

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Инженерно-экономический факультет

Кафедра экономической информатики

Дисциплина: Распределенные системы обработки информации

Индивидуальная практическая работа №1
на тему

«Создание веб-страниц с использованием языка разметки HTML
и каскадной таблицы стилей CSS»

Выполнил студент 5-го курса
группы 694051 специальности
«Электронный маркетинг»

Кузьмич Павел
Валерьевич

Проверил старший преподаватель
кафедры экономической
информатики

Атрощенко Натэлла
Александровна

Минск, 2020

Содержание

1. Описание HTML кода страницы.....	3
2. Описание CSS кода каскадных таблиц стилей.....	4
Выводы.....	8
Приложение А.....	9

1. Описание HTML кода страницы

Формат контента и название страницы, путь к favicon'у и его тип, а также путь к файлу с описанием каскадных таблиц CSS приведены в блоке `<head>` (см. листинг 1.1).

Листинг 1.1 Блок `<head>` HTML-страницы.

```
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="content.css">
  <link rel="icon" href="favicon.png" type="image/x-icon" />
  <title>Уровни тестирования</title>
</head>
```

Тело документа – блок `<body>` – состоит из одного блока `<div>`, описывающего строку итоговой таблицы, внутри которого описаны два `<div>`'а – левого и правого столбцов таблицы – которые заполнены контентом согласно заданию. Структура `<body>` приведена в листинге 1.2.

Листинг 1.2 Структура `<body>` HTML-страницы.

```
<body>
  <div class="row">
    <div class="column left">
      // содержание первой колонки
    </div>
    <div class="column right">
      // содержание второй колонки
    </div>
  </div>
</body>
```

Структура первой колонки задана с помощью тега таблицы `<table>` (см. листинг 1.3). Данная таблица разделена на три столбца и три строки, содержащих текстовую информацию, в данном случае с описанием уровней тестирования программного продукта.

Листинг 1.3 Структура первой колонки.

```
<table>
  <tr class="table_headers">
    <th></th>
    <th></th>
    <th></th>
  </tr>
```

Листинг 3. Продолжение.

```
<tr class="first_paragraph">
  <td class="split"></td>
  <td class="split"></td>
  <td></td>
</tr>
<tr class="second_paragraph">
  <td class="split"></td>
  <td class="split"></td>
  <td></td>
</tr>
</table>
```

Вторая колонка содержит три изображения и анимированную строку, которые заданы с помощью тегов <div>, и <p> соответственно. Код приведён в листинге номер 1.4.

Листинг 1.4 Структура второй колонки.

```
<div class="container">
  <div class="image1 image">
    
  </div>
  <div class="image2 image">
    
  </div>
  <div class="image3 image">
    
  </div>
</div>
<div style="animation-name: slide-right; animation-duration: 10s;">
  <p>Уровни тестирования ...</p>
</div>
```

Весь код HTML страницы представлен в приложении А.

2. Описание CSS кода каскадных таблиц стилей

Описание стиля, свойств расположения, толщины и цвета границ, а также фонового изображения и пути к нему приведены в листинге 2.1. Толщина границ таблицы задана в размере 1 пиксель, фоновое изображение во второй колонке задано из файла «background1.jpg» из корневой директории проекта. Параметр flex в размере 50% позволяет задать колонки одинаковой

ширины в рамках страницы. Класс «split» описывает разделить для текста в первой колонке.

Листинг 2.1 CSS для внешней таблицы.

```
.row {
    border: 1px solid #ADD8E6;
    display: flex;
    clear: both;
}
.column {
    float: left;
    border: 1px solid #ADD8E6;
}
.left {
    background-origin: padding-box;
    background-image: url('background1.jpg');
    flex: 50%;
}
.right {
    flex: 50%;
}
.split {
    border-right: 2px solid bisque;
}
```

Форматирование для разных абзацев текста, расположенного в первой колонке, включая его размер, тип и размер шрифт, а также расположение и дополнительные атрибуты, приведены в листинге 2.2.

Листинг 2.2 CSS для текста первой колонки.

```
.table_headers {
    text-align: center;
    font-family: Georgia, 'Times New Roman', Times, serif;
    font-size: 18px;
    font-style: italic;
    font-weight: bold;
    font-variant: small-caps;
    color: whitesmoke;
    text-shadow: 1px 1px gray;
}
.first_paragraph {
    text-indent: 10%;
    text-align: justify;
}
```

Листинг 2.2 Продолжение.

```
        font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande',  
                    'Lucida Sans', Arial, sans-serif;  
        font-size: 15px;  
        color: black;  
    }  
    .second_paragraph {  
        text-indent: 10%;  
        text-align: justify;  
        font-style: italic;  
        font-family: Verdana, Geneva, Tahoma, sans-serif;  
        font-size: 14px;  
        color: white;  
        background-color: rgba(0, 0, 0, 0.15);  
        text-shadow: 0.5px 0.5px black;  
    }
```

Для различного форматирования изображений во второй колонке используется CSS код, описанный в листинге 2.3. Для расположения изображений по возрастанию от нижнего левого угла используются атрибуты «bottom» и «left», прозрачность и толщина, цвет границ заданы с помощью атрибутов «opacity» и «border».

Листинг 2.3 CSS для второй колонки.

```
.container {  
    height: 550px;  
    width: 900px;  
    position: relative;  
}  
.container>.image {  
    position: absolute;  
}  
.image1 {  
    bottom: 3rem;  
    left: 2rem;  
    border: 3px solid lightcyan;  
    opacity: 0.7;  
}  
.image2 {  
    bottom: 10rem;  
    left: 17rem;  
    border: 2px solid lightblue;  
    opacity: 0.8;  
}
```

Листинг 2.3 Продолжение.

```
.image3 {  
    bottom: 17rem;  
    left: 27rem;  
    border: 1px solid lightskyblue;  
    opacity: 0.9;  
}
```

Для создания анимированной строки, передвигающейся от левого края блока к правому используется аннотация «@keyframes». При этом атрибут «margin-left» в начальной точке задан в размере 5%, в конечной – 75%. Код приведён в листинге 2.4.

Листинг 2.4 CSS для анимации.

```
@keyframes slide-right {  
    from {  
        margin-left: 5%;  
        color: white;  
    }  
    to {  
        margin-left: 75%;  
        color: mediumblue;  
    }  
}
```

Исходный код был протестирован в двух браузерах под операционной системой Linux (ubuntu 20.04) – Firefox, версии 81.0, результат представлен на рисунке 1, и Chromium, версии 85.0.4183.121, результат представлен на рисунке 2.

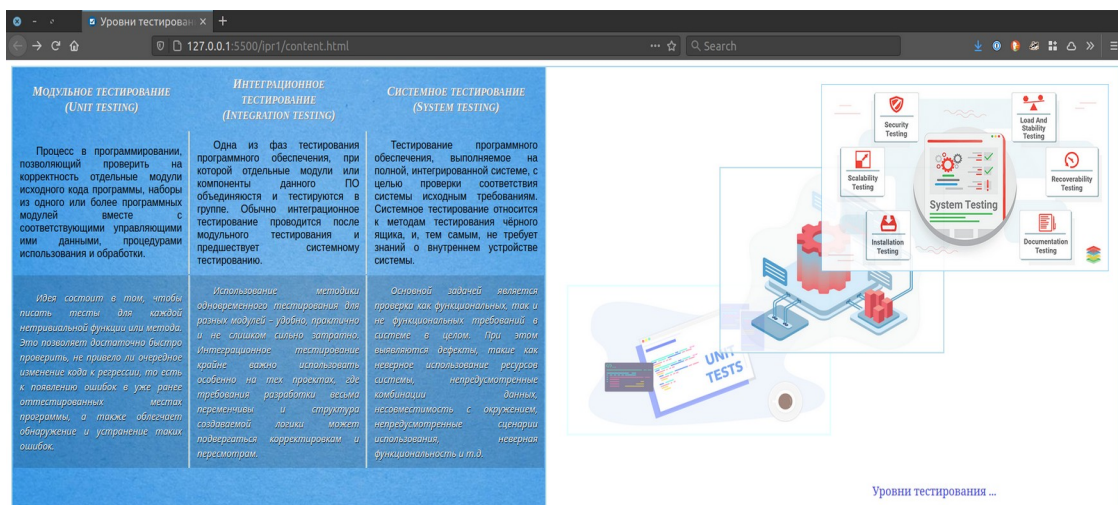


Рисунок 1 — Скриншот проекта в Mozilla Firefox.



Рисунок 2 — Скриншот проекта в Chromium Browser.

Выводы

В результате выполнения практического задания был разработан проект, содержащий файлы content.html и content.css. С помощью функциональных возможностей языков HTML и CSS была создана страница, содержащая текстовую и графическую информацию, оформленную с помощью каскадных таблиц стилей.

Приложение А

Листинг кода из файла content.html

```
<html>
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="content.css">
  <link rel="icon" href="favicon.png" type="image/x-icon" />
  <title>Уровни тестирования</title>
</head>
<body>
  <div class="row">
    <div class="column left">
      <table>
        <tr class="table_headers">
          <th>Модульное тестирование<br>(Unit testing)</th>
          <th>Интеграционное тестирование<br>(Integration testing)</th>
          <th>Системное тестирование<br>(System testing)</th>
        </tr>
        <tr class="first_paragraph">
          <td class="split">Процесс в программировании, позволяющий
проверить на корректность отдельные модули исходного кода программы,
наборы из одного или более программных модулей вместе с
соответствующими управляющими ими данными, процедурами
использования
и обработки.
          </td>
          <td class="split">Одна из фаз тестирования программного обеспечения,
при которой отдельные модули или компоненты данного ПО объединяются и
тестируются в группе. Обычно интеграционное тестирование проводится
после модульного тестирования и предшествует
системному тестированию.
          </td>
          <td>Тестирование программного обеспечения, выполняемое на полной,
интегрированной системе, с целью проверки соответствия системы исходным
требованиям. Системное тестирование относится к методам тестирования
чёрного ящика, и, тем самым, не требует знаний о
внутреннем устройстве системы.
          </td>
        </tr>
        <tr class="second_paragraph">
          <td class="split">Идея состоит в том, чтобы писать тесты для каждой
нетривиальной функции или метода. Это позволяет достаточно быстро
```

проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже ранее оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

</td>

<td class="split">Использование методики одновременного тестирования для разных модулей – удобно, практично и не слишком сильно затратно. Интеграционное тестирование крайне важно использовать особенно на тех проектах, где требования разработки весьма переменчивы и структура создаваемой логики может подвергаться корректировкам и пересмотрам.

</td>

<td>Основной задачей является проверка как функциональных, так и нефункциональных требований в системе в целом. При этом выявляются дефекты, такие как неверное использование ресурсов системы, непредусмотренные комбинации данных, несовместимость с окружением, непредусмотренные сценарии использования, неверная функциональность и т.д.

</td>

</tr>

</table>

</div>

<div class="column right">

<div class="container">

<div class="image1 image">

</div>

<div class="image2 image">

</div>

<div class="image3 image">

</div>

</div>

<div style="animation-name: slide-right; animation-duration: 10s;">

<p>Уровни тестирования ...</p>

</div>

</div>

</div>

</body>

</html>