МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

Институт информационных технологий и управления в технических системах

кафедра Информационные системы

09.03.02 Информационные системы и технологии (уровень бакалавриата)

Лабораторная работа №2

Исследование возможностей программирования на стороне клиента. Основы языка JavaScript

по дисциплине: «Веб-технологии»

Вариант – 2

Выполнил

студент 3 курса группы ИС/б-33-о

Генералов Николай Николаевич

Отметка о зачете\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(дата)

Проверил

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ­ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность) (подпись) (фамилия, инициалы)

г. Севастополь

2018 г.

1. ЦЕЛЬ РАБОТЫ

Изучить основы языка JavaScript и объектной модели браузера, приобрести практические навыки проверки HTML-форм с использованием JavaScript.

1. ПОСТАНОВКА ЗАДАЧИ

* Модифицировать страницу «Фотоальбом» (использовать HTMLстраницы, разработанные при выполнении предыдущей лабораторной работы), реализовав вывод таблицы, содержащей фото, с использованием операторов циклов. Значения имен файлов фото и подписей к фото предварительно разместить в массивах fotos и titles.
* Модифицировать страницу «Мои интересы», реализовав вывод списков с использованием JavaScript-функции с переменным числом аргументов.
* Добавить на страницах «Контакт» и «Тест по дисциплине» функции проверки заполненности форм. В случае если какое-либо из полей формы осталось незаполненным при нажатии на кнопку отправить, вывести сообщение об ошибке и установить фокус на незаполненный элемент.
* Добавить на странице «Контакт» текстовое поле «Телефон». И для полей “Фамилия Имя Отчество” и “Телефон” добавить функции специфической проверки значений. В случае если какое-либо из полей формы заполнено не верно, при нажатии на кнопку отправить, вывести сообщение об ошибке и установить фокус на неверно заполненный элемент. Формат правильных значений полей: Фамилия Имя Отчество – введено три слова, разделенные одним пробелом. Телефон – строка может состоять только из цифр; начинаться только с последовательности «+7» или «+3»; не содержит пробелов; количество цифр в строке от 9 до 11.
* Добавить на странице «Тест по дисциплине» функции проверки, что в вопросе №2 установлены как минимум два переключателя. В случае не выполнения условия сформировать сообщение об ошибке.
* Необходимо выполнить проверку разработанных JavaScript файлов с использованием сервиса jshint.

1. ХОД РАБОТЫ

В начале выполнения задания была модифицирована страница «Галерея» содержащая набор фотографий, которые располагались непосредственно в HTML документе. Для модификации данной страницы был написан скрипт “gallery.js”, который создает 15 объектов <img” в качестве атрибута “src” которому передаются имена фотографий, которые хранятся в заранее инициализированном массиве.

На втором этапе был разработан скрипт “interests.js” для вывода списка хобби на соответствующей странице, с помощью вызова функции с переменным числом аргументов. Данная функция в качестве первого параметра получает идентификатор родительского блока, и переменное количество строк, которые будут представлены как элементы списка.

На третьем этапе были разработан скрипт для проверки заполнения текстовых полей на формах расположенных на HTML-страницах “contacts.html” и “test.html”.

На заключительном этапе были разработаны скрипты “ContactsFormValidation.js” и “ TestFormValidation.js”, которые осуществляют проверку на соответствие значений полей, условиям поставленным для данной лабораторной работы.

1. ТЕКСТ ПРОГРАММНЫХ МОДУЛЕЙ

“custom.js”

'use strict';

function checkInputsFeeling(form){

var elements = form.elements;

for(var el in elements){

var element = elements[el];

if (element.tagName == 'INPUT' && element.value == ""){

alert("Поле " + element.name + " не может быть пустым");

element.focus();

element.classList.add("red-border");

return false;

}

if(!!element.classList && element.classList.contains('red-border')){

element.classList.remove('red-border');

}

}

return true;

};

“gallery.js”

'use strict';

(function () {

    var fotos = [];

    for (var i = 0; i < 15; ++i) {

        fotos.push((i + 1) + ".jpg");

    }

    var titles = [];

    for (var j = 0; j < 15; ++j) {

        titles.push("Это фотография " + (j + 1) + " места, где я когда-либо буду");

    }

var gallery\_block = window.document.getElementById('gallery\_block');

for(var k = 0; k < fotos.length; ++k){

var figure = document.createElement('figure');

figure.className = "gallery\_page";

var div = document.createElement('div');

div.className = "gallery\_page photo\_container";

var img = new Image();

img.src = "img/gallery\_img/" + fotos[k];

img.alt = "It\'s my photo";

img.className = 'gallery\_page photo\_img';

var figcaption = document.createElement('figcaption');

figcaption.className = 'gallery\_page';

var p = document.createElement('p');

p.className = 'gallery\_page description\_img';

p.innerHTML = titles[k];

figure.appendChild(div);

div.appendChild(img);

div.appendChild(figcaption);

figcaption.appendChild(p);

gallery\_block.appendChild(figure);

}

})();

“interests.js”

'use strict';

/\*

Функция заполняет список хобби

Список параметров:

blockHeader - текст заголовка

остальные параметры заполняются в li

\*/

function generateHobbyList(blockHeader) {

var resoursesBlock = document.getElementById('hobbys');

var h3 = document.createElement('h3');

h3.classname = 'interests\_page';

h3.innerHTML = blockHeader;

var div = document.createElement('div');

div.classname = 'interests\_page';

var ul = document.createElement('ul');

ul.classname = 'interests\_page hobbys\_list';

for (var i in arguments) {

var li = document.createElement('li');

li.classname = 'interests\_page';

li.innerHTML = arguments[i];

ul.appendChild(li);

}

div.appendChild(ul);

resoursesBlock.appendChild(h3);

resoursesBlock.appendChild(div);

}

generateHobbyList("Мои хобби",

                 "Велопоездки",

                 "Автопутешествия",

                 "Коддинг",

                 "Активный отдых");

“ContactsFormValidation.js”

'use strict';

(function addListeners(){

    var mainForm = document.forms[0];

    var TelNumberInput = mainForm.phoneNumber;

    var FioInput = mainForm.FIO;

    mainForm.addEventListener('submit', function(e){

        var result = checkInputsFeeling(this)

            & checkTelNumber(TelNumberInput)

            & checkFIO(FioInput);

        if (!result){

            e.preventDefault();

        }

    });

    TelNumberInput.addEventListener('change', function(e){

        checkTelNumber(this);

    });

    FioInput.addEventListener('change', function(e){

        checkFIO(this);

    });

})();

function checkTelNumber(TelNumberInput){

        var value = TelNumberInput.value;

        var regExp = /^\+[73]\d{9,11}$/;

        if (value.search(regExp) == -1){

            alert("Неверный формат номера телефона");

            TelNumberInput.focus();

            TelNumberInput.classList.add("red-border");

            return false;

        }

        if(TelNumberInput.classList.contains('red-border')){

            TelNumberInput.classList.remove('red-border');

        }

        return true;

};

function checkFIO(FioInput){

    var value = FioInput.value;

    var regExp = /^[А-ЯЁ][а-яё]+ [А-ЯЁ][а-яё]+ [А-ЯЁ][а-яё]+$/;

    if (value.search(regExp) == -1){

        alert("ФИО введено неверно");

        FioInput.focus();

        FioInput.classList.add("red-border");

        return false;

    }

    if(FioInput.classList.contains('red-border')){

        FioInput.classList.remove('red-border');

    }

    return true;

};

“TestFormValidation.js”

'use strict';

(function addListeners() {

var mainForm = document.forms[0];

mainForm.addEventListener('submit', function (e) {

var result = checkInputsFeeling(this) &&

checkCheckboxChecked(this, 2);

if (!result) {

e.preventDefault();

}

});

})();

function checkCheckboxChecked(form, minCheckedCount) {

var elements = form.elements;

var checkedCheckboxCount = 0;

var tempElement = null;

for (var el in elements) {

var element = elements[el];

if (element.type == 'checkbox') {

tempElement = element;

if (element.checked) {

++checkedCheckboxCount;

}

}

}

if (checkedCheckboxCount < minCheckedCount) {

alert('Должно быть выбрано не менее ' +

minCheckedCount + ' вариантов');

return false;

}

return true;

}

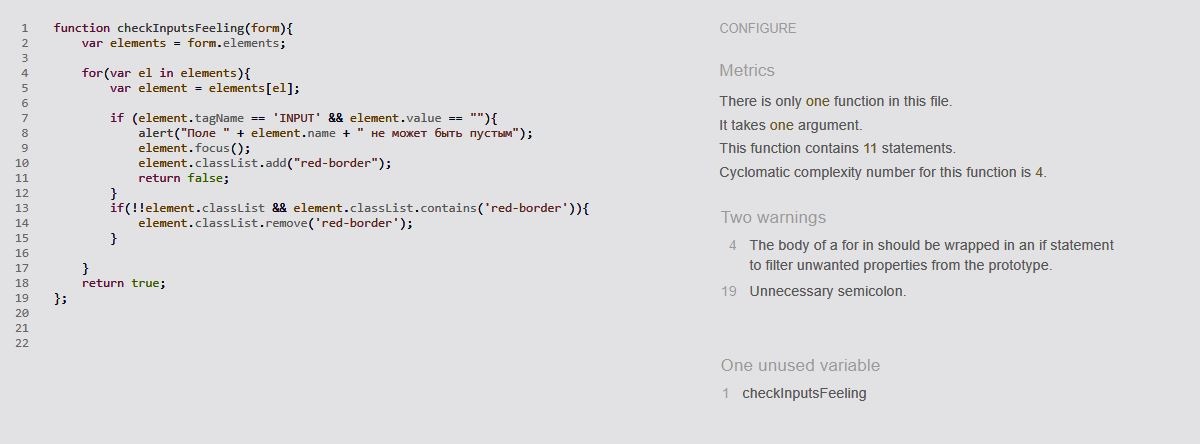
1. РЕЗУЛЬТАТЫ ПРОВЕРОК

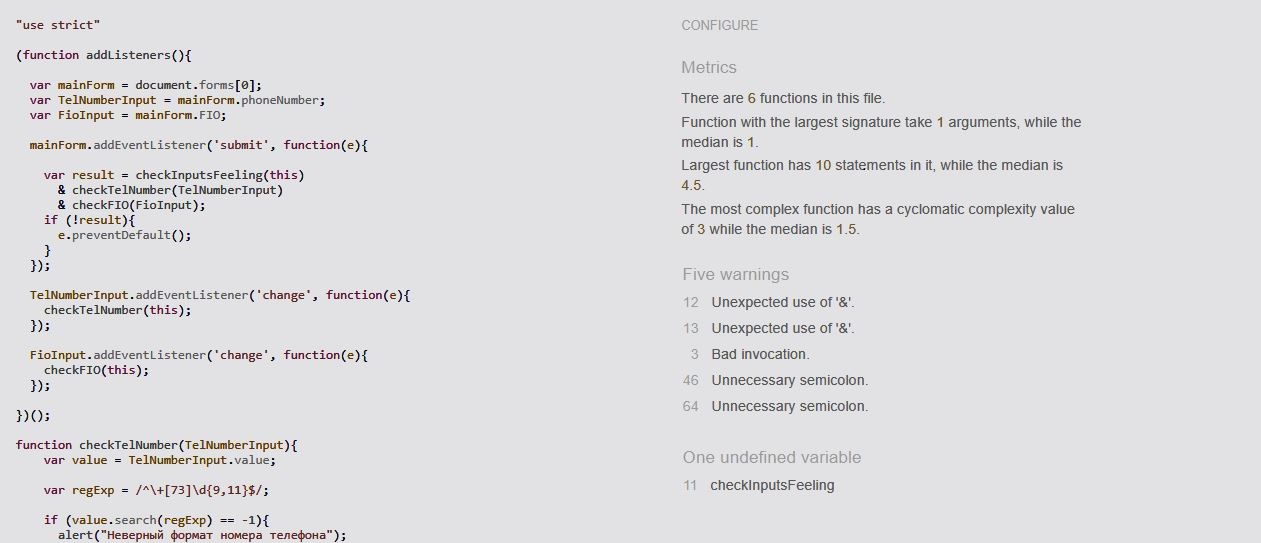
В данном разделе представлены результаты проверки программных модулей с помощью онлайн сервиса jshint.com

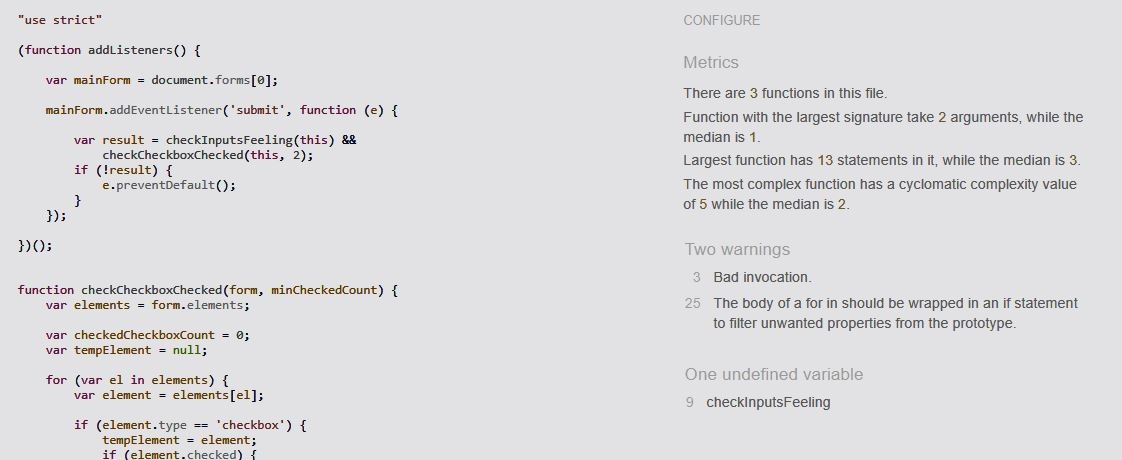


Рисунок 1 – Результат проверки файла “gallery.js”

Рисунок 2 – Результат проверки файла “interests.js”

Рисунок 3 – Результат проверки файла “custom.js”

Рисунок 4 – Результат проверки файла “ ContactsFormValidation.js”

Рисунок 5 – Результат проверки файла “ TestFormValidation.js”

ВЫВОД

В ходе выполнения лабораторной работы были изучены основы языка JavaScript и объектной модели браузера, а так же были приобретены практические навыки проверки HTML-форм с использованием JavaScript.

Для закрепления практических навыков, был модифицирован личный сайт, разрабатываемый в предыдущей лабораторной работе. В модификацию были включены несколько файлов осуществляющих динамическое добавление элементов страницы, а так же проверку корректности данных введённых пользователем.