Список лабораторных работ (ИТиРОД)

№ п/п	Тема работы	К-во занятий	Баллы
1	Элементы сетевого программирования	2	4
2	Создание простой HTML-страницы	1	4
3	Формы, изображения, внедрённые объекты	1	4
4	Использование CSS	1	4
5	Возможности CSS3	1	4
6	Базовый синтаксис JavaScript.	1	4
7	Работа с функциями в JavaScript.	1	4
8	Массивы в JavaScript.	1	4
9	ООП в JavaScript	2	6
10	Элементы ECMAScript 2015	1	4
11	Манипуляция DOM	1	4
12	Библиотека jQuery	2	4

Работа №1. Элементы сетевого программирования

Цели работы.

1. Изучить классы и библиотеки, используемые в популярных платформах для работы с сетью и сетевыми протоколами.

Постановка задачи.

Создать приложение для чата по локальной сети, используя протокол UDP. Приложение обеспечивает публикацию сообщений в общий чат. Приложение может иметь консольный или оконный интерфейс.

Платформа разработки – на выбор студента.

Работа №2. Создание простой HTML-страницы

Цели работы.

1. Изучить базовый синтаксис HTML и структуру HTML-документа.
2. Изучить элементы для выделения текста, ссылки и якоря.
3. Изучить списковые и табличные элементы.

Постановка задачи.

Компания ISsoft (для примера) собирается проводить для студентов университета тренинг по JavaScript. Необходимо создать HTML-страницу с объявлением о тренинге.

Страница должна содержать заголовок, общее описание тренинга (несколько абзацев), листинг маленькой программы на JavaScript, список тем, рассматриваемых на тренинге (с расшифровкой каждой темы), расписание занятий (список дней).

Обеспечить выравнивание элементов на странице при помощи таблиц. Уделить внимание шрифтовому оформлению страницы.

Работа №3. Формы, изображения, внедрённые объекты

Постановка задачи.

Компания ISsoft (для примера) предоставляет возможность студентам старших курсов профильных вузов пройти практику в компании. Необходимо создать страницу для ввода анкетных данных претендента на прохождение практики.

Структура страницы следующая:

1. Вверху располагается изображение-баннер с логотипом компании.
2. Далее следует область с видеороликом. Его можно проиграть, чтобы получить более полное представление о компании, практике, анкете.
3. Далее расположена собственно анкета. Претендент вводит контактную информацию, указывает свой вуз, специальность, обозначает вид практики (преддипломная, производственная). Также указываются даты начала и окончания практики, желаемое время присутствия в офисе компании.
4. Присутствует возможность указать несколько скилов (фиксированное количество, например 3, из фиксированного набора) и выбрать степень владения каждым скилом.
5. Можно выбрать JPEG-изображение (фотографию), чтобы загрузить на сервер вместе с анкетными данными и изготовить пропуск.
6. Анкета завершается стандартным «подвалом» (копирайт).

При верстке страницы для улучшения внешнего вида и упорядочивания контента использовать таблицы. Элементы управления можно использовать любые (разнообразие приветствуется). В случае применения новых элементов и атрибутов из HTML5 обязательно протестировать внешний вид и работу страницы в различных браузерах. После этого сделать самостоятельные выводы о возможности использования того или иного элемента в реальных проектах.

Работа №4. Использование CSS

Постановка задачи.

Основой работы являются страницы, выполненные в работах №2 и №3.

Необходимо настроить их внешний вид при помощи CSS. Вёрстку страниц также постараться выполнить при помощи CSS, а не таблиц.

Работа №5. Возможности CSS3

Постановка задачи.

Реализовать средствами CSS трансформацию элементов страницы и эффекты переходов. Реализовать простейшую анимацию.

Работа №6. Базовый синтаксис JavaScript.

Рассматриваемые темы

- идентификаторы, ключевые и зарезервированные слова
- тип данных в JavaScript, литералы
- выражения
- инструкции

Общие замечания

Каждая задача оформляется в виде отдельного скрипта. Для запуска и тестирования задачи создайте страницу, подключающую скрипт с задачей. Для ввода данных используйте функцию prompt(), для вывода – функцию alert().

Постановка задачи 1

В бесконечном цикле делается запрос на ввод двух чисел (два отдельных prompt). Числа сравниваются и выводится одна из трёх фраз: «числа равны», «первое число меньше», «второе число меньше». Если пользователь ввёл не число, выводится фраза «первый ввод – не число» (или «второй ввод – не число»), и выполнение скрипта прекращается.

Постановка задачи 2

Многоквартирный дом характеризуется следующими тремя показателями: этажность, число подъездов, количество квартир на этаже. Скрипт запрашивает эти показатели и номер квартиры. Выводится номер подъезда, в котором находится указанная квартира. При вводе некорректных данных предусмотреть генерацию исключительных ситуаций.

Постановка задачи 3

Вычисление и вывод i-го числа Фибоначчи, где параметр i вводится пользователем. При вводе некорректных данных предусмотреть генерацию исключительной ситуации.

Постановка задачи 4

1 января 2015 года – это был четверг. Скрипт запрашивает номер месяца (1..12) и число в этом месяце (1..31). Выведите имя дня недели.

Работа №7. Работа с функциями в JavaScript.

Рассматриваемые темы

- функции
- замыкания
- глобальный объект и его элементы

Постановка задачи 1

На декартовой плоскости прямоугольник задаётся четырьмя точками – своими вершинами (у каждой точки две числовые координаты). Вершины перечисляются последовательно, в порядке обхода по часовой стрелке, начиная с произвольной вершины.

1. Написать функцию, проверяющую, образуют ли заданные восемь чисел вершины некоего прямоугольника.
2. Написать функцию, проверяющую принадлежность указанной точки заданному прямоугольнику.

Постановка задачи 2

Декоратор функции получает на вход существующую функцию и возвращает новую функцию, изменяющую (дополняющую) поведение исходной.

1. Создать декоратор для функции с одним параметром, дополняющий переданную функцию проверкой аргумента на тип number.
2. Создать декоратор для функции с произвольным количеством параметров, дополняющий переданную функцию проверкой всех аргумента на указанный тип.

Работа №8. Массивы в JavaScript.

Рассматриваемые темы

- создание массивов
- доступ к элементам и индексы
- методы массивов
- объекты, подобные массивами

Постановка задачи 1

Напишите функцию range(), принимающую два аргумента, начало и конец диапазона, и возвращающую массив, который содержит все числа из диапазона, включая начальное и конечное. Третий необязательный аргумент функции range() – шаг для построения массива. Убедитесь, что функция range() работает с отрицательным шагом: например, вызов range(5, 2, -1) возвращает [5, 4, 3, 2].

Постановка задачи 2

Самостоятельно реализуйте любой алгоритм сортировки массива. Желательно, чтобы ваша функция сортировки могла принимать компаратор для сравнения элементов (как это делает стандартный метод sort()).

Постановка задачи 3

Создать функцию createMatrix(), принимающую количество строк и количество столбцов матрицы и возвращающее матрицу (массив массивов), заполненную случайными числами в диапазоне от 0 до 100. Написать функцию, выполняющую суммирование двух таких «матриц».

Работа №9. ООП в JavaScript.

Рассматриваемые темы

- создание объектов
- работа со свойствами
- атрибуты свойств и объектов
- прототипы
- конструкторы
- моделирование элементов классического ООП

Постановка задачи 1

Создайте класс Vector для представления вектора в трехмерном пространстве (свойства для координат x, y, z). Добавьте в прототип Vector два метода plus() и scalar() для вычисления суммы двух векторов и скалярного произведения двух векторов. Добавьте в прототип свойство только для чтения length, подсчитывающее длину вектора. Переопределите в классе Vector методы toString() и valueOf(). Протестируйте созданный класс.

Постановка задачи 2

Задача имеет название, описание, дату начала и дату окончания. Любая задача может иметь набор дочерних подзадач. Создайте класс для представления задачи.

Выполняемая задача – наследник задачи с дополнительным свойствами: процент выполнения (число) и флагом задача завершена. Реализуйте данное наследование.

Работа №10. Элементы ECMAScript 2015.

Постановка задачи 1

Модифицируйте код скриптов, созданных при решении задач из работ №№7-9, используя новые синтаксические элементы ECMAScript 2015.

Постановка задачи 2

Реализуйте на ECMAScript 2015 алгоритм сжатия информации по Хаффману (https://www.wikiwand.com/ru/%D0%9A%D0%BE%D0%B4_%D0%A5%D0%B0%D1%84%D1%84%D0%BC%D0%B0%D0%BD%D0%B0).

Работа №11. Манипуляция DOM.

Постановка задачи

Необходимо создать скрипты, демонстрирующее изменение набора DOM при наступлении определённого события (или событий). Пример: добавление текстового поля в форму при нажатии кнопки Add.

Работа №12. Библиотека jQuery.

Постановка задачи

Переписать лабораторную работу №11 с использованием библиотеки jQuery.

Выполнить AJAX-запрос с использованием jQuery.