

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Современные технологии
программирования»
для направления подготовки 09.03.02
«Информационные системы и технологии»

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
Тема 1. Введение в web-технологии: структура и принципы Web	
ЛАБОРАТОРНАЯ РАБОТА 1. HTML5. Работа с Web-	
формами	10
Тема 1. Введение в web-технологии: структура и принципы Web	
ЛАБОРАТОРНАЯ РАБОТА 2. HTML 5. Drag and Drop.	
Оформление страницы в HTML 5. Мультимедиа	34
Тема 1. Введение в web-технологии: структура и принципы Web	
ЛАБОРАТОРНАЯ РАБОТА 3. HTML 5. Работа с графикой.	
Canvas. Анимация	49

ВВЕДЕНИЕ

Целью изучения дисциплины «Современные технологии программирования» является формирование навыков по разработке, документированию и сопровождению сетевых приложений, расширение профессионального кругозора студентов, повышение программистской культуры. формирование набора общекультурных, общепрофессиональных и профессиональных компетенций будущего магистра по направлению подготовки 09.03.02 «Информационные системы и технологии».

Задачами изучения дисциплины «Современные технологии программирования» являются

- 1) формирование представлений об основных принципах функционирования Internet-приложений и навыков их разработки,
- 2) изучение основных подходов, платформ, технологий и инструментов проектирования Internet-приложений;
- 3) формирование набора профессиональных компетенций.

Тема 1. Введение в web-технологии: структура и принципы Web

ЛАБОРАТОРНАЯ РАБОТА 1

HTML5. РАБОТА С WEB-ФОРМАМИ

Цель работы: познакомить студентов с базовыми приемами создания Web-страниц средствами HTML, CSS и JavaScript, инструментальным средством WebMatrix; основами работы с Web-формами: создание Web-форм; стилизация форм; валидация вводимых значений; задание маски ввода.

Формируемые компетенции или их части:

1. ОПК-5 – владением методами и средствами получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе, в глобальных компьютерных сетях.

2. ПК-3 – умением разрабатывать новые технологии проектирования информационных систем.

3. ПК-15 – способностью разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач.

Теоретическая часть

Алгоритм установки WebMatrix может быть представлен следующей последовательностью действий:

1. Загрузите установщик по ссылке

<http://www.microsoft.com/web/gallery/install.aspx?appid=webmatrix>.

2. Запустите установщик.

3. Ознакомьтесь с параметрами установки и устанавливаемыми компонентами.

4. Для дальнейшей работы создайте сайт на основе шаблона (рисунок 1.1), а именно на основе шаблона "Пустой сайт" (рисунок – 1.2).

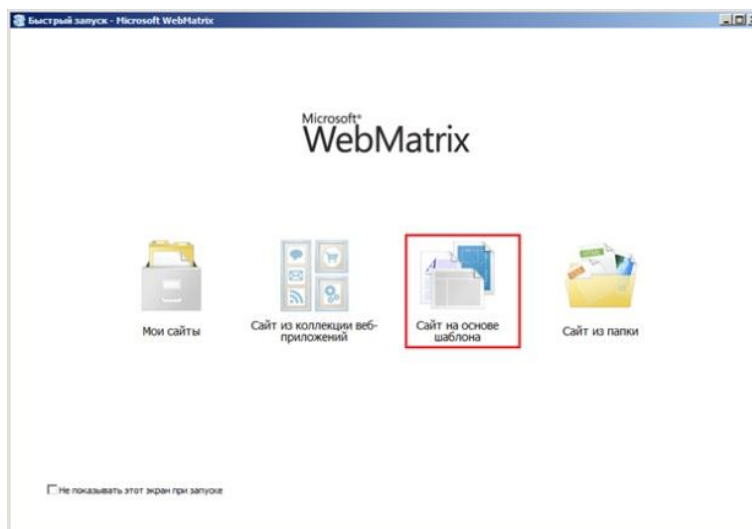


Рисунок 1.1 – Создание сайта на основе шаблона

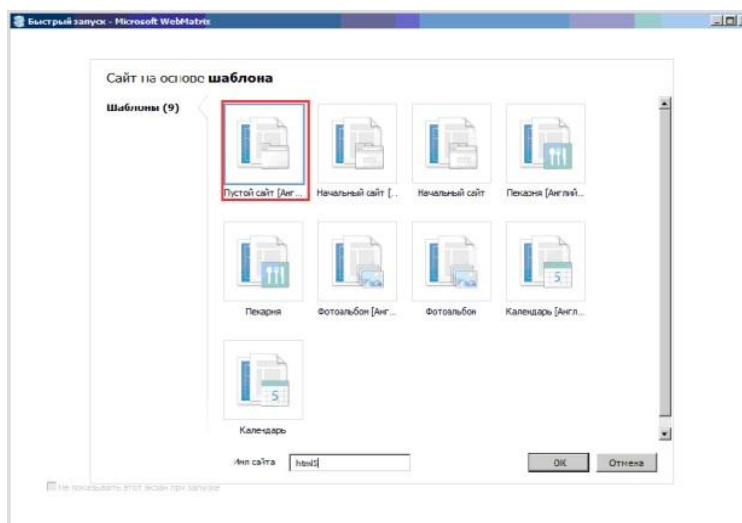


Рисунок 1.2 – Создание сайта на основе пустого шаблона

5. После окончания процесса создания проекта сайта, перейдите к разделу **Файлы** (рисунок 1.3) и создайте папку для создаваемого сайта (рисунок 1.4).

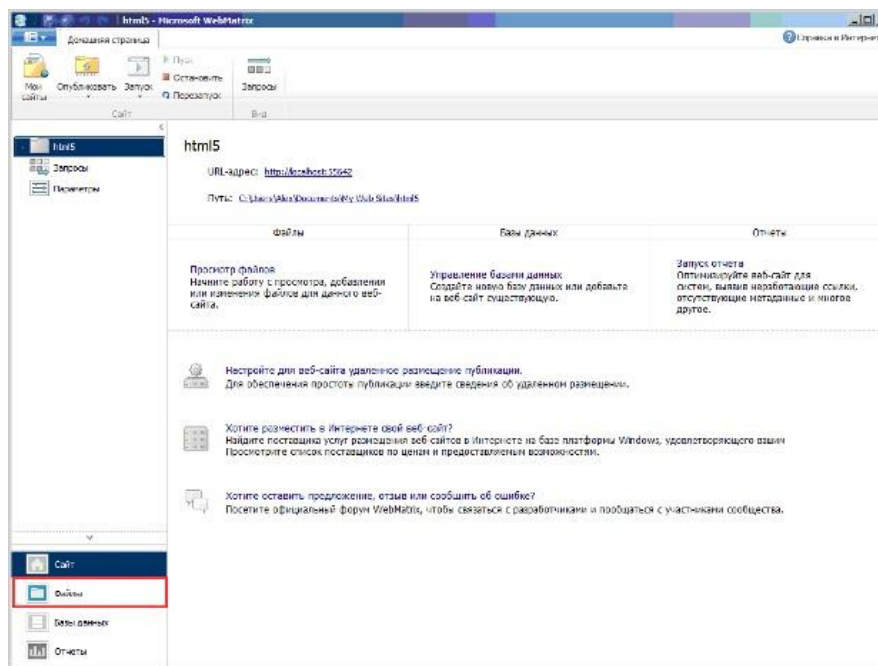


Рисунок 1.3 – Переход к просмотру и изменению файлов сайта

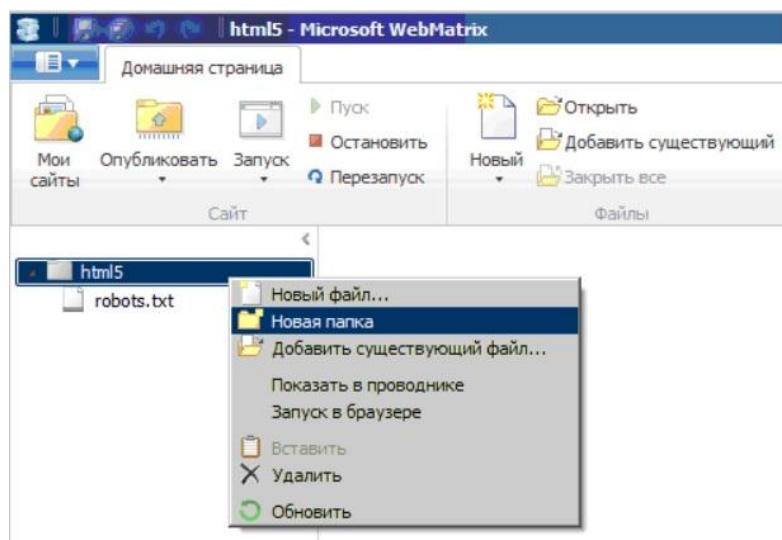


Рисунок 1.4 – Создание новой папки

Создание Web- страницы с подгружаемым содержимым

В качестве примера рассмотрим создание страницы с подгружаемым содержимым, например, теоретический материал к одной из лекций курса

Для удобства необходимо создать еще несколько папок, которые будут использоваться для хранения стилей, Web- сценариев и, собственно, для тех страниц, содержимое которых будем подгружать (рисунок 1.5).

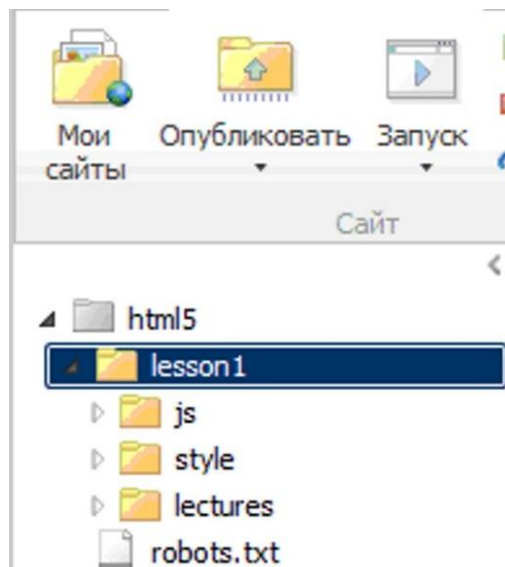


Рисунок 1.5 – Иерархия каталогов
Создание страниц - содержимого

Для начала создадим несколько Web-страниц, содержимое которых, затем будем погрузать в основную Web - страницу.

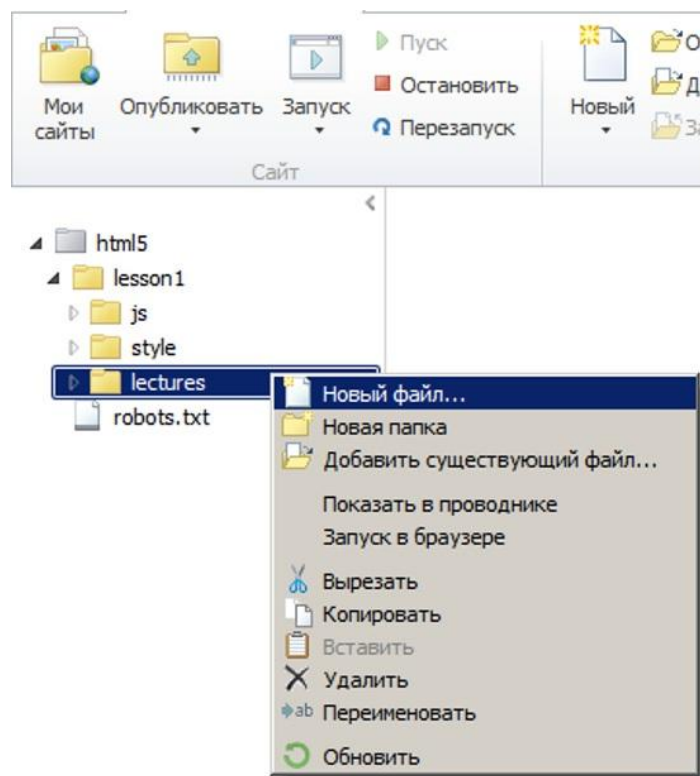


Рисунок 1.6 – Добавление нового файла в папку

В открывшемся окне, выберем тип создаваемого файла (рисунок 1.7)

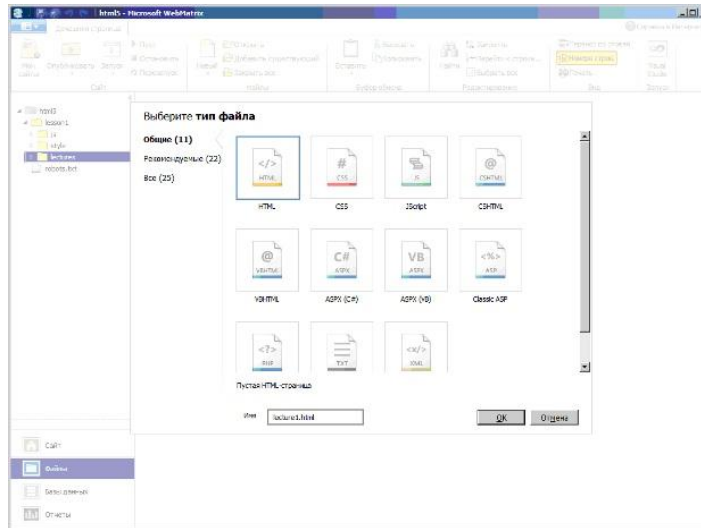


Рисунок 1.7 – Выбор типа создаваемого файла

Создание стилей содержимого

Для управления содержимым создадим CSS - файл:

```
h1, h2 { text-align:center; }
```

```
.text { text-align:justify}
```

```
.annotation { font-style:italic;}
```

```
table.normal { border: 1px solid black; border-right: none; border-bottom: none }
```

```
table.normal tr td { border-bottom: 1px solid black; border-right: 1px solid black }
```

```
table.normal thead { text-align:center; font-weight:bold }
```

```
table.normal caption { text-align:right }
```

```
dt.terms { font-style:italic; font-weight:bold }
```

Создание основной страницы

Создадим основную *HTML* страницу для выполнения задания. Для этого, во-первых, на страницу необходимо добавить четыре контейнера, во-вторых, создать *css-файл* для управления размещением контейнеров.

С добавлением контейнеров на страницу все просто:

```
<div id="header">HEADER</div>
```

Также не должно быть сложностей и с их позиционированием относительно друг друга:

```
#top { height: 15%; width:100%; }
```

Реализация подгрузки содержимого

Осталось немного, а именно: создать полосу навигации и организовать подгрузку содержимого в блок *content*.

Полоса навигации, к примеру может быть создана следующим образом:

```
<div id="navigation"><a href="javascript:/" on-  
click="loadContent('lectures/lecture1.html')" > Лекция № 1</a></div>
```

Теперь необходимо поместить элемент *iframe* в контейнер *content*:

```
<div id="content"><iframe id="target" src=""></iframe></div>
```

И, наконец, создать *JavaScript* - файл, в котором разместим функцию, обрабатывающую клик по ссылке:

```
function loadContent(path)
{
  document.getElementById('target').src = path;
}
```

Для завершения задания необходимо:

- создать несколько страниц – источников;
- оформить панель навигации;
- создать стили для оформления содержимого контейнеров.

Создание Web-форм

По аналогии с предыдущим практическим заданием создадим папку для стилей и основную *HTML* - страницу.

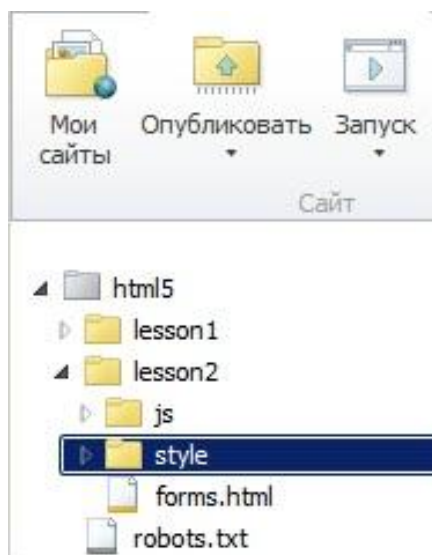


Рисунок – 1.8 – Иерархия каталогов

Рассмотрим по шагам создание требуемой страницы с регистрационной формой.

Шаг 1. Создание формы регистрации

```
<form id="registration">
  <fieldset>
    <legend>Форма регистрации</legend>
  </fieldset>
</form>
```

Тег `<fieldset>` используется для логической группировки объектов формы. Тег `<legend>` определяет заголовок.

Шаг 2. Добавление поля для ввода ФИО пользователя:

```
<fieldset>
  <legend>Форма регистрации</legend>
  <label for=name>ФИО</label>
  <input id=name name=name type=text>
</fieldset>
```

На данном шаге форма будет выглядеть в браузере следующим образом:

Форма регистрации

ФИО

Рисунок 1.9 – Промежуточный результат на шаге 2

Шаг 3. Создание поля для ввода *@-mail* и номера телефона, почтового индекса и города:

```
<fieldset>
  <legend>Форма регистрации</legend>
  <label for=name>ФИО</label>
  <input id=name name=name type=text>
  <label for=email>Email</label>
  <input id=email name=email type=email>
  <label for=phone>Номер телефона</label>
  <input id=phone name=phone type=tel>
</fieldset>
```

Форма регистрации

ФИО Email Номер телефона

Рисунок 1.10 – Промежуточный результат на шаге 2

Шаг 4. Для выравнивания элементов формы построчно и относительно друг друга необходимо создать таблицу с двумя колонками и без видимых границ внесем изменения в код следующим образом:


```
<fieldset>
  <legend>Форма регистрации</legend>
  <table class="alignment">
    <tr>
      <td> <label for=name>ФИО</label> </td>
      <td> <input id=name name=name type=text> </td>
    </tr>
    <tr>
```

```

<td> <label for=email>Email</label> </td>
<td> <input id=email name=email type=email> </td>
</tr>
<tr>
<td> <label for=phone>Номер телефона</label> </td>
<td> <input id=phone name=phone type=tel> </td>
</tr>
</table>
</fieldset>

```

В результате получим следующее:



Форма регистрации

ФИО	<input type="text"/>
Email	<input type="text"/>
Номер телефона	<input type="text"/>

Рисунок 1.11 – Промежуточный результат на шаге 4

Шаг 5. Добавление поля для ввода даты рождения с помощью `type = date` (работает не со всеми браузерами). Для этого к таблице добавим следующую строку:

```

<tr>
<td><label for=dateofbirth>Дата рождения</label></td>
<td><input id=dateofbirth name=dateofbirth type="date" /></td>
</tr>

```

В случае, если браузер поддерживает данный тип элементов управления, то получим следующее:

Форма регистрации

ФИО

Email

Номер телефона

Город

Дата рождения Day.Month.Year ▾

<< < Май 2002 ▾ > >>

Пн	Вт	Ср	Чт	Пт	Сб	Вс
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Today Clear

Рисунок 1.12 – Промежуточный результат на шаге 5

Шаг 6. Добавление элемента *fieldset* для логической группировки, объединяющий элементы, указывающие адрес и почтовый индекс:

```

<fieldset>
  <legend>Параметры доставки</legend>
  <table>
    <tr>
      <td><label for=address>Адрес</label></td>
      <td><textarea id=address name=address rows=5 ></textarea></td>
    </tr>
    <tr>
      <td><label for=postcode>Почтовый индекс</label></td>
      <td><input id=postcode name=postcode type=text ></td>
    </tr>
  </table>
</fieldset>

```

Поскольку написание полного адреса, скорее всего займет не одну строку, зададим атрибут `rows`, со значением равным 5 для соответствующего тега.

Форма регистрации	
ФИО	<input type="text"/>
Email	<input type="text"/>
Номер телефона	<input type="text"/>
Город	<input type="text"/>
Дата рождения	<input type="text"/>
Параметры доставки	
Адрес	<input rows="5" type="text"/>
Почтовый индекс	<input type="text"/>

Рисунок 1.13 – Промежуточный результат на шаге 6

Шаг 7. Выделение в отдельную логическую группу элементов управления, с помощью которого пользователь укажет предпочитаемый способ получения уведомлений:

```
<fieldset>
```

```
<legend>Предпочитаемый способ получения уведомлений</legend>
```

```
<table>
```

```
<tr>
```

```
<td><input id=emailmessage name=emailmessage type=radio></td>
```

```
<td><label for=emailmessage>По Email</label></td>
```

```
</tr>
```

```
<tr>
```

```
<td><input id=phonemessage name=phonemessage type=radio></td>
```

```
<td><label for=phonemessage>По телефону</label></td>
```

```
</tr>
```

```

<tr>
  <td><input id=nomessagee name=nomessage type=radio></td>
  <td><label for=nomessage>Не уведомлять меня</label></td>
</tr>
</table>
</fieldset>

```

The image shows a web form with three distinct sections, each enclosed in a box with a title bar:

- Форма регистрации (Registration Form):** Contains five input fields:
 - ФИО (Full Name)
 - Email
 - Номер телефона (Phone Number)
 - Город (City)
 - Дата рождения (Date of Birth)
- Параметры доставки (Delivery Parameters):** Contains two input fields:
 - Адрес (Address): A large text area with a vertical scrollbar.
 - Почтовый индекс (Postal Index): A standard text input field.
- Предпочитаемый способ получения уведомлений (Preferred notification method):** Contains three radio button options:
 - По Email
 - По телефону
 - Не уведомлять меня

Рисунок 1.14 – Промежуточный результат на шаге 7

Шаг 8. Добавление на страницу кнопки

```

<fieldset>
  <button type=submit>Отправить данные</button>
</fieldset>

```

Форма регистрации

ФИО

Email

Номер телефона

Город

Дата рождения

Параметры доставки

Адрес

Почтовый индекс

Предпочитаемый способ получения уведомлений

По Email

По телефону

Не уведомлять меня

Рисунок 1.15 – Промежуточный результат на шаге 8

Стилизация форм

Шаг 1. Создание стиля для всей формы

```
form#registration {
  background: #1E90FF;
  - moz-border-radius: 5px;
  - webkit-border-radius: 5px;
  - khtml-border-radius: 5px;
  border-radius: 5px;
  counter-reset: fieldsets;
  padding: 20px;
  width: 400px;
```

Шаг 2. Удаление границы у *fieldset* и добавление отступа

```
form#registration fieldset {
  border: none;
  margin-bottom: 10px;
```

Шаг 3. Стилизация legends:

```
form#registration legend {
  color: #384313;
  font-size: 16px;
  font-weight: bold;
  padding-bottom: 10px;
  text-shadow: 0 1px 1px #C1F7FF;
```

Шаг 4. Стилизация элементы label, input и button. Все label должны выглядеть одинаково, кроме поля label, которое используется для элементов radio. Выравниваем их по левому краю и придадим ширину:

```
form#registration label {
  float: left;
  font-size: 13px;
  width: 110px;
}

form#registration fieldset fieldset label {
  background:none no-repeat left 50%;
  line-height: 20px;
  padding: 0 0 0 30px;
  width: auto;
}

  form#registration button {
  background: #87CEEB;
  border: none;
  - moz-border-radius: 20px;
  - webkit-border-radius: 20px;
  - khtml-border-radius: 20px;
```

```
border-radius: 20px;
color: #ffffff;
display: block;
font: 18px Georgia, "Times New Roman", Times, serif;
letter-spacing: 1px;
margin: auto;
padding: 7px 25px;
text-shadow: 0 1px 1px #C1F7FF;
text-transform: uppercase;
```

Валидация вводимых значений

Рассмотрим следующий участок *HTML* кода:

```
<input id=email name=email type=email />
```

Атрибут типа равен «@-mail», а не «text». Самое ценное в новых HTML-типах input в том, что вы можете использовать их сейчас и они будут работать на том или ином уровне в любом браузере. Когда браузер встречает один из этих типов, происходит одно из двух.

Если браузер не поддерживает новые input-типы, объявление типа не распознается. В таком случае браузер корректно сокращает функциональность и интерпретирует элемент как type = "text" .

Чтобы сделать элемент управления обязательным к заполнению, достаточно вставить в создающий его тег атрибут REQUIRED. Это атрибут тега без значения.

```
<input id=name name=name type=text required>
```

Задать минимальное и максимальное значение и шаг числовых значений можно только для полей ввода числовых величин. Другие элементы управления, в том числе и обычные поля вида, эту возможность не поддерживают.

Для задания минимального значения числа используется атрибут тега MIN, а для задания максимального значения – атрибут тега MAX. В качестве их значений указываются числа.

```
<INPUT TYPE=" number" ID=" txtAge" REQUIRED MIN=" 1" MAX=" 100" >
```

Задание маски ввода

Маска ввода задаёт формат, которому должно соответствовать вводимое значение. Как правило, она указывается для обычных полей ввода.

Для указания маски используется атрибут стиля PATTERN. В качестве его значения указывается регулярное выражение, собственно, и задающее маску ввода.

```
<input id=phone name=phone type=tel pattern=" (d{3,4}) d{2,3}-d{2}-d{2}" >
```

Эта Web-форма содержит поле ввода, в котором указывается номер телефона в формате ([x]xxx) [x]xx-xx-xx, где x- цифра.

Если посетитель введёт в поле ввода с указанной маской значение, не соответствующее данной маске, то увидит всплывающее сообщение с предупреждением о неверном формате введенного значения.

Для окончания формирования формы необходимо:

- поля ФИО, @-mail и номер телефона, определить, как обязательные к заполнению;
- создать маску ввода для поля «Номер телефона»;
- осуществить проверку вводимых значений поля «Почтовый индекс»;
- ознакомиться с материалами для самостоятельного изучения.

Создание Web-форм

Рассмотрим по шагам создание требуемой страницы с регистрационной формой.

Шаг 1. Создание формы регистрации

```
<form id="registration">
  <fieldset>
    <legend>Форма регистрации</legend>
  </fieldset>
</form>
```

Тег <fieldset> используется для логической группировки объектов формы. Тег <legend> определяет заголовок.

Шаг 2. Добавление поля для ввода ФИО пользователя:

```
<fieldset>
  <legend>Форма регистрации</legend>
  <label for=name>ФИО</label>
  <input id=name name=name type=text>
</fieldset>
```

Шаг 3. Создание поля для ввода @-mail и номера телефона, почтового индекса и города:

```
<fieldset>
  <legend>Форма регистрации</legend>
  <label for=name>ФИО</label>
  <input id=name name=name type=text>
  <label for=email>Email</label>
  <input id=email name=email type=email>
  <label for=phone>Номер телефона</label>
  <input id=phone name=phone type=tel>
</fieldset>
```

Шаг 4. Для выравнивания элементов формы построчно и относительно друг друга необходимо создать таблицу с двумя колонками и без видимых границ внесем изменения в код следующим образом:

```
<fieldset>
  <legend>Форма регистрации</legend>
  <table class="alignment">
    <tr>
      <td <label for=name>ФИО</label> </td>
      <td <input id=name name=name type=text> </td>
    </tr>
    <tr>
      <td <label for=email>Email</label> </td>
      <td <input id=email name=email type=email> </td>
```

```

</tr>
<tr>
  <td> <label for=phone>Номер телефона</label> </td>
  <td> <input id=phone name=phone type=tel> </td>
</tr>
</table>
</fieldset>

```

Шаг 5. Добавление поля для ввода даты рождения с помощью `type = date` (работает не со всеми браузерами). Для этого к таблице добавим следующую строку:

```

<tr>
  <td><label for=dateofbirth>Дата рождения</label></td>
  <td><input id=dateofbirth name=dateofbirth type="date" /></td>
</tr>

```

Шаг 6. Добавление элемента *fieldset* для логической группировки, объединяющий элементы, указывающие адрес и почтовый индекс:

```

<fieldset>
  <legend>Параметры доставки</legend>
  <table>
    <tr>
      <td><label for=address>Адрес</label></td>
      <td><textarea id=address name=address rows=5 ></textarea></td>
    </tr>
    <tr>
      <td><label for=postcode>Почтовый индекс</label></td>
      <td><input id=postcode name=postcode type=text ></td>
    </tr>
  </table>
</fieldset>

```

Поскольку написание полного адреса, скорее всего займет не одну строку, зададим атрибут rows, со значением равным 5 для соответствующего тега.

Шаг 7. Выделение в отдельную логическую группу элементов управления, с помощью которого пользователь укажет предпочитаемый способ получения уведомлений:

```
<fieldset>
<legend>Предпочитаемый способ получения уведомлений</legend>
<table>
  <tr>
    <td><input id=emailmessage name=emailmessage type=radio></td>
    <td><label for=emailmessage>По Email</label></td>
  </tr>
  <tr>
    <td><input id=phonemessage name=phonemessage type=radio></td>
    <td><label for=phonemessage>По телефону</label></td>
  </tr>
  <tr>
    <td><input id=nomessagee name=nomessage type=radio></td>
    <td><label for=nomessage>Не уведомлять меня</label></td>
  </tr>
</table>
</fieldset>
```

Шаг 8. Добавление на страницу кнопки

```
<fieldset>
  <button type=submit>Отправить данные</button>
</fieldset>
```

Стилизация форм

Шаг 1. Создание стиля для всей формы

```
form#registration {
```

```

background: #1E90FF;
- moz-border-radius: 5px;
- webkit-border-radius: 5px;
- khtml-border-radius: 5px;
border-radius: 5px;
counter-reset: fieldsets;
padding: 20px;
width: 400px;

```

Шаг 2. Удаление границы у *fieldset* и добавление отступа

```

form#registration fieldset {
border: none;
margin-bottom: 10px;

```

Шаг 3. Стилизация legends:

```

form#registration legend {
color: #384313;
font-size: 16px;
font-weight: bold;
padding-bottom: 10px;
text-shadow: 0 1px 1px #C1F7FF;

```

Шаг 4. Стилизация элементы label, input и button. Все label должны выглядеть одинаково, кроме поля label, которое используется для элементов radio. Выравниваем их по левому краю и придадим ширину:

```

form#registration label {
float: left;
font-size: 13px;
width: 110px;
}
form#registration fieldset fieldset label {
background:none no-repeat left 50%;
line-height: 20px;

```

```
padding: 0 0 0 30px;
width: auto;
}
form#registration button {
background: #87CEEB;
border: none;
- moz-border-radius: 20px;
- webkit-border-radius: 20px;
- khtml-border-radius: 20px;
border-radius: 20px;
color: #ffffff;
display: block;
font: 18px Georgia, "Times New Roman", Times, serif;
letter-spacing: 1px;
margin: auto;
padding: 7px 25px;
text-shadow: 0 1px 1px #C1F7FF;
text-transform: uppercase;
```

Валидация вводимых значений

Рассмотрим следующий участок *HTML* кода:

```
<input id=email name=email type=email />
```

Атрибут типа равен «@-mail», а не «text». Самое ценное в новых HTML-типах `input` в том, что их можно использовать сейчас и они будут работать на том или ином уровне в любом браузере. Когда браузер встречает один из этих типов, происходит одно из двух.

Если браузер не поддерживает новые `input`-типы, объявление типа не распознается. В таком случае браузер корректно сокращает функциональность и интерпретирует элемент как `type = "text"` .

Чтобы сделать элемент управления обязательным к заполнению, достаточно вставить в создающий его тег атрибут `REQUIRED`. Это атрибут тега без значения.

```
<input id=name name=name type=text required>
```

Задать минимальное и максимальное значение и шаг числовых значений можно только для полей ввода числовых величин. Другие элементы управления, в том числе и обычные поля вида, эту возможность не поддерживают.

Для задания минимального значения числа используется атрибут тега `MIN`, а для задания максимального значения – атрибут тега `MAX`. В качестве их значений указываются числа.

```
<INPUT TYPE=" number" ID=" txtAge" REQUIRED MIN=" 1" MAX=" 100" >
```

Задание маски ввода

Маска ввода задаёт формат, которому должно соответствовать вводимое значение. Как правило, она указывается для обычных полей ввода.

Для указания маски используется атрибут стиля `PATTERN`. В качестве его значения указывается регулярное выражение, собственно, и задающее маску ввода.

```
<input id=phone name=phone type=tel pattern=" (d{3,4}) d{2,3}-d{2}-d{2}" >
```

Эта Web-форма содержит поле ввода, в котором указывается номер телефона в формате `([x]xxx) [x]xx-xx-xx`, где `x` - цифра.

Если посетитель введёт в поле ввода с указанной маской значение, не соответствующее данной маске, то увидит всплывающее сообщение с предупреждением о неверном формате введенного значения.

Оборудование и материалы. Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 2000 МГц и выше; оперативная память – не менее 1024 Мбайт; свободное дисковое пространство – не менее 1,2 Гбайт; устройство для чтения компакт-дисков; монитор типа Super VGA (число цветов – 256) с диагональю не менее 17 ". **Программное обеспечение** –

операционная система WINDOWS XP и выше, программы для просмотра Web-страниц.

Указания по технике безопасности. Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для работы с персональным компьютером. Самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения. В случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору). Соблюдать правила техники безопасности при работе с электрооборудованием. Не касаться электрических розеток металлическими предметами. Рабочее место пользователя персонального компьютера должно содержаться в чистоте. Не разрешается возле персонального компьютера принимать пищу, напитки.

Задания

Задание 1. Подготовить рабочее место для выполнения практических занятий.

Задание 2. Создать Web-страницу на основе контейнерного дизайна.

Задание 3. Создайте страницу регистрации с Web-формами, по образцу многочисленных регистрационных Web-страниц, вида представленного на рисунке.

Необходимо также создать соответствующие стили для оформления Web-формы и валидацию вводимых значений.

Для окончания формирования формы необходимо:

- поля ФИО, @-mail и номер телефона, определить, как обязательные к заполнению;
- создать маску ввода для поля «Номер телефона»;
- осуществить проверку вводимых значений поля «Почтовый индекс».

<i>ФИО</i>	<input type="text"/>
<i>Email</i>	<input type="text"/>
<i>Телефон</i>	<input type="text"/>
<i>Почтовый индекс</i>	<input type="text"/>
<i>Адрес</i>	<input type="text"/>
<i>Дата</i>	<input type="text"/>
<i>Предпочитаемый способ получения уведомлений</i>	<input type="radio"/> <i>По телефону</i> <input type="radio"/> <i>email</i>
<input type="button" value="Отправить данные"/>	

Указания по порядку выполнения работы

При выполнении заданий к лабораторной работе использовать теоретический материал и практические примеры, приведенные в нем.

Содержание отчета

Отчет по лабораторной работе должен состоять из:

- 1) названия лабораторной работы;
- 2) ответов на контрольные вопросы;
- 3) формулировки индивидуальных заданий и порядка их выполнения.

Контрольные вопросы

1. Каким образом можно установить WebMatrix?
2. Дайте характеристику алгоритмам создания «страниц с подгружаемым содержимым», «страниц – содержимого».
3. Охарактеризуйте процесс создания стилей содержимого.
4. Алгоритм создания основной страницы.
5. Каким образом можно создать Web-форму в HTML 5?
6. Охарактеризуйте процесс стилизации (оформления) форм.

7. С какой целью используется валидация вводимых значений?
8. Каким образом можно задать маску ввода в Web-форме?

СПИСОК ЛИТЕРАТУРЫ: основная – 2, 3, 4, 5; дополнительная – 2, 3.

Тема 1. Введение в web-технологии: структура и принципы Web

ЛАБОРАТОРНАЯ РАБОТА 2

HTML 5. DRAG AND DROP. ОФОРМЛЕНИЕ СТРАНИЦЫ В

HTML 5. МУЛЬТИМЕДИА

Цель работы: является формирования навыков работы с методами и элементами разметки, позволяющими менять местоположение любых элементов страницы при помощи мыши, на основе событийного механизма и *JavaScript API*, определенных спецификацией *HTML 5*; сформировать базовые навыки работы с элементами `<audio>` и `<video>`.

Формируемые компетенции или их части:

1. ОПК-5 – владением методами и средствами получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе, в глобальных компьютерных сетях.
2. ПК-3 – умением разрабатывать новые технологии проектирования информационных систем.
3. ПК-15 – способностью разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач.

Теоретическая часть

Создадим два контейнера, в один из которых (источник) поместим три элемента, элементы можно будет свободно перемещать из контейнера – источника, в целевой контейнер и обратно.

Для этого необходимо:

- создать контейнеры и элементы для перемещения;
- стилизовать элементы Web-страницы;

- создать функции для обработки следующих событий: перетаскивание объекта внутрь границ элемента; прохождение курсора "над" элементом во время осуществления операции перемещения; "освобождение" перетаскиваемого элемента в пределах элемента – цели; начало операции перемещения; окончание операции перетаскивания.

Создание элементов страницы для перемещения

Для того, чтобы можно было перемещать элемент в рамках страницы, достаточно просто добавить атрибут `draggable = true`. Создадим контейнеры для размещения элементов и элементы для последующего перемещения:

```
<div id="source" class="container" >
  <div id="firstDragElement" class="element" draggable="true">Text
1</div>
  <div id="secondDragElement" class="element" draggable="true">Text
2</div>
  <div id="thirdDragElement" class="element" draggable="true">Text
3</div>
</div>
<div id="target" class="container"></div>
```

Стилизация элементов

Для простоты восприятия примера, создадим следующие стили для элементов:

```
.element {
border: 2px solid black;
height: 50px;
width: 50px;
margin-left:20px;
margin-bottom: 10px;
text-align:center;
}
.container {
```

```
border: 2px solid red;
height: 200px;
width: 100px;
float:left;
margin: 50px;
```

Создание JavaScript-функции

Отслеживать процесс *Drag and drop* позволяют следующие события: dragstart; drag; dragenter; dragleave; dragover; drop; dragend.

События dragenter, dragover и dragleave можно использовать для того, чтобы сделать процесс переноса более наглядным.

Для обработки процесса бросания элемента, надо назначить обработчик событий drop и dragend. В нем надо отменить поведение браузера по умолчанию, т.к. это может быть переход на другую страницу.

Объект DataTransfer хранит данные отсылаемые в процессе перетаскивания. DataTransfer задается в событии dragset и используется в событии drop. Вызов e.dataTransfer.setData(format, data) устанавливает mimetype и данные нужные для перетаскивания.

Свойство dataTransfer имеет функцию getData(format) для получения сохраненных данных.

Свойство dataTransfer предоставляет возможность тонкой настройки отображения процесса переноса:

- dataTransfer.effectAllowed – эффект, поддерживаемый целевым элементом перетаскивания. Как правило, это значение задается обработчиком события dragstart. Может принимать следующие значения: none, copy, copyLink, copyMove, link, linkMove, move, all и uninitialized;

- dataTransfer.dropEffect – эффект, выбранный пользователем или целевым элементом. Может принимать следующие значения: none, copy, link, move;

- dataTransfer.setDragImage(i, x, y). Вместо того чтобы использовать при перетаскивании полупрозрачную картинку по умолчанию

Алгоритм создания функций:

```
function dragStart(ev)
{
    ev.dataTransfer.effectAllowed='link';
    ev.dataTransfer.setData("Text", ev.target.getAttribute('id'));
    ev.dataTransfer.setDragImage(ev.target,0,0);
    return true;
}

function dragEnter(ev)
{
    var idelt = ev.dataTransfer.getData("Text");
    return true;
}

function dragOver(ev)
{
    var idelt = ev.dataTransfer.getData("Text");
    var id = ev.target.getAttribute('id');
    return false;
}

function dragEnd(ev)
{
    ev.dataTransfer.clearData("Text");
    return true
}

function dragDrop(ev)
{
    var idelt = ev.dataTransfer.getData("Text");
    ev.target.appendChild(document.getElementById(idelt));
    ev.stopPropagation();
    return false;
}
```

```
}
```

Ниже приведен программный код для вызова соответствующих функций при возникновении событий:

```
<div id="source" class="container" ondragenter="return dragEnter(event)"
ondrop="return dragDrop(event)" ondragover="return dragOver(event)">
  <div id="firstDragElement" class="element" ondragstart="return drag-
Start(event)"
ondragend="return dragEnd(event)" draggable="true">Text 1</div>
  <div id="secondDragElement" class="element" ondragstart="return
dragStart(event)
" ondragend="return dragEnd(event)" draggable="true">Text 2</div>
  <div id="thirdDragElement" class="element" ondragstart="return drag-
Start(event)"
ondragend="return dragEnd(event)" draggable="true">Text 3</div>
</div>
<div id="target" class="container" ondragenter="return dragEnter(event)"
ondrop="return dragDrop(event)" ondragover="return dragO-
ver(event)"></div>
```

Вставка элементов мультимедиа

Для того, чтобы не загружать главную Web-страницу избыточным кодом, для каждого альбома создадим отдельную *HTML*-страницу, которую затем будем подгружать.

Таким образом, для выполнения задания, необходимо сделать следующее:

- создать страницы альбомов и главную страницу;
- стилизовать страницы;
- создать *JavaScript* функции для: добавления трека в очередь воспроизведения, предварительного прослушивания трека, перехода между альбомами и воспроизведения трека из очереди.

Для того, чтобы пользователю не пришлось каждый раз заново создавать свой список воспроизведения, будем использовать localStorage.

Рассмотрим по шагам выполнение задания.

Шаг 1.

```
<body>
  <div id="tracks">
    <form id="albumtracks">
      <fieldset>
        <legend id="albumname">The Genius Hits The Road</legend>
        
        <table>
          <tr>
            <td>01-Alabamy Bound</td>
            <td><a class="addtoPLlink" onclick="addtoPL('Ray Charles-Alabamy
Bound',
audio/Ray Charles/The Genius Hits The Road/01-Alabamy Bound.mp3');
">Add to playlist</a></td>
            <td><a class="listenlink" onclick="testListen
('audio/Ray Charles/The Genius Hits The Road/01-Alabamy Bound.mp3');
">Listen</a></td>
          </tr>
          <tr>
            <td>02-Georgia On My Mind</td>
            <td><a class="addtoPLlink" onclick="addtoPL('Ray Charles-Georgia On My
Mind',
'audio/Ray Charles/The Genius Hits The Road/02-Georgia On My Mind.mp3');
">Add to playlist</a></td>
            <td><a class="listenlink" onclick="testListen
('audio/Ray Charles/The Genius Hits The Road/02-Georgia On My Mind.mp3');
```

```

">Listen</a></td>
</tr>
<tr>
<td>03-Basin Street Blues</td>
<td><a class="addtoPLlink" onclick="addtoPL('Ray Charles-Basin Street
Blues',
'audio/Ray Charles/The Genius Hits The Road/03-Basin Street Blues.mp3');
">Add to playlist</a></td>
<td><a class="listenlink" onclick="testListen
('audio/Ray Charles/The Genius Hits The Road/03-Basin Street Blues.mp3');
">Listen</a></td>
</tr>
<tr>
<td>04-Mississippi Mud</td>
<td><a class="addtoPLlink" onclick="addtoPL('Ray Charles-Mississippi
Mud',
'audio/Ray Charles/The Genius Hits The Road/04-Mississippi Mud.mp3');
">Add to playlist</a></td>
<td><a class="listenlink" onclick="testListen
('audio/Ray Charles/The Genius Hits The Road/04-Mississippi Mud.mp3');
">Listen</a></td>
</tr>
.....
</body>

```

Обратите внимание, что сразу же заданы все атрибуты для дальнейшей стилизации, а также вызова *JavaScript*, а именно:

- addtoPL() – функция для добавления указанного трека в пользовательскую очередь воспроизведения;
- testListen() – функция для предварительного прослушивания трека (в течении 10 секунд, например).

Шаг 2. Создание *HTML*- документа для главной страницы:

```
<body onload="load();">
  <div id="header">
    <h1>Audio example</h1>
  </div>
  <div id="albums">
    <form id="albumform">
      <fieldset>
        <legend class="singer">Ray Charles</legend>
        <ul class="albumname">
          <li><a class="albumlink" onclick="loadAlbum('RC_hitstheroad.html');
"> The Genius Hits The Road</a></li>
          <li><a class="albumlink" onclick="loadAlbum('RC_smile.html')>Have
A Smile With Me</a></li>
        </ul>
      </fieldset>
    </form>
  </div>
  <div id="tracks">
    <iframe id="target" src=""></iframe>
  </div>
```

Поясним имена *JavaScript* – функций:

- load() – загрузка списка воспроизведения;
- loadAlbum() – загрузка внешнего *HTML* – документа (альбома).

Шаг 3. Создание минимально необходимых стилей:

```
#albums
{
  float:left;
}

#tracks
```

```
{
  float:left;
}
#audioPanel
{
  clear:left;
}
#albumform
{
  width:350px;
}
audio
{
  width:100%
}
a.albumlink:hover
{
  cursor:pointer;
  font-style:italic;
  font-weight:bolder;
}
#playlistpanel
{
  border:3px double black;
}
#target
{
  width:800px;
  height:600px;
  border:0px;
```

```
}  
a.listenlink  
{  
  margin:10px;  
}  
a.addtoPLink:hover, a.listenlink:hover  
{  
  cursor:pointer;  
  font-style:italic;  
  font-weight:bolder;  
}  
#albumtracks  
{  
  width:500px;  
}  
li.PL:hover  
{  
  background-color:Blue;  
  cursor:pointer;  
  color:White;  
}  
body  
{  
  background-color:#F7F7F7;  
}  
h1  
{  
  text-align:center;
```

Шаг 4. Создание *JavaScript* функции

Функция добавления трека в очередь воспроизведения:

```
function addtoPL(name, path)
{
    localStorage[name] = path; /*добавляем в localStorage путь к аудио
файлу*/
    window.parent.location.href=window.parent.location.href);
    /*поскольку iframe подгружает внешнюю html страницу, то после до-
бавления трека в список, необходимо явно перезагрузить основную страни-
цу*/
```

Функция загрузки внешнего содержимого:

```
function loadAlbum(path)
{
    var iframe = document.getElementById('target');
    iframe.setAttribute('src', path);
    sessionStorage["album"] = path; /*чтобы при перезагрузке родителя,
текущая веб – страница альбома не исчезала, используем
sessionStorage, для сохранения состояния*/
```

Функция предварительного прослушивания:

```
function testListen(path)
{
    var audio = document.createElement('audio'); /*создаем новый audio el-
ement*/
    var div = document.getElementById('tracks');
    /*получаем доступ к контейнеру с треками*/
    audio.src= path; /*задаем источник для воспроизведения*/
    audio.controls = false;
    /*отключаем элементы управления аудио, фактически элемент не бу-
дет отображаться*/
    div.appendChild(audio); /*добавляем созданный audio элемент контей-
неру*/
    audio.addEventListener('timeupdate', function()
```

*/*обработчик событий, вызываемый в течении всего воспроизведения аудио, каждые 250 мс*/*

```
{
  if (audio.currentTime > 10) {audio.pause();}
  /*прерываем воспроизведение после 10 секунд прослушивания*/
}, false);
audio.play(); /*начало воспроизведения аудио*/
```

Функция воспроизведения трека из пользовательского списка:

```
function playtrack(track)
```

```
{
  var b = false;
  for(var I in localStorage)
```

```
{
  /*в данном цикле мы находм текущий трек и «запоминаем» следующий*/
```

```
  if (b) {localStorage['next'] = localStorage[i]; break; }
  if (I == track) { b=true; }
}
```

```
var audio = document.getElementById('audio'); /*получаем доступ к элементу audio*/
```

```
audio.src = localStorage[track]; /*задаем источник воспроизведения*/
audio.controls = true;
audio.play(); /*начинаем воспроизведение*/
```

Функция загрузки списка воспроизведения:

```
function load()
```

```
{
  if (sessionStorage["album"] != undefined)
```

*/*если документ открывается впервые, либо после долгого перерыва, то sessionStorage*

не будет содержать информацию о последнем открытом альбоме/*

```

{
  loadAlbum(sessionStorage[«album»]);
  /*открываем последний альбом, к которому обратился пользователь,
таким образом
  после принудительного обновления главной страницы в функции
adtoPL
  пользователю не придется делать это самостоятельно*/
}
var audio = document.getElementById('audio');
/*получаем доступ к audio элементу*/
audio.addEventListener('ended', function()
/*добавляем обработчик события – окончания текущего воспроизведе-
ния*/
{
  playtrack('next');
  /*иницируем воспроизведение следующего трека из пользователь-
ского списка*/
}, false);
/*переходим к загрузке пользовательского списка воспроизведения*/
var list = document.getElementById('playlist');
/*получаем доступ к списку воспроизведения*/
for(var i in localStorage) /*добавляем треки, сохраненные пользовате-
лем*/
{
  var element = document.createElement('li');
  element.setAttribute('class', 'PL');
  element.setAttribute('onclick', "playtrack('"+i+"')");
  element.innerText = I;
  if ( I != 'next')
  {

```

```
list.appendChild(element);  
}  
}
```

Оборудование и материалы. Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 2000 МГц и выше; оперативная память – не менее 1024 Мбайт; свободное дисковое пространство – не менее 1,2 Гбайт; устройство для чтения компакт-дисков; монитор типа Super VGA (число цветов – 256) с диагональю не менее 17 ". **Программное обеспечение** – операционная система WINDOWS XP и выше, программы для просмотра Web-страниц.

Указания по технике безопасности. Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для работы с персональным компьютером. Самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения. В случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору). Соблюдать правила техники безопасности при работе с электрооборудованием. Не касаться электрических розеток металлическими предметами. Рабочее место пользователя персонального компьютера должно содержаться в чистоте. Не разрешается возле персонального компьютера принимать пищу, напитки.

Задания

Задание 1. Реализовать функцию перемещения элементов Web-страницы между группой контейнерами, в следующих вариантах:

- все элементы могут свободно перемещаться между тремя контейнерами;

- ряд элементов, относящихся к одному классу нельзя перенести в один из контейнеров;

- элементы не могут быть возвращены в контейнер, в котором находились первоначально.

Задание 2. Создать HTML – документ, содержащий список альбомов и позволяющий формировать пользовательскую очередь воспроизведения мультимедийных материалов.

Указания по порядку выполнения работы

При выполнении заданий к лабораторной работе использовать материал, приведенный в ее теоретической части.

Содержание отчета

Отчет по лабораторной работе должен состоять из:

- 1) названия лабораторной работы;
- 2) ответов на контрольные вопросы;
- 3) формулировки индивидуальных заданий и порядка их выполнения.

Контрольные вопросы

1. Охарактеризуйте алгоритм создания элементов перемещения внутри Web-формы.

2. При помощи каких тегов можно описать стилизацию элементов перемещения?

3. Дайте характеристику алгоритмам создания и использования JavaScript-функций в Web-формах.

4. Охарактеризуйте объект DataTransfer. Приведите его свойства.

5. Каким образом можно осуществить вставку элементов мультимедиа в Web-форму?

6. Характеристика и назначение JavaScript-функций при вставке объектов мультимедиа в Web-форму.

СПИСОК ЛИТЕРАТУРЫ: основная – 2, 3, 4, 5; дополнительная – 2, 3.

Тема 1. Введение в web-технологии: структура и принципы Web

ЛАБОРАТОРНАЯ РАБОТА 3

HTML 5. РАБОТА С ГРАФИКОЙ. CANVAS. АНИМАЦИЯ

Цель работы: ознакомление с элементом `<canvas>`, ей `<canvas>` и создание простейших *JavaScript*-функций для генерации изображений в рамках элемента `<canvas>`; формирование базовых навыков рисования простых фигур на холсте.

Формируемые компетенции или их части:

1. ОПК-5 – владением методами и средствами получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе, в глобальных компьютерных сетях.

2. ПК-3 – умением разрабатывать новые технологии проектирования информационных систем.

3. ПК-15 – способностью разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач.

Теоретическая часть

Размещение `<canvas>` на странице

Холст размещается на странице при помощи соответствующего тега:

```
<canvas id="lesson6" width="150" height="150"></canvas>
```

Атрибуты `width` и `height` не являются обязательными. По умолчанию высота и ширина холста равны 150 и 300 пикселям соответственно. При явном задании размеров холста его картинка масштабируется соответствующим образом.

По аналогии с такими новыми тегами как `<audio>` и `<video>`, элементы, размещенные в рамках парного тега `<canvas>` будут отображаться на странице в том случае, если используемый пользователем браузер не поддерживает `<canvas>`.

Стилизация <canvas>

Стилизация холста не отличается от таковой для любого другого элемента Web-страницы. Могут быть заданы величины отступа, выравнивания, рамки и т.п.

Поскольку еще не создана функция по отрисовке содержимого холста, то и в браузере мы ничего не увидим. Создадим стиль для элемента <canvas>, задающий параметры границы вокруг элемента:

```
canvas { border: 3px double black;
```

Рисование на холсте

<canvas> создаёт поверхность для рисования, предоставляющую один или более контекст отрисовки, который используется для создания отображаемого контента и манипуляций с ним.

Элемент <canvas> изначально пустой, и для того, чтобы что-либо отобразить, скрипту необходимо получить контекст отрисовки и рисовать уже на нём. Элемент canvas имеет *DOM-метод*, называемый `getContext`, и предназначенный для получения контекста отрисовки вместе с его функциями рисования. `getContext()` принимает один параметр – тип контекста. Осуществляется все вышесказанное следующим образом, в рамках *JavaScript*-функции:

```
var canvas = document.getElementById("lesson6");
var ctx = canvas.getContext("2d")
```

Первая строка приведенного фрагмента создает переменную, фактически являющуюся созданным холстом. Вторая – получает доступ к контексту рисования.

Рассмотрим пример создания изображения. Пусть это будет – прямоугольник багрового цвета.

Для этого, создадим *JavaScript* – функцию:

```
function createImage()
{
    var canvas = document.getElementById("lesson6");
```

```

    var ctx = canvas.getContext("2d");
  }

```

Прямоугольную закрашенную область можно нарисовать при помощи функции `fillRect (x,y,w,h)`, где `x` и `y` – координаты левой верхней вершины прямоугольника (по горизонтали и вертикали соответственно), а `w` и `h` – значения ширины и высоты прямоугольника, соответственно.

Функция `strokeRect (x,y,w,h)` рисует границы прямоугольника, `clearRect (x,y,w,h)` – очищает заданную прямоугольную область.

Функция `fillStyle` задает цвет рисования.

Таким образом, функция принимает вид:

```

function createImage()
{
    var canvas = document.getElementById("lesson6");
    var ctx = canvas.getContext("2d");
    ctx.fillStyle = "rgb(100,0,0)"
    ctx.fillRect (20, 10, 50, 70);
}

```

Функция готова, холст уже давно размещен на странице, осталось обеспечить выполнение функции. Для того, чтобы изображение обрисовывалось в момент загрузки страницы, добавим в тег `<body>` вызов функции `createImage` при загрузке. HTML-код будет выглядеть следующим образом:

```

<body onload="createImage();">
<canvas id="text" width="150" height="150"></canvas>
</body>

```

Рисование фигур

Каждый треугольник может быть нарисован последовательным вызовом трех методов `lineTo`.

Шаг 1. Создание пути для отрисовки первого треугольника:

```

function draw(){

```

```

var canvas = document.getElementById('text');
var ctx = canvas.getContext('2d');
ctx.beginPath();
ctx.moveTo(20,20); // задаем начальную точку для "пера"
ctx.lineTo(20, 120); // сторона А
ctx.lineTo(120,120); // сторона В
ctx.lineTo(20,20); // сторона С
ctx.stroke(); // рисуем незаполненную фигуру

```

Шаг 2. Создание залитого треугольника отличается от предыдущего шага незначительно, а именно, вызовом метода `fill`, вместо `stroke`. Добавим соответствующий код в функцию `draw`:

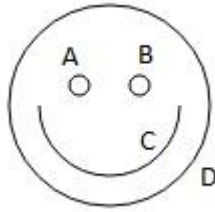
```

function draw(){
var canvas = document.getElementById('lesson6');
var ctx = canvas.getContext('2d');
ctx.beginPath();
ctx.moveTo(20,20); // задаем начальную точку для "пера"
ctx.lineTo(20, 120); // сторона А
ctx.lineTo(120,120); // сторона В
ctx.lineTo(20,20); // сторона С
ctx.stroke(); // рисуем незаполненную фигуру
ctx.beginPath();
ctx.moveTo(40,20); // задаем начальную точку для отрисовки второго
треугольника
ctx.lineTo(140, 20); // сторона D
ctx.lineTo(140,120); // сторона E
ctx.lineTo(40,20); // сторона F
ctx.fill(); // рисуем залитую цветом фигуру

```

Рисование дуг

Например, используем процесс построения дуг на следующем изображении. В нем можно выделить всего четыре дуги, которые необходимо нарисовать:

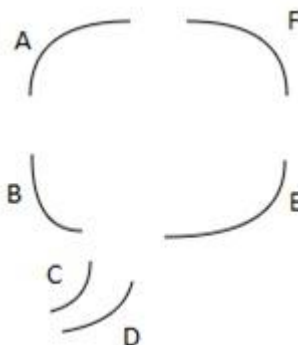


Основная сложность этого задания в переносе "пера" в начальные позиции для рисования, поэтому обратите внимание на координаты, указанные в методах `moveTo` и `arc`:

```
function draw(){
var canvas = document.getElementById('lesson6');
var ctx = canvas.getContext('2d');
ctx.beginPath();
ctx.moveTo(65,65); // начальная позиция для рисование
ctx.arc(60,65,5,0,Math.PI*2,true); // дуга A
ctx.moveTo(95,65); // перенос "пера" в позицию для отрисовки дуги B
ctx.arc(90,65,5,0,Math.PI*2,true); // дуга B
ctx.moveTo(110,75); // перенос "пера" для отрисовки дуги C
ctx.arc(75,75,35,0,Math.PI,false); // дуга C
ctx.moveTo(125,75); // перенос "пера" для отрисовки дуги D
ctx.arc(75,75,50,0,Math.PI*2,true); // дуга D
ctx.stroke(); // отрисовка незаполненной фигуры
```

Кривые Безье

Отметим только, что для отрисовки каждой "половинки сердца" используется три кривые (метод `bezierCurveTo`).



Если рисовать кривые последовательно от А к F, то понадобится только один вызов функции `moveTo` для задания начальных координат для первой кривой (А). Как и в предыдущем примере нужно уделить внимание задаваемым координатам для кривых:

```
function draw(){
    var canvas = document.getElementById('lesson6');
    var ctx = canvas.getContext('2d');
    ctx.beginPath();
    ctx.moveTo(75,25); // задание начальных координат
    ctx.quadraticCurveTo(25,25,25,62.5); // кривая А
    ctx.quadraticCurveTo(25,100,50,100); // кривая В
    ctx.quadraticCurveTo(50,120,30,125); // кривая С
    ctx.quadraticCurveTo(60,120,65,100); // кривая D
    ctx.quadraticCurveTo(125,100,125,62.5); // кривая E
    ctx.quadraticCurveTo(125,25,75,25); // кривая F
    ctx.stroke(); // отрисовка незаполненной фигуры
```

Оборудование и материалы. Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 2000 МГц и выше; оперативная память – не менее 1024 Мбайт; свободное дисковое пространство – не менее 1,2 Гбайт; устройство для чтения компакт-дисков; монитор типа Super VGA (число цветов – 256) с диагональю не менее 17 ". **Программное обеспечение** – операционная система WINDOWS XP и выше, программы для просмотра Web-страниц.

Указания по технике безопасности. Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для работы с персональным компьютером. Самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения. В случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору). Соблюдать правила техники безопасности при работе с электрооборудованием. Не касаться электрических розеток металлическими предметами. Рабочее место пользователя персонального компьютера должно содержаться в чистоте. Не разрешается возле персонального компьютера принимать пищу, напитки.

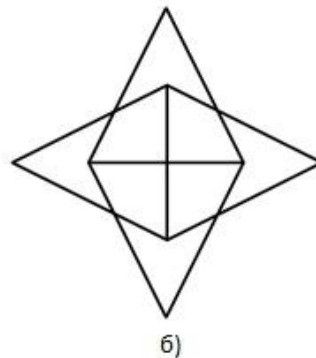
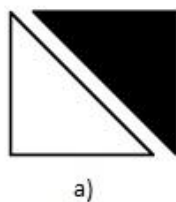
Задания

Задание 1. Разместите элемент `<canvas>` на странице.

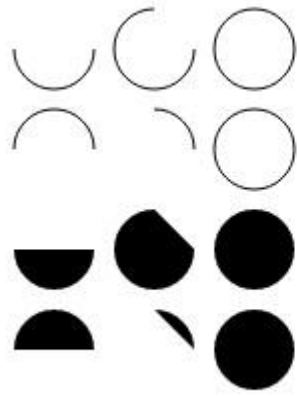
Задание 2. Создать файл стилей для `<canvas>`.

Задание 3. При помощи JavaScript-функции реализовать генерацию следующих изображений на странице:

1. Рисование путей



2. Дуги



a)



б)

3. Кривые Безье



a)



б)

4. Комплексное задание



Указания по порядку выполнения работы

При выполнении заданий к лабораторной работе следует использовать материал и примеры, приведенные в ее теоретической части.

Содержание отчета

Отчет по лабораторной работе должен состоять из:

- 1) названия лабораторной работы;
- 2) ответов на контрольные вопросы;
- 3) формулировки индивидуальных заданий и порядка их выполнения.

Контрольные вопросы

1. При помощи какого тега осуществляется размещение холста на странице?
2. С какой целью в HTML-документах проводится стилизация холста?
3. Охарактеризуйте процесс рисования на холсте. Назовите функции, которые для этого используются.
4. Приведите алгоритм рисования простейших фигур на холсте.
5. Каким образом осуществляется рисование дуг?
6. Алгоритм рисования «кривой Безье».

СПИСОК ЛИТЕРАТУРЫ: основная – 2, 3, 4, 5; дополнительная – 2, 3

СПИСОК ЛИТЕРАТУРЫ

Основная литература

1. Информационные Web-технологии / Ю. Громов, О.Г. Иванова, Н.Г. Шахов, В.Г. Однолько; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2014. - 96 с. : ил. - Библиогр. в кн. - ISBN 978-5-8265-1365-1 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=277935>

2. Тузовский А.Ф. Проектирование и разработка web-приложений [Электронный ресурс]: учебное пособие/ Тузовский А.Ф.— Электрон. текстовые данные.— Томск: Томский политехнический университет, 2014.— 219 с.— Режим доступа: <http://www.iprbookshop.ru/34702>.— ЭБС «IPRbooks»

Дополнительная литература

1. Информационные Web-технологии / Ю. Громов, О. Г. Иванова, Н. Г. Шахов, В. Г. Однолько; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». – Тамбов: Издательство ФГБОУ ВПО «ТГТУ», 2014. – 96 с. [Электронный ресурс]. Режим доступа: <http://biblioclub.ru/index.php?page=book&id=277935>

2. Сухов, К. HTML5 – путеводитель по технологии / К. Сухов. – М.: ДМК Пресс, 2014. – 352 с. [Электронный ресурс]. Режим доступа: <http://biblioclub.ru/index.php?page=book&id=260322>

3. Котов, О. М. Язык C#: краткое описание и введение в технологии программирования: учебное пособие / О. М. Котов; Министерство образова-

ния и науки Российской Федерации, Уральский федеральный университет имени первого Президента России Б. Н. Ельцина. – Екатеринбург: Издательство Уральского университета, 2014. – 209 с. [Электронный ресурс]. Режим доступа: <http://biblioclub.ru/index.php?page=book&id=275809>

4. Громов, Ю. Ю. Основы Web-инжиниринга: разработка клиентских приложений: учебное пособие / Ю. Ю. Громов, О. Г. Иванова, С. В. Данилкин; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». – Тамбов: Издательство ФГБОУ ВПО «ТГТУ», 2012. – 240 с. [Электронный ресурс]. Режим доступа: <http://biblioclub.ru/index.php?page=book&id=277648>

5. Кузин, А. В. Базы данных: учеб. пособие для вузов / А.В. Кузин, С.В. Левонисова. – 5-е изд., испр. – Москва: Академия, 2012. – 314 с. (электронный каталог библиотеки СКФУ)

6. Джонсон, Г. Разработка клиентских веб-приложений на платформе Microsoft NET FRAMEWORK: учеб. курс Microsoft: пер. с англ. / Гленн Джонсон, Тони Нортроп. – М.: Русская Редакция; СПб.: Питер, 2007. – 768 с. (электронный каталог СКФУ)

7. Базы данных: учебник для вузов / [Хомоненко А. Д., Цыганков В. М., Мальцев М. Г]; под ред. Хомоненко А. Д. - 3-е изд., перераб. и доп. - СПб.: КОРОНА принт, 2003. – 672 с (электронный каталог библиотеки СКФУ).

8. Основы WEB-технологий: Курс лекций. Специальность "Интернет-технологии" / П. Б. Храмцов, С. А. Брик, А. М. Русак, А. И. Сурин. - М.: Интернет-Университет Информационных Технологий, 2003. – 512 с (электронный каталог библиотеки СКФУ)

9. Шумаков, П. В. ADO.Net и создание приложений баз данных в среде Microsoft Visual .Net. Руководство разработчика с примерами на C# / П. В. Шумаков. – М.: Диалог- МИФИ, 2003. – 528 с (электронный каталог библиотеки СКФУ)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Методические указания к самостоятельным работам

Направление подготовки 09.03.02 Информационные системы и технологии

Направленность (профиль) «Информационные системы управления
технологическими и сервисными процессами»

Квалификация выпускника – бакалавр

Невинномысск 2026

Методические указания предназначены для студентов направления подготовки 09.03.02 Информационные системы и технологии и других технических специальностей. Они содержат рекомендации по организации самостоятельных работ студента для дисциплины «Современные технологии программирования».

Методические указания разработаны в соответствии с требованиями ФГОС ВО в части содержания и уровня подготовки выпускников направления 09.03.02 Информационные системы и технологии

Содержание

1 Подготовка к лекциям.....	4
2 Подготовка к лабораторным работам	6
3 Самостоятельное изучение темы. Конспект.....	9
4 Подготовка к экзамену.....	12

1 Подготовка к лекциям

Главное в период подготовки к лекционным занятиям – научиться методам самостоятельного умственного труда, сознательно развивать свои творческие способности и овладевать навыками творческой работы. Для этого необходимо строго соблюдать дисциплину учебы и поведения. Четкое планирование своего рабочего времени и отдыха является необходимым условием для успешной самостоятельной работы. В основу его нужно положить рабочие программы изучаемых в семестре дисциплин.

Каждому студенту следует составлять еженедельный и семестровый планы работы, а также план на каждый рабочий день. С вечера всегда надо распределять работу на завтрашний день. В конце каждого дня целесообразно подводить итог работы: тщательно проверить, все ли выполнено по намеченному плану, не было ли каких-либо отступлений, а если были, по какой причине это произошло. Нужно осуществлять самоконтроль, который является необходимым условием успешной учебы. Если что-то осталось невыполненным, необходимо изыскать время для завершения этой части работы, не уменьшая объема недельного плана.

Слушание и запись лекций – сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность студента. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим студентом. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Запись лекций рекомендуется вести по возможности собственными формулировками. Желательно запись осуществлять на одной странице, а следующую оставлять для проработки учебного материала самостоятельно в домашних условиях.

Конспект лекций лучше подразделять на пункты, параграфы, соблюдая красную строку. Этому в большой степени будут способствовать пункты плана лекции, предложенные преподавателям. Принципиальные места, опре-

деления, формулы и другое следует сопровождать замечаниями «важно», «особо важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек. Лучше если они будут собственными, чтобы не приходилось присить их у однокурсников и тем самым не отвлекать их во время лекции. Целесообразно разработать собственную «маркографию» (значки, символы), сокращения слов. Не лишним будет и изучение основ стенографии. Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть знаниями.

2 Подготовка к лабораторным работам

Подготовку к каждому практическому занятию студент должен начать с ознакомления с методическими указаниями, которые включают содержание работы. Тщательное продумывание и изучение вопросов основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованную к данной теме. На основе индивидуальных предпочтений студенту необходимо самостоятельно выбрать тему доклада по проблеме и по возможности подготовить по нему презентацию.

Если программой дисциплины предусмотрено выполнение практического задания, то его необходимо выполнить с учетом предложенной инструкции (устно или письменно). Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы семинара, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий и контрольных работ.

В зависимости от содержания и количества отведенного времени на изучение каждой темы практическое занятие может состоять из четырех-пяти частей:

1. Обсуждение теоретических вопросов, определенных программой дисциплины.
2. Доклад и/ или выступление с презентациями по выбранной проблеме.
3. Обсуждение выступлений по теме – дискуссия.
4. Выполнение практического задания с последующим разбором полученных результатов или обсуждение практического задания.
5. Подведение итогов занятия.

Первая часть – обсуждение теоретических вопросов – проводится в виде фронтальной беседы со всей группой и включает выборочную проверку преподавателем теоретических знаний студентов. Примерная продолжительность — до 15 минут. Вторая часть — выступление студентов с докладами, которые должны сопровождаться презентациями с целью усиления наглядности восприятия, по одному из вопросов практического занятия. Обязательный элемент доклада – представление и анализ статистических данных, обоснование социальных последствий любого экономического факта, явления или процесса. Примерная продолжительность — 20-25 минут. После докладов следует их обсуждение – дискуссия. В ходе этого этапа практического занятия могут быть заданы уточняющие вопросы к докладчикам. Примерная продолжительность – до 15-20 минут. Если программой предусмотрено выполнение практического задания в рамках конкретной темы, то преподавателями определяется его содержание и дается время на его выполнение, а затем идет обсуждение результатов. Подведением итогов заканчивается практическое занятие.

В процессе подготовки к практическим занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме семинарского или практического занятия, что позволяет студентам проявить свою индивидуальность в рамках выступления на данных занятиях, выявить широкий спектр мнений по изучаемой проблеме.

3 Самостоятельное изучение темы. Конспект

Конспект – наиболее совершенная и наиболее сложная форма записи. Слово «конспект» происходит от латинского «conspectus», что означает «обзор, изложение». В правильно составленном конспекте обычно выделено самое основное в изучаемом тексте, сосредоточено внимание на наиболее существенном, в кратких и четких формулировках обобщены важные теоретические положения.

Конспект представляет собой относительно подробное, последовательное изложение содержания прочитанного. На первых порах целесообразно в записях ближе держаться тексту, прибегая зачастую к прямому цитированию автора. В дальнейшем, по мере выработки навыков конспектирования, записи будут носить более свободный и сжатый характер.

Конспект книги обычно ведется в тетради. В самом начале конспекта указывается фамилия автора, полное название произведения, издательство, год и место издания. При цитировании обязательная ссылка на страницу книги. Если цитата взята из собрания сочинений, то необходимо указать соответствующий том. Следует помнить, что четкая ссылка на источник – неперемutable правило конспектирования. Если конспектируется статья, то указывается, где и когда она была напечатана.

Конспект подразделяется на части в соответствии с заранее продуманным планом. Пункты плана записываются в тексте или на полях конспекта. Писать его рекомендуется четко и разборчиво, так как небрежная запись с течением времени становится малопонятной для ее автора. Существует правило: конспект, составленный для себя, должен быть по возможности написан так, чтобы его легко прочитал и кто-либо другой.

Формы конспекта могут быть разными и зависят от его целевого назначения (изучение материала в целом или под определенным углом зрения, подготовка к докладу, выступлению на занятии и т.д.), а также от характера произведения (монография, статья, документ и т.п.). Если речь идет просто об изложении содержания работы, текст конспекта может быть сплошным, с

выделением особо важных положений подчеркиванием или различными значками.

В случае, когда не ограничиваются переложением содержания, а фиксируют в конспекте и свои собственные суждения по данному вопросу или дополняют конспект соответствующими материалами их других источников, следует отводить место для такого рода записей. Рекомендуется разделить страницы тетради пополам по вертикали и в левой части вести конспект произведения, а в правой свои дополнительные записи, совмещая их по содержанию.

Конспектирование в большей мере, чем другие виды записей, помогает вырабатывать навыки правильного изложения в письменной форме важные теоретических и практических вопросов, умение четко их формулировать и ясно излагать своими словами.

Таким образом, составление конспекта требует вдумчивой работы, затраты времени и труда. Зато во время конспектирования приобретаются знания, создается фонд записей.

Конспект может быть текстуальным или тематическим. В текстуальном конспекте сохраняется логика и структура изучаемого произведения, а запись ведется в соответствии с расположением материала в книге. За основу тематического конспекта берется не план произведения, а содержание какой-либо темы или проблемы.

Текстуальный конспект желательно начинать после того, как вся книга прочитана и продумана, но это, к сожалению, не всегда возможно. В первую очередь необходимо составить план произведения письменно или мысленно, поскольку в соответствии с этим планом строится дальнейшая работа. Конспект включает в себя тезисы, которые составляют его основу. Но, в отличие от тезисов, конспект содержит краткую запись не только выводов, но и доказательств, вплоть до фактического материала. Иначе говоря, конспект – это расширенные тезисы, дополненные рассуждениями и доказательствами, мыслями и соображениями составителя записи.

Как правило, конспект включает в себя и выписки, но в него могут войти отдельные места, цитируемые дословно, а также факты, примеры, цифры, таблицы и схемы, взятые из книги. Следует помнить, что работа над конспектом только тогда будет творческой, когда она не ограничена текстом изучаемого произведения. Нужно дополнять конспект данными из другими источниками.

В конспекте необходимо выделять отдельные места текста в зависимости от их значимости. Можно пользоваться различными способами: подчеркиваниями, вопросительными и восклицательными знаками, репликами, краткими оценками, писать на полях своих конспектов слова: «важно», «очень важно», «верно», «характерно».

В конспект могут помещаться диаграммы, схемы, таблицы, которые придадут ему наглядность.

Составлению тематического конспекта предшествует тщательное изучение всей литературы, подобранной для раскрытия данной темы. Бывает, что какая-либо тема рассматривается в нескольких главах или в разных местах книги. А в конспекте весь материал, относящийся к теме, будет сосредоточен в одном месте. В плане конспекта рекомендуется делать пометки, к каким источникам (вплоть до страницы) придется обратиться для раскрытия вопросов. Тематический конспект составляется обычно для того, чтобы глубже изучить определенный вопрос, подготовиться к докладу, лекции или выступлению на семинарском занятии. Такой конспект по содержанию приближается к реферату, докладу по избранной теме, особенно если включает и собственный вклад в изучение проблемы.

4 Подготовка к экзамену

Экзаменационная сессия – очень тяжелый период работы для студентов и ответственный труд для преподавателей. Главная задача экзаменов – проверка качества усвоения содержания дисциплины.

На основе такой проверки оценивается учебная работа не только студентов, но и преподавателей: по результатам экзаменов можно судить и о качестве всего учебного процесса. При подготовке к экзамену студенты повторяют материал курсов, которые они слушали и изучали в течение семестра, обобщают полученные знания, выделяют главное в предмете, воспроизводят общую картину для того, чтобы яснее понять связь между отдельными элементами дисциплины.

При подготовке к экзаменам основное направление дают программы курса и конспект, которые указывают, что в курсе наиболее важно. Основной материал должен прорабатываться по учебнику, поскольку конспекта недостаточно для изучения дисциплины. Учебник должен быть проработан в течение семестра, а перед экзаменом важно сосредоточить внимание на основных, наиболее сложных разделах. Подготовку по каждому разделу следует заканчивать восстановлением в памяти его краткого содержания в логической последовательности.

До экзамена обычно проводится консультация, но она не может возместить отсутствия систематической работы в течение семестра и помочь за несколько часов освоить материал, требующийся к экзамену. На консультации студент получает лишь ответы на трудные или оставшиеся неясными вопросы. Польза от консультации будет только в том случае, если студент до нее проработает весь материал. Надо учиться задавать вопросы, вырабатывать привычку пользоваться справочниками, энциклопедиями, а не быть на иждивении у преподавателей, который не всегда может тут же, «с ходу» назвать какой-либо факт, имя, событие. На экзамене нужно показать не только знание предмета, но и умение логически связно построить устный ответ.

Получив билет, надо вдуматься в поставленные вопросы для того, чтобы правильно понять их. Нередко студент отвечает не на тот вопрос, который поставлен, или в простом вопросе ищет скрытого смысла. Не поняв вопроса и не обдумав план ответа, не следует начинать писать. Конспект своего ответа надо рассматривать как план краткого сообщения на данную тему и составлять ответ нужно кратко. При этом необходимо показать умение выражать мысль четко и доходчиво.

Отвечать нужно спокойно, четко, продуманно, без торопливости, придерживаясь записи своего ответа. На экзаменах студент показывает не только свои знания, но и учится владеть собой. После ответа на билет могут следовать вопросы, которые имеют целью выяснить понимание других разделов курса, не вошедших в билет. Как правило, на них можно ответить кратко, достаточно показать знание сути вопроса. Часто студенты при ответе на дополнительные вопросы проявляют поспешность: не поняв смысла того, что у них спрашивают, начинают отвечать и нередко говорят не по сути.

Следует помнить, что необходимым условием правильного режима работы в период экзаменационной сессии является нормальный сон, поэтому подготовка к экзаменам не должна быть в ущерб сну. Установлено, что сильное эмоциональное напряжение во время экзаменов неблагоприятно отражается на нервной системе и многие студенты из-за волнений не спят ночи перед экзаменами. Обычно в сессию студенту не до болезни, так как весь организм озабочен одним - сдать экзамены. Но это еще не значит, что последствия неправильно организованного труда и чрезмерной занятости не скажутся потом. Поэтому каждый студент помнить о важности рационального распорядка рабочего дня и о своевременности снятия или уменьшения умственного напряжения.