

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 «Интерактивное программирование web-приложений»

Направления подготовки/специальность
«09.03.02 Информационные системы и технологии»

Основная профессиональная образовательная программа
«Информационные системы и технологии»

Квалификация выпускника
Бакалавр

Форма обучения
Заочная

Цель освоения дисциплины

Целями освоения дисциплины «Интерактивное программирование web-приложений» в области обучения, воспитания, развития, соотносённые 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 обобщённые трудовые функции: Разработка архитектуры ИС, Проектирование и дизайн ИС, Сбор информации для инициации проекта в соответствии с полученным заданием

Компетенции обучающегося, формируемые в результате освоения дисциплины

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 дея-

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

Структура и содержание учебной дисциплины

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

Календарный план

№ Р а з д е л а	№ Т е м ы	Наименование раздела (темы) дисциплины	Виды учебной деятельности (в часах)					Атте-ста-ция раздела (форма)	Мак-си-мальный 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	Продвинутая работа с JavaScript							КИ	25
	4	Рекурсия и стек. Остаточные параметры и оператор расширения. Свойства объекта, их конфигурация. Прототипы, наследование.	30/3	1/1	-	4/2	25		
	5	Классы, примеси, обработка ошибок, промины, генераторы, модули, динамические импорты	26/1	1/1	-	2	23		
Вид промежуточной аттестации								3	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
3	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 3. Переменные в JavaScript 4. Типы данных переменных.	2	1-4

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. Комментарии в JavaScript. 5. Цикл for. Цикл for...in. Цикл while. Цикл do...while. 6. Бесконечный цикл. Вложенный цикл. Управление циклом.	2	1-4
Лекция 3. Рекурсия и стек. Остаточные параметры и оператор расширения. Свойства объекта, их конфигурация. Прототипы, наследование. 1. Рекурсия и стек. 2. Остаточные параметры и оператор расширения. 3. Замыкание. 4. Устаревшее слово «var». 5. Глобальный объект. 6. Объект функции. 7. Синтаксис «new Function». 8. Планирование: setTimeout и setInterval 9. Флаги и дескрипторы свойств 10. Свойства - геттеры и сеттеры 11. Прототипом наследование 12. Встроенные прототипы 13. Методы прототипов	1	1-4
Лекция 4. Классы, примеси, обработка ошибок, промисы, генераторы, модули, динамические импорты. 1. Класс: базовый синтаксис 2. Наследование классов 3. Статические свойства и методы 4. Приватные и защищенные методы и свойства 5. Расширение встроенных классов 6. Проверка класса 7. Примеси 8. Обработка ошибок 9. Пользовательские ошибки, расширение Error 10. Колбэки 11. Промисы, цепочка промисов 12. Промисы: обработка ошибок 13. Промисификация 14. Микрозадачи 15. Модули, экспорт и импорт 16. Генераторы 17. Асинхронные итераторы и генераторы	1	1-4

Перечень практических занятий

Тема практического занятия. Вопросы, отрабатываемые на практическом занятии	Всего часов	Учебно-методическое обеспечение
Основы взаимодействия с web-документами в JavaScript	2	1-4
Взаимодействие с функциями и свойствами в JavaScript	2	1-4
Работа с объектами в JavaScript	4	1-4
Работа с классами и модулями в JavaScript	2	1-4

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 - "usestrict". Переменные. Типы данных. Взаимодействие alert, prompt, confirm. Базовые операторы, математика, операторы сравнения. Условное ветвление if, '?. Логические операторы, оператор объединения с null '??'.	22	1-4
Конструкция "switch". Функции, Function Expression. Функции-стрелки. Отладка в браузере. Автоматическое тестирование с использованием фреймворка Mocha. Полифилы. Объекты, копирование объектов 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.	22	1-4
Создание, вставка, изменение элементов на странице. Атрибуты, свойства, управление классами. Простые задачи по генерации компонентов интерфейса. DOM и инструменты разработчика в браузере. Динамическая генерация элементов интерфейса. Установка обработчиков событий. Свойства событий. Погружение и всплытие. Приём проектирования "делегирование" для работы с большим количеством элементов, упрощения сложных интерфейсов. Приём проектирования "поведение" для добавления функционала при помощи HTML-атрибутов. Практика: меню, карусель изображений.	25	1-4
Классы, ООП в JavaScript. Модули в JavaScript. Drag'n'drop при помощи событий, компонент "слайдер" для проекта. Обработка пользовательского ввода, события клавиатуры. Компоненты "модальное окно", "ленточное меню». Создание и отправка форм, динамические формы, их валидация. DOM-свойства и методы для форм. Чтение данных из формы. Метод fetch и, основы Promise для сетевых запросов. Обмен данными с сервером в формате JSON. Событийный цикл браузера. Объекты Promise в деталях, их методы, цепочки Promise. Промисификация функций. Синтаксис async/await.	23	1-4

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-педагогическую работу.

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 вы-

полнения 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	3-УК-2, У-УК-2, В-УК-2, 3-ПК-6.1, У-ПК-6.1, В-ПК-6.1, 3-ПК-12, У-ПК-12, В-ПК-12	Контроль по итогам в форме тестирования (письменно)
3	Реализация сценариев jQuery	3-УК-2, У-УК-2, В-УК-2, 3-ПК-6.1, У-ПК-6.1, В-ПК-6.1, 3-ПК-12, У-ПК-12, В-ПК-12	Контроль по итогам в форме тестирования (письменно)
Промежуточная аттестация			
4	Зачет	3-УК-2, У-УК-2, В-УК-2, 3-ПК-6.1, У-ПК-6.1, В-ПК-6.1, 3-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. Собираемое 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. Что такое куки (cookie)?
2. Зачем нужны куки?
3. Какими возможностями обладают куки?
4. Как удалить куки?
5. В чем разница между куки, локальным и сессионным хранилищами?
6. В чем главное отличие между локальным и сессионным хранилищами?
7. Как получить доступ к веб-хранилищу?
8. Какие методы предоставляет сессионное хранилище?
9. Какое событие возникает при работе с веб-хранилищем?
10. Для чего используется веб-хранилище?
11. Как определить поддержку веб-хранилища браузером?
12. Как определить поддержку сервис-воркеров браузером?
13. Приведите пример веб-воркера
14. Назовите ограничения веб-воркеров по работе с DOM
15. Что такое промис (обещание, promise)?
16. Зачем нужны промисы?
17. Назовите три возможных состояния промиса
18. Что такое функция обратного вызова (колбек)?
19. Зачем нужны колбеки?
20. Что такое ад колбеков?
21. Что такое события, отправленные сервером (server-sent events, SSE)?
22. Как получать сообщения (уведомления или события), отправленные сервером?
23. Как проверить поддержку SSE браузером?
24. Какие события возникают при работе с SSE?
25. Назовите основные правила работы с промисами
26. Что такое колбек в колбеке?
27. Что такое цепочка из промисов?
28. Что такое Promise.all()?
29. Что такое Promise.race()?
30. Что такое строгий режим?

31. Зачем нужен строгий режим?
32. Как включить строгий режим?
33. Для чего используется двойное отрицание?
34. Для чего используется оператор «delete»?
35. Для чего используется оператор «typeof»?
36. Что такое undefined?
37. Что такое null?
38. В чем разница между null и undefined?
39. Что такое eval?
40. В чем разница между window и document?
41. Как получить доступ к истории браузера?
42. Какие типы данных существуют в JavaScript?
43. Что делает isNaN()?
44. В чем разница между необъявленными и неопределенными переменными?
45. Что такое глобальные переменные?
46. Какие проблемы влечет за собой создание глобальных переменных?
47. Что такое NaN?
48. Что делает isFinite()?
49. Что такое поток событий (event flow)?
50. Что такое всплытие события?
51. Что такое погружение или захват события?
52. Как отправить форму на обработку?
53. Как получить информацию об операционной системе?
54. В чем разница между событиями «DOMContentLoaded» и «load»?
55. В чем разница между нативными, хостовыми (принадлежащими среде выполнения кода) и пользовательскими объектами?
56. Какие средства используются для отладки кода?
57. В чем заключаются преимущества и недостатки промисов по сравнению с колбеками?
58. В чем разница между атрибутом и свойством элемента?
59. Что такое политика общего происхождения (same-origin policy)?
60. Что делает void 0?
61. JavaScript — это компилируемый или интерпретируемый язык программирования?
62. Чувствителен ли JavaScript к регистру?
63. Связаны ли Java и JavaScript?
64. Что такое событие (event)?

Перечень практических заданий

1. Написать функцию, которая принимает массив сотрудников, каждый сотрудник имеет имя и возраст ([{name: 'Иван', age: 23},...]) и возвращает массив, где каждый элемент представляет из себя строку "Имя: Иван, возраст: 23».

2. У нас есть объект, в котором хранятся зарплаты нашей команды:

```
let salaries = {
  John: 100,
  Ann: 160,
  Pete: 130
}
```

Напишите код для суммирования всех зарплат и сохраните результат в переменной sum. Должно получиться 390.

Если объект salaries пуст, то результат должен быть 0.

3. Создайте функцию multiplyNumeric(obj), которая умножает все числовые свойства объекта obj на 2.

Например:

```
// до вызова функции
let menu = {
  width: 200,
  height: 300,
```

```

    title: "My menu"
  };
  multiplyNumeric(menu);
  // после вызова функции
  menu = {
    width: 400,
    height: 600,
    title: "My menu"
  };

```

Обратите внимание, что `multiplyNumeric` не нужно ничего возвращать. Следует напрямую изменять объект.

P.S. Используйте `typeof` для проверки, что значение свойства числовое.

4. Написать объект `ladder` - объект, который позволяет подниматься вверх и спускаться.

Пример работы должен быть таким:

```

ladder.showStep(); // 0 (выводит ступеньку на которой мы находимся)
ladder.up();
ladder.up();
ladder.showStep(); // 2
ladder.down();
ladder.showStep(); // 1

```

5. Вывести кнопку с текстом "Привет", при нажатии на неё выводим `alert` с текстом "Привет Мир!"

6. Выводим кнопку с текстом "Заполнить" и незаполненный инпут, при клике на кнопку, заполняем инпут текстом "test@email.ru"

7. Выводим кнопку и инпут, в инпуте можно ввести любой текст и при нажатии на кнопку, выводится алерт с текстом "Вы ввели 'текст инпута'" или "Вы ничего не ввели в поле".

8. Выводим кнопку с текстом "Поменять" и два инпута, при клике на кнопку инпуты меняются своим введенным текстом

9. Выводим две кнопки "заблокировать" и "разблокировать" и инпут. Одна из них блокирует инпут с помощью атрибута `disabled`, а другая разблокирует

10. Вывести любой квадрат и кнопку "скрыть квадрат". Когда мы нажимаем на скрыть, квадрат исчезает и текст кнопки меняется на "показать квадрат" и так можно кликать сколько угодно раз.

11. Выводим красный квадрат, при наведении на него он становится зеленым, а когда увидим курсор от него, обратно красным.

12. Вывести 4 красных квадрата, при клике на любой, он становится зеленым, при этом если есть уже зеленый квадрат, то он становится обратно красным и так можно кликать на любой квадрат, он становится зеленым, а старый зеленый квадрат обратно красным и тд. (Сделать задачу так, чтобы можно было добавить ещё хоть 100 квадратов при этом скрипт не надо менять).

13. Реализовать калькулятор. Выводим 10 кнопок цифр от 0 до 10. Кнопки: умножить, поделить, сложить, вычесть, вычислить. При нажатии на кнопки в любом порядке выводиться в отдельном блоке строка (5-4+3*2...) и при нажатии на кнопку посчитать, заменяем в блоке данную строку на результат её вычисления, при этом можно потом дальше вычислять с уже этим вычислением.

14. Проверка URL на валидность. Передаем url функции и проверяем валидное оно или нет (возвращаем `true` или `false`). Критерии: Урл должен начинаться с `http://` и или `https://` и заканчиваться на `.php` или `.html`. Например - `http://site.ru/index.php` - валидный, `http://site.com` - не валидный, `site.ru/index.php` - не валидный

15. Проверка номера телефона на валидность. Передаем телефон функции и проверяем валидное оно или нет (возвращаем `true` или `false`). Такие номера должны быть валидными: 89991112233, 8 (999) 1112233, +7 (999) 111-22-33, +7 (999) 111 22 33. Т.е. телефон может начинаться как и с +7, так и с 8. код оператора, может быть в скобках и без, и с пробелами. Оставшиеся часть может быть с дефисами и пробелами

Аттестация раздела по дисциплине проводится как контроль по итогам в форме тестирования. Тест содержит от 15 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) False

3. Чему равна сумма `[] + 1 + 2`?

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

4. Что делает код:

```
1 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

```
1 let a = [1,2]
2
3 (function() { alert(a) })()
```

подвох.

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

7. Есть ли различия между проверками:

```
1 if( x <= 100 ) {...}
2 // и
3 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?

```

1 <p id="elem">Привет, мир!</p>
2
3 <script>
4   elem.outerHTML = "Прощай, жестокий мир!"
5   alert(elem.outerHTML);
6 </script>

```

- a) Выведет undefined, теперь elem – текстовый узел.
 - b) Прощай, жестокий мир!
 - c) <p id="elem">Привет, мир!</p>
 - d) <p id="elem">Прощай, жестокиймир!</p>
 - e) Будет ошибка.
9. 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?

10. Результат elem.getBoundingClientRect() содержит отрицательное значение bottom: -10.

- a) Что это означает?
- b) Что низ элемента вылезает за нижнюю границу документа.
- c) Что низ элемента вылезает за нижнюю границу окна.
- d) Что низ элемента вылезает за верхнюю границу документа.
- e) Что низ элемента вылезает за верхнюю границу окна.
- f) Что содержимое элемента прокручено на 10px.
- g) Такого не может быть.

11. 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.

12. Как получить атрибут test у элемента <div id="elem" test="5">?

- a) elem.test
- b) elem.getAttribute("test")
- c) elem.attributes.test.value
- d) elem.attributes[1].value
- e) elem.dataset.test

13. Вызов setTimeout(func, 0) вызовет func...

- a) Сразу после setTimeout, до следующей строки кода.
- b) Сразу после текущего скрипта, когда браузер сможет выполнить JavaScript.
- c) Ровно через 4 миллисекунды.
- d) Через 4 миллисекунды или больше, в зависимости от пожеланий браузера.

14. Как правильно вывести приветствие через 5 секунд?

- a) sleep(5); alert("Привет!");
- b) sleep(5000); alert("Привет!");
- c) setTimeout(function() {alert("Привет!")}, 5000);
- d) setTimeout(function() {alert("Привет!")}, 5);

15. Сколько потомков будет у <div> после кода ниже?

```
1 <div id="div"></div>
2 <p id="p">Привет</p>
3 <script>
4   div.appendChild(p);
5   p.appendChild(div);
6 </script>
```

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

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?

```
1 [].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?

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

- a) Hello
- b) undefined
- c) Будет ошибка.

4. Что выведет этот код?

```
1 if (function f(){}) {
2   alert(typeof f);
3 }
```

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

5. Чему равно $a + b + c$?

```
1 let a = 1;
2 let b = { toString() {return '1'} };
3 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. Можно ли из JavaScript получить содержимое комментария?

- a) Да, комментарий – DOM-узел, можно получить всегда.
- b) Нет, комментарии есть в HTML, но отсутствуют в DOM.
- c) Можно получить, но только до окончания загрузки страницы.

9. Вызов `setInterval(func, 100)` вызовет `func`...

- a) Ровно каждые 100мс.
- b) Примерно каждые 100мс.
- c) Обычно каждые 100мс, но возможно, что вызовы будут происходить гораздо реже.

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

`offsetHeight`

`outerHeight`

`clientHeight`

`scrollHeight`

Ни одно из вышеперечисленных.

11. Как получить размер отступа `margin-top` у `<div>`?

```
1 <style>
2   div { margin: 10px; }
3 </style>
4
5 <div id="div"></div>
```

e) 0

f) 1

g) 2

h) В коде ошибка.

12. Какое событие не может быть вызвано кликом мыши ?

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

13. Какие из этих обработчиков сработают?

```
1 div.onclick = function() { alert(1) };
2 div.onclick = function() { alert(2) };
3 div.addEventListener('click', function() { alert(3) });
```

a) Первый.

b) Второй.

c) Третий.

14. Есть два вложенных элемента:

```
1 <a id="outer" href="http://site.com">
2   <em id="inner">Текст ссылки</em>
3 </a>
```

На внутреннем стоит обработчик:

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

Произойдёт ли при клике на #inner переход по ссылке?

- a) да.
- b) нет.
- c) будет ошибка

15. Есть такой элемент:

```
1 <input id="input" value="Привет">
```

Какие вызовы поменяют значение в нём?

- a) input.value = "Пока"
- b) input.setValue("Пока")
- c) input.setAttribute("value", "Пока");
- d) input.dataset.value = "Пока";
- e) input["Пока"] = true;
- f) Все вышеперечисленные.

Критерии оценки тестовых заданий:

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

Тестовое задание 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.

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 бинарного +
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. Самовывзывающиеся функция

Примерный перечень тестовых заданий к зачету:

тестовое задание (Т1).

1. Что выведет выражение ниже?

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

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

2. Что выведет этот код?

```
1 let f = function g() { return 23; };  
2  
3 alert( typeof g() );
```

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

3. Что выведет этот код?

```
1 function User() { }  
2 User.prototype = { admin: false };  
3  
4 let user = new User();  
5 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?

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

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

6. Чему равно arr.length?

```
1 function MyArray() { }  
2 MyArray.prototype = [];  
3  
4 let arr = new MyArray();  
5 arr.push(1, 2, 3);  
6 alert(arr.length);
```

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

7. Что выведет этот код?

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

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

8. 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.

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) onmousecroll
- b) onclick
- c) onmouseover
- d) onmousemove
- e) onwheel
- f) все существуют.

11. Сколько детей будет у <div> после кода ниже?

```

1 <div id="div">текст</div>
2 <script>
3   div.insertAdjacentHTML('afterBegin', '<p>Привет</p>');
4   div.insertAdjacentHTML('beforeEnd', '<p>Пока</p>');
5 </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:

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

На документе стоит обработчик:

```
1 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 <div>?

```

1 <style>
2   div { margin: 10px; }
3 </style>
4
5 <div id="div"></div>

```

- div.style.marginTop
- div.style["margin-top"]
- getComputedStyle(div, "margin").top
- getComputedStyle(div).marginTop
- getComputedStyle(div).margin.top

15. Что выведет этот код?

```

1 <p id="elem">Привет!</p>
2
3 <script>
4   elem.textContent = elem.outerHTML
5   alert(elem.innerHTML);
6 </script>

```

- Привет!
- <p id="elem">Привет!</p>
- <p id="elem">Привет!</p>
- Будет ошибка.

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

- offsetHeight
- outerHeight
- clientHeight
- scrollHeight
- Ни одно из вышеперечисленных.

17. Какие из этих свойств элемента могут указывать на текстовый узел?

- firstElementChild
- firstChild
- parentNode
- parentElement
- 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'));

```

- Из JavaScript нельзя генерировать встроенные события.
- Обработчик через on-свойство не срабатывает для событий, сгенерированных из JavaScript.
- Встроенное событие нужно генерировать иначе, конструктор CustomEvent надо заменить на другой.
- При генерации события не указано, что оно должно всплывать.
- Боже, что это?

19. Можно ли из JavaScript получить содержимое комментария?

- Да, комментарий – DOM-узел, можно получить.
- Нет, комментарии есть в HTML, но отсутствуют в DOM.
- Можно получить, но только до окончания загрузки страницы.

20. Что будет, если вызвать document.write(str) после загрузки страницы?

- Строка str допишется в конец документа.
- Содержимое документа будет полностью заменено на строку str.

с) 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?

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

- a) Правда.
 - b) Неправда.
 - c) Как повезёт.
4. Есть ли различия между проверками:

```
1 if( x <= 100 ) {...}
2 // и
3 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) выведет false?

```
1 let a = X;
2 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?

```
1 !( a && b )
2 ( a && b )
```

- a) Да.
- b) Нет.

с) 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. Чему равной в конце кода?

```
1 for(let i=0; i<10; i++) {
2   console.log(i);
3 }
4 // i = ?
```

a) undefined

b) 9

c) 10

d) Нет такой переменной после цикла.

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

```
1 let f = function(x) {
2   alert(x)
3 }
4
5 (function() {
6   f(1)
7 })()
```

a) Никакого результата.

b) Выведет 1.

c) Выведет undefined.

d) Будет ошибка.

10. Дан такой HTML:

```
1 <div id="outer">
2   <div id="inner"></div>
3 </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:

```
1 <!--+ run -->
2 <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:

```
1 elem.addEventListener("click", handler1, false);
2 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 <div> после кода ниже?

```
1 <div id="div"></div>
2 <p id="p">Привет</p>
3 <script>
4   div.appendChild(p);
5   div.appendChild(p);
6 </script>
```

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

20. Какие события из этого списка возникнут при нажатии и отпуске 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

Лекционные занятия проводятся в аудиториях: учебная мебель, учебная доска, комплект мультимедийного оборудования

Практические занятия проводятся в компьютерных классах: учебная мебель, учебная доска, комплект мультимедийного оборудования, персональные компьютеры.

Для самостоятельной работы обучающихся имеется: читальный зал с выходом в сеть Интернет: Учебная мебель, комплект мультимедийного оборудования, персональные компьютеры, МФУ.

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, подчеркивать 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 М.А. Несытых

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

Председатель учебно-методической комиссии О.В. Виштак