Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования «Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра «Информационная безопасность»

Отчет по лабораторной работе 7

*по курсу*

СЕТЕВОЕ ПРОГРАММИРОВАНИЕ

Выполнил: студент группы 221061-ПБ Савельева В.А.\_\_\_\_\_\_\_\_\_\_\_

Проверил: асс.каф.ИБ Поляничко К.В.\_\_\_\_\_\_\_\_\_\_\_

Тула, 2019 г.

**ХОД РАБОТЫ**

**Практическое задание 1**

* 1. Написать функцию, преобразующую число в объект. Передавая на вход число от 0 до 999, надо получить на выходе объект, в котором в соответствующих свойствах описаны единицы, десятки и сотни. Например, для числа 245 надо получить следующий объект: {‘единицы’: 5, ‘десятки’: 4, ‘сотни’: 2}. Если число превышает 999, необходимо выдать соответствующее сообщение с помощью console.log и вернуть пустой объект. Результат представлен на рисунке 1.

function getNum( num ) {

var obj = {};

if ( num < 1 || num > 999 || isNaN( num )) {

console.log( 'Ошибка. Введите число от 1 до 999' );

return obj;

}

else {

obj[ 'едининцы' ] = num % 10;

obj[ 'десятки' ] = ( num - obj[ 'едининцы' ]) % 100 / 10;

obj[ 'сотни' ] = ( num - obj[ 'едининцы' ] - obj[ 'десятки' ] \* 10 ) % 1000 / 100;

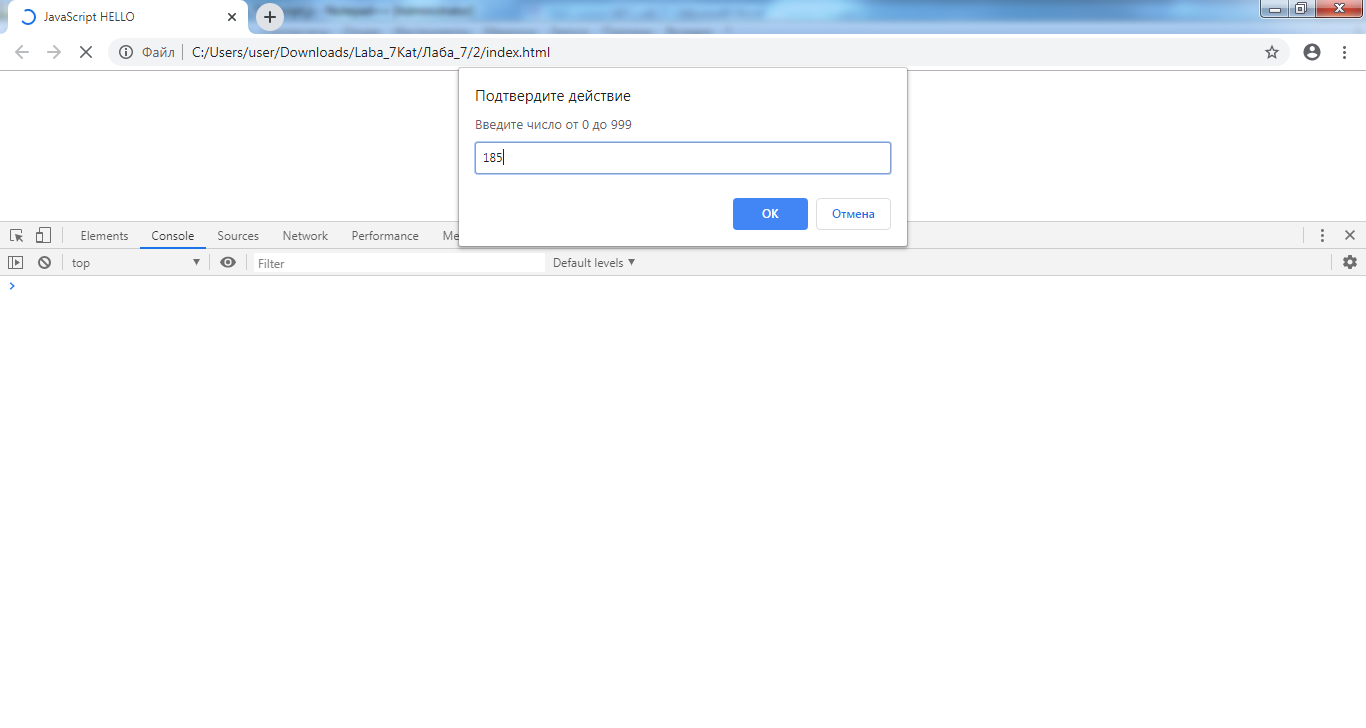
return obj;

}

}

var a = prompt('Введите число от 0 до 999');

console.log( getNum(a));



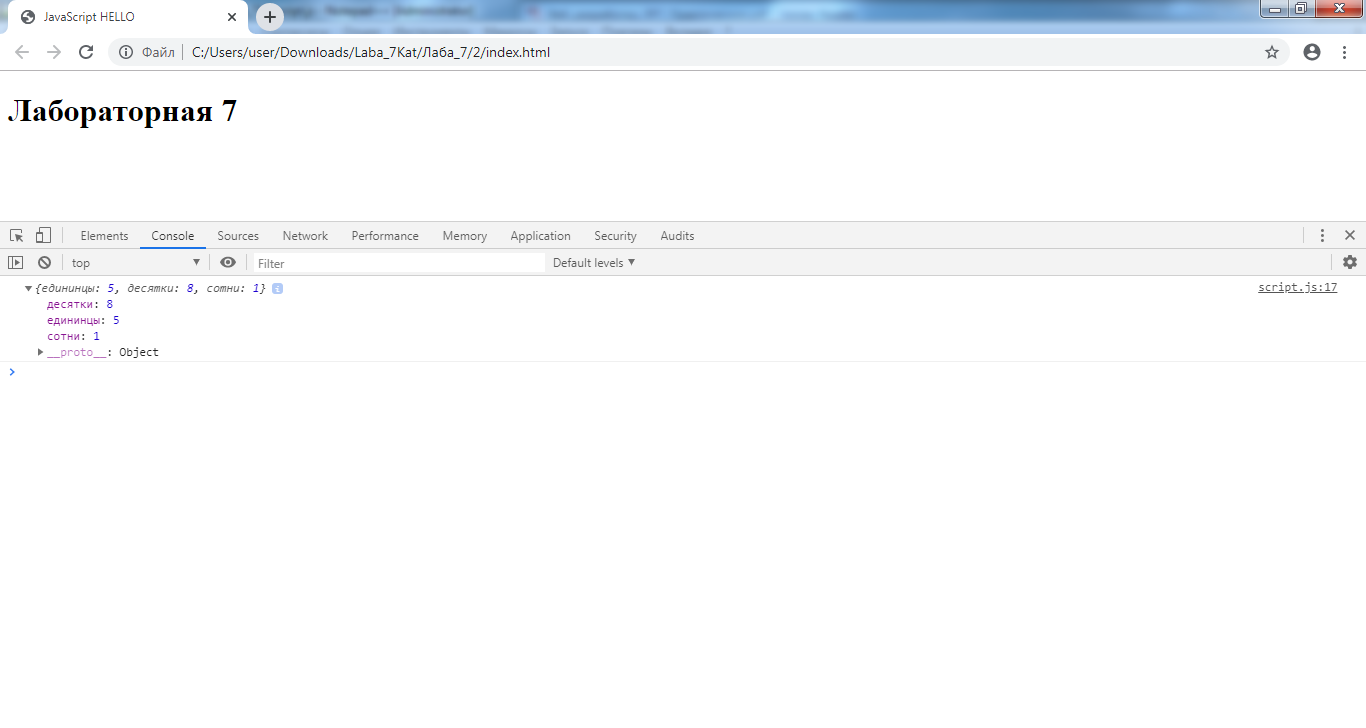


Рисунок 1 – Результат разделения числа на сотни, десятки и единицы

1.2 Продолжить работу с интернет-магазином, а именно, перенести функционал подсчета корзины на объектно-ориентированную базу:

* создать объект Basket с полем goodList и методами countTotalPrice,

countTotalNumber и putProduct (методы должны считать общую стоимость того, что в goodList, общее количество того, что в goodList и добавлять товар в корзину)

* создать массив товаров с полями name и price
* после этого пройтись по нему в цикле и положить в корзину по одному товару на четной позиции и по 2 товара на нечетной позиции
* внутри корзины в массиве goodList у каждого товара должно быть поле

count – количество товаров этого вида в корзине (например, 5 красных шаров)

* вывести общую стоимость корзины и общее количество товаров в корзине.

Результат работы программы представлен на рисунке 2.

let basket = {

goodList: [],

countTotalPrice() {//общая стоимость

let totalPrice = 0;

for (let i = 0; i < products.length; i++) {

let goodsItem = this.goodList[i];

totalPrice += goodsItem.price \* goodsItem.count;

}

return totalPrice;

},

countTotalNumber() {//общее количество

let totalCount = 0;

for (let i = 0; i < products.length; i++) {

let goodsItem = this.goodList[i];

totalCount += goodsItem.count;

}

return totalCount;

},

putProduct(product, count) {//добавление товара в корзину

let goodsItem = {

product,

price: product.price,

count

};

this.goodList.push(goodsItem);

}

};

let products = [];//

for (let i = 1; i < (parseInt(Math.random() \* 10) + 2); i++) {//количество товаров от 1 до 10

let product = {//заполняем продукты

name: 'name' + i,

price: parseInt(Math.random() \* 10000) / 100

};

products.push(product);

}

console.log(products);

for (let i = 1; i <= products.length; i++) {

// положить в корзину по одному товару на четной позиции и по 2 товара на нечетной позиции

let count = i % 2 ? 2 : 1;

basket.putProduct(products[i - 1], count);

}

console.log(basket);

console.log('Общая цена: ' + basket.countTotalPrice().toFixed(2));

console.log('Общее количество: ' + basket.countTotalNumber());

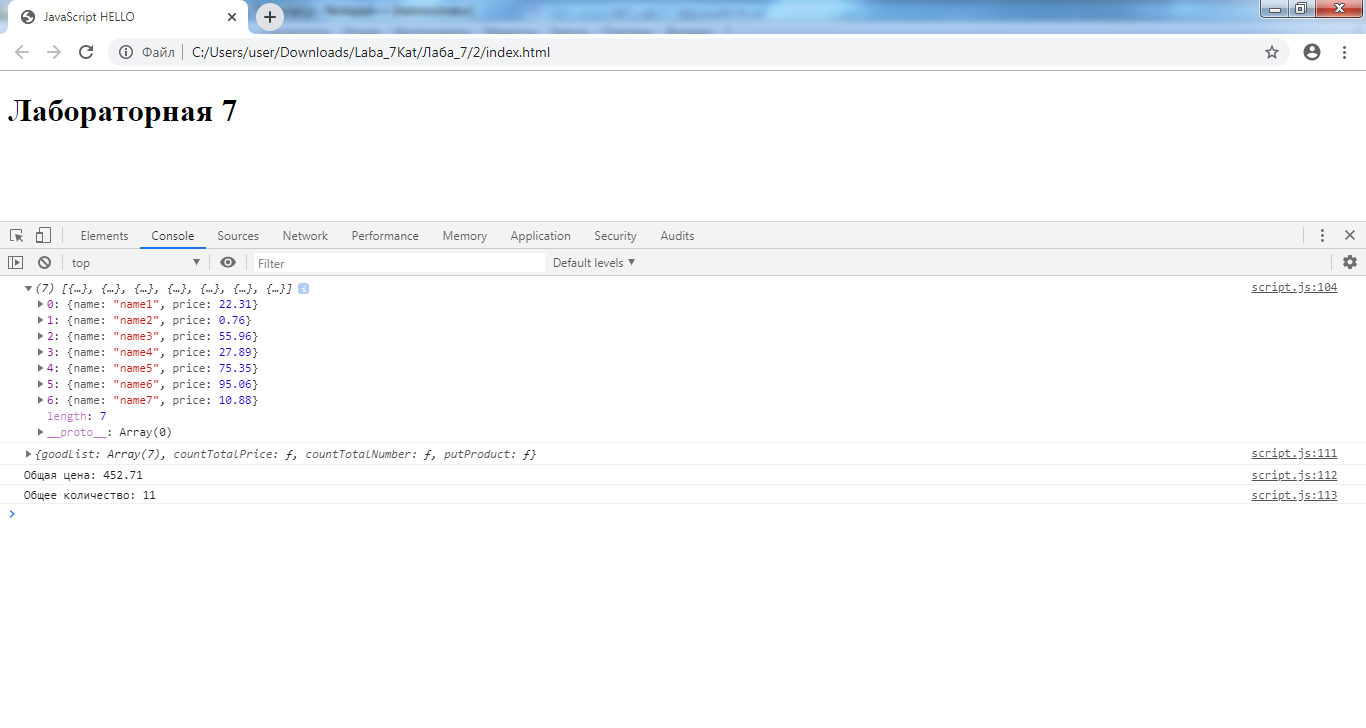


Рисунок 2 – Результат работы корзины

**Практическое задание 2**

2.1 Создать функцию, генерирующую шахматную доску. Можно использовать любые html-теги. Доска должна быть верно разлинована на черные и белые ячейки. Строки должны нумероваться числами от 1 до 8, столбцы — латинскими буквами A, B, C, D, E, F, G, H. Результат работы представлен на рисунке 3.

function chessboard() {

var newTable = document.createElement( 'table' ),//10\*10

lets = [ '','A','B','C','D','E','F','G','H','' ],

blackFigs1 = [ '8','','','','','','','','','8' ],

whiteFigs1 = [ '1','','','','','','','','','1' ],

blackFigs2 = [ '7','','','','','','','','','7' ],

whiteFigs2 = [ '2','','','','','','','','','2' ];

for ( var i = 0, a = 9; i < 10, a >= 0; i++, a-- ) {

var newTr = newTable.insertRow(i);//добавление строки в таблицу

for ( var j = 0; j < 10; j++ ) {

var newTd = newTr.insertCell( j );//вставляем новую ячейку в последнюю позицию

switch (i) {

case 0:

newTd.innerText = lets[ j ];

break;

case 1:

newTd.innerHTML = blackFigs1[ j ];

break;

case 2:

newTd.innerHTML = blackFigs2[ j ];

break;

case 7:

newTd.innerHTML = whiteFigs2[ j ];

break;

case 8:

newTd.innerHTML = whiteFigs1[ j ];

break;

case 9:

newTd.innerText = lets[ j ];

break;

default:

if ( j === 0 || j === 9 ) {

newTd.innerHTML = a;

}

break;

}

}

}

var elem = document.getElementsByClassName( 'chess' );

document.body.appendChild( newTable );

};

chessboard();

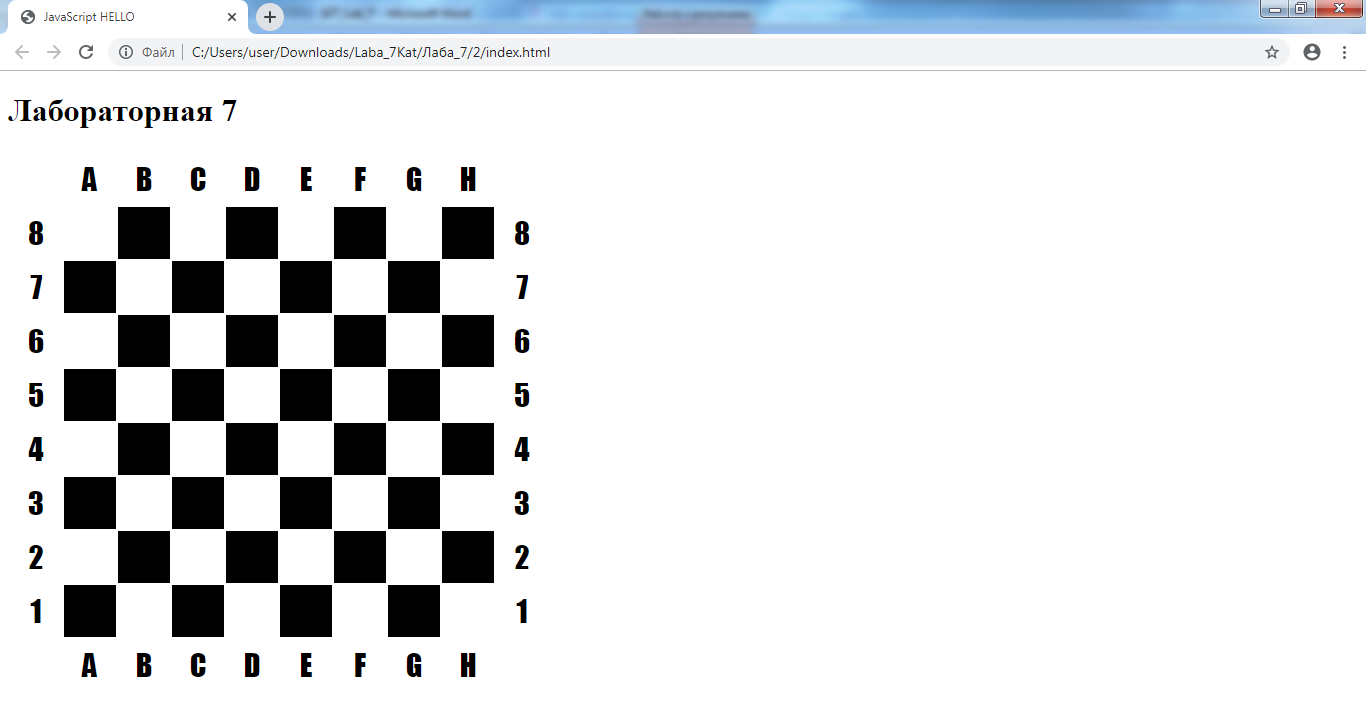


Рисунок 3 – Полученная шахматная доска

# ВЫВОД

# При выполнении данной лабораторной работы было продолжено изучение языка JavaScript, изучено понятие и реализация объектов в JS, модель документа и работа с ней, а также написаны демонстрационные программы для закрепления материала.