

Работа №2 «Объектная модель документа. Формы и интерфейс пользователя. Формы и алгоритмы.

Понятие спецификации программного продукта на примере HTML5 API.  
Составление алгоритмов с учётом спецификаций ES-6 (ES-2015)»

Цель работы: освоить принципы создания интерфейса пользователя для выполнения алгоритмов различного типа.

Задачи:

получить представление о соответствии элементов управления HTML-формы объектной модели документа;

освоить способы доступа из кода на языке *Javascript* к объектам DOM и элементам управления формы;

научиться манипулировать элементами управления форм для проверки правильности ввода и динамического ввода-вывода информации на стороне клиента;

ознакомиться с принципами обработки данных формы на стороне сервера;

освоить постановку задачи по разработке динамических документов и форм.

Время выполнения: 2 часа.

Работа состоит из нескольких заданий и ответов на контрольные вопросы. В результате выполнения заданий необходимо разработать веб-документ с формой для ввода среднемесячных значений температуры воды за некоторый текущий год в некоторой точке моря. В форме должны быть заданы неизменяемые значения климатических среднемесячных значений в точке. После ввода очередного текущего значения должно вычисляться отклонение от нормы.

Основное внимание в работе уделяется освоению методов проверки правильности ввода информации. Необходимо разработать (или использовать готовые библиотеки) подпрограммы (функции), проверяющие правильность вводимых данных в соответствии с тремя основными способами проверки.

### **Краткие теоретические сведения**

Исчерпывающий материал по теме работы находится при переходе по ссылке: [https://pvn.org.ru/www\\_JS/lecture/4\\_html/4.pdf](https://pvn.org.ru/www_JS/lecture/4_html/4.pdf). До выполнения настоящей работы необходимо освоить все разделы лекции №4.

Рекомендуется также вспомнить принципы построения форм в HTML-документах (смотрите подраздел 2.7 лекции №2), а также просмотреть материал и примеры лекции 4, в которой изучаются принципы взаимодействия *Javascript* и объектов объектной

модели документа (HTML DOM).

Необходимо вспомнить три основных дополняющих друг друга, способа проверки информации, вводимой пользователем в форму:

- 1) проверка в целом всей формы вместе со всеми, имеющимися в ней полями, в момент отправки формы на обработку серверу;
- 2) проверка в момент набора на клавиатуре вводимого символа;
- 3) проверка в момент попытки перехода к следующему полю.

Эти способы проверки рассматриваются в лекции №5 при прохождении подраздела 5.3. В этом подразделе приведены и подробно описаны три функции, реализующие перечисленные способы проверки, а именно:

функция `checkForm ()` выполняет проверку всех полей при отправке формы на обработку серверу;

функция `no_empty()` срабатывает при переходе к следующему полю, выполняя проверку значения, введённого до перехода;

функция `no_number()` срабатывает мгновенно, по нажатию любой клавиши клавиатуры.

При выполнении настоящей лабораторной работы студентам разрешается использовать эти функции.

Для желающих получить дополнительные поощрительные баллы разрешается использовать специально разработанные библиотеки, реализующие более сложные алгоритмы проверок форм. Одну из таких библиотек, Javascript\_form, с описанием (англ.) можно взять по адресу: [http://www.lifeinweb.biz/wp-content/uploads/2012/06/javascript\\_form.zip](http://www.lifeinweb.biz/wp-content/uploads/2012/06/javascript_form.zip). В библиотеке Javascript\_form размещён класс javascript под названием javascript Form Validation. Краткое описание этой библиотеки на русском языке с примером с целью обучения её использованию имеется по адресу: <http://www.lifeinweb.biz/kak-proverit-formu-s-ispolzovaniem-javascriptredaktirovat/>.

В качестве дополнительного задания можно также разработать функцию выводящую данные формы в графическом виде с использованием графической библиотеки *Highcharts* (или любой другой библиотеки). Эта библиотека рассматривается в подразделе 4.5 конспекта лекций, где приводятся примеры её применения. Библиотеку можно взять на сайте владельца по адресу: <http://www.highcharts.com/download>. Можно не скачивать, а подключить, указав URL. Примеры подключения имеются в многочисленных демонстрационных примерах. Основные возможности библиотеки с

примерами использования изложены по адресу: Описание:

<http://troitskiy.net/2010/10/13/highcharts-javascript-biblioteka-postroeniya-grafikov/>.

## Задания

### Постановка задачи.

1. Имеются ряды среднемесячных значений температуры воды в некоторой прибрежной точке Керченского пролива по данным за некоторый год.
2. Известны также среднемесячные климатические значения (нормы) температуры в этой точке, полученные усреднением по данным за многолетний период.
3. Необходимо написать программу для ввода данных за текущий год, вычисления отклонений от норм и средних за 12 месяцев и вывода вводимых и вычисляемых значений в виде таблицы.
4. Все рутинные операции должны быть автоматизированы. Необходимо предусмотреть проверку вводимых данных, исключая возможность ввода пустых и нецифровых значений.

Первоначальный вид формы может иметь примерно такой вид:

**Отклонения среднемесячных температур воды от нормы в г.Керчь (Опасное) за  -й год.**

Месяц	Нормы	Разница
01	-0.1	
02	0.2	
03	3.2	
04	8.8	
05	13.1	
06	17.2	
07	18.2	
08	15.2	
09	5.1	
10	2.0	
11	1.2	
12	-0.4	
<b>Средние:</b>		

Рис. 3.7.1 Рекомендуемый вид формы при открытии веб-документа.

При попытке ввода не цифры вместо вводимого символа должно появляться окно

предупреждения с текстом: “Недопустимый символ! В это поле разрешается вводить только лишь цифры и точку”.

При попытке перейти к другому полю, не заполнив предыдущее поле должно появляться окно предупреждения с текстом: “Поле обязательно для заполнения! Повторите ввод.”

Очередное значение отклонения от нормы должно появляться сразу же после окончания ввода очередного значения среднемесячной температуры по переходу по клавишам “Tab”, “Enter” или с помощью мышки.

В результате ввода данных и всех вычислений выводимая на экран таблица должна иметь вид, показанный на рис. 3.7.2.

**Отклонения среднемесячных температур воды от нормы в г.Керчь (Опасное) за 2009 -й год.**

Месяц		Нормы	Разница
01	-0.2	-0.1	-0.1
02	0.2	0.2	0.0
03	4.7	3.2	1.5
04	10.2	8.8	1.4
05	14.5	13.1	1.4
06	18.5	17.2	1.3
07	20.2	18.2	2.0
08	18.2	15.2	3.0
09	6.2	5.1	1.1
10	3.2	2.0	1.2
11	1.8	1.2	0.6
12	0.4	-0.4	0.8
<b>Средние:</b>	8.2	7.0	1.2

**Рис. 3.7.2** Вид формы после ввода всех значений среднемесячной температуры.

При нажатии на расположенную внизу кнопку “Средние” должны вычисляться средние годовые значения (при условии заполнения всех полей, иначе – сообщение с напоминанием о необходимости заполнения). Кнопку “Отклонения”, показанную в

формах на рис.3.7.1-3.7.2 можно не создавать.

Вместо “Керчь” и “Опасное” нужно вывести № точки (номер варианта).

Климатические нормы должны соответствовать варианту из таблицы 3.7.1. Номер варианта (это и номер точки в проливе) должен быть равен двум последним цифрам вашей зачётной книжки.

Таблица 3.7.1 Климатические нормы в точках для вариантов

№ варианта	Месяцы												За год *)
	01	02	03	04	05	06	07	08	09	10	11	12	
00	-0.1	0.2	3.2	8.8	13.1	17.2	18.2	15.2	5.1	2.0	1.2	-0.4	7.0
01	-0.2	-1.9	-0.7	3.1	7.9	12.5	15.4	17.1	15.5	12.4	8.1	3.4	7.7
02	-0.3	-0.7	1.6	4.9	9.6	13.5	17.5	17.2	15.3	11.2	6.3	2.0	8.2
03	-0.7	1.0	4.4	7.8	12.0	15.7	17.3	16.1	12.1	8.6	3.9	0.8	8.2
04	-1.9	0.2	3.2	7.8	12.7	16.1	17.0	15.7	13.7	8.6	3.1	-0.4	8.0
05	0.9	0.9	2.1	5.7	9.2	12.2	15.2	14.8	13.0	10.8	6.8	3.4	7.9
06	1.3	-0.5	-0.1	3.0	7.7	11.7	15.2	17.5	15.5	13.0	8.4	3.9	8.1
07	2.4	1.0	3.1	6.3	9.5	12.3	14.2	15.1	13.4	10.3	6.4	3.7	8.2
08	2.1	1.1	0.6	3.7	8.1	11.9	14.5	16.5	14.5	12.1	7.9	3.8	8.1
09	-1.2	-0.1	3.0	7.7	12.3	15.9	17.1	16.4	12.2	8.0	3.6	0.6	8.0
10	-0.7	-0.9	2.2	6.2	11.4	15.2	16.9	16.2	14.4	9.5	4.5	0.1	7.9
11	-0.5	-1.2	2.4	7.0	11.8	15.0	17.0	16.4	13.4	9.1	4.1	0.2	7.9
12	-1.2	-2.3	-0.2	3.8	8.7	14.2	17.6	17.3	16.8	12.3	7.3	1.6	8.0
13	0.2	-0.6	0.4	3.4	7.8	12.0	15.0	16.9	15.9	12.8	7.9	3.3	7.9
14	1.8	-0.1	-0.4	2.5	7.3	12.2	15.7	17.6	16.8	13.6	9.3	4.4	8.4
15	2.6	1.0	0.9	3.5	7.0	11.2	13.9	16.0	14.8	12.7	8.7	4.1	8.0
16	1.4	0.8	4.3	7.2	10.5	14.2	14.8	15.1	12.1	8.9	4.8	1.7	8.0
17	0.3	1.6	3.9	7.7	11.2	13.8	14.8	14.0	12.6	8.3	4.3	1.6	7.8
18	0.4	-1.7	0.7	4.8	9.5	13.4	17.5	17.4	15.9	11.3	6.4	1.5	8.1
19	-1.8	-1.8	1.8	6.4	11.6	15.9	18.9	16.9	14.9	9.8	4.3	-0.5	8.0
20	-0.6	-1.7	1.5	4.9	9.3	13.3	16.4	18.0	15.3	11.1	6.7	2.2	8.0

\*) Последний столбец (“За год”) может использоваться для дополнительной проверки правильности ввода месячных норм.

**Задание №1.** Разработать веб-документ с HTML-формой, вид которой представлен на рис. 3.7.1.

Веб-документ должен соответствовать стилю шаблона, разработанного вами ранее по заданию работы №4. Веб-документ необходимо сохранить в файле Petrov.html в папке

Petrov.

В форме должен быть предусмотрен столбец с неизменяемыми значениями норм по варианту из таблицы 3.7.1.

**Задание №2.** Разработать подпрограммы (функции) на языке Javascript, с помощью которых должны выполняться проверки ввода в форму, аналогичные проверкам, выполняемым в функциях `checkForm()`, `no_empty()` и `no_number()` (см. лекции №5 подраздел 5.3).

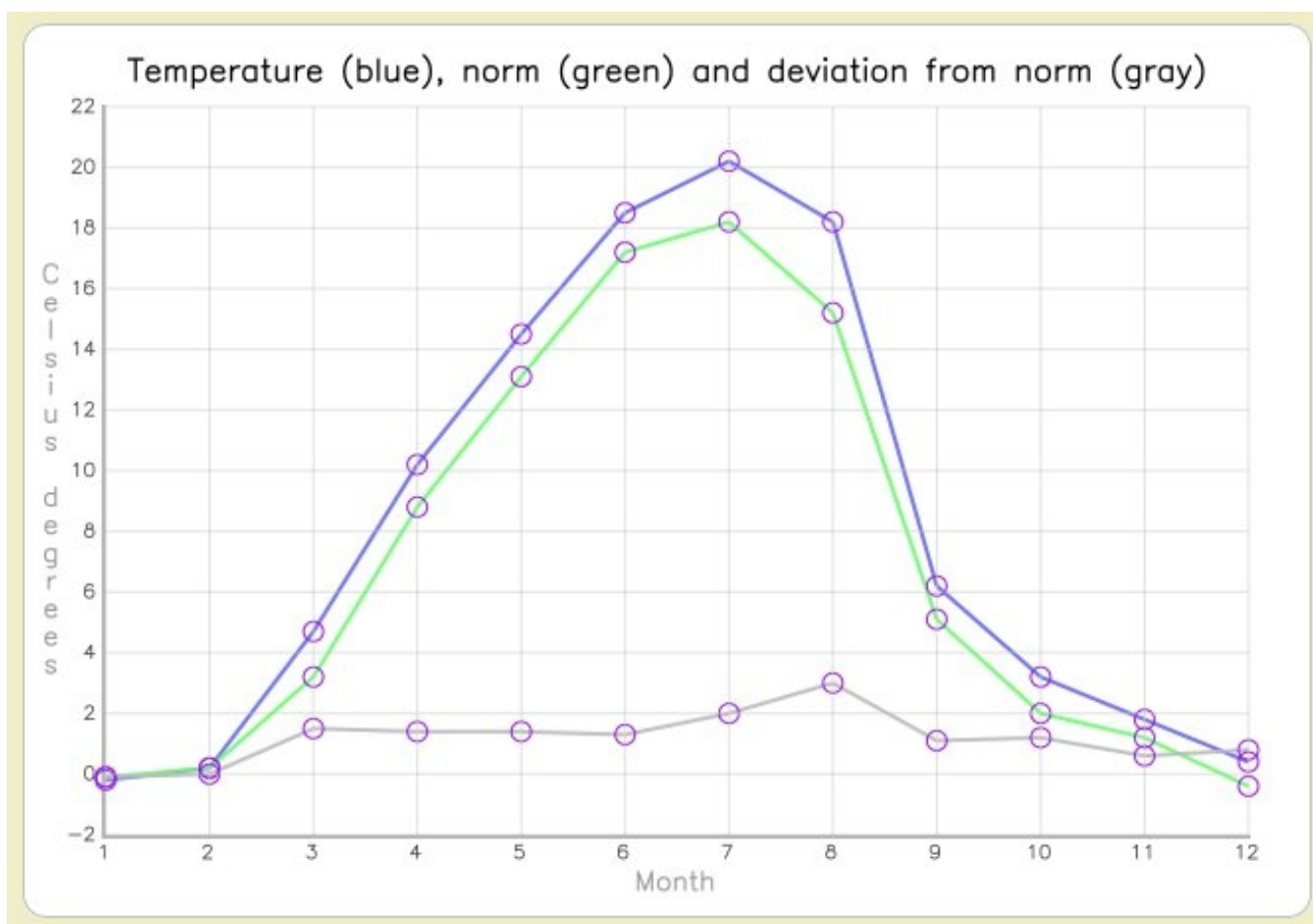
При выполнении настоящего задания разрешается использовать эти функции или функции любых других библиотек, но они (эти функции) должны быть помещены в вашу собственную библиотеку (в подключаемый файл с расширением `js`) и этот файл должен быть помещён в папку Petrov вместе с HTML-файлом с формой.

Для желающих получить дополнительные поощрительные баллы разрешается использовать специально разработанные библиотеки, реализующие более сложные алгоритмы проверок форм. Одну из таких библиотек, `Javascript_form`, с описанием (англ.) можно взять по адресу: [http://www.lifeinweb.biz/wp-content/uploads/2012/06/javascript\\_form.zip](http://www.lifeinweb.biz/wp-content/uploads/2012/06/javascript_form.zip). В библиотеке `Javascript_form` размещён класс `javascript` под названием `javascript Form Validation`. Краткое описание этой библиотеки на русском языке с примером с целью обучения её использованию имеется по адресу: <http://www.lifeinweb.biz/kak-proverit-formu-s-ispolzovaniem-javascriptredaktirovat/>.

В качестве дополнительного задания можно также разработать функцию выводящую данные формы в графическом виде с использованием графической библиотеки *Highcharts* (или любой другой библиотеки). С этой библиотекой вы уже имели возможность научиться работать при выполнении лабораторной работы №5. Эта, и другие графические библиотеки, рассматривается в подразделе 4.5 конспекта лекций, где приводятся примеры применения. Библиотеку можно взять на сайте владельца по адресу: <http://www.highcharts.com/download>. Можно не скачивать, а подключить, указав URL. Примеры подключения имеются в многочисленных демонстрационных примерах. Основные возможности библиотеки с примерами использования изложены по адресу: Описание: <http://troitskiy.net/2010/10/13/highcharts-javascript-biblioteka-postroeniya-grafikov/>.

Ниже расположен рекомендуемый вид графиков, которые вы можете также увидеть в примере по ссылке:

используется библиотека *JSCharts*).



**Рис. 3.7.3.** Графики среднемесячных значений температуры воды: климатических норм, за текущий год, отклонений от норм.

#### Методика выполнения

Методика выполнения подробно описана при переходе по ссылке: [https://pvn.org.ru/www\\_JS/lection/4\\_html/4.pdf](https://pvn.org.ru/www_JS/lection/4_html/4.pdf). Настоящая работа выполняется после прохождения всех подразделов лекции №4.

В целях усвоения материала лекции №4 для самостоятельной работы студентов предусмотрено выполнение упражнений, в которых имеются фрагменты, полностью охватывающие задания настоящей лабораторной работы.

#### 3.7.4 Рекомендации по обработке и оформлению полученных результатов

Работа оформляется в виде веб-документа и сохраняется в файле Petrov.html с использованием шаблона, который должен был быть создан ранее.

Все функции должны быть помещены в файл с расширением js, присоединяемый к файлу Petrov.html.

Все файлы лабораторной работы ( Petrov.html, css-файл и js-файл) должны быть

помещены в папку Petrov.

В конце файла Petrov.html должны быть помещены ответы на контрольные вопросы, помещённые ниже.

### 3.7.5 Вопросы для самоконтроля

1. Можно ли создавать с помощью атрибута `name` одинаковые имена в элементах управления формы?
2. Можно ли создавать с помощью атрибута `id` одинаковые идентификаторы в элементах управления формы?
3. Объекты, создаваемые при каждом добавлении в HTML-документ набора тэгов `<form> ...</form>`?
4. Как получить доступ ко второй форме страницы?
5. Как определить количество форм на странице?
6. Как определить количество управляющих элементов во второй форме страницы?
7. Назначение атрибутов `name` и `id`?
8. Свойство, позволяющее получить доступ к значению элемента формы?
9. Оператор присваивания произвольной переменной значения элемента формы с именем `in1`?
10. Свойство, позволяющее получить доступ к значению атрибута `action="url"` формы?
11. Оператор присваивания произвольной переменной значения атрибута `action="url"` формы с именем `in1`? В операторе должно использоваться имя формы.
12. Свойство, являющееся массивом, позволяющим получить доступ к каждому элементу формы?
13. Свойство, являющееся массивом, содержащим элемент в поле со списком, создаваемым с помощью набора тэгов `<select> ...</select>`?
14. Методы объекта формы?
15. Метод позволяющий сбросить форму, используя код на языке `Javascript`?
16. Метод, позволяющий отправить форму, используя код на языке `Javascript`?
17. Какое булево значение должна вернуть функция проверки формы, если введённые пользователем значения ошибочны и нужно отменить отправку формы серверу?
18. Как задаются методы отправки формы серверу при использовании атрибута `method`?
19. Составные части URL?
20. Разница между URL, отправленными при различных значениях атрибута `method`?
21. Можно ли просмотреть в браузере код PHP-скрипта, внедрённого в HTML-документ, также, как это можно сделать при просмотре исходного кода на языке `Javascript`?
22. Какое расширение имеют файлы HTML-документов с внедрёнными скриптами на PHP?
23. Какие тэги используются для внедрения PHP-скриптов?
24. Как образуются имена переменных в скриптах на PHP?



25. В чём состоит разница между операторами `if` в языках `Javascript` и `PHP` ?
26. В чём состоит разница между операторами `for` в языках `Javascript` и `PHP` ?
27. В чём состоит разница между описанием и вызовом функций в языках `Javascript` и `PHP`?
28. Как образуются и используются массивы в скриптах на `PHP`?
29. В чём состоит разница в использовании функций из подключаемых файлов в языках `Javascript` и `PHP`?
30. Новый тэг для вывода информации в `HTML5` ?
31. Новые типы элементов ввода форм, появившиеся в `HTML5`?
32. Новые атрибуты элементов ввода форм, появившиеся в `HTML5`?
33. Новые псевдоклассы `CSS3` для работы с элементами ввода форм, появившимися в `HTML5`?