблицу со сложной структурой, содержащую расписание занятий

Описание показателей и критериев оценивания с указанием шкалы оценивания для очной и других форм обучения (с применением больно-рейтинговой системы и/или без ее использования):

Оцениваются следующие показатели: понимание вопросов, правильность, полнота и логическое изложение ответов.

Оценка по дисциплине складывается из текущего рейтинга и рейтинга промежуточной аттестации.

Зачетный рейтинг промежуточной аттестации определяется следующим образом:

Ответы на 1, 2 вопрос – до 20 баллов, выполнение 3 задания – до 10 баллов, дополнительные вопросы в рамках курса до 10 баллов.

Оценивание ответов на любой из теоретических вопросов:

8-10 баллов выставляется, если студент демонстрирует полное понимание вопросов, правильность ответов, полное и логически последовательное изложение материала.

6-8 баллов выставляется, если студент демонстрирует: значительное понимание вопросов, правильность, но недостаточную полноту ответов на заданные теоретические вопросы; допущение неточности ответа;

4-6 баллов выставляется, если студент демонстрирует: понимание вопросов, по существу излагает материал, но не усвоил его деталей, есть погрешности в ответах; допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении материала;

Менее 4 баллов выставляется, если студент демонстрирует: непонимание вопросов; студент не знает значительной части материала, не ответил на дополнительные вопросы или отказался от ответов на вопросы и задания.

Оценивание третьего задания:

9-10 баллов выставляется, если студент демонстрирует полное понимание заданий, правильность ответов; полное, точное и логически последовательное изложение материала;

7-8 баллов выставляется, если студент демонстрирует: значительное понимание заданий, правильность, но недостаточную полноту ответов на заданные задания; допущение неточности ответа;

5-6 баллов выставляется, если студент демонстрирует: понимание заданий, основные этапы задания выполнены, но есть погрешности в ответах

Менее 5 баллов выставляется, если студент демонстрирует: непонимание заданий; основные шаги задания не выполнены или выполнены неправильно, не ответил на дополнительные вопросы или отказался от ответов на вопросы и задания.

Минимальный балл текущего рейтинга в соответствии с положением о рейтинге равен 36, максимальный – 60.

Минимальный балл зачетного рейтинга с оценкой в соответствии с положением о рейтинге равен 24.

В итоге по курсу, суммируя итоги текущего рейтинга и рейтинга промежуточной аттестации:

- оценка «отлично» выставляется обучающемуся, если он набрал 87-100 баллов;
- оценка «хорошо» выставляется обучающемуся, если он набрал 73-86 баллов;
- оценка «удовлетворительно» выставляется обучающемуся, если он набрал 60-72 балла;
- оценка «неудовлетворительно» выставляется обучающемуся, если он набрал менее 60 баллов;

Вопросы для защиты лабораторных работ и контроль выполнения лабораторных работ

по дисциплине: **Б1.В.12 Web-программирование**

Лабораторная работа 1.

Проверка хода выполнения лабораторной работы.

Вопросы для защиты лабораторной работы:

Переменные и числа JavaScript

- Даны 2 константы, `firstValue`, равная 121 и `secondValue`, значение которой равно 300. Необходимо сделать переменную, значение которой будет равно результату деления `secondValue` на `firstValue`, округленного до 3х знаков после запятой. Вывести данную переменную в консоль.
- Даны 2 константы, `firstValue`, равная 32 и `secondValue`, значение которой равно 150. Необходимо сделать переменную, значение которой будет равно результату деления `firstValue` на `secondValue`, округленного в большую сторону. Вывести данную переменную в консоль.
- Даны 2 константы, `firstValue`, равная 0.2 и `secondValue`, значение которой равно 123. Необходимо сделать переменную, значение которой будет равно результату умножения `firstValue` на `secondValue`, округленного в меньшую сторону. Вывести данную переменную в консоль.
- Даны 2 константы, `firstValue`, равная 4 и `secondValue`, значение которой равно 5. Необходимо сделать переменную, значение которой будет равно результату суммы квадратов `firstValue` и `secondValue`. Вывести данную переменную в консоль.
- Необходимо получить случайное число, округлить его до двух знаков после запятой. После чего вывести число в консоль.
- Даны 2 константы, это `firstValue`, равная 1 и `secondValue`, равная 10. Необходимо написать код, задача которого получить случайное число в заданном интервале. Интервал формируют константы. Вывести полученное число в консоль.

Строки JavaScript

- Дана строка 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.'. Ваша задача проверить, встречаются ли в данной строке слова: `incidunt`, `dolore`, `exercitation`, `world`, `tea`. Если слово встретилось — вывести в консоль позицию слова в строке. Если слово не обнаружилось — вывести в консоль -1 соответственно.
- Вам дается любая строка в верхнем регистре. Необходимо вывести в консоль исходную строку, преобразовав первую букву в нижний регистр.
- Вам дана строка 'Hello world', вам нужно вывести в консоль строку, содержащую коды символов второго слова. Коды должны разделяться пробелами.
- Даны 2 строки — 'Hello world' и 'Web programming'. Ваша задача сделать из них одну строку 'Hello programming' и вывести ее в консоль.
- Дана строка 'Start web right now'. Ваша задача сделать из нее строку 'Start web programming right now' и вывести ее в консоль.
- Вам дана строка 'Привет, как дела?'. Вам нужно составить из нее слово «ветка» и вывести в консоль.

- Вам дается любая строка в нижнем регистре. Необходимо вывести в консоль исходную строку, преобразовав первую букву в верхний регистр.
- Вам дана строка 'Два слова'. Сложите коды всех символов второго слова и выведите в консоль итоговое значение.

Условия, ветвления JavaScript

1. Даны переменные:

```
const firstCheck = 1,
const secondCheck = true,
let accumulator = null.
```

Задание — Создать ветвление, в котором:

1) Если firstCheck строго равно 1, то открывается блок, в котором идет вторая проверка. Иначе в консоль выводится “false”

2) Вторая проверка — если secondCheck строго равно true, в консоль выводится 'Ассерт', если данное условие не выполняется, то в консоль выводится 'Decline'.

2. Даны 3 константы: first, second, third. Ваша задача написать код, определяющий максимальное значение. После чего вывести значение в консоль.

3. Даны 4 константы: first, second, third, fourth. Ваша задача написать код, который отсортирует числа в порядке возрастания. Каждое число в порядке возрастания вывести в консоль.

4. Необходимо написать код, который по номеру дня определяет его название. Ввести 2 примера, ответы вывести в консоль в формате - («номер дня недели» - «название дня недели»).

Пример — «1 – Понедельник»

5. Необходимо написать код, который по номеру месяца определяет его название. Ввести 2 примера, ответы вывести в консоль в формате - («номер месяца» - «название месяца»).

Пример — «1 – Январь»

6. Даны 4 константы: first, second, third, fourth. Ваша задача написать код, который отсортирует числа в порядке убывания. Каждое число в порядке убывания вывести в консоль.

Циклы JavaScript

- У вас есть константа processedNumber. Напишите код, который определяет факториал заданного в константе натурального числа.
- У вас есть две переменные, firstValue и secondValue. Напишите код, который определяет наибольший общий делитель.
- У вас есть две переменные, firstValue и secondValue. Напишите код, который определяет наименьшее общее кратное.
- У вас есть переменная ticketNumber. Она содержит в себе шестизначный номер автобусного билета. Напишите код, который определяет, является ли номер автобусного билета «счастливым». Счастливым билетом является билет, сумма цифр которого на нечетных местах равна сумме цифр, расположенных на четных местах.
- У вас есть константа ticketNumber. Она содержит в себе шестизначный номер автобусного билета. Напишите код, который определяет, является ли номер автобусного билета «счастливым». Счастливым билетом является билет, сумма цифр которого на первых трех позициях равна сумме цифр, расположенных на последних трех местах.
- У вас есть константа processedNumber. Напишите код, который определяет все делители заданного натурального числа. Формирует строку, содержащую необходимые числа через запятую. После чего должен производиться вывод в консоль сформированной строки.

- Пример работы программы для числа 24 — '1, 2, 4, 6, 8, 12, 24'
- У вас есть константа `processedNumber`. Напишите код, который определяет, является ли заданное в константе число совершенным. Совершенным числом является натуральное число, равное сумме всех своих собственных делителей.

Функции JavaScript

- Напишите сценарий, который осуществляет визуальное удаление изображения с помощью рекурсии.
- Напишите сценарий, который для любого натурального числа строит его двоичное представление.
- Напишите сценарий, который вычисляет корень уравнения методом половинного деления с помощью рекурсии.
- Напишите сценарий, который осуществляет визуальное увеличение изображения с помощью рекурсии.
- Напишите сценарий, который рекурсивно определяет число Фибоначчи с номером `n`.
- Напишите сценарий, который для двух заданных натуральных чисел определяет их наибольший общий делитель.
- Напишите сценарий, в котором присутствует рекурсивная функция `sum(s, i, j)`. Необходимо проверить, является ли симметричной часть строки `s`, начинающаяся `i`-м и заканчивающаяся `j`-м ее элементами.
- Напишите рекурсивную функцию, «переворачивающую» заданное натуральное число.
- 9. Напишите рекурсивную функцию, которая формирует в обратном порядке текст, задаваемый пользователем в текстовом поле формы.

Объекты JavaScript

- Напишите объект, в котором находятся ключи с данными о человеке (фамилия, имя, отчество, возраст) и метод, реализующий вывод в консоль строки формата «Меня зовут “Фамилия” “Имя” “Отчество”, мне “Возраст” лет»
- У вас есть объект со случайными ключами, в количестве 7шт (придумайте их сами и дайте им любые значения). Полностью скопируйте объект в другую константу, так, чтобы оба одинаковых объекта лежали в разных ячейках памяти и не ссылались на одно и то же. (`объект1 !== объект2`). Вывести результат данного логического сравнения в консоль
- У вас есть 2 объекта, каждый имеет по 3 ключа, ключами являются числа внутри строк. Идут они по возрастанию от 1 до 6 соответственно. Значения — рандомные булевы значения (`true/false`).
- Пример первого объекта — `{'1': true, '2': false, '3': true}`.
- Ваша задача – сделать функцию, которая будет принимать в себя 2 параметра (это и будут ваши объекты). Задача функции - сделать строку, которая будет хранить в себе ключи из каждого объекта, значения которых равны `true`. Значения должны идти через запятую. Строку нужно вывести в консоль.
- У вас есть 3 объекта, у каждого по 2 ключа. Ключи и их значения придумайте сами. Ваша задача создать один новый объект, содержащий в себе все ключи с их значениями из заданных 3х объектов. Вывести итоговый объект в консоль.
- Вам дан объект. Ключами являются числа внутри строк. Идут они по возрастанию от 1 до 10 соответственно.
- Значения — рандомные булевы значения (`true/false`).
- Ваша задача — удалить с объекта все ключи, значения которых равны `false`. Итоговый объект вывести в консоль.

Массивы JavaScript

- Дан массив `array` равный `[-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]`. Необходимо создать константу и положить в нее отфильтрованный массив. Фильтрация производится по одному условию — если число больше нуля — оно попадает в массив. Вывести полученный массив в консоль.
- Дан массив `array` равный `[0, 1, 2, 3, 4, 5]`. Необходимо создать константу и скопировать в нее исходный массив. Единственное условие, которое необходимо соблюдать при копировании — каждое число в новом массиве должно быть больше числа исходного массива на собственный индекс. Вывести полученный массив в консоль.
- Дан массив `array` равный `[0, 1, 2, 3, 4, 5]`. Необходимо создать константу и скопировать в нее исходный массив. Единственное условие, которое необходимо соблюдать при копировании — каждое число в новом массиве должно быть больше числа исходного массива на значение предыдущего элемента массива. Вывести полученный массив в консоль.
- Дан массив `array` равный `[1, 0, 0, 0, 0, 0]`. Необходимо создать константу и скопировать в нее исходный массив. Единственное условие, которое необходимо соблюдать при копировании — каждое число в новом массиве (кроме начального элемента, он остается без изменений) должно равняться значению предыдущего элемента, умноженному на индекс текущего числа. Вывести полученный массив в консоль.
- Необходимо создать массив, заполнив его 10-ю нулями (заполнять нужно функционально). Так же необходимо создать константу и скопировать в нее исходный массив. Единственное условие, которое необходимо соблюдать при копировании — каждое число в новом массиве должно быть равным значению своего индекса. Вывести полученный массив в консоль.
- Дан массив `array` равный `[-5, -4, -3, -2, -1, 1, 2, 3, 4, 5]`. Необходимо найти сумму квадратов всех элементов и вывести значение в консоль.
- Дан массив `array` равный `[3, 8, 95, 64, 72, 55, 49, 300]`. Необходимо найти среднее арифметическое значение его элементов. Результат необходимо вывести в консоль
- Дан массив `array` равный `[180, 300, 750, 1050]`. Необходимо найти наибольший общий делитель для всех чисел данного массива. Результат вывести в консоль.

Работа с DOM

- Создайте прямоугольник. При клике на прямоугольник он должен плавно переместиться на 500 пикселей вправо от своей изначальной позиции.
- Создайте квадрат. При клике на квадрат он плавно должен превратиться в окружность.
- У вас есть пустая страница с тегом `div`, покрывающим 100% видимого в окне контента. И есть массив с цветами.
- `const colors = ['red', 'green', 'purple', 'black', 'white']`
- Ваша задача при клике на область страницы поменять цвет вашего `div` на один из элементов массива `colors`
- Создайте 3 прямоугольника. При клике на прямоугольник он должен начать плавно вращаться вокруг своей оси.
- Создайте окружность. При клике на нее она должна плавно переместиться в противоположный угол экрана (к примеру, она находится в левом верхнем углу, ее конечная позиция должна быть в правом нижнем углу)
- У вас есть страница с несколькими заголовками и параграфами. Также одна кнопка до текстов.
- Цвет фона — белый, цвет текста — черный. Ваша задача по нажатию на кнопку сделать плавную смену цветовой темы (Так, чтобы, цвет текста стал белым, а цвет

фона — черным). Кнопка рассчитана на неограниченное количество использований, т.е. Темы должны чередоваться.

Лабораторная работа 2.

Проверка хода выполнения лабораторной работы.

Вопросы для защиты лабораторной работы:

- Что такое nodeJS и для чего он используется?
- Что такое npm и для чего он используется?
- Зачем нужен package.json файл?
- Что хранится в папке node_modules/?
- Как установить какой-либо пакет, используя пакетный менеджер npm?
- Как выполнить установку всех пакетов из их списка в файле package.json?
- Как выполнить запуск скрипта, описанного в файле package.json?
- Что такое VDOM и DOM? Для чего они нужны?
- Что такое JSX и для чего он нужен?
- Опишите принцип добавления корневого React-компонента в корневой html-документ.
- Опишите последовательность действий для создания нового функционального компонента.
- Что такое React hooks и для чего они нужны?
- Опишите принцип работы хука useState. Для чего он нужен?
- Опишите принцип работы хука useEffect. Для чего он нужен?
- Расскажите, в чем различия эффектов с очисткой и без.

Лабораторная работа 3.

Проверка хода выполнения лабораторной работы.

Вопросы для защиты лабораторной работы:

- Что такое Node.js?
- Как создать HTTP-сервер на Node.js? Приведите пример кода.

Описание показателей и критериев оценивания с указанием шкалы оценивания для очной и других форм обучения (с применением больно-рейтинговой системы и /или без ее использования)

Оцениваются следующие показатели: знание теоретических основ лабораторной работы, умение применить их на практике, обосновать используемое решение, выполнение в установленные сроки. В рамках защиты по каждой лабораторной работе задается несколько вопросов.

Для лабораторных работ 1-6

8 баллов выставляется, если студент выполнил работы в установленный срок, правильно и полно отвечает на вопросы по каждой лабораторной работе, объясняет их на примерах, связывает с программной реализацией.

6-7 баллов выставляется, если студент отвечает на вопросы недостаточно полно или с неточностями, или не отвечает на часть заданных вопросов, не может объяснить их на примере, есть недочеты в лабораторной работе.

0-5 балла выставляется, если студент не отвечает на вопросы, не может объяснить их на примере, лабораторная работа выполнена некорректно.

Для лабораторной работы 7

10-12 баллов выставляется, если студент выполнил работы в установленный срок, правильно и полно отвечает на вопросы по каждой лабораторной работе, объясняет их на примерах, связывает с программной реализацией.

6-9 баллов выставляется, если студент отвечает на вопросы недостаточно полно или с неточностями, или не отвечает на часть заданных вопросов, не может объяснить их на примере, есть недочеты в лабораторной работе.

0-5 балла выставляется, если студент не отвечает на вопросы, не может объяснить их на примере, лабораторная работа выполнена некорректно.

**Оформление сведений о дополнениях и изменениях, внесенных в ФОС дисциплины
Б1.В.12 Web-программирование**

Сведения о дополнениях и изменениях, внесенных в ФОС дисциплины

Учебный год	Решение кафедры (№ протокола, дата)	Внесенные в ФОС дополнения и изменения	Подпись заведующего кафедрой