Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гриб Владислав Валерьевич
Должность: Ректор
Дата подписания: 14.10.2026 17:08:07
Уникальный программный ключ:
637517d24e103c3db032acf37e839d98ec1c5bb2f5eb89c29abfcd7f43985447

Образовательное частное учреждение высшего образования
«МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ А.С. ГРИБОЕДОВА»
(ИМПЭ им. А.С. Грибоедова)

МЕЖДУНАРОДНЫЙ ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И БИЗНЕС-ИНФОРМАТИКИ

УТВЕРЖДАЮ
Директор Международного института информационных технологий и бизнес-информатики
_____ Т.В. Новикова
«17» декабря 2025г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по учебной дисциплине:
ЯЗЫК ПРОГРАММИРОВАНИЯ JAVA SCRIPT

по направлению подготовки:

38.03.05
БИЗНЕС-ИНФОРМАТИКА
(уровень бакалавриат)

направленность (профиль):
«Цифровая трансформация бизнеса»

Форма обучения: очная, очно-заочная

Москва

Фонд оценочных средств для дисциплины «Язык программирования JavaScript». Направление подготовки/специальность 38.03.05 Бизнес-информатика, направленность (профиль): «Цифровая трансформация бизнеса»/ Р.Р. Абзалимов – М.: ИМПЭ им. А.С. Грибоедова

Фонд оценочных средств является неотъемлемой частью рабочей программы дисциплины.

Разработчик: Р.Р. Абзалимов



Заведующий кафедрой _____/Т.В. Новикова, к. э. н., доцент

1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Настоящий Фонд оценочных средств (ФОС) по дисциплине является неотъемлемым приложением к рабочей программе дисциплины (РПД) «Язык программирования JavaScript». На данный ФОС распространяются все реквизиты утверждения, представленные в РПД по данной дисциплине.

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

Для определения качества освоения обучающимися учебного материала по дисциплине используются следующие оценочные средства:

№ п/п	Оценочное средство	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Тестирование	Вид контроля, позволяющий оценить изученный теоретический материал.	Вопросы для проведения тестирования
2	Практические задания	Вид контроля, позволяющий оценить умение обучающегося применять осваиваемую компетенцию в практических ситуациях и при решении производственных задач	Задания к практическому занятию
3	Контрольная работа	Вид контроля, позволяющий определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций	Задания контрольной работы
4	Самостоятельная работа	Вид контроля, позволяющий оценить проработку теоретического материала, изучение рекомендуемой литературы, выполнение практико-ориентированных заданий (заполнение таблиц, проведение сравнительного анализа, составление схем и др.), решение практических задач, создание презентаций, написание рефератов, подборку нормативного и иного материала и выполнение других заданий	Задания самостоятельной работы
5	Курсовая работа	Вид контроля, позволяющий выявить степень владения базовыми знаниями, умениями и навыками, необходимыми для обучения, и определить уровень владения новым материалом	Индивидуальные задания (темы) для курсовой работы
6	Зачет/Зачет с оценкой/Экзамен	Вид контроля, позволяющий выявить степень овладения знаниями, умениями и навыками, необходимыми для дальнейшего освоения образовательной программы подготовки	Вопросы для подготовки к зачету/зачету с оценкой/экзамену

3. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЕ

3.1. Сопроводительная информация.

Разработчик	Р.Р. Абзалимов
Кафедра	Прикладной информатики и информационных технологий
Наименование дисциплины	Язык программирования JavaScript
Факультет / институт	Институт международной экономики, лидерства и менеджмента
Направление подготовки / специальность	38.03.05 Бизнес-информатика
Количество вопросов в оценочных заданиях (диапазон)	От 35 до 70
Общее время тестирования (мин)	90
Общее количество вопросов/заданий в ФОС	70
Размещенность на сайте Университета примерного перечня вопросов, заданий ФОС – для подготовки обучающихся к прохождению оценки (да / нет)	Да

3.2. Характеристика оцениваемых компетенций

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
ПК-2.	Способен осуществлять оценку времени и трудоёмкости реализации требований к компьютерному программному обеспечению	**ИПК-2.1** Возможности современных и перспективных средств разработки программных продуктов и технических средств **ИПК-2.2** Выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению **ИПК-2.3 Владеть:** Методами осуществления оценки времени и трудоемкости реализации требований к компьютерному программному обеспечению
ПК-6.	Способен осуществлять проектирование баз данных	**ИПК-6.1** Методы и средства проектирования баз данных **ИПК-6.2** **ИПК-6.3 Владеть:** Навыками проектирования баз данных

4. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ТЕКУЩЕГО КОНТРОЛЯ

4.1. ТИПОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ.

Тесты содержат набор вопросов, в полном объеме охватывающие изученный теоретический материал по указанной теме. Выполнение тестов позволяет определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций. Индивидуальный тестовый сеанс для каждого обучающегося формируется по специальному алгоритму, обеспечивающему заданную тематическую структуру и пропорциональное наличие вопросов разного типа и сложности.

При формировании тестов необходимо использовать задания следующих типов:

Тип задания 1. Задания закрытого типа на установление соответствия.

Тип задания 2. Задания закрытого типа на установление последовательности.

Тип задания 3. Задания комбинированного типа, предполагающие выбор одного правильного ответа из предложенных

Тип задания 4. Задания комбинированного типа, предполагающие выбор нескольких ответов из предложенных

Тип задания 5. Задания открытого типа с развернутым ответом.

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td>Тема 1. Введение в JavaScript</td><td>ПК-2.</td><td>ИПК-2.1</td><td>1. Задание на установление соответствия (Тип 1)
Соотнесите базовые типы данных JavaScript с их описаниями:
<table><tr><th>Код</th><th>Тип данных</th><th>Код</th><th>Описание</th></tr><tr><td>А</td><td>number</td><td>1</td><td>Логический тип true/false</td></tr><tr><td>Б</td><td>string</td><td>2</td><td>Целочисленные и дробные числа</td></tr><tr><td>В</td><td>boolean</td><td>3</td><td>Строковый тип</td></tr><tr><td>Г</td><td>undefined</td><td>4</td><td>Неопределенное значение</td></tr><tr><td>Д</td><td>null</td><td>5</td><td>Пустое значение</td></tr></table>
ОТВЕТ: А-2, Б-3, В-1, Г-4, Д-5</td></tr>
<tr><td></td><td>ПК-2.</td><td>ИПК-2.1</td><td>2. Задание на установление последовательности (Тип 2)
Расположите этапы выполнения цикла for в правильной последовательности:
1. Проверка условия
2. Выполнение тела цикла
3. Инициализация переменной
4. Изменение переменной
5. Завершение цикла
ОТВЕТ: 3 → 1 → 2 → 4 → 1 → 5</td></tr>
<tr><td></td><td>ПК-2.</td><td>ИПК-2.1</td><td>3. Задание с выбором одного ответа (Тип 3)
Что выведет следующий код: console.log(2 + "2")?

4. Ошибку
ОТВЕТ: 2) "22" — в JavaScript при сложении числа и строки происходит конкатенация, число преобразуется в строку</td></tr>
</table>

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td></td><td>ПК-2.</td><td>ИПК-2.1</td><td>4. Задание с выбором нескольких ответов (Тип 4)
Какие из следующих операторов используются для сравнения в JavaScript?

ОТВЕТ: 1, 2, 4, 5</td></tr>
<tr><td></td><td>ПК-2.</td><td>ИПК-2.1</td><td>5. Задание открытого типа (Тип 5)
Изложите разницу между операторами == и === в JavaScript.
ОТВЕТ:
• == — нестрогое сравнение (с приведением типов)
• === — строгое сравнение (без приведения типов)
Примеры:

// true (приведение типов)
console.log(2 === "2"); // false (разные типы)
console.log(null == undefined); // true
console.log(null === undefined); // false</td></tr>
<tr><td>Тема 2
Функции</td><td>ПК-2.
ПК-6.</td><td>ИПК-2.1</td><td>6. Задание на установление соответствия (Тип 1)
Соотнесите понятия функций JavaScript с их описаниями:
<table><tr><th>Код</th><th>Понятие</th><th>Код</th><th>Описание</th></tr><tr><td>А</td><td></td><td></td><td>Функция без имени</td></tr><tr><td>Б</td><td></td><td></td><td>Объявление функции через переменную</td></tr><tr><td>В</td><td></td><td></td><td>Классическое объявление функции</td></tr><tr><td>Г</td><td></td><td></td><td>Функция, передаваемая как аргумент</td></tr><tr><td>Д</td><td></td><td></td><td>Замыкание - функция с запомненным окружением</td></tr></table>ОТВЕТ: А-3, Б-2, В-1, Г-4, Д-5</td></tr>
</table>

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
	ПК-2.	**ИПК-2.1**	**7. Задание с выбором одного ответа (Тип 3)** Что такое замыкание (closure) в JavaScript? пособ объявления функции ункция с доступом к переменным внешней функции етод для закрытия окна браузера ип данных для хранения значений **ОТВЕТ:** 2) Функция с доступом к переменным внешней функции — замыкание сохраняет лексическое окружение, в котором было создано
	ПК-2.	**ИПК-2.1**	**8. Задание с выбором нескольких ответов (Тип 4)** Какие способы объявления функций существуют в JavaScript? **ОТВЕТ:** 1, 2, 3
Тема 3 Объекты и массивы	ПК-2.	**ИПК-2.1**	**9. Задание открытого типа (Тип 5)** Опишите различия между объектами и массивами в JavaScript. В каких случаях предпочтительнее использовать каждый из этих типов данных? **ОТВЕТ:** • **Объекты**: для хранения именованных свойств, неупорядоченная коллекция • **Массивы**: для хранения упорядоченных списков, доступ по индексу **Примеры использования:** *// Объект для данных пользователя* const user = { name: "John", age: 30, city: "Moscow" }; *// Массив для списка элементов* const fruits = ["apple", "banana", "orange"];
	ПК-2.	**ИПК-2.1**	**10. Задание на установление последовательности (Тип 2)** Расположите этапы работы с массивом в правильной последовательности: оздание массива обавление элементов бращение к элементам зменение элементов спользование методов массива **ОТВЕТ:** 1 → 2 → 3 → 4 → 5

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
Тема 4. Объектная модель браузера	ПК-2.	**ИПК-2.1**	**Задание 11 (Тип 3 - Выбор одного ответа)** Какой объект BOM является главным и содержит все остальные объекты? 1. document 2. window 3. navigator 4. screen **Ответ:** 2) window — глобальный объект, представляющий окно браузера
Тема 4. Объектная модель браузера	ПК-2.	**ИПК-2.1**	**Задание 12 (Тип 4 - Выбор нескольких ответов)** Какие объекты входят в Browser Object Model (BOM)? 1. window 2. navigator 3. screen 4. history 5. location **Ответ:** 1, 2, 3, 4, 5
Тема 5. Объектная модель документа	ПК-2.	**ИПК-2.1**	**Задание 13 (Тип 5 - Открытый тип)** Опишите, что такое DOM и как осуществляется доступ к элементам страницы с помощью JavaScript. **Ответ:** DOM (Document Object Model) — программный интерфейс для работы с HTML-документом. Методы доступа: • getElementById() • getElementsByClassName() • getElementsByTagName() • querySelector(), querySelectorAll()
Тема 5. Объектная модель документа	ПК-2.	**ИПК-2.1**	**Задание 14 (Тип 1 - Установление соответствия)** Соотнесите методы DOM с их назначением: А) createElement() - 2) Создание нового элемента Б) appendChild() - 1) Добавление дочернего элемента В) removeChild() - 3) Удаление дочернего элемента Г) setAttribute() - 4) Установка атрибута

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			Д) addEventListener() - 5) Добавление обработчика события **Ответ:** А-2, Б-1, В-3, Г-4, Д-5
Тема 6. События и их обработка	ПК-2.	**ИПК-2.1**	**Задание 15 (Тип 3 - Выбор одного ответа)** Что такое всплытие событий (event bubbling) в JavaScript? 1. Создание нового события 2. Перехват события дочерним элементом 3. Передача события от целевого элемента вверх по DOM 4. Отмена действия по умолчанию **Ответ:** 3) Передача события от целевого элемента вверх по DOM — событие сначала обрабатывается на целевом элементе, затем на его родителях
Тема 6. События и их обработка	ПК-2. ПК-6.	**ИПК-2.1**	**Задание 16 (Тип 4 - Выбор нескольких ответов)** Какие способы назначения обработчиков событий существуют в JavaScript? 1. Через атрибут HTML 2. Через свойство DOM-элемента 3. С помощью addEventListener() 4. Через CSS-стили 5. С помощью setAttribute() **Ответ:** 1, 2, 3

Тема 1. Введение в JavaScript	ПК-2.	**ИПК-2.1**	**Задание 17 (Тип 5 - Открытый тип)**
			Напишите функцию, которая проверяет, является ли строка палиндромом. Учтите регистр и пробелы.
			Ответ:
			javascript
			function isPalindrome(str) {
			const cleanedStr = str.toLowerCase().replace(/\s/g, '');
			return cleanedStr === cleanedStr.split('').reverse().join('');
			}

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
Тема 2 Функции	ПК-2.	**ИПК-2.1**	**Задание 18 (Тип 2 - Установление последовательности)** Расположите этапы выполнения функции в правильной последовательности: 1. Объявление функции 2. Вызов функции 3. Передача аргументов 4. Выполнение тела функции 5. Возврат результата **Ответ:** 1 → 2 → 3 → 4 → 5
Тема 3. Объекты и массивы	ПК-2.	**ИПК-2.1**	**Задание 19 (Тип 3 - Выбор одного ответа)** Какой метод массива используется для преобразования каждого элемента? 1. filter() 2. map() 3. reduce() 4. forEach() **Ответ:** 2) map() — создает новый массив с результатами вызова функции для каждого элемента
Тема 4. Объектная модель браузера	ПК-2.	**ИПК-2.1**	**Задание 20 (Тип 4 - Выбор нескольких ответов)** Какие свойства объекта location используются для работы с URL? 1. href 2. protocol 3. host 4. pathname 5. search **Ответ:** 1, 2, 3, 4, 5
Тема 5. Объектная модель документа	ПК-2.	**ИПК-2.1**	**Задание 21 (Тип 5 - Открытый тип)** Напишите код, который создает новый элемент div, добавляет ему класс и текст, и вставляет его в body страницы. **Ответ:** javascript const newDiv = document.createElement('div'); newDiv.className = 'my-class';

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			newDiv.textContent = 'Новый элемент'; document.body.appendChild(newDiv);
Тема 6. События и их обработка	ПК-2.	**ИПК-2.1**	**Задание 22 (Тип 1 - Установление соответствия)** Соотнесите события с их описанием: А) click - 2) Щелчок мышью Б) load - 1) Загрузка страницы В) submit - 3) Отправка формы Г) keydown - 4) Нажатие клавиши Д) change - 5) Изменение значения элемента формы **Ответ:** А-2, Б-1, В-3, Г-4, Д-5
Тема 1. Введение в JavaScript	ПК-2.	**ИПК-2.1**	**Задание 23 (Тип 3 - Выбор одного ответа)** Что такое hoisting (поднятие) в JavaScript? 1. Поднятие переменных и функций в начало области видимости 2. Увеличение значения переменной 3. Поднятие элемента в DOM 4. Улучшение производительности кода **Ответ:** 1) Поднятие переменных и функций в начало области видимости — до выполнения кода интерпретатор "поднимает" объявления
Тема 2. Функции	ПК-2.	**ИПК-2.1**	**Задание 24 (Тип 4 - Выбор нескольких ответов)** Какие особенности имеют стрелочные функции? 1. Не имеют своего this 2. Не имеют arguments 3. Не могут быть конструкторами 4. Всегда возвращают значение 5. Имеют сокращенный синтаксис **Ответ:** 1, 2, 3, 5
Тема 3. Объекты и массивы	ПК-2. ПК-6.	**ИПК-2.1**	**Задание 25 (Тип 5 - Открытый тип)** Создайте объект "книга" с свойствами и методами. Добавьте метод для вывода информации о книге.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме

			Ответ:
			javascript
			const book = {
			title: "JavaScript для начинающих",
			author: "Иван Иванов",
			year: 2023,
			pages: 350,
			getInfo: function() {
			return `${this.title} - ${this.author} (${this.year}), ${this.pages} стр.`;
			}
			};

Тема 4. Объектная модель браузера	ПК-2.	**ИПК-2.1**	**Задание 26 (Тип 2 - Установление последовательности)** Расположите этапы загрузки веб-страницы в правильной последовательности: 1. Парсинг HTML 2. Запрос к серверу 3. Построение DOM 4. Загрузка ресурсов 5. Выполнение JavaScript **Ответ:** 2 → 1 → 3 → 4 → 5
Тема 5. Объектная модель документа	ПК-2.	**ИПК-2.1**	**Задание 27 (Тип 3 - Выбор одного ответа)** Какой метод используется для делегирования событий? 1. addEventListener() на родительском элементе 2. onclick на каждом дочернем элементе 3. createEvent() 4. dispatchEvent() **Ответ:** 1) addEventListener() на родительском элементе — обработчик на родителе + проверка event.target
Тема 6. События и их	ПК-2.	**ИПК-2.1**	**Задание 28 (Тип 4 - Выбор нескольких ответов)** Какие методы объекта Event используются для управления событиями?

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
обработка			1. preventDefault() 2. stopPropagation() 3. stopImmediatePropagation() 4. cancelBubble() 5. stopEvent() **Ответ:** 1, 2, 3

Комплексное задание	ПК-2.	**ИПК-2.1**	**Задание 29 (Тип 5 - Открытый тип)**
			Напишите функцию, которая создает список элементов на странице на основе массива данных.
			Ответ:
			javascript
			function createList(items, containerId) {
			const container = document.getElementById(containerId);
			const list = document.createElement('ul');
			items.forEach(item => {
			const listItem = document.createElement('li');
			listItem.textContent = item;
			list.appendChild(listItem);
			});
			container.appendChild(list);
			}

Итоговое задание	ПК-2.	**ИПК-2.1**	**Задание 30 (Тип 5 - Открытый тип)**
			Создайте интерактивную форму с валидацией. Форма должна содержать поля: имя, email, пароль.
			Ответ:
			javascript
			document.getElementById('myForm').addEventListener('submit', function(event) {
			event.preventDefault();
			// Валидация полей формы
			if (name.length < 2) alert('Имя должно содержать минимум 2 символа');

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме

			if (!email.includes('@')) alert('Введите корректный email');
			if (password.length < 6) alert('Пароль должен содержать минимум 6 символов');
			alert('Форма успешно отправлена!');
			});

Критерии оценивания тестового задания:

Оценка	Критерии оценивания
отлично	от 90 до 100 % правильно выполненных заданий
хорошо	от 70 до 89 % правильно выполненных заданий
удовлетворительно	от 50 до 69 % правильно выполненных заданий
неудовлетворительно	менее 50 % правильно выполненных заданий

4.2 ТИПОВЫЕ ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Практические задания должны отражать умение обучающегося применять осваиваемую компетенцию в практических ситуациях и при решении производственных задач.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

Тема 1. Введение в JavaScript	ПК-2.	**ИПК-2.2**	**Задание 1 (Тип 5 - Практическая программа)**
			Программа работы с базовыми типами данных
			javascript
			function demoDataTypes() {
			let num = 10, floatNum = 3.14, str = "Hello", bool = true;
			console.log(`number: ${num}, float: ${floatNum}, string: "${str}", boolean: ${bool}`);
			console.log(`Types: ${typeof num}, ${typeof floatNum}, ${typeof str}, ${typeof bool}`);
			let converted = num + str;
			console.log(`Конкатенация: ${converted}, тип: ${typeof converted}`);
			let userInput = parseFloat(prompt("Введите число:"));
			console.log(`Введено: ${userInput}, тип: ${typeof userInput}`);
			}

Тема 1. Введение в JavaScript	ПК-2.
	ПК-6.
		ИПК-2.2	**Задание 2 (Тип 5 - Практическая программа)**
			Программа с операторами сравнения и ветвлениями
			javascript
			function demoBranching() {
			let x = 10, y = 5;
			console.log(`Числа: ${x}, ${y}`);
			if (x > y) console.log(`${x} > ${y}`);
			else if (x < y) console.log(`${x} < ${y}`);
			else console.log(`${x} = ${y}`);
			
			let age = parseInt(prompt("Введите возраст:"));
			let status = age >= 18 ? "взрослый" : "ребенок";
			console.log(`Статус: ${status}`);
			
			let value = "123";

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)
			if (typeof value === "string") console.log(`"${value}" - это строка`); }

Тема 1. Введение в JavaScript	ПК-2.	**ИПК-2.2**	**Задание 3 (Тип 5 - Практическая программа)**
			Программа с циклами for и while
			javascript
			function demoLoops() {
			console.log("Цикл for:");
			for (let i = 1; i <= 3; i++) console.log(i);
			console.log("Цикл while:");
			let n = 1;
			while (n <= 3) {
			console.log(n);
			n++;
			}
			let nums = [2, 4, 6];
			console.log("For...of по массиву:");
			for (let num of nums) console.log(`Квадрат ${num}: ${num * num}`);
			console.log("Break и continue:");
			for (let i = 1; i <= 5; i++) {
			if (i === 3) continue;
			if (i === 5) break;
			console.log(i);
			}
			}

Тема 2 Функции	ПК-2.	**ИПК-2.2**	**Задание 4 (Тип 5 - Практическая программа)** Программа с различными типами функций javascript

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			// Function Declaration
			function greet(name) {
			return `Привет, ${name}!`;
			}
			// Function Expression
			const multiply = function(a, b) {
			return a * b;
			};
			// Arrow Function
			const divide = (a, b) => a / b;
			function functionsDemo() {
			console.log(greet("Анна"));
			console.log(`Умножение: ${multiply(3, 4)}`);
			console.log(`Деление: ${divide(10, 2)}`);
			const numbers = [1, 2, 3];
			const squares = numbers.map(x => x * x);
			console.log(`Квадраты: ${squares}`);
			}

Тема 2
Функции
	ПК-2.
	
	ПК-6.
		ИПК-2.2	**Задание 5 (Тип 5 - Практическая программа)**
			Программа с замыканиями
			javascript
			function createCounter() {
			let count = 0;
			return function() {
			count++;
			return count;
			};
			}
			function closureDemo() {
			const counter = createCounter();

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			console.log("Замыкания:");
			console.log(counter()); // 1
			console.log(counter()); // 2
			console.log(counter()); // 3
			function createMultiplier(factor) {
			return function(x) {
			return x * factor;
			};
			}
			const double = createMultiplier(2);
			const triple = createMultiplier(3);
			console.log(`Удвоение: ${double(5)}`);
			console.log(`Утроение: ${triple(5)}`);
			}

Тема 3. Объекты и массивы	ПК-2.	**ИПК-2.2**	**Задание 6 (Тип 5 - Практическая программа)**
			Программа работы с объектами
			javascript
			function objectDemo() {
			const student = {
			name: "Иван",
			age: 20,
			grades: [5, 4, 3],
			getInfo: function() {
			return `${this.name}, ${this.age} лет`;
			},
			getAverage: function() {
			const sum = this.grades.reduce((acc, grade) => acc + grade, 0);
			return sum / this.grades.length;
			}
			};

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			console.log("Объект студента:", student);
			console.log(student.getInfo());
			console.log(`Средний балл: ${student.getAverage().toFixed(1)}`);
			student.group = "ПИ-101";
			console.log("После добавления группы:", student);
			console.log("Свойства объекта:");
			for (let key in student) {
			if (typeof student[key] !== 'function') {
			console.log(`${key}: ${student[key]}`);
			}
			}
			}

Тема 3. Объекты и массивы	ПК-2.	**ИПК-2.2**	**Задание 7 (Тип 5 - Практическая программа)**
			Программа работы с массивами
			javascript
			function arrayDemo() {
			let fruits = ["apple", "banana", "orange"];
			console.log("Исходный массив:", fruits);
			fruits.push("grape");
			fruits.unshift("kiwi");
			console.log("После добавления:", fruits);
			let removed = fruits.pop();
			console.log(`Удален: ${removed}`, fruits);
			fruits.splice(1, 1);
			console.log("После удаления:", fruits);
			const numbers = [1, 2, 3, 4, 5];
			console.log("Методы массивов:");
			console.log("map:", numbers.map(x => x * 2));
			console.log("filter:", numbers.filter(x => x % 2 === 0));
			console.log("reduce:", numbers.reduce((sum, x) => sum + x, 0));

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			const [first, second, ...rest] = numbers;
			console.log(`Деструктуризация: first=${first}, second=${second}, rest=${rest}`);
			}

Тема 4. Объектная модель браузера	ПК-2.	**ИПК-2.2**	**Задание 8 (Тип 5 - Практическая программа)**
			Программа работы с BOM
			javascript
			function bomDemo() {
			console.log("Объектная модель браузера (BOM):");
			console.log("Ширина окна:", window.innerWidth);
			console.log("Высота окна:", window.innerHeight);
			console.log("Браузер:", navigator.userAgent);
			console.log("Язык:", navigator.language);
			console.log("URL:", location.href);
			console.log("Протокол:", location.protocol);
			console.log("Хост:", location.host);
			let counter = 0;
			const timerId = setInterval(() => {
			counter++;
			console.log(`Секунда: ${counter}`);
			if (counter >= 3) clearInterval(timerId);
			}, 1000);
			}

Тема 5. Объектная модель документа	ПК-2.	**ИПК-2.2**	**Задание 9 (Тип 5 - Практическая программа)** Программа работы с DOM javascript function domDemo() {

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			const container = document.createElement('div');
			container.id = 'demo-container';
			container.innerHTML = ' Демонстрация DOM ';
			const list = document.createElement('ul');
			const items = ['Первый элемент', 'Второй элемент', 'Третий элемент'];
			items.forEach(text => {
			const li = document.createElement('li');
			li.textContent = text;
			li.className = 'list-item';
			list.appendChild(li);
			});
			container.appendChild(list);
			document.body.appendChild(container);
			const listItems = document.querySelectorAll('.list-item');
			console.log(`Найдено элементов: ${listItems.length}`);
			listItems.forEach((item, index) => {
			item.style.color = index % 2 === 0 ? 'blue' : 'green';
			});
			setTimeout(() => {
			if (listItems.length > 0) {
			list.removeChild(listItems[0]);
			console.log('Первый элемент удален');
			}
			}, 2000);
			}

Тема 6. События и их обработка	ПК-2.	**ИПК-2.2**	**Задание 10 (Тип 5 - Практическая программа)** Программа обработки событий javascript function eventsDemo() {

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			const button = document.createElement('button');
			button.textContent = 'Нажми меня!';
			button.id = 'demo-button';
			const input = document.createElement('input');
			input.type = 'text';
			input.placeholder = 'Введите текст...';
			input.id = 'demo-input';
			const output = document.createElement('div');
			output.id = 'demo-output';
			button.addEventListener('click', function(event) {
			output.textContent = `Кнопка нажата! Координаты: ${event.clientX}, ${event.clientY}`;
			output.style.color = 'red';
			});
			input.addEventListener('input', function(event) {
			output.textContent = `Введен текст: ${event.target.value}`;
			output.style.color = 'blue';
			});
			input.addEventListener('keydown', function(event) {
			if (event.key === 'Enter') {
			output.textContent = `Нажат Enter: ${event.target.value}`;
			}
			});
			const container = document.createElement('div');
			container.appendChild(button);
			container.appendChild(input);
			container.appendChild(output);
			document.body.appendChild(container);
			}

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

Комплексное задание	ПК-2.	**ИПК-2.2**	**Задание 11 (Тип 5 - Практическая программа)**
			Программа "Калькулятор оценок"
			javascript
			function gradeCalculator() {
			const grades = [4, 5, 3, 4, 5];
			const student = "Иван";
			const avgGrade = grades.reduce((sum, grade) => sum + grade, 0) / grades.length;
			const maxGrade = Math.max(...grades);
			const minGrade = Math.min(...grades);
			console.log(`Студент: ${student}`);
			console.log(`Оценки: ${grades}`);
			console.log(`Средний балл: ${avgGrade.toFixed(1)}`);
			console.log(`Максимальная: ${maxGrade}`);
			console.log(`Минимальная: ${minGrade}`);
			if (avgGrade >= 4.5) console.log("Отличник!");
			else if (avgGrade >= 3.5) console.log("Хорошист!");
			else console.log("Нужно подтянуть!");
			}

Тема 1. Введение в JavaScript	ПК-2.	**ИПК-2.2**	**Задание 12 (Тип 5 - Практическая программа)**
			Программа "Классификатор чисел"
			javascript
			function numberClassifier() {
			const num = parseInt(prompt("Введите число:"));
			if (isNaN(num)) {
			console.log("Ошибка: введите число!");
			return;
			}
			if (num > 0) console.log("Положительное");
			else if (num < 0) console.log("Отрицательное");

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			else console.log("Ноль");
			if (num % 2 === 0) console.log("Четное");
			else console.log("Нечетное");
			if (num > 1) {
			let isPrime = true;
			for (let i = 2; i <= Math.sqrt(num); i++) {
			if (num % i === 0) {
			isPrime = false;
			break;
			}
			}
			console.log(isPrime ? "Простое число" : "Составное число");
			}
			}

Тема 2 Функции	ПК-2.	**ИПК-2.2**	**Задание 13 (Тип 5 - Практическая программа)**
			Программа "Калькулятор"
			javascript
			function createCalculator() {
			return {
			add: (a, b) => a + b,
			subtract: (a, b) => a - b,
			multiply: (a, b) => a * b,
			divide: (a, b) => b !== 0 ? a / b : 'Ошибка: деление на ноль',
			power: (a, b) => Math.pow(a, b),
			sqrt: (a) => a >= 0 ? Math.sqrt(a) : 'Ошибка: отрицательное число'
			};
			}
			function calculatorDemo() {
			const calc = createCalculator();
			console.log("Калькулятор:");

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			console.log(`5 + 3 = ${calc.add(5, 3)}`);
			console.log(`10 - 4 = ${calc.subtract(10, 4)}`);
			console.log(`6 * 7 = ${calc.multiply(6, 7)}`);
			console.log(`15 / 3 = ${calc.divide(15, 3)}`);
			console.log(`2 ^ 8 = ${calc.power(2, 8)}`);
			console.log(`√25 = ${calc.sqrt(25)}`);
			console.log(`10 / 0 = ${calc.divide(10, 0)}`);
			console.log(`√(-1) = ${calc.sqrt(-1)}`);
			}

Тема 3 Объекты и массивы	ПК-2.	**ИПК-2.2**	**Задание 14 (Тип 5 - Практическая программа)**
			Программа "Анализатор текста"
			``javascript
			function textAnalyzer() {
			const text = prompt("Введите текст:").toLowerCase();
			const words = text.split(/\s+/).filter(word => word.length > 0);
			const uniqueWords = [...new Set(words)];
			
			console.log(Текст: "${text}");
			console.log(Слов: ${words.length}, Символов: ${text.length});
			console.log(Первое слово: '${words[0]}', Последнее:
			'${words[words.length - 1]}');
			console.log(Уникальных слов: ${uniqueWords.length});
			console.log(Реверс текста: '${text.split('').reverse().join('')}');
			
			const wordFrequency = {};
			words.forEach(word => {
			wordFrequency[word] = (wordFrequency[word]

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

Комплексное задание	ПК-2.	**ИПК-2.2**	**Задание 15 (Тип 5 - Практическая программа)**
			Программа "Управление задачами"
			javascript
			function taskManagerDemo() {
			const taskManager = {
			tasks: [],
			addTask: function(title, priority = 'normal') {
			const task = {
			id: Date.now(),
			title,
			priority,
			completed: false,
			createdAt: new Date().toLocaleString()
			};
			this.tasks.push(task);
			console.log(`Задача "${title}" добавлена`);
			},
			completeTask: function(id) {
			const task = this.tasks.find(t => t.id === id);
			if (task) {
			task.completed = true;
			console.log(`Задача "${task.title}" выполнена`);
			}
			},
			showTasks: function() {
			console.log('Список задач:');
			this.tasks.forEach(task => {
			const status = task.completed ? '✓' : '○';
			console.log(`${status} [${task.priority}] ${task.title} (${task.createdAt})`);
			});
			},

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			getStats: function() {
			const total = this.tasks.length;
			const completed = this.tasks.filter(t => t.completed).length;
			const pending = total - completed;
			console.log(`Статистика: Всего ${total}, Выполнено ${completed}, Осталось ${pending}`);
			}
			};
			
			taskManager.addTask('Изучить JavaScript', 'high');
			taskManager.addTask('Сделать практические задания');
			taskManager.addTask('Прочитать документацию', 'low');
			taskManager.showTasks();
			taskManager.completeTask(taskManager.tasks[0].id);
			taskManager.getStats();
			taskManager.showTasks();
			}

Критерии оценивания практических занятий:

Оценка	Критерии оценивания
отлично	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи, формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, обосновывает свои суждения и даёт правильные ответы на вопросы преподавателя
хорошо	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи и формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, но содержание ответов имеют некоторые неточности и требуют уточнения и комментария со стороны преподавателя
удовлетворительно	Выставляется, если обучающийся знает и понимает материал по заданной теме, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, студент не может обосновать свои ответы на уточняющие вопросы преподавателя
неудовлетворительно	Выставляется, если обучающийся допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал. Делает ошибки в ответах на уточняющие вопросы преподавателя

4.3. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ КОНТРОЛЬНЫХ РАБОТ

Контрольные работы содержат несколько практических заданий по индивидуальным вариантам, в полном объеме охватывающих изученный материал по указанной теме. Выполнение контрольных работ позволяет определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

Тема 1 Введение в JavaScript	ПК-2.	**ИПК-2.2**	Вариант 1 (Тип 5 - Практическая программа)
			javascript
			// 1. Условный оператор: Проверка возраста
			let age = parseInt(prompt("Введите возраст:"));
			if (age >= 18) console.log("Совершеннолетний");
			else console.log("Несовершеннолетний");
			
			// 2. Логические операторы: Проверка диапазона
			let num = parseInt(prompt("Введите число:"));
			if (num >= 10 && num <= 50) console.log("В диапазоне 10-50");
			else console.log("Вне диапазона");
			
			// 3. Вложенные условия: Классификация оценок
			let grade = parseInt(prompt("Оценка:"));
			if (grade === 5) console.log("Отлично");
			else if (grade === 4) console.log("Хорошо");
			else if (grade === 3) console.log("Удовлетворительно");
			else console.log("Неудовлетворительно");

Тема 1 Введение в JavaScript	ПК-2.	**ИПК-2.2**	Вариант 2 (Тип 5 - Практическая программа)
			javascript
			// 1. Цикл for: Сумма четных чисел
			let total = 0;
			for (let i = 2; i <= 100; i += 2) total += i;
			console.log(`Сумма четных 2-100: ${total}`);
			
			// 2. Цикл while: Обратный отсчет
			let n = 5;
			while (n > 0) {
			console.log(n);

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			n--;
			}
			console.log("Старт!");
			
			// 3. Вложенные циклы: Таблица умножения
			for (let i = 1; i <= 3; i++) {
			let row = "";
			for (let j = 1; j <= 3; j++) {
			row += `${i}×${j}=${i * j} `;
			}
			console.log(row);
			}

Тема 3 Объекты и массивы	ПК-2.	**ИПК-2.2**	Вариант 3 (Тип 5 - Практическая программа)
			javascript
			// 1. Работа с массивом: Поиск максимального
			let numbers = [3, 7, 2, 9, 1];
			let maxVal = Math.max(...numbers);
			console.log(`Максимальный элемент: ${maxVal}`);
			
			// 2. Срезы массива: Обратный порядок
			let arr = [1, 2, 3, 4, 5];
			let reversedArr = [...arr].reverse();
			console.log(`Обратный массив: ${reversedArr}`);
			
			// 3. Методы строк: Анализ текста
			let text = "Программирование на JavaScript";
			console.log(`Длина: ${text.length}`);
			console.log(`Верхний регистр: ${text.toUpperCase()}`);
			console.log(`Замена: ${text.replace('JavaScript', 'Java')}`);

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

Тема 3 Объекты и массивы	ПК-2.	**ИПК-2.2**	Вариант 4 (Тип 5 - Практическая программа)
			javascript
			// 1. Работа с объектом: Данные студента
			let student = {name: "Иван", age: 20, group: "ПИ-101"};
			console.log(`Студент: ${student.name}`);
			student.gpa = 4.5;
			console.log(`Обновленные данные:`, student);
			
			// 2. Операции с множествами: Уникальные элементы
			let set1 = new Set([1, 2, 3, 3, 4]);
			let set2 = new Set([3, 4, 5, 6]);
			console.log(`Уникальные: ${Array.from(set1)}`);
			console.log(`Пересечение: ${Array.from(new Set([...set1].filter(x => set2.has(x))))}`);
			
			// 3. Методы объекта: Перебор элементов
			for (let key in student) {
			console.log(`${key}: ${student[key]}`);
			}

Тема 3 Объекты и массивы	ПК-2.	**ИПК-2.2**	Вариант 5 (Тип 5 - Практическая программа)
			javascript
			// 1. Вложенные массивы: Матрица
			let matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]];
			let submatrix = matrix.slice(0, 2).map(row => row.slice(0, 2));
			console.log(`Матрица 2x2:`, submatrix);
			
			// 2. Вложенные объекты: База данных
			let library = {
			book1: {title: "JavaScript", year: 2020},

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)
			book2: {title: "Java", year: 2019} }; console.log(`Название книги: ${library.book1.title}`); // 3. Доступ к вложенным элементам let grades = {math: [5, 4, 3], physics: [4, 5]}; console.log(`Оценка по математике: ${grades.math[0]}`);

Тема 2 Функции	ПК-2.	**ИПК-2.2**	Вариант 6 (Тип 5 - Практическая программа)
			javascript
			// 1. Простая функция: Калькулятор
			function multiply(a, b) {
			return a * b;
			}
			let result = multiply(5, 3);
			console.log(`Произведение: ${result}`);
			
			// 2. Функция с параметром по умолчанию
			function greet(name = "Гость") {
			return `Привет, ${name}!`;
			}
			console.log(greet());
			console.log(greet("Анна"));
			
			// 3. Использование Math
			console.log(`Квадратный корень: ${Math.sqrt(25)}`);
			console.log(`Округление вверх: ${Math.ceil(4.3)}`);

Тема 4. Объектная модель документа	ПК-2.	**ИПК-2.2**	Вариант 7 (Тип 5 - Практическая программа) javascript

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			// 1. Создание элемента
			let div = document.createElement('div');
			div.textContent = "Новый элемент";
			document.body.appendChild(div);
			console.log("Элемент создан");
			
			// 2. Поиск элементов
			let elements = document.getElementsByTagName('div');
			console.log(`Найдено элементов div: ${elements.length}`);
			
			// 3. Изменение стилей
			let firstDiv = elements[0];
			if (firstDiv) {
			firstDiv.style.color = "red";
			firstDiv.style.fontSize = "20px";
			console.log("Стили применены");
			}

Комплексное задание	ПК-2. ПК-6.	ИПК-2.2 ИПК-6.1	Вариант 8 (Тип 5 - Практическая программа) javascript // 1. Обработка пользовательского ввода let name = prompt("Введите имя:"); let age = parseInt(prompt("Введите возраст:")); console.log(`Данные: ${name}, ${age} лет`); // 2. Работа с коллекциями let input = prompt("Введите числа через пробел:"); let numbers = input.split(' ').map(Number).filter(n => !isNaN(n)); console.log(`Сумма: ${numbers.reduce((sum, n) => sum + n, 0)}`);

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			// 3. Условная логика и циклы
			numbers.forEach(num => {
			if (num % 2 === 0) console.log(`${num} - четное`);
			else console.log(`${num} - нечетное`);
			});

Тема 3. Объекты и массивы	ПК-2.	**ИПК-2.2**	Вариант 9 (Тип 5 - Практическая программа) javascript // 1. Создание и модификация массива let fruits = ["apple", "banana", "cherry"]; fruits.push("orange"); fruits.splice(fruits.indexOf("banana"), 1); console.log(`Фрукты: ${fruits}`); // 2. Работа с множествами let uniqueFruits = new Set(fruits); uniqueFruits.add("grape"); console.log(`Уникальные фрукты: ${Array.from(uniqueFruits)}`); // 3. Преобразование типов let numbersStr = ["1", "2", "3"]; let numbersInt = numbersStr.map(Number); console.log(`Числа: ${numbersInt}`);
Интегрированное задание	ПК-2.	**ИПК-2.2**	Вариант 10 (Тип 5 - Практическая программа) javascript // 1. Функция для работы с объектом function getStudentInfo() {

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)
			return {name: "Мария", subjects: ["math", "physics"]}; } let student = getStudentInfo(); console.log(`Студент: ${student.name}`); // 2. Обработка вложенных данных let subjects = student.subjects; console.log(`Предметы: ${subjects.join(', ')}`); // 3. Создание элемента с данными let studentDiv = document.createElement('div'); studentDiv.innerHTML = ` <h3>${student.name}</h3> <p>Предметов: ${subjects.length}</p> `; document.body.appendChild(studentDiv); console.log("Информация отображена");

Тема 6. События и их обработка	ПК-2.	**ИПК-2.2**	Вариант 11 (Тип 5 - Практическая программа)
			javascript
			// 1. Обработчик клика
			let button = document.createElement('button');
			button.textContent = 'Нажми меня';
			button.addEventListener('click', function() {
			console.log('Кнопка нажата!');
			this.style.backgroundColor = 'green';
			});
			document.body.appendChild(button);
			
			// 2. Обработчик ввода
			let input = document.createElement('input');

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			input.placeholder = 'Введите текст...';
			input.addEventListener('input', function(event) {
			console.log(`Введен текст: ${event.target.value}`);
			});
			document.body.appendChild(input);
			
			// 3. Обработчик загрузки страницы
			window.addEventListener('load', function() {
			console.log('Страница загружена');
			});

Тема 2 Функции	ПК-2.	**ИПК-2.2**	Вариант 12 (Тип 5 - Практическая программа)
			javascript
			// 1. Стрелочная функция
			const square = x => x * x;
			console.log(`Квадрат 5: ${square(5)}`);
			
			// 2. Функция высшего порядка
			function applyOperation(x, y, operation) {
			return operation(x, y);
			}
			let addition = (a, b) => a + b;
			console.log(`Результат: ${applyOperation(10, 5, addition)}`);
			
			// 3. Замыкание
			function createMultiplier(factor) {
			return function(x) {
			return x * factor;
			};
			}
			let double = createMultiplier(2);

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)
			console.log(`Удвоение: ${double(7)}`);

Тема 1. Введение в JavaScript	ПК-2.	**ИПК-2.2**	Вариант 13 (Тип 5 - Практическая программа)
			javascript
			// 1. Работа с различными типами
			let values = [42, "Hello", true, null, undefined];
			values.forEach(value => {
			console.log(`Значение: ${value}, тип: ${typeof value}`);
			});
			
			// 2. Преобразование типов
			let str = "123";
			let num = Number(str);
			let bool = Boolean(str);
			console.log(`Строка: "${str}", число: ${num}, булево: ${bool}`);
			
			// 3. Операторы сравнения
			console.log(`"5" == 5: ${"5" == 5}`);
			console.log(`"5" === 5: ${"5" === 5}`);
			console.log(`0 == false: ${0 == false}`);
			console.log(`0 === false: ${0 === false}`);

Тема 4 Объектная модель браузера	ПК-2.	**ИПК-2.2**	Вариант 14 (Тип 5 - Практическая программа) javascript // 1. Работа с window console.log(`Размер окна: ${window.innerWidth}x${window.innerHeight}`); console.log(`URL: ${window.location.href}`);

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)
			// 2. Работа с navigator console.log(`Браузер: ${navigator.userAgent}`); console.log(`Платформа: ${navigator.platform}`); // 3. Работа с localStorage localStorage.setItem('username', 'Иван'); let savedName = localStorage.getItem('username'); console.log(`Сохраненное имя: ${savedName}`);

Тема 3 Объекты и массивы	ПК-2.	**ИПК-2.2**	Вариант 15 (Тип 5 - Практическая программа)
			javascript
			// 1. Методы массивов
			let numbers = [1, 2, 3, 4, 5];
			console.log(`Исходный: ${numbers}`);
			console.log(`map: ${numbers.map(x => x * 2)}`);
			console.log(`filter: ${numbers.filter(x => x % 2 === 0)}`);
			console.log(`reduce: ${numbers.reduce((sum, x) => sum + x, 0)}`);
			
			// 2. Деструктуризация
			let [first, second, ...rest] = numbers;
			console.log(`first: ${first}, second: ${second}, rest: ${rest}`);
			
			// 3. Spread оператор
			let arr1 = [1, 2, 3];
			let arr2 = [4, 5, 6];
			let combined = [...arr1, ...arr2];
			console.log(`Объединенный: ${combined}`);

Критерии оценивания контрольной работы:

Оценка	Критерии оценивания
отлично	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи, формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, обосновывает свои суждения и даёт правильные ответы на вопросы преподавателя
хорошо	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи и формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, но содержание ответов имеют некоторые неточности и требуют уточнения и комментария со стороны преподавателя.
удовлетворительно	Выставляется, если обучающийся знает и понимает материал по заданной теме, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, студент не может обосновать свои ответы на уточняющие вопросы преподавателя
неудовлетворительно	Выставляется, если обучающийся допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал. Делает ошибки в ответах на уточняющие вопросы преподавателя

4.4. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа включает в себя проработку теоретического материала, изучение рекомендуемой литературы, выполнение практико-ориентированных заданий (заполнение таблиц, проведение сравнительного анализа, составление схем и др.), решение практических задач, создание презентаций, написание рефератов, подборка нормативного и иного материала и выполнение других заданий.

<table>
<tr><th>Раздел/ Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Задание (Тип и формулировка)</th></tr>
<tr><td>Тема 1 Введение в JavaScript</td><td>ПК-2.</td><td>ИПК-2.1</td><td>Задание 1 (Тип 1 - Установление соответствия)
Установите соответствие между типами данных JavaScript и их характеристиками:
<table><tr><th>Пункт</th><th>Тип данных</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>number</td><td>2</td><td>Целочисленные и дробные числа</td></tr><tr><td>Б</td><td>string</td><td>3</td><td>Строковый тип Unicode</td></tr><tr><td>В</td><td>boolean</td><td>1</td><td>Логический тип true/false</td></tr><tr><td>Г</td><td>undefined</td><td>4</td><td>Неопределенное значение</td></tr><tr><td>Д</td><td>null</td><td>5</td><td>Пустое значение</td></tr><tr><td>Е</td><td>object</td><td>6</td><td>Ссылочный тип для коллекций</td></tr><tr><td>Ж</td><td>symbol</td><td>7</td><td>Уникальный и неизменяемый идентификатор</td></tr></table>
Решение: А-2, Б-3, В-1, Г-4, Д-5, Е-6, Ж-7</td></tr>
<tr><td>Тема 1 Введение в JavaScript</td><td>ПК-2.</td><td>ИПК-2.2</td><td>Задание 2 (Тип 5 - Практическая программа)
Исследование основных конструкций JavaScript:
javascript
function demonstrateBasics() {
let name = prompt("Введите ваше имя:");
let age = parseInt(prompt("Введите ваш возраст:"));
console.log(`Привет, ${name}! Тебе ${age} лет.`);
console.log(`Через 5 лет тебе будет ${age + 5} лет.`);
console.log(`Тип переменной name: ${typeof name}, тип переменной age: ${typeof age}`);
let numStr = "123";
let convertedNum = Number(numStr);
console.log(`Преобразование строки "${numStr}" в число: ${convertedNum}, тип: ${typeof convertedNum}`);
}</td></tr>
</table>

Раздел/ Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

Тема 1 Введение в JavaScript	ПК-2.	**ИПК-2.1**	**Задание 3 (Тип 5 - Практическая программа)**
			Анализ операторов и приведения типов:
			function typeConversionResearch() {
			console.log("=== Исследование операторов ===");
			console.log(`5 + "5" = ${5 + "5"}`);
			console.log(`5 - "5" = ${5 - "5"}`);
			console.log(`"5" * 2 = ${"5" * 2}`);
			console.log(`"5" / 2 = ${"5" / 2}`);
			console.log(`5 == "5": ${5 == "5"}`);
			console.log(`5 === "5": ${5 === "5"}`);
			console.log(`0 == false: ${0 == false}`);
			console.log(`0 === false: ${0 === false}`);
			console.log(`true && "Hello": ${true && "Hello"}`);
			console.log(`false || "Default": ${false || "Default"}`);
			console.log(`null ?? "Fallback": ${null ?? "Fallback"}`);
			}
			// Вызов функции исследования операторов и приведения типов
			typeConversionResearch();

Тема 2 Функции	ПК-2.	**ИПК-2.1**	**Задание 4 (Тип 1 - Сравнительный анализ)** Сравнительный анализ типов функций в JavaScript: **Решение:** Function Declaration - Полное поднятие, есть собственный this, есть arguments, может быть конструктором Function Expression - Частичное поднятие, есть собственный this, есть arguments, может быть конструктором Arrow Function - Частичное поднятие, нет собственного this, нет arguments, не может быть конструктором
Тема 2 Функции	ПК-2.	**ИПК-2.1**	**Задание 5 (Тип 1 - Аналитическая таблица)** Анализ областей видимости в JavaScript: **Решение:** Глобальная - Доступна везде в программе

Раздел/ Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)
			Функциональная - Внутри функции Блочная - Внутри блока {} Модульная - В пределах модуля

Тема 2 Функции	ПК-2.	**ИПК-2.2**	**Задание 6 (Тип 5 - Практическая программа)**
			Исследование замыканий и каррирования:
			javascript
			function closureResearch() {
			function createCounter() {
			let count = 0;
			return function() {
			count++;
			return count;
			};
			}
			const counter = createCounter();
			console.log("Счетчик:", counter(), counter(), counter());
			
			function createMultiplier(factor) {
			return function(x) {
			return x * factor;
			};
			}
			const double = createMultiplier(2);
			const triple = createMultiplier(3);
			console.log("Удвоение 5:", double(5));
			console.log("Утроение 5:", triple(5));
			}

Тема 3 Объекты и массивы	ПК-2.	**ИПК-2.2**	**Задание 7 (Тип 1 - Сравнительная таблица)** Сравнительный анализ методов работы с объектами и массивами: **Решение:**

Раздел/ Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)
			Объекты - obj.key = value, delete obj.key, 'key' in obj, Object.keys(obj).length, for...in Массивы - arr.push(value), arr.splice(index, 1), arr.includes(value), arr.length, for...of

Тема 3 Объекты и массивы	ПК-2.	**ИПК-2.2**	**Задание 8 (Тип 5 - Практическая программа)**
			Исследование методов работы с объектами:
			javascript
			function objectMethodsResearch() {
			const person = {name: "Иван", age: 30, city: "Москва"};
			console.log("Ключи объекта:", Object.keys(person));
			console.log("Значения объекта:", Object.values(person));
			console.log("Пары ключ-значение:", Object.entries(person));
			const additionalInfo = {hobby: "чтение"};
			const extendedPerson = Object.assign({}, person, additionalInfo);
			console.log("Расширенный объект:", extendedPerson);
			
			const {name, age, ...rest} = person;
			console.log("Деструктуризация:", {name, age, rest});
			}

Тема 3 Объекты и массивы	ПК-2.	**ИПК-2.2**	**Задание 9 (Тип 5 - Практическая программа)**
			Анализ производительности методов массивов:
			javascript
			function arrayPerformanceResearch() {
			const largeArray = Array.from({length: 10000}, (_, i) => i);
			
			console.time('for loop');
			let sum1 = 0;
			for (let i = 0; i < largeArray.length; i++) {
			sum1 += largeArray[i];
			}

Раздел/ Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			console.timeEnd('for loop');
			console.time('forEach');
			let sum2 = 0;
			largeArray.forEach(num => sum2 += num);
			console.timeEnd('forEach');
			console.time('reduce');
			const sum3 = largeArray.reduce((acc, num) => acc + num, 0);
			console.timeEnd('reduce');
			}

Тема 4 Объектная модель браузера	ПК-2.	**ИПК-2.2**	**Задание 10 (Тип 5 - Практическая программа)**
			Исследование объектной модели браузера:
			function bomStructureResearch() {
			// Анализ основных свойств объекта Window
			console.log("Window properties:", window.innerWidth, window.innerHeight);
			// Исследование свойств объекта Navigator
			console.log("Navigator properties:", navigator.userAgent, navigator.language);
			// Получение информации об экране пользователя
			console.log("Screen properties:", screen.width, screen.height);
			// Анализ текущего местоположения в браузере
			console.log("Location properties:", location.href, location.protocol);
			// Функция для определения типа устройства
			function getDeviceType() {
			const userAgent = navigator.userAgent.toLowerCase();
			if (/mobile|android|iphone|ipad/.test(userAgent)) return 'mobile';
			else if (/tablet/.test(userAgent)) return 'tablet';
			else return 'desktop';
			}
			const deviceType = getDeviceType();
			console.log("Тип устройства:", deviceType);

Раздел/ Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			}
			// Вызов функции исследования BOM
			bomStructureResearch();

Тема 4 Объектная модель браузера	ПК-2.	**ИПК-2.2**	**Задание 11 (Тип 1 - Сравнительная таблица)** Сравнительный анализ объектов хранения данных: **Решение:** localStorage - ~5-10MB, постоянно, в пределах домена, ключ-значение, синхронный sessionStorage - ~5-10MB, до закрытия вкладки, в пределах вкладки, ключ-значение, синхронный Cookies - ~4KB, задается, с сервером, строка, синхронный IndexedDB - >250MB, постоянно, в пределах домена, объекты, асинхронный

Тема 5 Объектная модель документа	ПК-2.	**ИПК-2.2**	**Задание 12 (Тип 5 - Практическая программа)**
			Исследование методов работы с DOM:
			javascript
			function domMethodsResearch() {
			const testContainer = document.createElement('div');
			testContainer.innerHTML = `<div class="item" id="item1">Элемент 1</div>`;
			document.body.appendChild(testContainer);
			
			console.time('getElementById');
			const byId = document.getElementById('item1');
			console.timeEnd('getElementById');
			
			console.time('querySelector');
			const byQuery = document.querySelector('.item');
			console.timeEnd('querySelector');
			
			const newElement = document.createElement('div');
			newElement.textContent = 'Новый элемент';
			testContainer.appendChild(newElement);

Раздел/ Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)
			}

Тема 5 Объектная модель документа	ПК-2.	**ИПК-2.2**	**Задание 13 (Тип 5 - Практическая программа)**
			Исследование производительности DOM операций:
			javascript
			function domPerformanceResearch() {
			const iterations = 1000;
			console.time('create elements');
			for (let i = 0; i < iterations; i++) {
			const div = document.createElement('div');
			}
			console.timeEnd('create elements');
			console.time('DocumentFragment');
			const fragment = document.createDocumentFragment();
			for (let i = 0; i < iterations; i++) {
			const div = document.createElement('div');
			fragment.appendChild(div);
			}
			console.timeEnd('DocumentFragment');
			}

Тема 6 События и их обработка	ПК-2.	**ИПК-2.2**	**Задание 14 (Тип 5 - Практическая программа)**
			Исследование обработки событий:
			javascript
			function eventHandlingResearch() {
			const eventContainer = document.createElement('div');
			eventContainer.innerHTML = `<button id="btn1">Кнопка 1</button>`;
			document.body.appendChild(eventContainer);
			document.getElementById('btn1').onclick = function() {
			console.log('Традиционная модель');
			};

Раздел/ Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			document.getElementById('btn1').addEventListener('click', function(event) {
			console.log('Современная модель', event);
			});
			eventContainer.addEventListener('click', function(event) {
			if (event.target.tagName === 'BUTTON') {
			console.log('Делегирование - нажата кнопка:', event.target.id);
			}
			});
			}

Тема 6 События и их обработка	ПК-2.	**ИПК-2.1**	**Задание 15 (Тип 1 - Сравнительная таблица)** Сравнительный анализ методов обработки событий: **Решение:** HTML-атрибуты - onclick="handler()", нет множественных обработчиков, нет удаления, только всплытие DOM-свойства - elem.onclick = handler, нет множественных обработчиков, elem.onclick = null, только всплытие addEventListener - elem.addEventListener('click', handler), да множественные обработчики, removeEventListener, любая фаза

Тема 6 События и их обработка	ПК-2.	**ИПК-2.2**	**Задание 16 (Тип 5 - Практическая программа)**
			Исследование производительности обработки событий:
			javascript
			function eventPerformanceResearch() {
			const iterations = 1000;
			const container = document.createElement('div');
			console.time('individual handlers');
			for (let i = 0; i < iterations; i++) {
			const btn = document.createElement('button');
			btn.addEventListener('click', function() {});
			container.appendChild(btn);
			}

Раздел/ Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задание (Тип и формулировка)

			console.timeEnd('individual handlers');
			
			console.time('event delegation');
			const container2 = document.createElement('div');
			container2.addEventListener('click', function(event) {
			if (event.target.tagName === 'BUTTON') {}
			});
			console.timeEnd('event delegation');
			}

Критерии оценивания самостоятельной работы:

Оценка	Критерии оценивания
отлично	выставляется если работа носит научно-исследовательский характер, проанализирован и сделан сравнительный анализ нескольких литературных источников, приведены примеры
хорошо	выставляется если проанализирован и сделан сравнительный анализ нескольких литературных источников, приведены примеры
удовлетворительно	выставляется если проведен сравнительный анализ научно-методической литературы, приведены примеры
неудовлетворительно	выставляется если работа прошла проверку на антиплагиат и соответствует требованиям оформления

4.5. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ ДЛЯ КУРСОВЫХ РАБОТ

Курсовая работа позволяет выявить степень владения базовыми знаниями, умениями и навыками, необходимыми для обучения, и определить уровень владения новым материалом.

Примерные индивидуальные задания (темы) для курсовых работ:

КУРСОВАЯ РАБОТА НЕ ПРЕДУСМОТРЕНА УЧЕБНЫМ ПЛАНОМ

Критерии оценивания курсовой работы:

Оценка	Критерии оценивания
отлично	Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Представлены результаты структурированного и логически последовательного обзора литературных и иных источников по теме исследования. Структура курсовой работы логически и методически выдержана. Верно определены исходные данные для расчетов. Все аналитические расчеты выполнены верно, корректно применены методы экономического анализа, не нарушена методика анализа предмета исследования. Все выводы и предложения убедительно аргументированы. При защите курсовой работы обучающийся правильно и уверенно отвечает на вопросы преподавателя, демонстрирует глубокое знание теоретического материала, способен аргументировать собственные утверждения и выводы
хорошо	Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Представлены результаты структурированного и логически последовательного обзора литературных и иных источников по теме исследования. Структура курсовой работы логически и методически выдержана. Верно определены исходные данные для расчетов. В расчетах допускаются незначительные (не искажающие общего итога оценки) погрешности/ошибки. Большинство выводов и предложений аргументировано, корректно применены методы экономического анализа, не нарушена методика анализа предмета исследования. Оформление курсовой работы и полученные результаты в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две несущественные ошибки в использовании терминов, в построенных диаграммах и схемах, в оформлении таблиц. Наличествует незначительное количество грамматических и/или стилистических ошибок. При защите курсовой работы обучающийся правильно и уверенно отвечает на большинство вопросов преподавателя, демонстрирует хорошее знание теоретического материала, но не всегда способен аргументировать собственные утверждения и выводы. При наводящих вопросах преподавателя исправляет ошибки в ответе

удовлетворительно	Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Результаты обзора литературных и иных источников представлены недостаточно полно, недостаточно логично и последовательно. Верно определены исходные данные для расчетов, но имеются грубые ошибки в расчетах. Аргументация выводов и предложений слабая или отсутствует. Экономические выводы носят констатирующий (описательный) характер. Имеются одно-два существенных отклонений от требований в оформлении курсовой работы. Полученные результаты в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две существенных ошибки в использовании терминов, в построенных диаграммах и схемах. Много грамматических и/или стилистических ошибок. При защите курсовой работы обучающийся допускает грубые ошибки при ответах на вопросы преподавателя, демонстрирует слабое знание теоретического материала, в большинстве случаев не способен уверенно аргументировать собственные утверждения и выводы
неудовлетворительно	Содержание курсовой работы не соответствует заданию, содержащемуся в методических указаниях, и плану. Неверно определены исходные данные для расчетов, неверно и не корректно применены методы экономического анализа. Экономические выводы содержат неверную экономическую оценку. Имеются более двух существенных отклонений от требований в оформлении курсовой работы. Большое количество существенных ошибок по сути работы, много грамматических и стилистических ошибок и др. Полученные результаты не отвечают требованиям, изложенным в методических указаниях. При защите курсовой работы обучающийся демонстрирует слабое понимание программного материала, студент не может защитить свои решения, допускает грубые фактические ошибки при ответах на поставленные вопросы или вовсе не отвечает на них. Курсовая работа не представлена преподавателю. Обучающийся не явился на защиту курсовой работы

5. МАТЕРИАЛЫ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Промежуточная аттестация проводится в форме зачета/зачета с оценкой/экзамена.

5.1. Вопросы к зачету с оценкой:

1. Что включает в себя экосистема JavaScript и каковы этапы выполнения кода?
2. Какие типы данных существуют в JavaScript и каковы их особенности?
3. Как объявляются функции в JavaScript и какие различия между Function Declaration и Function Expression?
4. Что такое замыкания и как они работают в JavaScript?
5. Как работают объекты в JavaScript и какие методы для работы с ними существуют?
6. Какие методы массивов наиболее часто используются и в каких случаях?
7. Что такое BOM и какие объекты в него входят?
8. Как осуществляется взаимодействие с браузером через JavaScript?
9. Что такое DOM и как осуществляется работа с элементами страницы?
10. Какие методы используются для создания и модификации DOM-элементов?
11. Как работают события в JavaScript и какие модели событий существуют?
12. Что такое делегирование событий и в каких случаях оно применяется?
13. Что такое прототипное наследование и как оно реализовано в JavaScript?
14. Как работают Promise и для чего используется async/await?
15. Что такое hoisting (поднятие) и как оно работает с переменными и функциями?

16. Какие различия между var, let и const?
17. Что такое this в JavaScript и как определяется его значение?
18. Как работают методы call, apply и bind?
19. Что такое CORS и как обрабатываются кросс-доменные запросы?
20. Какие методы хранения данных в браузере вы знаете и в чем их различия?

Критерии оценивания зачета с оценкой/экзамена:

Оценка	Критерии оценивания
зачтено	Обучающийся должен: уметь строить ответ в соответствии со структурой излагаемого вопроса; продемонстрировать прочное, достаточно полное усвоение знаний программного материала; продемонстрировать знание основных теоретических понятий; правильно формулировать определения; последовательно, грамотно и логически стройно изложить теоретический материал; продемонстрировать умения самостоятельной работы с литературой; уметь сделать достаточно обоснованные выводы по излагаемому материалу.
не зачтено	Обучающийся демонстрирует: незнание значительной части программного материала; не владение понятийным аппаратом дисциплины; существенные ошибки при изложении учебного материала; неумение строить ответ в соответствии со структурой излагаемого вопроса; неумение делать выводы по излагаемому материалу.

Критерии оценивания зачета с оценкой/экзамена:

Оценка	Критерии оценивания
отлично	Выставляется, если обучающимся правильно и полностью раскрыто содержание материала в пределах программы, чётко и правильно даны определения и раскрыто содержание понятий, точно использованы научные и технические термины, в ответе использованы ранее приобретённые теоретические знания, сделаны необходимые выводы и обобщения
хорошо	Выставляется, если обучающимся раскрыто основное содержание материала в пределах программы, даны определения и раскрыто содержание понятий, в ответе использованы ранее приобретённые теоретические знания, сделаны необходимые выводы и обобщения, но присутствуют незначительные нарушения в последовательности изложения, имеются одна-две неточности в содержании ответа
удовлетворительно	Выставляется, если обучающимся содержание учебного материала изложено фрагментарно, не всегда последовательно, не даны определения, не раскрыто содержание понятий, или они изложены с ошибками, допускаются ошибки и неточности в использовании научной терминологии, отсутствуют выводы и обобщения из предыдущего материала, или возможны ошибки в их изложении
неудовлетворительно	Выставляется, если обучающимся основное содержание учебного материала не раскрыто, не даются ответы на основные вопросы, допускаются грубые ошибки в определении понятий, в использовании терминологии, отсутствуют выводы и обобщения

6. МАТЕРИАЛЫ ДЛЯ ДИАГНОСТИЧЕСКОЙ РАБОТЫ

Задания для диагностической работы должны обеспечивать оценку полностью или частично сформированных компетенций. Каждое задание должно быть привязано к тому или иному индикатору сформированности компетенций.

При формировании заданий для диагностической работы необходимо использовать тестовые задания следующих типов:

Тип задания 1. Задания закрытого типа на установление соответствия.

Тип задания 2. Задания закрытого типа на установление последовательности.

Тип задания 3. Задания комбинированного типа, предполагающие выбор одного правильного ответа из предложенных

Тип задания 4. Задания комбинированного типа, предполагающие выбор нескольких ответов из предложенных

Тип задания 5. Задания открытого типа с развернутым ответом.

Все типы заданий должны быть представлены не менее одного раза.

<table>
<tr><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th>Вариант 1</th><th>Вариант 2</th><th>Уровень освоения компетенции</th></tr>
<tr>
<td>Тема 1. Введение в JavaScript</td>
<td>ПК-2</td>
<td>ИПК-2.1</td>
<td>1. Задания закрытого типа на установление соответствия</td>
<td>Задание 1: Соотнесите типы данных JavaScript с их описанием:
<table>
<tr><th>Пункт</th><th>Тип данных</th><th>Пункт</th><th>Описание</th></tr>
<tr><td>А</td><td>number</td><td>1</td><td>Логический тип true/false</td></tr>
<tr><td>Б</td><td>string</td><td>2</td><td>Целочисленные и дробные числа</td></tr>
<tr><td>В</td><td>boolean</td><td>3</td><td>Строковый тип</td></tr>
<tr><td>Г</td><td>undefined</td><td>4</td><td>Неопределенное значение</td></tr>
<tr><td>Д</td><td>null</td><td>5</td><td>Пустое значение</td></tr>
</table>
Ответ: А-2, Б-3, В-1, Г-4, Д-5</td>
<td>Задание 1: Соотнесите операторы JavaScript с их описанием:
<table>
<tr><th>Пункт</th><th>Оператор</th><th>Пункт</th><th>Описание</th></tr>
<tr><td>А</td><td>==</td><td>1</td><td>Строгое равенство</td></tr>
<tr><td>Б</td><td>===</td><td>2</td><td>Нестрогое равенство</td></tr>
<tr><td>В</td><td>!=</td><td>3</td><td>Строгое неравенство</td></tr>
<tr><td>Г</td><td>!==</td><td>4</td><td>Нестрогое неравенство</td></tr>
</table>
Ответ: А-2, Б-1, В-4, Г-3</td>
<td>Базовый (1-3 минуты)</td>
</tr>
<tr>
<td>Тема 1. Введение в JavaScript</td>
<td>ПК-2</td>
<td>ИПК-2.2</td>
<td>3. Задания комбинированного типа с</td>
<td>Задание 2: Что выведет код: console.log(typeof NaN)?
1. "number"
2. "NaN"
3. "undefined"</td>
<td>Задание 2: Какой результат выражения: "5" + 3?
1. 8
2. "53"
3. Ошибка</td>
<td>Базовый (1-3 минуты)</td>
</tr>
</table>

<table>
<tr><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th>Вариант 1</th><th>Вариант 2</th><th>Уровень освоения компетенции</th></tr>
<tr><td></td><td></td><td></td><td>выбором одного ответа</td><td>4. "string"
Ответ: 1) "number" - NaN является числовым типом, представляющим не-число</td><td>4. "8"
Ответ: 2) "53" - в JavaScript при сложении строки и числа происходит конкатенация</td><td></td></tr>
<tr><td>Тема 1. Введение в JavaScript</td><td>ПК-2</td><td>ИПК-2.2</td><td>5. Задания открытого типа с развернутым ответом</td><td>Задание 3: Напишите программу, которая определяет тип данных переменной
Решение:
javascript
let x = 3.14;
console.log(`Тип переменной: ${typeof x}`);</td><td>Задание 3: Напишите программу, которая демонстрирует преобразование типов
Решение:
javascript
let numStr = "123";
let numInt = parseInt(numStr);
console.log(`Результат: ${numInt}, тип: ${typeof numInt}`);</td><td>Повышенный (3-5 минут)</td></tr>
<tr><td>Тема 1. Введение в JavaScript</td><td>ПК-2</td><td>ИПК-2.1</td><td>1. Задания закрытого типа на установление соответствия</td><td>Задание 4: Соотнесите способы объявления переменных с их характеристиками:<table><tr><th>Пункт</th><th>Ключевое слово</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>var</td><td>1</td><td>Блочная область видимости, нельзя переопределять</td></tr><tr><td>Б</td><td>let</td><td>2</td><td>Блочная область видимости, можно изменять</td></tr><tr><td>В</td><td>const</td><td>3</td><td>Функциональная область видимости</td></tr></table>Ответ: А-3, Б-2, В-1</td><td>Задание 4: Соотнесите методы строк с их описанием:<table><tr><th>Пункт</th><th>Метод</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>length</td><td>1</td><td>Разделение строки на массив</td></tr><tr><td>Б</td><td>split()</td><td>2</td><td>Длина строки</td></tr><tr><td>В</td><td>toUpperCase()</td><td>3</td><td>Преобразование в верхний регистр</td></tr><tr><td>Г</td><td>includes()</td><td>4</td><td>Проверка наличия подстроки</td></tr></table>Ответ: А-2, Б-1, В-3, Г-4</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>Тема 1. Введение в</td><td>ПК-2</td><td>ИПК-2.2</td><td>3. Задания комбин</td><td>Задание 5: Что выведет console.log(0.1 + 0.2 === 0.3)?
1. true</td><td>Задание 5: Какой результат: console.log([] == false)?
1. true</td><td>Повышенный (3-5 минут)</td></tr>
</table>

Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
JavaScript			ированного типа с выбором одного ответа	2. false 3. undefined 4. Ошибка Ответ: 2) false - из-за особенностей представления чисел с плавающей точкой	2. false 3. undefined 4. Ошибка Ответ: 1) true - пустой массив преобразуется в 0, false в 0	
Тема 2 Функции	ПК-2	ИПК-2.1	2. Задания закрытого типа на установление последовательности	Задание 6: Расположите этапы выполнения функции в правильной последовательности: 1. Возврат результата 2. Вызов функции 3. Объявление функции 4. Выполнение тела функции 5. Передача аргументов Ответ: 3-2-5-4-1	Задание 6: Расположите типы функций в порядке их создания: 1. Function Expression 2. Arrow Function 3. Function Declaration 4. Callback Function Ответ: 3-1-4-2	**Базовый (1-3 минуты)**
Тема 2 Функции	ПК-2	ИПК-2.3	4. Задания комбинированного типа с выбором нескольких ответов	Задание 7: Какие из перечисленных способов являются правильными для объявления функций? 1. function myFunc() {} 2. let myFunc = function() {} 3. const myFunc = () => {} 4. class myFunc {} 5. myFunc: function() {} Ответ: 1, 2, 3 - основные способы объявления функций в JavaScript	Задание 7: Какие особенности имеют стрелочные функции? 1. Не имеют своего this 2. Не имеют arguments 3. Не могут быть конструкторами 4. Всегда возвращают значение 5. Имеют сокращенный синтаксис Ответ: 1, 2, 3, 5 - характеристики стрелочных функций	**Базовый (1-3 минуты)**

<table>
<tr><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th>Вариант 1</th><th>Вариант 2</th><th>Уровень освоения компетенции</th></tr>
<tr><td>Тема 2 Функции</td><td>ПК-2</td><td>ИПК-2.2</td><td>5. Задания открытого типа с развернутым ответом</td><td>Задание 8: Напишите функцию, которая вычисляет факториал числа
Решение:
javascript
function factorial(n) {
if (n === 0) return 1;
return n * factorial(n - 1);
}
console.log(factorial(5));</td><td>Задание 8: Напишите функцию, которая проверяет, является ли строка палиндромом
Решение:
javascript
function isPalindrome(str) {
return str === str.split('').reverse().join('');
}
console.log(isPalindrome("топот"));</td><td>Повышенный (3-5 минут)</td></tr>
<tr><td>Тема 2 Функции</td><td>ПК-2</td><td>ИПК-2.1</td><td>1. Задания закрытого типа на установление соответствия</td><td>Задание 9: Соотнесите понятия функций с их описанием:<table><tr><th>Пункт</th><th>Термин</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>Параметр</td><td>1</td><td>Значение, передаваемое в функцию</td></tr><tr><td>Б</td><td>Аргумент</td><td>2</td><td>Переменная в определении функции</td></tr><tr><td>В</td><td>Замыкание</td><td>3</td><td>Функция + лексическое окружение</td></tr><tr><td>Г</td><td>Рекурсия</td><td>4</td><td>Вызов функции самой себя</td></tr></table>Ответ: А-2, Б-1, В-3, Г-4</td><td>Задание 9: Соотнесите методы функций с их назначением:<table><tr><th>Пункт</th><th>Метод</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>call()</td><td>1</td><td>Вызов с массивом аргументов</td></tr><tr><td>Б</td><td>apply()</td><td>2</td><td>Создание новой функции с привязанным контекстом</td></tr><tr><td>В</td><td>bind()</td><td>3</td><td>Вызов с указанием контекста</td></tr></table>Ответ: А-3, Б-1, В-2</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>Тема 2 Функции</td><td>ПК-2</td><td>ИПК-2.2</td><td>3. Задания комбинированного типа с выбором</td><td>Задание 10: Что такое hoisting в JavaScript?
1. Поднятие объявлений переменных и функций
2. Удаление неиспользуемого кода
3. Оптимизация производительности
4. Сборка мусора
Ответ: 1) Поднятие объявлений переменных и функций</td><td>Задание 10: Какая функция будет вызвана первой при hoisting?
1. Function Declaration
2. Function Expression
3. Arrow Function
4. Все одновременно
Ответ: 1) Function Declaration - поднимается полностью</td><td>Повышенный (3-5 минут)</td></tr>
</table>

<table>
<tr><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th>Вариант 1</th><th>Вариант 2</th><th>Уровень освоения компетенции</th></tr>
<tr><td></td><td></td><td></td><td>одного ответа</td><td></td><td></td><td></td></tr>
<tr><td>Тема 3 Объекты и массивы</td><td>ПК-2</td><td>ИПК-2.1</td><td>1. Задания закрытого типа на установление соответствия</td><td>Задание 11: Соотнесите методы массивов с их описанием:
<table><tr><th>Пункт</th><th>Метод</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>push()</td><td>1</td><td>Добавление элемента в начало</td></tr><tr><td>Б</td><td>pop()</td><td>2</td><td>Удаление элемента с конца</td></tr><tr><td>В</td><td>shift()</td><td>3</td><td>Удаление элемента с начала</td></tr><tr><td>Г</td><td>unshift()</td><td>4</td><td>Добавление элемента в конец</td></tr></table>
Ответ: А-4, Б-2, В-3, Г-1</td><td>Задание 11: Соотнесите методы объектов с их назначением:
<table><tr><th>Пункт</th><th>Метод</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>Object.keys()</td><td>1</td><td>Получение значений свойств</td></tr><tr><td>Б</td><td>Object.values()</td><td>2</td><td>Получение пар ключ-значение</td></tr><tr><td>В</td><td>Object.entries()</td><td>3</td><td>Получение ключей свойств</td></tr></table>
Ответ: А-3, Б-1, В-2</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>Тема 3 Объекты и массивы</td><td>ПК-2</td><td>ИПК-2.2</td><td>3. Задания комбинированного типа с выбором одного ответа</td><td>Задание 12: Какой метод массива используется для преобразования каждого элемента?
1. filter()
2. map()
3. reduce()
4. forEach()
Ответ: 2) map() - создает новый массив с результатами вызова функции для каждого элемента</td><td>Задание 12: Что вернет typeof []?
1. "array"
2. "object"
3. "list"
4. "undefined"
Ответ: 2) "object" - массивы в JavaScript являются объектами</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>Тема 3 Объекты и массивы</td><td>ПК-2</td><td>И ПК-2.2</td><td>5. Задания открытого типа с разверн</td><td>Задание 13: Напишите программу, которая находит среднее арифметическое элементов массива
Решение:
javascript
let numbers = [1, 2, 3, 4, 5];</td><td>Задание 13: Напишите программу, которая создает объект студента
Решение:
javascript
let student = {
name: "Иван",</td><td>Высокий (5-10 минут)</td></tr>
</table>

Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
			утым ответом	let average = numbers.reduce((sum, num) => sum + num, 0) / numbers.length; console.log(`Среднее: ${average}`);	age: 20, subjects: ["Математика", "Физика"] }; console.log(`Студент: ${student.name}, предметов: ${student.subjects.length}`);	
Тема 3 Объекты и массивы	ПК-2	ИПК-2.1	4. Задания комбинированного типа с выбором нескольких ответов	Задание 14: Какие методы массивов возвращают новый массив? 1. map() 2. filter() 3. forEach() 4. reduce() 5. slice() Ответ: 1, 2, 5 - методы, возвращающие новый массив	Задание 14: Какие способы создания объектов существуют в JavaScript? 1. Литерал объекта 2. Конструктор Object 3. Функция-конструктор 4. Класс 5. Object.create() Ответ: 1, 2, 3, 4, 5 - все способы создания объектов	**Базовый (1-3 минуты)**

Тема 3 Объекты и массивы	ПК-2	ИПК-2.2	5. Задания открытого типа с развернутым ответом	Задание 15: Напишите программу, которая фильтрует четные числа из массива
				Решение:
				javascript
				let numbers = [1, 2, 3, 4, 5, 6];
				let even = numbers.filter(num => num % 2 === 0);
				console.log(`Четные числа: ${even}`);
					Задание 15: Напишите программу, которая объединяет два объекта
					Решение:
					javascript
					let obj1 = {a: 1, b: 2};
					let obj2 = {c: 3, d: 4};
					let merged = {...obj1, ...obj2};
					console.log(merged);
						Повышенный (3-5 минут)

Тема 4. Объектная модель	ПК-2	ИПК-2.1	1. Задания закрытого типа	Задание 16: Соотнесите асинхронные конструкции с их описанием:	Задание 16: Соотнесите состояния Promise с их описанием:	**Базовый (1-3 минуты)**

<table>
<tr><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th>Вариант 1</th><th>Вариант 2</th><th>Уровень освоения компетенции</th></tr>
<tr><td>браузера</td><td></td><td></td><td>на установление соответствия</td><td>
<table>
<tr><th>Пункт</th><th>Концепция</th><th>Пункт</th><th>Описание</th></tr>
<tr><td>А</td><td>Callback</td><td>1</td><td>Промис с улучшенным синтаксисом</td></tr>
<tr><td>Б</td><td>Promise</td><td>2</td><td>Функция, передаваемая как аргумент</td></tr>
<tr><td>В</td><td>async/await</td><td>3</td><td>Объект для отложенных вычислений</td></tr>
</table>
Ответ: А-2, Б-3, В-1
</td><td>
<table>
<tr><th>Пункт</th><th>Состояние</th><th>Пункт</th><th>Описание</th></tr>
<tr><td>А</td><td>pending</td><td>1</td><td>Успешное выполнение</td></tr>
<tr><td>Б</td><td>fulfilled</td><td>2</td><td>Ожидание выполнения</td></tr>
<tr><td>В</td><td>rejected</td><td>3</td><td>Выполнено с ошибкой</td></tr>
</table>
Ответ: А-2, Б-1, В-3
</td><td></td></tr>
<tr><td>Тема 4. Объектная модель браузера</td><td>ПК-2</td><td>ИПК-2.2</td><td>3. Задания комбинированного типа с выбором одного ответа</td><td>Задание 17: Что такое Event Loop?
1. Механизм обработки асинхронных операций
2. Цикл событий в DOM
3. Алгоритм сортировки
4. Метод отладки
Ответ: 1) Механизм обработки асинхронных операций</td><td>Задание 17: Какой метод Promise используется для обработки ошибок?
1. then()
2. catch()
3. finally()
4. error()
Ответ: 2) catch() - для обработки отклоненных промисов</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>Тема 4. Объектная модель браузера</td><td>ПК-2</td><td>ИПК-2.2</td><td>5. Задания открытого типа с развернутым ответом</td><td>Задание 18: Напишите программу с использованием Promise для загрузки данных
Решение:
javascript
function loadData() {
return new Promise((resolve, reject) => {
setTimeout(() => resolve("Данные загружены"), 1000);
});
}</td><td>Задание 18: Напишите программу с использованием async/await
Решение:
javascript
async function fetchData() {
let response = await fetch('/api/data');
let data = await response.json();
console.log(data);
}
fetchData();</td><td>Высокий (5-10 минут)</td></tr>
</table>

Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
				loadData().then(data => console.log(data));		
Тема 4. Объектная модель браузера	ПК-2	ИПК-2.1	4. Задания комбинированного типа с выбором нескольких ответов	Задание 19: Какие методы есть у Promise? 1. then() 2. catch() 3. finally() 4. resolve() 5. reject() Ответ: 1, 2, 3 - методы экземпляра Promise	Задание 19: Какие статические методы есть у Promise? 1. Promise.all() 2. Promise.race() 3. Promise.resolve() 4. Promise.reject() 5. Promise.delay() Ответ: 1, 2, 3, 4 - статические методы Promise	**Базовый (1-3 минуты)**

Тема 4. Объектная модель браузера	ПК-2	ИПК-2.2	5. Задания открытого типа с развернутым ответом	Задание 20: Напишите программу, которая обрабатывает несколько промисов параллельно
				Решение:
				javascript
				Promise.all([
				fetch('/api/users'),
				fetch('/api/posts')
				]).then(responses => {
				console.log('Все данные загружены');
				});
					Задание 20: Напишите программу с обработкой ошибок в асинхронном коде
					Решение:
					javascript
					async function loadWithErrorHandling() {
					try {
					let data = await fetch('/api/data');
					console.log(data);
					} catch (error) {
					console.error('Ошибка:', error);
					}
					}
						Высокий (5-10 минут)

Тема 4. Объектная модель	ПК-2	ИПК-2.1	2. Задания закрытого типа	Задание 21: Расположите объекты BOM в иерархическом порядке: 1. window 2. navigator	Задание 21: Расположите этапы загрузки страницы: 1. Парсинг HTML 2. Запрос к серверу	**Базовый (1-3 минуты)**

Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
браузера			на установление последовательности	3. document 4. screen Ответ: 1-2-4-3	3. Построение DOM 4. Загрузка ресурсов Ответ: 2-1-3-4	
Тема 4. Объектная модель браузера	ПК-2	ИПК-2.2	4. Задания комбинированного типа с выбором нескольких ответов	Задание 22: Какие объекты входят в Browser Object Model (BOM)? 1. window 2. navigator 3. screen 4. history 5. location Ответ: 1, 2, 3, 4, 5 - все являются частью BOM	Задание 22: Какие методы window используются для работы с диалоговыми окнами? 1. alert() 2. prompt() 3. confirm() 4. console.log() 5. document.write() Ответ: 1, 2, 3 - методы для взаимодействия с пользователем	**Базовый (1-3 минуты)**
Тема 4. Объектная модель браузера	ПК-2	ИПК-2.2	5. Задания открытого типа с развернутым ответом	Задание 23: Напишите программу, которая показывает информацию о браузере Решение: javascript console.log(`Браузер: ${navigator.userAgent}`); console.log(`Язык: ${navigator.language}`); console.log(`Платформа: ${navigator.platform}`);	Задание 23: Напишите программу, которая работает с localStorage Решение: javascript localStorage.setItem('username', 'Иван'); let user = localStorage.getItem('username'); console.log(`Пользователь: ${user}`);	**Повышенный (3-5 минут)**
Тема 4. Объектная	ПК-2	ИПК-2.1	1. Задания закрыто	Задание 24: Соотнесите объекты BOM с их назначением:	Задание 24: Соотнесите методы хранения данных с их характеристиками:	**Базовый (1-3 минуты)**

<table>
<tr><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th>Вариант 1</th><th>Вариант 2</th><th>Уровень освоения компетенции</th></tr>
<tr><td>модель браузера</td><td></td><td></td><td>го типа на установление соответствия</td><td>
<table>
<tr><th>Пункт</th><th>Объект</th><th>Пункт</th><th>Описание</th></tr>
<tr><td>А</td><td>location</td><td>1</td><td>Информация об экране</td></tr>
<tr><td>Б</td><td>history</td><td>2</td><td>URL текущей страницы</td></tr>
<tr><td>В</td><td>screen</td><td>3</td><td>История браузера</td></tr>
<tr><td>Г</td><td>navigator</td><td>4</td><td>Информация о браузере</td></tr>
</table>
Ответ: А-2, Б-3, В-1, Г-4
</td><td>
<table>
<tr><th>Пункт</th><th>Технология</th><th>Пункт</th><th>Описание</th></tr>
<tr><td>А</td><td>localStorage</td><td>1</td><td>Временное хранение в сессии</td></tr>
<tr><td>Б</td><td>sessionStorage</td><td>2</td><td>Постоянное хранение</td></tr>
<tr><td>В</td><td>cookies</td><td>3</td><td>Хранение с сроком действия</td></tr>
</table>
Ответ: А-2, Б-1, В-3
</td><td></td></tr>
<tr><td>Тема 4. Объектная модель браузера</td><td>ПК-2</td><td>ИПК-2.2</td><td>3. Задания комбинированного типа с выбором одного ответа</td><td>Задание 25: Какой объект содержит информацию об истории браузера?
1. history
2. location
3. navigator
4. window
Ответ: 1) history - управляет историей браузера</td><td>Задание 25: Какой метод используется для перезагрузки страницы?
1. location.reload()
2. window.refresh()
3. document.reload()
4. browser.reload()
Ответ: 1) location.reload() - перезагружает текущую страницу</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>Тема 5. Объектная модель документа</td><td>ПК-2

ПК-6.</td><td>ИПК-2.1

ИПК-6.1</td><td>1. Задания закрытого типа на установление соответствия</td><td>Задание 26: Соотнесите методы DOM с их назначением:
<table>
<tr><th>Пункт</th><th>Метод</th><th>Пункт</th><th>Описание</th></tr>
<tr><td>А</td><td>getElementById()</td><td>1</td><td>Поиск по классу</td></tr>
<tr><td>Б</td><td>getElementsByClassName()</td><td>2</td><td>Поиск по тегу</td></tr>
<tr><td>В</td><td>getElementsByTagName()</td><td>3</td><td>Поиск по CSS-селектору</td></tr>
<tr><td>Г</td><td>querySelector()</td><td>4</td><td>Поиск по id</td></tr>
</table>
Ответ: А-4, Б-1, В-2, Г-3</td><td>Задание 26: Соотнесите свойства элемента с их описанием:
<table>
<tr><th>Пункт</th><th>Свойство/Метод</th><th>Пункт</th><th>Описание</th></tr>
<tr><td>А</td><td>innerHTML</td><td>1</td><td>Текстовое содержимое</td></tr>
<tr><td>Б</td><td>textContent</td><td>2</td><td>HTML содержимое</td></tr>
<tr><td>В</td><td>className</td><td>3</td><td>Значение атрибута</td></tr>
<tr><td>Г</td><td>getAttribute()</td><td>4</td><td>CSS классы</td></tr>
</table>
Ответ: А-2, Б-1, В-4, Г-3</td><td>Базовый (1-3 минуты)</td></tr>
</table>

Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
Тема 5. Объектная модель документа	ПК-2	ИПК-2.2	3. Задания комбинированного типа с выбором одного ответа	Задание 27: Какой метод используется для создания нового элемента? 1. document.createElement() 2. document.newElement() 3. document.addElement() 4. document.buildElement() Ответ: 1) document.createElement() - стандартный метод создания элементов	Задание 27: Как добавить элемент в конец другого элемента? 1. appendChild() 2. insertBefore() 3. addToEnd() 4. putElement() Ответ: 1) appendChild() - метод для добавления дочернего элемента	**Базовый (1-3 минуты)**

Тема 5. Объектная модель документа	ПК-2	ИПК-2.2	5. Задания открытого типа с развернутым ответом	Задание 28: Напишите программу, которая создает и добавляет элемент на страницу
				Решение:
				javascript
				let div = document.createElement('div');
				div.textContent = 'Новый элемент';
				div.className = 'my-class';
				document.body.appendChild(div);
					Задание 28: Напишите программу, которая изменяет стили элемента
					Решение:
					javascript
					let element = document.getElementById('myElement');
					if (element) {
					element.style.color = 'red';
					element.style.fontSize = '20px';
					}
						Высокий (5-10 минут)

Тема 5. Объектная модель документа	ПК-2	ИПК-2.1	4. Задания комбинированного типа с выбором несколь	Задание 29: Какие методы используются для поиска элементов в DOM? 1. getElementById() 2. getElementsByClassName() 3. querySelector() 4. findElement() 5. searchElements() Ответ: 1, 2, 3 - стандартные методы поиска	Задание 29: Какие свойства используются для навигации по DOM? 1. parentNode 2. children 3. nextSibling 4. previousElementSibling 5. closest() Ответ: 1, 2, 3, 4, 5 - все свойства для навигации	**Повышенный (3-5 минут)**

Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
			ких ответов			

Тема 5. Объектная модель документа	ПК-2	ИПК-2.2	5. Задания открытого типа с развернутым ответом	Задание 30: Напишите программу, которая динамически создает список
				Решение:
				javascript
				let ul = document.createElement('ul');
				['Пункт 1', 'Пункт 2', 'Пункт 3'].forEach(text => {
				let li = document.createElement('li');
				li.textContent = text;
				ul.appendChild(li);
				});
				document.body.appendChild(ul);
					Задание 30: Напишите программу, которая изменяет атрибуты элемента
					Решение:
					javascript
					let img = document.getElementById('myImage');
					img.setAttribute('src', 'new-image.jpg');
					img.setAttribute('alt', 'Новое изображение');
						Повышенный (3-5 минут)

Тема 6. События и их обработка	ПК-2	ИПК-2.1	2. Задания закрытого типа на установление последовательности	Задание 31: Расположите фазы распространения события в правильном порядке: 1. Целевая фаза 2. Фаза перехвата 3. Фаза всплытия Ответ: 2-1-3	Задание 31: Расположите этапы работы с событием: 1. Назначение обработчика 2. Возникновение события 3. Выполнение обработчика 4. Получение объекта события Ответ: 1-2-4-3	**Базовый (1-3 минуты)**
Тема 6. События и их обработка	ПК-2	ИПК-2.2	4. Задания комбинированного типа	Задание 32: Какие способы назначения обработчиков событий существуют? 1. Через атрибут HTML 2. Через свойство DOM-элемента 3. С помощью addEventListener()	Задание 32: Какие методы объекта Event используются для управления событиями? 1. preventDefault() 2. stopPropagation() 3. stopImmediatePropagation()	**Базовый (1-3 минуты)**

<table>
<tr><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th>Вариант 1</th><th>Вариант 2</th><th>Уровень освоения компетенции</th></tr>
<tr><td></td><td></td><td></td><td>с выбором нескольких ответов</td><td>4. Через CSS
5. С помощью JavaScript-комментарии
Ответ: 1, 2, 3 - допустимые способы назначения обработчиков</td><td>4. cancelEvent()
5. stopBubbling()
Ответ: 1, 2, 3 - стандартные методы управления событиями</td><td></td></tr>
<tr><td>Тема 6. События и их обработка</td><td>ПК-2</td><td>ИПК-2.2</td><td>5. Задания открытого типа с развернутым ответом</td><td>Задание 33: Напишите программу, которая обрабатывает клик по кнопке
Решение:
javascript
document.getElementById('myButton').addEventListener('click', function(event) {
console.log('Кнопка нажата');
event.preventDefault();
});</td><td>Задание 33: Напишите программу, которая обрабатывает ввод в текстовое поле
Решение:
javascript
document.getElementById('myInput').addEventListener('input', function(event) {
console.log(`Введен текст: ${event.target.value}`);
});</td><td>Повышенный (3-5 минут)</td></tr>
<tr><td>Тема 6. События и их обработка</td><td>ПК-2</td><td>ИПК-2.1</td><td>1. Задания закрытого типа на установление соответствия</td><td>Задание 34: Соотнесите типы событий с их описанием:
<table><tr><th>Пункт</th><th>Событие</th><th>Пункт</th><th>Описание</th></tr><tr><td>А</td><td>click</td><td>1</td><td>Изменение значения поля</td></tr><tr><td>Б</td><td>input</td><td>2</td><td>Загрузка элемента</td></tr><tr><td>В</td><td>load</td><td>3</td><td>Щелчок мышью</td></tr><tr><td>Г</td><td>submit</td><td>4</td><td>Отправка формы</td></tr></table>
Ответ: А-3, Б-1, В-2, Г-4</td><td>Задание 34: Соотнесите объекты событий с их свойствами:
<table><tr><th>Пункт</th><th>Тип события</th><th>Пункт</th><th>Свойства</th></tr><tr><td>А</td><td>MouseEvent</td><td>1</td><td>clientX, clientY</td></tr><tr><td>Б</td><td>KeyboardEvent</td><td>2</td><td>key, code</td></tr><tr><td>В</td><td>InputEvent</td><td>3</td><td>data, target</td></tr></table>
Ответ: А-1, Б-2, В-3</td><td>Базовый (1-3 минуты)</td></tr>
<tr><td>Тема 6. События и их</td><td>ПК-2</td><td>ИПК-2.2</td><td>3. Задания комбинированн</td><td>Задание 35: Что такое делегирование событий?
1. Обработка событий на родительском элементе</td><td>Задание 35: Как остановить всплытие события?
1. event.stopPropagation()
2. event.preventDefault()</td><td>Базовый (1-3 минуты)</td></tr>
</table>

Тема занятия	Код компетенции	Индикатор	Тип задания	Вариант 1	Вариант 2	Уровень освоения компетенции
обработка			ого типа с выбором одного ответа	2. Передача событий между окнами 3. Отмена событий 4. Создание пользовательских событий Ответ: 1) Обработка событий на родительском элементе	3. event.stop() 4. event.cancelBubble() Ответ: 1) event.stopPropagation() - останавливает всплытие	

Критерии оценивания диагностической работы:

Оценка	Критерии оценивания
отлично	от 90 до 100 % правильно выполненных заданий
хорошо	от 70 до 89 % правильно выполненных заданий
удовлетворительно	от 50 до 69 % правильно выполненных заданий
неудовлетворительно	менее 50 % правильно выполненных заданий