|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ | | | | | | | | | | | | | | | | | | |
| Федеральное государственное автономное образовательное  учреждение высшего образования | | | | | | | | | | | | | | | | | | |
| **Дальневосточный федеральный университет** | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
| **ШКОЛА ЕСТЕСТВЕННЫХ НАУК** | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
| **Кафедра информационной безопасности** | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
| **О Т Ч Е Т** | | | | | | | | | | | | | | | | | | |
| о прохождении учебной практики (по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности) | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
|  | | | | | | | |  |  | Выполнил студент гр. Б8216а | | | | | | | | |
|  |  |  | | | | | | Гусев М.Д. | | |
|  | | | | | | | |  |  | (подпись) | | | | | |  | | |
|  | | | | | | | | | | | | | | | | | | |
| Отчет защищен с оценкой | | | | | | | |  |  | Руководитель практики | | | | | | |  | |
|  | | | | | | | |  |  | Доцент кафедры информационной безопасности ШЕН | | | | | | | | |
|  | | | |  |  | | |  |  |  | | | | |  | Г.П. Антонова | | |
| (подпись) | | | |  | (И.О. Фамилия) | | |  |  | (подпись) | | | | |  | (И.О. Фамилия) | | |
| « |  | » |  | | | | 2018 г. |  |  |  | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
| Регистрационный № | | | | | |  | |  |  | Практика пройдена в срок | | | | | | | | |
| « |  | » |  | | | | 2018 г. |  |  | с | « | 23 | » | июля | | | | 2018 г. |
|  | | | | | | | |  |  | по | « | 04 | » | августа | | | | 2018 г. |
|  | | | |  |  | | |  |  | на предприятии | | | | | | | | |
| (подпись) | | | |  | (И.О. Фамилия) | | |  |  |
|  | | | | | | | |  |  | Кафедра информационной | | | | | | | | |
|  | | | | | | | |  |  | безопасности ШЕН ДВФУ | | | | | | | | |
|  | | | | | | | |  |  |  | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
| г. Владивосток | | | | | | | | | | | | | | | | | | |
| 2018 | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
| **Содержание** | | | | | | | | | | | | | | | | | | |
| [Задание на практику 3](#_Toc520317732)  [Введение 4](#_Toc520317733)  [Языки программирования 5](#_Toc520317734)  [Pascal 5](#_Toc520317735)  [С++ и С# 5](#_Toc520317737)  [HTML 6](#_Toc520317738)  [Реализация теста 7](#_Toc520317739)  [HTML 7](#_Toc520317740)  [CSS 9](#_Toc520317741)  [JavaScript 11](#_Toc520317743)  [Заключение 15](#_Toc520317744)  [Список используемых источников 16](#_Toc520317745) | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | |

# Задание на практику

- Поиск и анализ методов составления тестов

- Программная реализация интерактивного теста

# Введение

Учебная практика (по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности) проходила на кафедре информационной безопасности ШЕН ДВФУ в период с 23 июля 2018 года по 4 августа 2018 года.

Целью прохождения практики является приобретение практических и теоретических навыков по специальности, а также улучшения навыков написания программ.

Задачи практики:

1. Ознакомиться с базовыми языками программирования.
2. Теоретически ознакомиться с их синтаксисом, а также методами написания программ на этих языках.
3. На основе полученных знаний составить и написать интерактивный тест, согласно варианту.

# Языки программирования

За время обучения в университете было изучено четыре базовых языка программирования:

1. Pascal;
2. C++;
3. С#;
4. HTML с дополнительно подключаемыми CSS и JavaScript;

## Pascal

## в программе обучения университета был первым изучаемым языком программирования высокого уровня, также он изучается в школьном курсе информатики, что говорит о его относительной простоте и примитивности. Но в этой же простоте заключается и главная его проблема. Для работы на Pascal обычно используют консольное приложение, которое подходит лишь для вывода результатов простейших программ, поэтому он и является базовым языком для изучения программирования. Для того чтобы в Pascal создать окно, требуется относительно большой пласт программного кода, что делает данный язык фактически непригодным для поставленной задачи.

## С++ и С#

являются языками программирования высокого уровня. Они имеют схожий синтаксис и в целом очень схожи, и многие делают выбор между этими языками только «на свой вкус и цвет», поэтому и рассматриваться будут вместе. Данные языки расширяют возможности в области программирования относительно того же Pascal. Базой для работы на С++ и С# также является консольное приложение, что также создает проблемы для реализации ранее поставленной задачи, но в отличие от Pascal, вместо консольного приложения может использоваться Windows Forms, который позволяет создавать окна, а также задавать им нужные свойства. В отличие от Pascal для этого не требуется большого пласта кода: задать нужные параметры можно наглядно прямо в интерфейсе Windows Forms (в стандартной среде разработки «visual studio»). Но тогда помимо базового синтаксиса этих языков, придется изучать и работу с Windows Forms, которая имеет свои особенности и нюансы. Но т.к. практика ограничена двумя неделями, времени на изучение с нуля новых методов программирования может просто не хватить.

## HTML

- язык программирования для разметки документов во всемирной паутине. Он изучается в школьном курсе информатики еще раньше, чем Pascal. Т.к. этот язык изначально предназначен для разметки документов (а в большей степени именно это и требуется для реализации задания), то он является очень наглядным и не требует относительно большого пласта кода для создания окон, как Pascal. Важной плюсом является то, что он куда более простой в освоении, чем Windows Form, для работы с которым нужно сначала освоить C++ либо C#. Но в отличие от того же Pascal, который нужен в основном для введения в высшие языки программирования, HTML является стандартизированным языком для создания web-страниц. Т.е. при всей своей простоте он является фактически незаменимым в своей области. Также код на языке HTML можно связать с такими языками программирования, как СSS и JavaScript, которые изучались в университете в последнюю очередь (на данном этапе обучения). С помощью CSS можно задать стили тем объектам, которые мы зададим в HTML. А благодаря JavaScript, который также является высшим языком программирования, задается работа элементов, прописанных в том же HTML. Как можно было понять, также огромным плюсом данного варианта является наличие четкой последовательности, которой не могут похвастаться С++ и С# при работе на Windows Forms и которую при желании можно не соблюдать:

1. C помощью HTML прописываются элементы страницы;
2. В СSS задается более точное расположение этих элементов, а также их визуальные стили;
3. Затем в работу вступает документ JavaScript, в котором прописывается работа элементов из HTML и стилей из CSS;

# Реализация теста

Для практической реализации поставленной задачи, учитывая все плюсы и минусы, очевидно, был выбран язык HTML, с подключаемыми к нему CSS и JavaScript. Также задействовался один из его плюсов – наличие четкой последовательности, которая была описана выше.

Для начала прорисовываем все схемы, которые будут присутствовать в вариантах ответов. Схемы составлялись в программе Paint, т.к. отвечает всем требованиям для составления данных схем, а также она по умолчанию есть почти на каждом компьютере, и ее интерфейс знаком большинству людей с детства.

Затем создаем три новых документа, с соответствующими расширениями: html, css, js.

## В документе HTML задается стандартная разметка страницы:

<html>  
<head>  
<title>Заголовок страницы</title>  
<meta charset="windows-1251">  
<meta charset="utf-8" />  
<meta http-equiv="content-language" content="ru">  
<link rel="stylesheet" href="Sources/style.css" >  
<script src="Sources/work.js"></script>  
</head>  
<body>  
</body>  
</html>

В блоке «head» (между тегами <head> и </head>) указывается заголовок страницы (между тегами <title> и </title>), затем идут подключения всех нужных библиотек, а именно: сначала задается кодировка страницы, после чего подключается библиотека для того чтобы страницы поддерживала русский язык. В последних двух строках перед закрытием тега <head> прописывается подключение CSS и JavaScript файлов.

В блоке «body» (между тегами <body> и </body>) будет прописываться основная часть страницы. В нем с помощью тегов <center> и </center> задаем то, что все последующие блоки будут располагаться по центру. Затем прописываем главный блок, в котором будут располагаться все вопросы, он понадобится в конце.

Для начала прописываем структуру одного вопроса в тесте. Т.к. все вопросы визуально друг на друга похожи лишь с некоторыми различиями, то структуру этого вопроса можно будет использовать как шаблон для остальных, внося некоторые корректировки.

<div class="question">  
<div id="1" style="text">  
<p>1)Мостовая однофазная схема выпрямителя имеет вид:</p>  
</div>  
<div id="2" style="image">  
<img style="miniimage" src="Sources/1.jpg" >  
<img style="miniimage" src="Sources/2.jpg" >  
<img style="miniimage" src="Sources/3.jpg" >  
</div>  
<div style="answer">  
<p>Варианты ответов:</p>  
<input type="button" id="option1" value=" 1 ">  
<input type="button" id="option2" value=" 2 ">  
<input type="button" id="option3" value=" 3 ">  
</div>  
</div>

Первым блоком является блок полного вопроса (между первым тегом <div> и его закрытием </div> в последней строке), в котором располагаются остальные элементы вопроса. Внутри главного блока задаем еще один блок, в котором располагается текст вопроса (между тегами <p> и </p>). После блока с текстом вопроса идет блок, внутри которого задаются 3 картинки со схемами в качестве возможных вариантов ответов (3 тега <img>). В других вопросах вместо трех картинок, может быть либо одна большая картинка (тег <img>, c другим стилем), либо три текстовых варианта (три тега <p>, внутри которых задаем другой стиль и цвет с помощью тегов <em> и <font>, чтобы отличалось от текста вопросов). Затем идет блок с 3 кнопками для выбора варианта ответов (3 тега <input>). Но перед ними, между тегами <p> и </p>, прописываем, что данные кнопки – варианты ответов.   
Блок вопроса копируется и вставляется 18 раз, т.к. тест содержит 18 вопросов. И редактируется лишь текст вопроса, а также в вариантах ответах задаются адреса для других картинок, либо прописываются другие текстовые варианты ответов. После 18 блоков вопросов закрываем основной блок и затем прописываем кнопку, при нажатии на которую будут подводиться итоги теста. В ней задаем другие стилевые параметры, чтобы отличалась от кнопок вариантов ответов.

<input type="button" id="i0" value="Завершить тест " style="font-size : 30px; width: 70%; height: 100px; background: red;">

На этом работа с документом HTML заканчивается.

## В документе CSS задаем стили для уже имеющихся блоков:

## Сначала задаем стили для основных блоков, а именно: основной фон страницы, стиль шрифта вопросов, размер шрифта и цвет текста:

body  
{  
background-color:#F0FFF0;  
}  
p  
{   
font-family: Times new roman;   
font-size: 140%;   
font-weight: bold;   
}

К остальным блокам обращаемся с помощью наименований классов, которые прописывались в тегах HTML документа:

.question {  
width: 700px;  
height: 350px;   
background-color:#D3D3D3;  
}  
Задаем размеры и цвет основного блока вопроса;

.miniimage{  
width: 220px;  
height: 180px;  
margin: 10px 10px 10px 10px;  
display: inline-block;  
}  
Задаем размеры картинок, которые являются вариантами ответа (там, где 3 картинки). Затем задаются промежутки между картинками и прописываются свойства картинок, чтобы они расставились в ряд:

.question1{  
width: 700px;  
height: 400px;   
background-color:#D3D3D3;  
}  
Задаем размеры и цвет блока вопроса, в котором присутствует одна большая картинка (размеры отличаются от основных блоков вопросов);

.miniimage1{  
width: 350px;  
height: 250px;   
}  
Задаем размер картинки в вопросе, где она одна;

.b1 {  
background: navy;   
color: white;   
font-size: 9pt;  
}  
Задается стиль, на который будет меняться обычный стиль кнопки, при нажатии на нее, а именно: цвет кнопки, цвет текста в кнопке и размер шрифта;

## В документе JS прописываем работу кнопок и динамическую смену стилей:

function question1()  
{  
var button1 = document.getElementById('option1');  
var button2 = document.getElementById('option2');  
var button3 = document.getElementById('option3');  
var flag = 0;  
button1.onclick = function()  
{   
button1.setAttribute('class', 'b1');  
button2.setAttribute('class', '');  
button3.setAttribute('class', '');  
if(flag==1) {  
flag = 0;  
count--;  
}  
if(flag==0) {  
flag = 0;   
}  
}  
button2.onclick = function()  
{   
button1.setAttribute('class', '');  
button2.setAttribute('class', 'b1');  
button3.setAttribute('class', '');  
if(flag==1) {  
flag = 0;  
count--;  
}  
if(flag==0) {  
flag = 0;   
}  
}  
button3.onclick = function()  
{  
button1.setAttribute('class', '');  
button2.setAttribute('class', '');  
button3.setAttribute('class', 'b1');  
if(flag==1) {  
flag = 1;  
}  
if(flag==0) {  
flag = 1;  
count++;  
}  
}  
}

Данная функция задает работу кнопок в первом вопросе. Таких функций столько же, сколько вопросов (18), и в зависимости от того первый, второй или третий вариант ответа, в ней будут меняться некоторые параметры.

Логика работы функции: обращаемся к трем кнопкам в вопросе по названию их id, которое было прописано в HTML, и запоминаем их в переменных button1, button2, button3. Затем прописываются функции с действиями, которые выполнятся при нажатии на какую-либо кнопку. Если была нажата кнопка с правильным вариантом ответа, то переменная, отвечающая за количество правильных ответов, заданная ранее, увеличивается на единицу и при этом активируется флаг того, что кнопка нажата (переходит из режима true в false). Этот флаг отвечает за то, чтобы количество верных ответов не увеличивалось при многократном нажатии на один и тот же верный ответ. Если нажимается кнопка, с неправильным вариантом ответа, то ничего не происходит. Но если она нажимается после того, как нажата кнопка с верным вариантом ответа, то переменная, отвечающая за количество правильных ответов, уменьшается на единицу, а флаг, отвечающий за нажатие верной кнопки, снова переходит в значение true (которое было изначально). Помимо этого при нажатии на любую кнопку ей задается стиль, прописанный ранее в CSS, который задает цвет кнопки, цвет и размер шрифта. На других же кнопках стиль меняется на первоначальный. Таким образом, при нажатии кнопка меняет свой визуальный стиль, а остальные кнопки переходят в первоначальный стиль (если стиль и был первоначальный, т.е. кнопки не были нажаты, то их стиль визуально не меняется).

Остается прописать работу последней кнопки, отвечающей за окончание теста:

var end = document.getElementById('i0');  
var flag = true;  
end.onclick = function()  
{   
if(flag==true)  
{  
flag=false;  
if (confirm("Ваш результат: "+count+" правильных ответов из 18 вопросов. Пройти тест заново?"))   
{  
location.reload();  
}   
else {  
document.getElementById("0").style.display = 'none';  
end.value = ('');  
end.value = ('Ваш результат: '+count+' правильных ответов из 18 ');  
}  
}  
}

Задаем переменную, в которой хранится элемент с кнопкой, к которому мы обратились по id, прописанному в HTML документе. Также задаем флаг, который отвечает за то, что кнопка еще не была нажата. При нажатии на кнопку, флаг активируется и, таким образом, нельзя сменить варианты ответов, а потом снова нажать на кнопку завершения. При нажатии на кнопку на экран выводится сообщение с количеством правильных ответов и вопросом, повторить ли тест. При положительном ответе, страница перезагружается, и тест можно пройти заново, при отрицательном ответе текст на кнопке сначала стирается, затем выводится новый текст с количеством правильных ответов. При этом мы обращаемся к главному блоку (заострял на нем внимание и писал, что он понадобится в конце), и меняем значение стиля «display» на «none», и этот блок просто убирается со страницы, что делает исправление вариантов ответов в вопросах невозможным. Также можно не убирать блок, а просто скрыть, меняя значение стиля блока «visibility» на «hidden», но тогда страница останется такой же громоздкой, что и с полноценным тестом, но будет иметь лишь табличку с результатом теста, поэтому, чтобы избавиться от громоздкости, был выбран первый вариант.

# Заключение

Для достижения данной цели, в процессе прохождения учебной практики (по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности) были рассмотрены различные языки программирования, определены их плюсы и минусы для того, чтобы выбрать самый оптимальный вариант реализации поставленной задачи.

В результате проделанной работы был написан интерактивный тест по электротехнике на языке программирования HTML, с подключением CSS и JavaScript.

В ходе прохождения практики все задачи были выполнены, цель достигнута.

# Список используемых источников

1. Работа с формами в Pascal - <http://pascalabc.net/wiki/index.php?title=FAQ_по_Windows_Forms>
2. C++ Window Forms - <https://code-live.ru/post/first-windows-form/>
3. C# Window Forms - <https://metanit.com/sharp/windowsforms/1.1.php>
4. Справочник по HTML, CSS, JavaScript - <http://htmlbook.ru>