Балаковский инженерно-технологический институт – филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»

Факультет атомной энергетики и технологий
Кафедра «Информационные системы и технологии»

РАБОЧАЯ ПРОГРАММА

по дисциплине «Создание интерактивных приложений для интернет»

Направления подготовки/специальность
«09.03.02 Информационные системы и технологии»

Основная профессиональная образовательная программа
«Информационные системы и технологии»

Квалификация выпускника
Бакалавр

Форма обучения
Заочная

Балаково

Цель освоения дисциплины

Целями освоения дисциплины «Создание интерактивных приложений для интернет» в области обучения, воспитания, развития, соотнесенные с общими целями ООП ВО и требованиями профессиональных стандартов («06.016. Руководитель проектов в области информационных технологий», «06.015. Специалист по информационным системам») являются: формирование профессиональной ответственности в области проектирования, разработки, эксплуатации и сопровождения информационных систем и технологий, используемых для информационного обеспечения деятельности различных предприятий и организаций, в том числе организаций атомной отрасли; формализация и алгоритмизация поставленных задач; написание программного кода с использованием языков программирования, определения и манипулирования данными; проверка и отладка программного кода; проверка работоспособности программного обеспечения; анализ требований к программному обеспечению.

Место дисциплины в структуре ООПВО

Необходимыми условиями для освоения дисциплины являются знания, умения и практические навыки по предшествующим дисциплинам и практикам в соответствии с требованиями освоения компетенций:

Учебная практика (по получению первичных профессиональных умений и навыков)

Знания, умения и практические навыки, полученные при освоении дисциплины, необходимы при изучении следующих дисциплин и прохождения практик:

Экономика

Современные среды визуального программирования

Методы и средства проектирования информационных систем и технологий

Управление данными

Современные технологии интернет-программирования

Информационный менеджмент

Технологии создания web –ресурсов /Основы web-программирования

Электронный документооборот / Системы электронного документооборота на предприятии

Производственная практика (проектно-технологическая)

Производственная практика (преддипломная)

Государственная итоговая аттестация

При освоении данной дисциплины студент сможет частично продемонстрировать следующие обобщенные трудовые функции: Разработка архитектуры ИС, Проектирование и дизайн ИС, Сбор информации для инициации проекта в соответствии с полученным заданием

Компетенции обучающегося, формируемые в результате освоениядисциплины

В процессе освоения данной дисциплины у студента формируются следующие компетенции:

универсальные

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	З-УК-2 Знать: виды ресурсов и ограничений для решения профессиональных задач; основные методы оценки разных способов решения задач; действующее законодательство и правовые нормы, регулирующие профессиональную деятельность У-УК-2 Уметь: проводить анализ поставленной цели и формулировать задачи, которые необходимо решить для ее достижения; анализировать альтернативные варианты решений для достижения намеченных результатов; использовать нормативно-правовую документацию в сфере профессиональной деятельности В-УК-2 Владеть: методиками разработки цели и задач проекта; методами оценки потребности в ресурсах, продолжительности и стоимости проекта, навыками работы с нормативно- правовой документацией

профессиональные

Задачи профессиональной деятельности (ЗПД)	Объект или область знания	Код и наименование компетенции	Индикаторы достижения компетенции
проектирование базовых и прикладных информационных технологий	информационные процессы, технологии, системы и сети, их инструментальное (программное, техническое, организационное) обеспечение, способы и методы проектирования, отладки, производства и эксплуатации информационных технологий и систем в различных областях и сферах деятельности.	ПК-6.1 Способен разрабатывать и адаптировать прикладное программное обеспечение	З-ПК-6.1 Знать: принципы и методы разработки, внедрения и адаптации прикладного программного обеспечения У-ПК-6.1 Уметь: программировать приложения и создавать программные прототипы решения прикладных задач В-ПК-6.1 Владеть: инструментальными средствами проектирования и разработки прикладного программного обеспечения
выбор исходных данных для проектирования	информационные процессы, технологии, системы и сети, их инструментальное (программное, техническое, организационное) обеспечение, способы и методы проектирования, отладки, производства и эксплуатации информационных технологий и систем в различных областях и сферах деятельности	ПК-12 Способен проводить выбор исходных данных для проектирования с учетом требований заказчика	З-ПК-12 Знать: требования к разработке технического задания, его структуру и принципы составления У-ПК-12 Уметь: анализировать исходную документацию заказчика В-ПК-12 Владеть: методикой составления технического задания

Задачи воспитания, реализуемые в рамках освоения дисциплины

Направление/цели	Создание условий, обеспечивающих	Использование воспитательного потенциала учебной дисциплины	Вовлечение в разноплановую внеучебную деятельность
Профессиональное воспитание	формирование профессиональной ответственности в области проектирования, разработки, эксплуатации и сопровождения информационных систем и технологий, используемых для информационного обеспечения деятельности различных предприятий и организаций, в том числе организаций атомной отрасли **(В26).**	1. Использование для формирования чувства личной ответственности в области проектирования, разработки, эксплуатации и сопровождения информационных систем и технологий, используемых для информационного обеспечения деятельности различных предприятий и организаций, в том числе организаций атомной отрасли воспитательного потенциала блока профессиональных дисциплин: Управление данными; Интерактивное программирование web-приложений; Создание интерактивных приложений для интернет; Технологии создания web –ресурсов; Основы web-программирования; Информационная теория управления 2. Развитие навыков творческого мышления путем содействия и поддержки	1. Организация и проведение научно-практических конференций, форумов, круглых столов, вебинаров по вопросам профессиональной деятельности 2. Участие в студенческих олимпиадах и конкурсах научных проектов, творческих мероприятиях, конкурсах профессионального мастерства, в том числе по стандартам WorldSkills. 3. Участие в подготовке публикаций в периодических научных изданиях; 4. Участие в деятельности студенческого

		участия студентов в научно-практических мероприятиях внутривузовского регионального и/или всероссийского уровня по информатике и информационным технологиям.	научного общества

Структура и содержание учебной дисциплины

Дисциплина преподается студентам в 3-ем семестре. Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 ак. часа.

Календарный план

№ Раздела	№ Темы	Наименование раздела (темы) дисциплины	Виды учебной деятельности (в часах)					Аттестация раздела (форма)	Максимальный балл за раздел
			Всего	Лекции	Лабораторные	Практические	СРС		
1	**Реализация сценариев JavaScript**							КИ	**25**
	1	Типы данных и переменные, типы данных, глобальные и локальные переменные	26	2	-	2	22		
	2	Арифметические операторы, операторы присваивания, операторы инкремента и декремента, операторы сравнения, логические операторы, цикл for, цикл while, вложенный цикл	26/2	2	-	2/2	22		
2	**Реализация сценариев jQuery**							КИ	**25**
	4	Методы jQuery, манипуляции HTML-элементами, чтение и изменение CSS-свойств, классов и атрибутов, обход DOM и выборка html-элементов	30/3	1/1	-	4/2	25		
	5	jQuery анимация, методы объекта windows, события jQuery, селекторы jQuery	26/1	1/1	-	2	23		
Вид промежуточной аттестации								**З**	**50**
Итого			**108/6**	**6/2**	**-**	**10/4**	**92**		**100**

* - занятия в интерактивной форме

Сокращенное наименование форм текущего контроля и аттестации разделов:

Обозначение	Полное наименование
КИ	Контроль по итогам
З	Зачет

Содержание лекционного курса

Темы лекции. Вопросы, отрабатываемые на лекции	Всего часов	Учебно-методическое обеспечение
1	**2**	**3**
Лекция 1. Типы данных и переменные, типы данных, глобальные и локальные переменные 1. Подключение сценариев к HTML-документу 2. Типы данных и переменные в JavaScript	2	1-3

3. Переменные в JavaScript 4. Типы данных переменных. 5. Глобальные и локальные переменные.		
Лекция 2. Арифметические операторы, операторы присваивания, операторы инкремента и декремента, операторы сравнения, логические операторы, цикл for, цикл while, вложенный цикл. 1. Арифметические операторы. Операторы присваивания. 2. Операторы инкремента и декремента. Операторы сравнения. 3. Логические операторы. Побитовые операторы. 4. Строковые операторы. Специальные операторы. Коментарии в JavaScript. 5. Цикл for. Цикл for...in. Цикл while. Цикл do...while. 6. Бесконечный цикл. Вложенный цикл. Управление циклом.	2	1-3
Лекция 3. Методы jQuery, манипуляции HTML-элементами, чтение и изменение CSS-свойств, классов и атрибутов, обход DOM и выборка html-элементов. 1. Методы jQuery. 2. Добавление содержимого на страницу. 3. Добавление элементов. 4. Замена и удаление элементов. 5. Добавление и удаление класса. Изменение атрибутов элементов. 6. Изменение свойств элемента. Получение и изменение размеров т координат элемента. 7. Обход DOM и выборка html-элементов.	1	1-3
Лекция 4. jQuery анимация, методы объекта windows, события jQuery, селекторы jQuery. 1. Создание эффектов с помощью .animate(). 2. Анимационные эффекты jQuery 3. Управление очередью анимации. 4. Управление анимацией через свойства объекта jQuery. 5. События мыши. 6. События клавиатуры.	1	1-3

Перечень практических занятий

Тема практического занятия. Вопросы, отрабатываемые на практическом занятии	Всего часов	Учебно-методическое обеспечение
Основы взаимодействия с web-документами в JavaScript	2	1-3
Операции с циклами. Введение в объектную модель в JavaScript	4	1-3
Принципы работы с формами, иерархией объектов web-страницы	4	1-3

Перечень лабораторных работ – не предусмотрены учебным планом

Задания для самостоятельной работы студентов

Вопросы для самостоятельного изучения (задания)	Всего часов	Учебно-методическое обеспечение
1	**2**	**3**
Справочники и спецификации. Редакторы кода. Консоль разработчика. Строгий режим - "use strict". Переменные. Типы данных. Взаимодействие alert, propt, confirm. Базовые операторы, математика, операторы сравнения. Условное ветвление if, '?'. Логические операторы, оператор объединения с null '??'.	22	1-3
Конструкция "switch". Функции, Function Expression. Функции-стрелки. Отладка в браузере. Автоматическое тестирование с использованием фреймворка Mocha. Полифилы. Объекты, копирование объ-	22	1-3

ектов и ссылки, сборка мусора, методы объекта "this", конструкторы, создание объекта через "new", опциональная цепочка '?'. Тип данных Symbol. Преобразование объектов в примитивы. Методы у примитивов, числа, строки, массивы, методы массивов, перебираемые объекты, деструктурирующее присваивание, дата и время, рекурсия и стек, остаточные параметры и оператор расширения, замыкание, устаревшее слово "var", глобальный объект, объект функции NFE, синтаксис "newFunction", привязка контекста к функции.		
Класс: базовый синтаксис, промисы, цепочка промисов, промисификация, наследование классов, статические свойства и методы, приватные и защищённые методы и свойства. Расширение встроенных классов, примеси. Обработка ошибок "try..catch", пользовательские ошибки.	48	1-3

Расчетно-графическая работа не предусмотрена учебным планом

Курсовая работа не предусмотрена учебным планом

Образовательные технологии

При реализации учебного материала курса используются различные образовательные технологии, способствующие созданию атмосферы свободной и творческой дискуссии как между преподавателем и студентами, так и в студенческой группе. Целью при этом является выработка у студентов навыков и компетенций, позволяющих самостоятельно вести исследовательскую и научно-педагогическуюработу.

Практическая подготовка при реализации учебной дисциплины организуется путем проведения практических занятий, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

Практическая подготовка также включает в себя занятия лекционного типа, которые предусматривают передачу учебной информации обучающимся, необходимой для последующего выполнения работ, связанных с будущей профессиональной деятельностью.

Аудиторные занятия проводятся в виде лекций, практических занятий с использованием ПК и компьютерного проектора. Самостоятельная работа студентов проводится под руководством преподавателей, с оказанием консультаций и помощи при подготовке к практическим занятиям.

Фонд оценочных средств

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

№ п/п	Наименование контролируемых разделов (темы)	Код и наименование индикатора достижения компетенций	Наименование оценочного средства
Входной контроль			
1	Входной контроль		Вопросы входного контроля (письменно)
Аттестация разделов, текущий контроль успеваемости			
2	Реализация сценариев JavaScript	З-УК-2, У-УК-2, В-УК-2, З-ПК-6.1, У-ПК-6.1, В-ПК-6.1, З-ПК-12, У-ПК-12, В-ПК-12	Контроль итогов в форме тестирования (письменно)
3	Реализация сценариев jQuery	З-УК-2, У-УК-2, В-УК-2, З-ПК-6.1, У-ПК-6.1, В-ПК-6.1, З-ПК-12, У-ПК-12, В-ПК-12	Контроль итогов в форме тестирования (письменно)

Промежуточная аттестация			
4	Зачет	З-УК-2, У-УК-2, В-УК-2, З-ПК-6.1, У-ПК-6.1, В-ПК-6.1, З-ПК-12, У-ПК-12, В-ПК-12	тест (письменно)

Входной контроль предназначен для выявления пробелов в знаниях студентов и готовности их к получению новых знаний. Оценочные средства для входного контроля представляют собой вопросы, которые задаются студентам в письменной форме.

Перечень вопросов входного контроля

1. Основные этапы в развитии HTML.
2. Теговая модель и базовая структура HTML-документов.
3. Основные требования к заглавной части HTML.
4. Дерево элементов HTML. Родственные связи между элементами. Принципы наследования.
5. Основные элементы HTML для форматирования текста.
6. Дополнительные (вспомогательные) элементы HTML для форматирования текста.
7. Основные элементы HTML для вставки изображений и создания гиперссылок.
8. Основные элементы HTML для работы со списками.
9. Основные элементы HTML для работы с таблицами.
10. Блочные и строчные элементы HTML. Определения и основные особенности.
11. Универсальные элементы HTML. Назначение и принципы использования.
12. Атрибуты элементов HTML. Принципы наследования. Универсальные атрибуты.
13. Адресация в HTML. Варианты и примеры абсолютной и относительной адресации.
14. Каскадные таблицы стилей CSS. Предпосылки появления и история развития.
15. Основы синтаксиса CSS. Назначение и особенности использования.
16. Методы определения CSS. Встраивание, вложение и связывание.
17. Методы определения CSS. Принципы каскадирования и наследования стилей.
18. Единицы измерения в CSS. Перечень абсолютных и относительных единиц измерения.
19. Способы задания цвета в CSS. Цветовые таблицы (палитры). Принципы подбора цвета.
20. Шрифтовое оформление в CSS. Гарнитуры. Семейство и тип шрифта. Понятие о «безопасных» шрифтах.
21. Шрифтовое оформление в CSS. Настройка типа, размера, начертания и модификации шрифта.
22. Собирательное шрифтовое оформление.
23. Оформление текста в CSS. Выравнивание, отступы и промежутки, трансформация, интервалы
24. и декорация.
25. Базовый синтаксис CSS. Селекторы тегов.
26. Базовый синтаксис CSS. Классы и идентификаторы.
27. Базовый синтаксис CSS. Контекстные, соседние и дочерние селекторы.
28. Базовый синтаксис CSS. Селекторы атрибутов.
29. Блочная модель CSS. Рамки, поля и отступы.
30. Блочная модель CSS. Позиционирование элементов.
31. Блочная модель CSS. Многослойность, выравнивание и обтекание.

Текущий контроль – это непрерывно осуществляемый мониторинг уровня усвоения знаний и формирования умений и навыков в течение семестра. Текущий контроль знаний, умений и навыков студентов осуществляется в ходе учебных (аудиторных) занятий, проводимых по расписанию. Формами текущего контроля выступают используются коллоквиум, выполнение практических контрольных заданий.

Перечень вопросов коллоквиума

1. Как создать объект?
2. Что такое прототипы?

3. В чем разница между Call, Apply и Bind?
4. Что такое JSON и какие у него есть методы?
5. Что делает метод Array.slice()?
6. Что делаем метод Array.splice()?
7. В чем разница между slice() и splice()?
8. Как сравниваются объекты (objects) и карты (maps)?
9. В чем разница между операторами "=="и "==="?
10. Что такое лямбда- или стрелочные функции?
11. Почему функции называют объектами первого класса?
12. Что такое функция первого порядка?
13. Что такое функция высшего порядка?
14. Что такое унарная функция?
15. Что такое каррирование (currying)?
16. Что такое чистая функция?
17. Для чего используется ключевое слово «let»?
18. В чем разница между let и var?
19. Почему в качестве ключевого слова было выбрано слово «let»?
20. Как переопределить переменную в блоке switch?
21. Что такое временная мертвая зона?
22. Что такое немедленно вызываемое функциональное выражение (IIFE, Immediately Invoked Function Expression)?
23. В чем заключаются преимущества использования модулей?
24. Что такое запоминание или мемоизация?
25. Что такое поднятие переменных (hoisting)?
26. Что такое класс?
27. Что такое замыкание?
28. Что такое модуль?
29. Зачем нужны модули?
30. Что такое область видимости?
31. Что такое сервис-воркер (service worker)?
32. Как взаимодействовать с объектной моделью документа (Document Object Model, DOM) с помощью сервис-воркеров?
33. Как повторно использовать информацию при перезапуске сервис-воркера?
34. Что такое индексированная база данных (IndexedDB)?
35. Что такое веб-хранилище (Web Storage)?
36. Что такое postMessage?
37. Как использовать jquery для изменения или удаления отдельных параметров из выбора?
38. Какие преимущества jQuery?
39. Чем отличается JavaScript от jQuery?
40. Является ли jQuery стандартом W3C?
41. Что означает символ доллара ($) в jQuery?
42. Можем ли мы иметь несколько DOCUMENT.READY() на одной и той же странице?
43. Что такое jQuery.NoConflict?
44. Какая разница между jQuery.js и jQuery.min.js?
45. Что такое селектор в jQuery, какие типы селекторов вы знаете?
46. Какие селекторы в jQuery самые быстрые?
47. Как выбрать все элементы с id = idname?
48. Как выбрать все элементы div с id = idname?
49. Как выбрать все элементы с class = classname?
50. Как выбрать все элементы div с class = classname?
51. Как выбрать предыдущий элемент от найденного?

Перечень практических заданий

1. Написать функцию, которая принимает два числа и возвращает результат их умножения

2. Написать функцию, которой передаем, имя, фамилия и возраст, и получаем строку "Привет Иван Петров с возрастом 17 лет" (только здесь данные, которые были переданы в функцию)

3. Написать функцию, которая принимает пол человека ('M','F') в виде строки, результат функции возвращает строку "Ваш пол мужской" (или женский) или же "Ваш пол не определен"

4. Сделайте функцию, которая принимает параметром число от 1 до 7, а возвращает день недели на русском языке.

5. Написать функцию, которая принимает строку (в этом тексте 3-5 предложений), верните каждое первое слово в каждом предложении, через запятую.

6. Написать функцию, которой передаем имя, и она возвращает приветствие в зависимости от времени суток (Доброе утро\день\вечер\ночи Иван)

7. Вывести числа от 1 до 100 в столбец. К каждой цифре подписать состояние возраста (1-17 ребенок, 18-30 - молодой, 30-55 - зрелый, от 55 - старый). Например. 33 - зрелый

8. Создай новую функцию, в которую передаешь имя и возраст человека и получаешь сообщение (Иван имеет возраст 44 и он зрелый). А также вызови внутри своей функции, функцию из прошлого задания

9. Сделай функцию, которая принимает массив любых целых чисел, которая возвращает истину, если все элементы четные, если бы хотя бы один элемент не четный, то false.

10. Сделай функцию, которая принимает массив любых целых чисел, которая возвращает истину, если хотя бы один элемент нечетный, если все четные, то false.

11. Сделай функцию, которая принимает массив любых целых чисел, которая возвращает новый массив, где все элементы кратны пяти. ([1,2,5,12,15,21] вернет [5,15])

12. Написать функцию, которая принимает массив чисел, например [1,2,3,4,5] и функция возвращает среднее арифметическое, (округлить результат до десятых)

13. Написать функцию, которая принимает массив чисел, например [1,2,3,4,5], и переносит первый элемент массива в конец (например можно засунуть первый элемент в конец, затем удалить первый элемент), попробуй несколькими способами сделать, если догадаешься

Аттестация раздела по дисциплине проводится как контроль по итогам в форме тестирования. Тест содержит от 10 вопросов. На выполнение задания отводится от 20 минут. Тест – это форма контроля, направленная на проверку уровня освоения контролируемого теоретического и практического материала по дидактическим единицам дисциплины (терминологический аппарат, основные методы).

Примерный перечень тестовых заданий:

Тестовые задания 1. (Т1)

1. Чему равна переменная name

```
1  let name = "пупкин".replace("п", "д")
```

a) дудкин.
b) дупкин.
c) пупкин.
d) ляпкин-тяпкин.

2. Чему равно 0 || "" || 2 || undefined || true || false ?

a) 0
b) «»
c) 2
d) Undefined
e) True
f) Falce

3. Чему равна сумма [] + 1 + 2?

a) 1
b) NaN
c) Undefined
d) 12
e) Другое

4. Что делает код:

```
break me;
```

a) Ломает интерпретатор javascript.
b) Выходит из текущего блока цикла или switch на метку «me».
c) Выдает ошибку.

5. Что делает оператор ===?

a) Сравнивает по ссылке, а не по значению.
b) Сравнивает без приведения типа.
c) Нет такого оператора.

6. Что выведет этот код? Посмотрите на него очень внимательно, в этом вопросе есть подвох.

```
let a = [1,2]

(function() { alert(a) })()
```

a) [object Array]
b) [object Object]
c) 1,2
d) Будет ошибка.

7. Есть ли различия между проверками:

```
if( x <= 100 ) {...}
// и
if( !(x > 100) ) {...}
```

a) Да, существует значение x, для которого они работают по-разному.
b) Нет, они полностью взаимозаменяемы.
c) Зависит от браузера.

8. Что выведет этот код?

```
alert( 20e-1['toString'](2) );
```

a) 2
b) 10
c) 20
d) NaN
e) В коде ошибка.

9. Какой результат будет у выражения ниже?

```
null + {0:1}[0] + [,[1],][1][0]
```

a) 0
b) 1
c) 2
d) undefined
e) NaN

10. Что выведет alert?

```
alert(str); // ?
let str = "Hello";
```

a) Hello
b) undefined
c) Будет ошибка.

11. Что выведет этот скрипт?

```
"use strict";

a = null + undefined;
alert(a);
```

a) null
b) undefined
c) NaN
d) В коде ошибка.

12. Что получится, если сложить true + false?
a) "truefalse"
b) 0
c) 1
d) NaN

13. Объявлена функция:

```
function F() {}
```

Верно ли, что F instanceof Function?
a) Да.
b) Нет.

14. Есть ли разница между вызовами i++ и ++i?
a) Разница в значении, которое возвращает такой вызов.
b) Разница в значении i после вызова.
c) Нет никакой разницы.

15. Какие вызовы parseInt возвратят число?
a) parseInt("1px")
b) parseInt("-1.2")
c) parseInt("0 минут")
d) parseInt("$1.2")

Тестовые задания 2 (Т2)

1. Выберите правильные варианты объявления массива, то есть такие, в результате которых мы получаем массив из двух чисел 1 и 2.
a) new Array.prototype.constructor(1, 2)
b) new Array(1, 2)
c) Array(1, 2)
d) [1, 2]
e) 1..2

Все варианты правильные.

2. Чему равно это выражение?

```
[].push(1,2).unshift(3).join()
```

a) 3,1
b) 1,2,3
c) 3,1,2
d) В коде ошибка.

3. Что выведет alert?

```
alert(str); // ?
var str = "Hello";
```

a) Hello
b) undefined
c) Будет ошибка.

4. Что выведет этот код?

```
if (function f(){}) {
  alert(typeof f);
}
```

a) undefined
b) function
c) null
d) object
e) В коде ошибка.

5. Чему равно a + b + c?

```
let a = 1;
let b = { toString() {return '1'} };
let c = 1;
```

a) 11[object Object]
b) 2[object Object]
c) 111
d) 3

6. Какая арифметическая операция приводит к ошибке в javascript?

a) Деление на ноль.
b) Умножение числа на строку.
c) Корень из отрицательного числа.
d) Никакая из вышеперечисленных.

7. Чему равно 2 && 1 && null && 0 && undefined ?

a) 2
b) 1
c) null
d) 0
e) undefined
f) false

8. Существует ли такое значение X, что после присваивания a = X вызов alert(a == X) выдаст false?

```
let a = X;
alert( a == X ); // false
```

a) Да, X – это undefined.
b) Да, X – это null.
c) Да, другое.
d) Нет, не бывает.

9. Что выведет этот код?

```
alert( "1"[0] );
```

a) 0
b) 1
c) 2
d) undefined
e) В коде ошибка.

10. Выберите правильные варианты объявления массива, то есть такие, в результате которых мы получаем массив из двух чисел 1 и 2.

a) new Array.prototype.constructor(1, 2)
b) new Array(1, 2)
c) Array(1, 2)
d) [1, 2]
e) 1..2

f) Все варианты правильные.

11. Чему равно 0 || 1 && 2 || 3 ?

a) 0
b) 1
c) 2
d) 3
e) true
f) false

12. Какое из этих слов не имеет специального использования в JavaScript, никак не упомянуто в стандарте?

a) this
b) instanceof
c) constructor
d) parent
e) new
f) Все имеют специальное использование.

13. Что выведет sayHi при вызове через setTimeout?

```
let name = "Вася";
function sayHi() {
  alert(name);
}

setTimeout(function() {
  let name = "Петя";
  sayHi();
}, 1000);
```

a) Вася.
b) Петя.
c) undefined
d) Будет ошибка.

14. Верно ли сравнение: "ёжик">«яблоко"?

a) Да.
b) Нет.
c) Зависит от локальных настроек браузера.

15. Верно ли, что null == undefined?

a) Да.
b) Нет.

Критерии оценки тестовых заданий, устных опросов:

1. Полнота знаний теоретического контролируемого материала.
2. Количество правильных ответов.

Тестовое задание / опрос считается сданным, если студент правильно ответил на 60 процентов от общего числа вопросов.

Критерии оценивания	Оценка
Студент ответил на 90 % (и более) вопросов	Отлично
Студент ответил на 70-89 % вопросов	Хорошо
Студент ответил на 60-69 % вопросов	Удовлетворительно
Студент ответил менее чем на 59 % вопросов	Неудовлетворительно

Сумма баллов по разделам дисциплины складывается из оценок, полученных обучающимся в течение семестра по всем формам текущего контроля. Каждая форма контроля оценивается баллом в интервале от 0 до 10.

При заочной форме обучения в качестве оценочного средства аттестации раздела используется также контрольная работа.

Контрольная работа – средство проверки умений применять полученные знания для выполнения заданий по теме или разделу. Выполняется по индивидуальному заданию, представленному в методических указаниях для выполнения контрольных работ по дисциплине.

Критерии оценки контрольной работы

Максимальное количество баллов, начисляемое за контрольную работу, составляет 20 баллов по системе ECTS.

Расшифровка уровня знаний, соответствующего баллам ECTS

Оценка (ECTS)	Сумма баллов	Требования к знаниям на устном зачёте
«Зачтено» – A – E	12 - 20	Оценка «Зачтено» выставляется студенту, если он выполнил не менее 60% заданий контрольной работы; верно ответил на вопросы преподавателя
«Не зачтено» – F	менее 12	Оценка «Не зачтено» выставляется студенту, если он выполнил менее 60% заданий контрольной работы; затрудняется с ответами на вопросы преподавателя

Промежуточная аттестация осуществляется в форме зачета. Зачет проводится в форме тестирования. Тест содержит от 20 вопросов. Тест – это форма контроля, направленная на проверку уровня освоения контролируемого теоретического и практического материала по дидактическим единицам дисциплины (терминологический аппарат, основные методы).

Перечень вопросов выходного контроля (зачет):

1. Что может JS в браузере?
2. Что не может JS в браузере?
3. Тег <script>
4. Современная разметка
5. Внешние скрипты
6. Структура кода. Инструкции
7. Структура кода. Точка с запятой
8. Структура кода. Комментарии
9. Переменные
10. Имена переменных
11. Константы
12. Тип данных “число”
13. Тип данных. “BigInt”
14. Тип данных “Строка”
15. Тип данных “Булевый”
16. Значение “null”
17. Значение “undefined”
18. Оператор typeof
19. Операторы alert, prompt, confirm
20. Строковое и логическое преобразования
21. Численное преобразование
22. Термины “унарный”, “бинарный”, “операнд”
23. Математические операции в JS, взятие остатка от деления, возведение в степень
24. Сложение при помощи банирного +
25. Приведение к числу, унарный +
26. Присваивание
27. Оператор “запятая”
28. Условные операторы
29. Условие if
30. Тернарный оператор
31. Switch
32. Оператор for

33. Оператор while и do while
34. Оператор for in
35. Оператор обработки исключения try catch
36. Объявление функции
37. Локальные переменные
38. Внешние переменные
39. Параметры функции
40. Параметры функции по умолчанию
41. Возврат значения функции
42. Выбор имен функций
43. Использование меток в функциях
44. Function Expression
45. Функции-«колбэки»
46. Function Expression в сравнении с Function Declaration
47. Необязательные параметры и значения по умолчанию
48. Массив arguments
49. Неопределенное количество параметров в функциях
50. Функции в качестве параметров
51. Результат функции
52. Возвращение функции из функции
53. Глобальные переменные
54. Определение локальной области видимости
55. Переменные и константы функции
56. Локальные переменные в блоках кода, условиях и циклах
57. Переменная let и константы
58. Скрытие переменных
59. Замыкания
60. Самовызывающиеся функция

Примерный перечень тестовых заданий к зачету:

тестовое задание (Т1).

1. Что выведет выражение ниже?

```
let a = new Array(1,2), b = new Array(3);
alert(a[0] + b[0]);
```

a) 1
b) 4
c) undefined
d) NaN

2. Что выведет этот код?

```
let f = function g() { return 23; };

alert( typeof g() );
```

a) number
b) undefined
c) function
d) ошибка.

3. Что выведет этот код?

```
function User() { }
User.prototype = { admin: false };

let user = new User();
alert(user.admin);
```

a) false
b) undefined
c) true

4. Какие варианты подключения скрипта являются корректными с точки зрения современного стандарта HTML?

a) <script type="text/javascript" src="my.js"></script>
b) <script src="my.js"></script>
c) <script src="my.js"/>
d) <хачу-javascript отсюда="my.js">

5. Чему равно такое выражение?

```
[] + false - null + true
```

a) 0
b) NaN
c) undefined
d) 1

6. Чему равно arr.length?

```
function MyArray() { }
MyArray.prototype = [];

let arr = new MyArray();
arr.push(1, 2, 3);
alert(arr.length);
```

a) 0
b) undefined
c) 3
d) В этом коде допущена ошибка.

7. Что выведет этот код?

```
alert( 20e-1['toString'](2) );
```

a) 2
b) 10
c) 20
d) NaN
e) В коде ошибка.

8. Что выведет этот код?

```
let obj = {
 "0": 1,
 0: 2
};

alert( obj["0"] + obj[0] );
```

a) 2
b) 3
c) 4
d) 12
e) В коде ошибка.

9. Сколько параметров можно передать функции ?

a) Ровно столько, сколько указано в определении функции.
b) Сколько указано в определении функции или меньше.
c) Сколько указано в определении функции или больше.
d) Любое количество.

10. Какое событие из этого списка не существует?

a) onmousescroll
b) onclick
c) onmouseover
d) onmousemove
e) onwheel
f) все существуют.

11. Сколько детей будет у после кода ниже?

```
<div id="div">текст</div>
<script>
  div.insertAdjacentHTML('afterBegin', '<p>Привет</p>');
  div.insertAdjacentHTML('beforeEnd', '<p>Пока</p>');
</script>
```

a) 0
b) 1
c) 2
d) 3
e) В коде ошибка

12. Как получить полную ширину блочного DOM-элемента, если у него есть border и padding?

parseInt(elem.style.width)
parseInt(getComputedStyle(elem).width)
elem.offsetWidth
elem.clientWidth

13. Есть ссылка:

```
<a href="http://site.com">Текст ссылки</a>
```

На документе стоит обработчик:

```
document.onclick = function(e) { e.preventDefault(); }
```

Произойдёт ли при клике на ссылку переход по ней?

a) да.
b) нет.
c) зависит от положения звёзд в момент клика.

14. Как получить размер отступа margin-top у ?

```
<style>
  div { margin: 10px; }
</style>

<div id="div"></div>
```

a) div.style.marginTop
b) div.style["margin-top"]
c) getComputedStyle(div, "margin").top
d) getComputedStyle(div).marginTop
e) getComputedStyle(div).margin.top

15. Что выведет этот код?

```
<p id="elem">Привет!</p>

<script>
  elem.textContent = elem.outerHTML
  alert(elem.innerHTML);
</script>
```

a) Привет!
b) Привет!
c) <p id="elem">Привет!</p>
d) Будет ошибка.

16. Какое значение содержит «внешнюю» высоту элемента, то есть вертикальный размер, который он занимает во внешнем контейнере?

a) offsetHeight
b) outerHeight
c) clientHeight

d) scrollHeight
e) Ни одно из вышеперечисленных.
17. Какие из этих свойств элемента могут указывать на текстовый узел?
a) firstElementChild
b) firstChild
c) parentNode
d) parentElement
e) nextSibling
18. Этот код ничего не выведет. Почему?

```
1 document.onclick = function(event) {
2   alert(event.type);
3 };
4
5 document.body.dispatchEvent(new CustomEvent('click'));
```

a) Из JavaScript нельзя генерировать встроенные события.
b) Обработчик через on-свойство не срабатывает для событий, сгенерированных из JavaScript.
c) Встроенное событие нужно генерировать иначе, конструктор CustomEvent надо заменить на другой.
d) При генерации события не указано, что оно должно всплывать.
e) Боже, что это?
19. Можно ли из JavaScript получить содержимое комментария?
a) Да, комментарий – DOM-узел, можно получить.
b) Нет, комментарии есть в HTML, но отсутствуют в DOM.
c) Можно получить, но только до окончания загрузки страницы.
20. Что будет, если вызвать document.write(str) после загрузки страницы?
a) Строка str допишется в конец документа.
b) Содержимое документа будет полностью заменено на строку str.
c) Будет ошибка.

Тестовое задание (Т2).
1. Каких операторов из этого списка нет в javascript?
a) *
b) ^
c) %
d) #
e) &
f) >>
g) >>>
h) !
2. Что выведет этот код?

```
1 let obj = {
2  "0": 1,
3  0: 2
4 };
5
6 alert( obj["0"] + obj[0] );
```

a) 2
b) 3
c) 4
d) 12
e) В коде ошибка
3. Правда ли что a == b?

```
a = [1, 2, 3];
b = [1, 2, 3];
```

a) Правда.
b) Неправда.
c) Как повезёт.

4. Есть ли различия между проверками:

```
if( x <= 100 ) {...}
// и
if( !(x > 100) ) {...}
```

a) Да, существует значение x, для которого они работают по-разному.
b) Нет, они полностью взаимозаменяемы.
c) Зависит от браузера.

5. Существует ли такое значение X, что после присваивания a = X вызов alert(a == X) выдаст false?

```
let a = X;
alert( a == X ); // false
```

a) Да, X – это undefined.
b) Да, X – это null.
c) Да, другое.
d) Нет, не бывает.

6. Есть ли разница между выражениями?

```
!!( a && b )
(a && b)
```

a) Да.
b) Нет.
c) В первом выражении ошибка, что еще за «!!» ??

7. Какие из этих вариантов задают массив из элементов «a», «b»?

a) let a = new Array("a","b")
b) let a = { "a", "b" }
c) let a = ("a", "b")
d) let a = ["a", "b"]
e) let a = "a,b".split(",")

8. Чемуравноі вконцекода?

```
for(let i=0; i<10; i++) {
  console.log(i);
}
// i = ?
```

a) undefined
b) 9
c) 10
d) Нет такой переменной после цикла.

9. Внимательно посмотрите на синтаксис этого кода. Что он выведет?

```
let f = function(x) {
  alert(x)
}

(function() {
  f(1)
}())
```

a) Никакого результата.
b) Выведет 1.
c) Выведет undefined.
d) Будет ошибка.

10. Дан такой HTML:

```
<div id="outer">
  <div id="inner"></div>
</div>
```

Какие вызовы вернут div#inner?

a) document.querySelector("#outer #inner")
b) document.getElementById("#outer").getElementById("#inner")
c) document.querySelector("#outer").children[0]
d) document.getElementById("#outer").firstChild

11. Можно ли скриптом перевести посетителя на другую страницу сайта?

a) Да, куда угодно.
b) Да, но только в рамках текущего сайта.
c) Нет, нельзя.

12. Какое событие из этого списка не существует?

a) onmousescroll
b) onclick
c) onmouseover
d) onmousemove
e) onwheel
f) все существуют.

13. Есть кнопка <button id="elem"></button>, как показать в ней строку "<hello>"?

Результат должен быть таким:

```
<!--+ run -->
<button id="elem">&lt;hello&gt;</button>
```

a) elem.innerHTML = "<hello>"
b) elem.innerText = "<hello>"
c) elem.textContent = "<hello>"
d) elem.setText("<hello>")

14. При каком условии elem.scrollHeight == elem.clientHeight ?

a) Содержимое элемента полностью прокручено вниз.
b) Страница прокручена так, что элемент полностью видим и находится в границах окна.
c) Высота элемента равна высоте полосы прокрутки.
d) Содержимое элемента полностью видимо, в нём нет прокрутки.
e) У элемента нет padding.

15. Может ли скрипт во время работы страницы подключить к ней другие внешние js-файлы?

a) Да, но только один раз.
b) Да, но только до полной загрузки страницы.
c) Да, сколько угодно файлов когда угодно.

16. В каких случаях для элемента elem верно elem.firstChild == elem.lastChild?

a) Когда у elem нет детей.
b) Когда у elem единственный потомок.
c) Когда у elem два или более потомков.

17. Какое событие не может быть вызвано кликом мыши ?

a) onfocus
b) onclick
c) onkeydown
d) onmousedown

18. Есть элемент <input type=«button» id=«elem»/>. На нём два обработчика:

```
elem.addEventListener("click", handler1, false);
elem.addEventListener("click", handler2, true);
```

Какой из них сработает раньше при клике на elem?

a) первый.
b) второй.
c) стандарт не указывает точную последовательность.

19. Сколько потомков будет у после кода ниже?

```
<div id="div"></div>
<p id="p">Привет</p>
<script>
  div.appendChild(p);
  div.appendChild(p);
</script>
```

a) 0
b) 1
c) 2
d) В коде ошибка.

20. Какие события из этого списка возникнут при нажатии и отпускании клавиши «пробел»? Другие клавиши не нажимаются.

a) keydown
b) keyup
c) keypress
d) никаких не возникнет.

Критерии оценки зачета

Шкалы оценки образовательных достижений

Сумма баллов	Оценка по 4-х балльной шкале	Зачет	Оценка (ECTS)
90-100	отлично	Зачтено	A
85-89	хорошо		B
75-84			C
70-74			D
65-69	удовлетворительно		
60-64			E
Ниже 60	неудовлетворительно	не зачтено	F

Баллы (итоговой рейтинговой оценки)	**Баллы** (зачет)	**Требования к знаниям**
100-90	Зачтено 24 – 40 баллов	теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному
85 - 89		теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному.
75 - 84		теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все

		предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками.
65 - 74		теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки.
60-64		теоретическое содержание курса освоено частично, некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены, либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному
Ниже 60	не зачтено 0-23 баллов	очень слабые знания, недостаточные для понимания курса, имеется большое количество основных ошибок и недочетов

Учебно-методическое и информационное обеспечение учебной дисциплины

Основная литература:

1. Янцев, В. В. JavaScript. Креативное программирование : учебное пособие для вузов / В. В. Янцев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2024. — 232 с. — ISBN 978-5-507-49267-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/383837

2. Янцев, В. В. JavaScript. Готовые программы : учебное пособие для вузов / В. В. Янцев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2024. — 200 с. — ISBN 978-5-507-49740-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/401207

Дополнительная литература:

3. Диков, А. В. Клиентские технологии веб-программирования: JavaScript и DOM : учебное пособие / А. В. Диков. — Санкт-Петербург : Лань, 2020. — 124 с. — ISBN 978-5-8114-4074-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/126934

4. Государев, И. Б. Введение в веб-разработку на языке JavaScript : учебное пособие / И. Б. Государев. — Санкт-Петербург : Лань, 2022. — 144 с. — ISBN 978-5-8114-3539-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/206588

Программное обеспечение и Интернет-ресурсы:

1. Электронно-библиотечная система «Лань» - http://e.lanbook.com.
2. Электронно-библиотечная система «Юрайт» - http://www.biblio-onlain.ru.
3. Российская национальная библиотека - http://www.nlr.ru/
4. Поисковая система - http://www.yandex.ru.
5. Гарант - http://base.garant.ru/.
6. Интернет-Университет Информационных Технологий -http://www.intuit.ru
7. Современный учебник Javascript - https://learn.javascript.ru/

Для проведения практических занятий и выполнения самостоятельной работы используются учебные компьютерные классы с выходом в Интернет и лицензионным программным обеспечением.

Материально-техническое обеспечение учебной дисциплины

Занятия проводятся в учебной аудитории, предназначенной для проведения занятий лекционного и практического типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Аудитория оснащена необходимым оборудованием. Лабораторные занятия проводятся в специализированных лабораториях, оснащенных необходимым оборудованием.

В распоряжении обучающихся студентов имеются электронно-библиотечные системы.

Учебно-методические рекомендации для студентов

1. Указания для прослушивания лекций

Перед началом занятий внимательно ознакомиться с учебным планом проведения лекций и списком рекомендованной литературы.

Перед посещением очередной лекции освежить в памяти основные концепции пройденного ранее материала. Подготовить при необходимости вопросы преподавателю. Не надо опасаться, что вопросы могут быть простыми.

На лекции основное внимание следует уделять не формулам и математическим выкладкам, а содержанию изучаемых вопросов, определениям и постановкам задач.

В процессе изучения лекционного курса необходимо по возможности часто возвращаться к основным понятиям и методам решения задач (здесь возможен выборочный контроль знаний студентов).

Желательно использовать конспекты лекций, в которых используется принятая преподавателем система обозначений.

Для более подробного изучения курса следует работать с рекомендованными литературными источниками и вновь появляющимися источниками.

2. Указания для участия в практических занятиях

Перед посещением уяснить тему практического занятия и самостоятельно изучить теоретические вопросы.

В конце практического занятия при необходимости выяснить у преподавателя неясные вопросы.

Основные результаты выполнения работы необходимо распечатать.

3. Указания для выполнения самостоятельной работы

Получить у преподавателя задание и список рекомендованной литературы. Изучение теоретических вопросов следует проводить по возможности самостоятельно, но при затруднениях обращаться к преподавателю.

Подготовить письменный отчет о проделанной работе.

При выполнении заданий по усмотрению преподавателя работа может быть оценена без письменного отчета на основе ответов на контрольные вопросы, при условии активной самостоятельной работы.

Подготовить ответы на контрольные вопросы.

Методические рекомендации для преподавателей

1. Указания для проведения лекций

На первой вводной лекции сделать общий обзор содержания курса и отметить новые методы и подходы к решению задач, рассматриваемых в курсе, довести до студентов требования кафедры, ответить на вопросы.

При подготовке к лекционным занятиям необходимо продумать план его проведения, содержание вступительной, основной и заключительной части лекции, ознакомиться с новинками учебной и методической литературы, публикациями периодической печати по теме лекционного занятия. Уточнить план проведения семинарского занятия по теме лекции. Перед изложением текущего лекционного материала напомнить об основных итогах, достигнутых на предыдущих лекциях. С этой целью задать несколько вопросов аудитории и осуществить выборочный контроль знания студентов.

В ходе лекционного занятия преподаватель должен назвать тему, учебные вопросы, ознакомить студентов с перечнем основной и дополнительной литературы по теме занятия. Раскрывая содержание учебных вопросов, акцентировать внимание студентов на основных категориях, явлениях и процессах, особенностях их протекания. Раскрывать сущность и содержание различных точек зрения и научных подходов к объяснению тех или иных явлений и процессов.

Следует аргументировано обосновать собственную позицию по спорным теоретическим вопросам. Приводить примеры. Задавать по ходу изложения лекционного материала риторические вопросы и самому давать на них ответ. Это способствует активизации мыслительной деятельности студентов, повышению их внимания и интереса к материалу лекции, ее содержанию. Преподаватель должен руководить работой студентов по конспектированию лекционного материала, под-

черкивать необходимость отражения в конспектах основных положений изучаемой темы, особо выделяя, категориальный аппарат. В заключительной части лекции необходимо сформулировать общие выводы по теме, раскрывающие содержание всех вопросов, поставленных в лекции. Объявить план очередного практического занятия, дать краткие рекомендации по подготовке студентов к семинару. Определить место и время консультации студентам, пожелавшим выступить на занятии с докладами.

На последней лекции уделить время для обзора наиболее важных положений, рассмотренных в курсе.

2. Указания для проведения практических занятий

Четко обозначить тему занятия.

Обсудить основные понятия, связанные с темой.

В процессе решения задач вести дискуссию со студентами о правильности применения теоретических знаний.

Отмечать студентов, наиболее активно участвующих в решении задач и дискуссиях.

В конце занятия задать аудитории несколько контрольных вопросов.

3.Указания по контролю самостоятельной работы студентов

По усмотрению преподавателя задание на самостоятельную работу может быть индивидуальным или фронтальным.

При использовании индивидуальных заданий требовать от студента письменный отчет о проделанной работе.

При применении фронтальных заданий вести коллективные обсуждения со студентами основных теоретических положений.

С целью контроля качества выполнения самостоятельной работы требовать индивидуальные отчеты (допустимо вместо письменного отчета применять индивидуальные контрольные вопросы).

Программа составлена в соответствии с требованиями ОС НИЯУ МИФИ по направлению 09.03.02 Информационные системы и технологии

Рабочую программу составил ст.пр. А.Г. Мотков

Рецензент: доцент О.А. Ларионова

Программа одобрена на заседании УМКН «Информационные системы и технологии».

Председатель учебно-методической комиссии О.В. Виштак