

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Комарова Светлана Юриевна

Должность: Проректор по образовательной деятельности

Дата подписания: 09.07.2023 12:40:16

Уникальный программный ключ:

43ba42f5deae4116bbfcb9e98e79108031227e81add307cbee41149f2098d7a

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Омский государственный аграрный университет имени П.А.Столыпина»**

## **Университетский колледж агробизнеса**

**ООП по специальности 09.02.07 Информационные системы и программирование**

### **МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ**

#### **САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ по учебной дисциплине**

#### **ОП.12 Основы гипертекстовой разметки**

**Специальность: 09.02.07 Информационные системы и программирование**

Ведущий преподаватель  
(руководитель) дисциплины

Д.И. Васина

**Омск 2023**

## **Пояснительная записка**

Методические рекомендации по учебной дисциплине трудовое право предназначены для выполнения самостоятельной работы обучающимися по специальности 09.02.07 Информационные системы и программирование.

Самостоятельная работа выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Целью самостоятельной работы является овладение обучающимся умениями работать с источниками, аргументации собственной точки зрения.

Методические рекомендации по самостоятельной работе студентов содержат материалы для подготовки к лекционным, практическим занятиям, к формам текущего и промежуточного контроля.

Предложенные в рекомендациях задания позволят успешно овладеть профессиональными знаниями, умениями и навыками, и направлены на формирование общих и профессиональных компетенций:

ОК 2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.

ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных.

ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации.

При выполнении самостоятельной работы обучающийся самостоятельно осуществляет сбор, изучение, систематизацию и анализ информации, а затем оформляет информацию и представляет на оценку преподавателя или группы.

### Виды самостоятельной работы

№ п/п	Вид самостоятельной работы	Форма контроля	Максимальное кол-во баллов
1.	Работа с источниками	Устный ответ на занятии Составление конспекта	5
2.	Составление опорного конспекта, логических схем	Опорный конспект	5
3.	Составление сравнительной таблицы	Сравнительная таблица	5
4.	Решение ситуационных задач	Письменный ответ	5

### Методические рекомендации по составлению опорного конспекта

Опорный конспект составляется с целью обобщения, систематизации и краткого изложения информации. Составление опорного конспекта способствует более быстрому запоминанию учебного материала.

Составление опорного конспекта включает следующие действия:

1. Изучение текста учебного материала.
2. Определение главного и второстепенного в анализируемом тексте.
3. Установление логической последовательности между элементами.
4. Составление характеристики элементов учебного материала в краткой форме.
5. Выбор опорных сигналов для расстановки акцентов.
6. Оформление опорного конспекта.

Опорный конспект может быть представлен в виде схемы с использованием стрелок для определения связи между элементами; системы геометрических фигур; логической лестницы и т.д.

Оценкой опорного конспекта может служить качество ответа, как самого студента, так и других студентов его использовавших. Преподаватель также может проверить опорные конспекты, сданные в письменной форме. Допускается проведение конкурса на самый лучший конспект по следующим критериям: краткость формы; логичность изложения; наглядность выполнения; универсальность содержания.

### Методические рекомендации по составлению сравнительной таблицы

Сравнительная таблица составляется с целью выявления сходств, отличий, преимуществ и недостатков анализируемых объектов.

Критерии для составления сравнительной таблицы предлагает преподаватель. Студент, самостоятельно сформулировавший критерии для сравнения, получает дополнительные баллы.

Проверка и оценка сравнительной таблицы осуществляется преподавателем в письменной форме.

## **Задания для самостоятельной работы**

### **Самостоятельная работа № 1**

#### **Изучение формата HTML**

***Цель работы:** изучить структуру HTML и основные тэги, научиться создавать свои документы в формате HTML.*

#### **Самостоятельная часть**

Взять любой текстовый файл размером 2-3 страницы. Создать из него HTML-файл, отформатированный следующим образом.

1. Текст начинается с заглавия. Заглавие должно отображаться в заголовке окна браузера.
2. Затем идет содержание из гиперссылок на нижеследующие абзацы.
3. Далее идет текст, состоящий из абзацев (не менее трех).
4. Каждый абзац начинается с заголовка. Заголовок выровнен по центру.
5. Первый абзац написан курсивом и подчеркнут, выровнен по левому краю.
6. Второй абзац написан полужирным шрифтом и подчеркнут, выровнен по правому краю.
7. Третий абзац написан в обычном стиле, выровнен по ширине страницы.
8. Текст должен содержать несколько таблиц, не менее чем 2х3 ячейки.
9. Текст должен содержать несколько рисунков.
10. В тексте должны встречаться ссылки на таблицы и рисунки.
11. Должна быть прописана используемая кодировка для формата HTML –“Windows-1251”.

Для выполнения работы используется текстовый редактор «Блокнот» и браузер Microsoft Internet Explorer.

Полное описание формата HTML изложено в прилагаемой документации «Спецификация HTML 4.0». Типовая структура HTML-документа и список используемых в работе тэгов изложен в теоретической части.

#### **Теоретическая часть**

##### **HTML**

Язык разметки документов - это набор специальных инструкций, называемых тэгами, предназначенных для формирования в документах какой-либо структуры и определения отношений между различными элементами этой структуры. Тэги языка, или, как их иногда

называют, управляющие дескрипторы, в таких документах каким-то образом кодируются, выделяются относительно основного содержимого документа и служат в качестве инструкций для программы, производящей показ содержимого документа на стороне клиента. В самых первых системах для обозначения этих команд использовались символы “<” и “>”, внутри которых помещались названия инструкций и их параметры. Сейчас такой способ обозначения тэгов является стандартным.

Использование гипертекстовой разбивки текстового документа в современных информационных системах во многом связано с тем, что гипертекст позволяет создавать механизм нелинейного просмотра информации. В таких системах данные представляются не в виде непрерывного потока текстовой информации, а набором взаимосвязанных компонентов, переход по которым осуществляется при помощи гиперссылок.

Самый популярный на сегодняшний день язык гипертекстовой разметки – HTML, был создан специально для организации информации, распределенной в сети Интернет, и является одной из ключевых составляющих технологии WWW. С использованием гипертекстовой модели документа способ представления разнообразных информационных ресурсов в сети стал более упорядочен, а пользователи получили удобный механизм поиска и просмотра нужной информации.

### Структура HTML-документа

```
<HTML>
<HEAD>
<META HTTP-EQUIV=Content-type CONTENT="text/html; charset=iso-8859-5">
<TITLE>Заголовок документа</TITLE>
</HEAD>
<BODY>    <! Тело документа >
Первая строка документа <BR>
Вторая строка документа <BR>
Третья строка документа ... <BR>
</BODY>
</HTML>
```

### Комментарии

```
<! Это простой комментарий >
<!-- <B>Это комментарий, который может содержать тэги внутри себя <BR>
<I>Простой комментарий не содержит тэгов</I> </B> -->
```

### Ссылки

```
<A HREF="url"> Microsoft </A> - на другой документ
<A HREF="#name"> local </A> - внутри этого же документа
```

### Стили текста

```
<B> Жирный </B>
<I> Курсив </I>
<U> Подчеркнутый </U>
<H1></H1>...<H6></H6> - Заголовки абзацев
```

### Параграфы и строки

```
<BR> - конец строки
<CENTER> - выравнивание по центру
<HR> - горизонтальная линия
<P> - конец абзаца
```

<DIV ALIGN="JUSTIFY"></DIV> - выравнивание блока

<TITLE>Заголовок документа</TITLE>

<IMG SRC="pic.gif" WIDTH=100 HEIGHT=100 ALT="альтернатива"> - отобразить рисунок

<PRE></PRE> - предварительно отформатированный текст

#### Таблица

<TABLE></TABLE> - тело таблицы

<TR></TR> - строка таблицы

<TD></TD> - ячейка таблицы

## Самостоятельная работа № 2

### Знакомство с языком PHP

**Задачи:** Изучить следующие базовые элементы языка:

1. Типы данных, идентификаторы, комментарии, объявление переменных.
2. Управляющие конструкции языка.
3. Функции форматированного/неформатированного вывода данных.

#### Самостоятельная часть

1. Динамически сгенерировать таблицу данных 50 строк на 15 столбцов с помощью генератора случайных чисел. Подсчитать статистические данные по строкам и столбцам – среднее значение, максимальное значение и сумма. Результат показать в браузере. Примерный вид таблицы:

	<i>1</i>	<i>2</i>	<i>...</i>	<i>15</i>	<i>среднее</i>	<i>максимальное</i>	<i>сумма</i>
<i>1</i>	12	29	...	45	34	67	345
<i>2</i>	43	23	...	24	54	89	254
<i>...</i>	...	...	...	...	...	...	...
<i>50</i>	12	87	...	27	23	95	259
<i>среднее</i>	34	56	...	45			
<i>максимальное</i>	98	89	...	87			
<i>сумма</i>	345	674	...	345			

2. Сгенерировать строку из 100 символов с различным типом форматирования. Символы и форматирование получаются случайным образом. Для форматирования применяются тэги <b></b>, <u></u>, <i></i> и их комбинации. Результат показать в браузере. Примерный вид строки:  
d**j**kls**g**j**k**l**f**m**b**d**g**c**j**d**f**g**k**q**w****x****f**d**g**z**d**faglj

#### Теоретическая часть

##### 1. Примеры объявления переменных:

\$a = 1; - целочисленный тип

\$b = 2.45; - вещественный тип

\$c = "string"; - строковый тип

##### Комментарии:

// однострочный комментарий

# это тоже однострочный комментарий

```
/* многострочный  
комментарий */
```

## 2. Основные команды и примеры:

```
if (<условие>) <команда> [ else <команда>]  
if (<условие>) <команда> elseif (<условие>) <команда>  
if (x > y) print "x>y";  
elseif (x<y) print "x<y";  
else print "x=y";  
for(<инициализация>;<условие>;<приращение>) <команда>  
for ($i=0; $i < 10; $i++) $fact = $fact * $i;  
continue; - продолжить цикл, пропустив нижележащие команды  
break; - прекратить цикл  
mt_rand (<минимум>, <максимум>); - сгенерировать случайное число  
$num = mt_rand(10,50);
```

## 3. Основные функции вывода:

```
print <аргумент>; - вывести аргумент  
Пример: print "Hello World!";
```

```
echo <аргумент>; - вывести группу аргументов  
Пример: echo "Hello World!", "Hello Earth!", "Hello Sky!";
```

```
printf (<формат>[, <аргументы>,...]); - C-подобная функция  
Пример: printf("my course is %d%s\n", 4, "<br>");
```

## 4. Пример PHP-скрипта:

```
<html>  
<body>  
<?php  
print "Hello Programmer!<br>";  
for ($i = 0; $i < 10; $i++)  
    printf("Serial number is %d<br>", $i);  
    printf("Random number is %d<br>", mt_rand(0,100));  
?>  
</body>  
</html>
```

## Самостоятельная работа № 3 Обмен информацией между WEB-сервером и клиентом

**Задачи:** Изучить следующие приемы:

- Изучить конструкции HTML для организации обмена между сервером и клиентом
- Разобраться с методами передачи данных GET и POST
- Изучить возможности PHP для получения информации от пользователя

## Самостоятельная часть

### Программа «Получение анкетных данных от пользователя».

1. Разработать HTML-файл для возможности получения от пользователя следующих данных:

- ФИО – поля редактирования
- Пол (мужской, женский) – радио-кнопки
- семейное положение (холост, женат/замужем) – радио-кнопки
- образование (среднее, высшее и т.п.) – комбинированный список
- хобби (спорт, охота, рыбалка, компьютеры, пиво, ТВ и т.п.) – список с множественным выбором
- есть ли компьютер – кнопка-переключатель

2. Разработать PHP-скрипт, анализирующий введенные данные и генерирующий страничку вида:

Иванов Иван Иванович

Пол: *мужской*

Семейное положение: *холост*

Образование: *высшее*

Хобби: *охота, рыбалка*

Наличие ПК дома: *есть*

3. Использовать поочередно в п.1 методы GET и POST. Посмотреть в чем различия. Разобрать формат передачи параметров в методе GET.

## Теоретическая часть

Примеры создания полей ввода и кнопок можно посмотреть в лекции или спецификации языка HTML. Используемые тэги:

<FORM> - создание формы

<INPUT> - создание полей редактирования, кнопок различных видов

<SELECT> - создание списка выбора и комбинированного списка

## Самостоятельная работа № 4

**Цель:** Разработка диалоговых систем с помощью Web-технологий.

### Задачи:

1. Научиться переносить данные между страницами
2. Научиться переадресовывать запросы

**Требования к системе.** Разработать систему, выполняющую следующие функции:

1. Авторизация введенных данных о пользователе. После авторизации каждый каждая страничка должна содержать ФИО «залогинившегося» пользователя
2. Ввод данных от пользователя
3. Редактирование введенных данных
4. Просмотр введенных данных

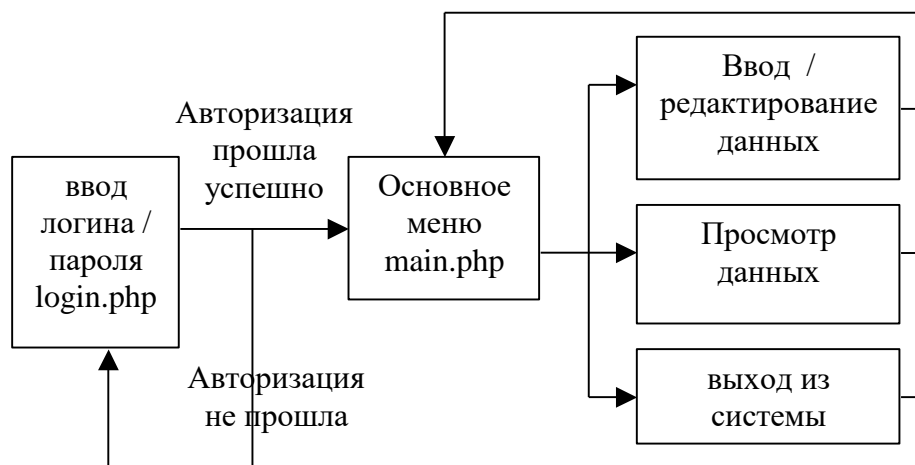
Система состоит из 4-х модулей:

1. Страница ввода идентификационных данных (логин/пароль)
2. Главное меню



3. Страница ввода/редактирования данных
4. Просмотр введенных данных

Функциональная схема работы:



### Самостоятельная реализация

Данная Самостоятельная работа опирается на предыдущую лабораторную работу №3 «Обмен информацией между WEB-сервером и клиентом». В качестве примера для реализации можно взять систему работы с анкетными данными. Работа системы начинается со страницы «ввод логина/пароля». В случае удачной аутентификации пользователь попадает на «основное меню», из которого он может ввести/поменять/просмотреть данные о себе или выйти из системы. Переадресация на другую страницу выполняется с помощью команды протокола HTTP, реализуемую через:

```
print header ("Location: login.php");
exit;
```

Данная команда должна идти первой из выводимого на поток вывода (до тега <html>), иначе переадресации не случится.

**Требование к безопасности:** «неавторизованные» пользователи не должны видеть ни одной странички, кроме «ввод логина/пароля». В случае, если «неавторизованный» пользователь пытается вызвать любую страницу, система должна адресовать его на страницу «ввод логина/пароля».

### Простейший пример реализации механизма аутентификации.

Создаются две функции:

**get\_user\_id(\$login, \$password);** - в случае успешной проверки возвращает идентификатор пользователя, в случае неудачи – возвращает -1.

**get\_correct\_login(\$user\_id);** - проверяет корректность идентификации. В случае неудачи переадресует на страницу «ввод логина/пароля».

Файл «authlib.inc»

```
<?php
$passwd[“Ivanov”][“id”] = 1289;
$passwd[“Ivanov”][“password”] = “pass1”;
$passwd[“Petrov”][“id”] = 1423;
```

```

$passwd["Petrov"]["password"] = "table";
$passwd["Sidorov"]["id"] = 4598;
$passwd["Sidorov"]["password"] = "nothing";

function get_user_id($login, $password) {
GLOBAL $passwd;
if ($passwd[$login]["password"] == $password) {
    return $passwd[$login]["id"];
} else {
    return -1;
}
}

function get_correct_login($user_id) {
GLOBAL $passwd;
return in_array($user_id, $passwd);
}

?>

```

После чего можно использовать эту библиотеку в страницах командой  
include("authlib.inc");

После чего нужно идентифицировать пользователя:

```

$id = get_user_id($login, $password);
if ($id == -1) {
print header ("Location: login.php");
exit;
}

```

Для передачи параметров между страницами в формах используется тэг

<INPUT type="HIDDEN">. Данный тег не отображается на форме визуально и предназначен для передачи скрытых параметров, принцип передачи – как у <INPUT type="TEXT">.

Пример передачи идентификатора пользователя:

```

<INPUT type="HIDDEN" name="user_id" value="<?php print $user_id; ?>">

```

После чего для организации безопасности каждый файл можно начинать со строки:

```

$id = get_correct_login($user_id);
if ($id == -1) {
print header ("Location: login.php");
exit;
}

```

## Самостоятельная работа № 5

### *Работа с файлами в PHP на примере организации чата*

## Постановка задачи

Реализовать свой чат на базе файлов. Чат – программа обмена сообщениями, мнениями и т.п., строящаяся по принципу доски объявлений: каждый посылает сообщение от своего имени, доступное для просмотра всем. Типичный чат обладает рядом следующих правил:

- чтобы войти в чат, необходимо пройти регистрацию: задать ваше сетевое имя и пароль (чтобы никто не мог писать сообщения от вашего имени)
- имя или сообщение каждого пользователя выделяется своим цветом, выбираемым пользователем на этапе регистрации
- на экране показываются только последние 10-30 сообщений
- через определенный период времени чат сам обновляет поле текста с сообщениями с сервера
- когда пользователь входит в чат или выходит с чата, появляется сообщение об этом

## Теоретическая часть

Команды работы с файлами изложены (или будут изложены) на лекции. Так же их можно найти в спецификации языка PHP или любой книге по PHP.

Для реализации периодического обновления страницы используются следующие приемы:

- обработка таймера на одном из скриптовых языков (JavaScript или VBScript) – выходит за рамки нашего курса
- посылка соответствующей команды HTTP-протокола браузеру перед выводом HTML-страницы – самый правильный способ, поскольку скрипты на стороне клиента можно отключать в настройках браузера. Реализуется следующим образом - в начале php-файла выполняются две команды:  
header("Refresh: 10"); // здесь 10 – время задержки, 10 секунд до обновления  
header("URL: chat.php"); // chat.php – url-адрес файла, на который перейти  
// после задержки

Переадресация (redirect) файла выполняется с помощью команды:

```
header ("Location: enter.php");
```

// enter.php – файл, на который выполняется переадресация

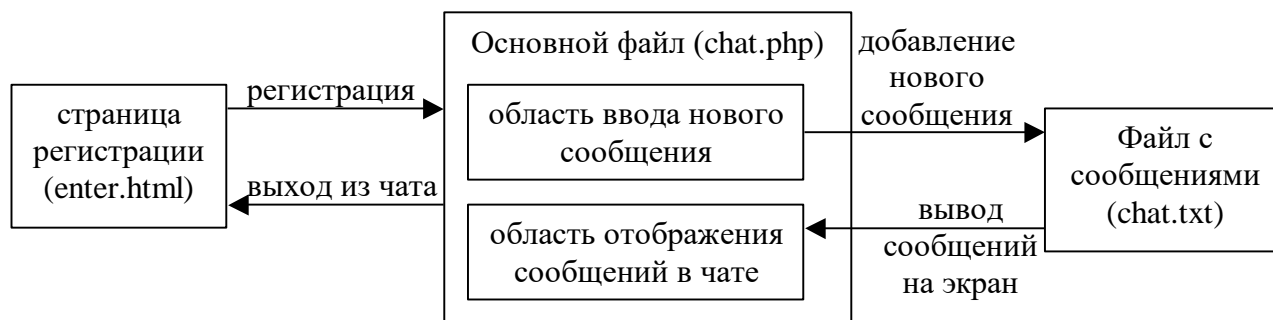
Следует помнить, что при переадресации полученные с помощью html-форм данные теряются, поэтому, если их нужно сохранить, то нужно их дописать в конец url по принципу метода GET: file?var1=val1&var2=val2&var3=val3 и т.п., т.е.

```
header ("Location: enter.php?family=Ivanov&name=Ivan");
```

Невозможно передать при переадресации данные методом POST!

## Простейший пример реализации

Простейшая реализация чата состоит из трех файлов и выглядит следующим образом:



Основной файл (chat.php) формирует экран пользователя, состоящий из двух частей:

- область ввода нового сообщения (форма с полем ввода сообщения и кнопкой отправки сообщения) отвечает за формирование (дополнение) файла chat.txt
- область показа существующих сообщений (интерпретация и отображение файла chat.txt)

Кроме того, файл chat.php сам же добавляет новые данные в файл chat.txt, для этого надо:

- в форме в свойстве action надо вписать "chat.php"
- в начале файла chat.php надо проверять ввод новых данных: если ввели, то добавить их в файл chat.txt

Чтобы постоянно пользователь видел самые новые сообщения, необходимо реализовать механизм, заставляющий браузер периодически обновлять данные (вызывать файл chat.php без помощи пользователя)

Простейшая схема файла chat.php выглядит следующим образом:

```
<?php
header("Refresh: 10"); // обновить чат через 10 секунд
header("URL: chat.php");
?>
<html> <body>
<form method="post" action="chat.php">
<input type="text" name="msg">
<input type="submit" value="Послать">
</form><br>
<?php
if ($msg) { // проверка ввода нового сообщения
    $fptr = fopen("chat.txt", "a"); // открыть файл на добавление
    fwrite($fptr, $msg."<br>");
    fclose($fptr);
}
// вывести файл на стандартный поток вывода
if (file_exists("chat.txt")) readfile("chat.txt");
?>
</body> </html>
```

**P.S.** На самом деле подобная реализация обладает массой существенных недостатков, которые не позволят «жить» этому чату в реальных условиях. Подумайте, что это за

недостатки и как их решить – обязательно спрошу.

### **Самостоятельная работа № 6** **Знакомство с СУБД MySQL**

#### **Задание**

Разработать систему «телефонный справочник».

База данных системы должна включать, как минимум, следующую информацию:

1. Фамилия, имя, отчество.
2. Телефон.
3. Адрес проживания.

Система должна выполнять следующие операции:

1. Ввод и корректировка данных администратором системы.
2. Поиск и вывод информации о человеке по фамилии.
3. Поиск и вывод информации о человеке по телефону.

Система состоит из двух независимых блоков:

1. Блок ввода информации в базу администратором
2. Блок вывода информации по запросам пользователей

**Блок вывода информации по запросам пользователей** состоит из двух модулей. Первый модуль запрашивает информацию для поиска в базе (фамилию или телефон). Второй модуль, на основании критериев поиска, формирует запрос к базе, получает ответ и формирует вывод результатов поиска в виде таблицы (см. л/р № 2).

**Блок ввода информации в базу** позволяет вводить и изменять данные только администратору БД. Перед работой в этом блоке необходима проверка прав доступа (см. л/р № 4)

Вся необходимая информация для выполнения лабораторной работы выложена на Avalon (это руководства по PHP и MySQL), а также была (или будет) изложена на лекциях.

### **Самостоятельная работа № 7** **Знакомство с СУБД PostgreSQL**

**Цель работы:** Познакомиться с СУБД PostgreSQL и разобраться с методикой работы с СУБД PostgreSQL из языка PHP.

#### **Задание**

Разработать систему «телефонный справочник».

База данных системы должна включать, как минимум, следующую информацию:

4. Фамилия, имя, отчество.
5. Телефон.
6. Адрес проживания.

Система должна выполнять следующие операции:

4. Ввод и корректировка данных администратором системы.
5. Поиск и вывод информации о человеке по фамилии.
6. Поиск и вывод информации о человеке по телефону.

Система состоит из двух независимых блоков:

3. Блок ввода информации в базу администратором
4. Блок вывода информации по запросам пользователей

**Блок вывода информации по запросам пользователей** состоит из двух модулей. Первый модуль запрашивает информацию для поиска в базе (фамилию или телефон). Второй модуль, на основании критериев поиска, формирует запрос к базе, получает ответ и формирует вывод результатов поиска в виде таблицы (см. л/р № 2).

**Блок ввода информации в базу** позволяет вводить и изменять данные только администратору БД. Перед работой в этом блоке необходима проверка прав доступа (см. л/р № 4)

Вся необходимая информация для выполнения лабораторной работы выложена на Avalon (это руководства по PHP и PostgreSQL), а также была (или будет) изложена на лекциях.

**Примечание к заданию:** Логика работы программы в данной лабораторной работе полностью совпадает с лабораторной работой № 6. Отличиями являются используемый SQL-сервер и, следовательно, функции работы с ним. Функции работы с PostgreSQL работают почти так же, как и аналогичные MySQL-функции.

## Самостоятельная работа № 8

### Разработка кросс платформенной информационной системы.

**Цель работы:** Познакомиться с библиотекой PEAR DB (унифицированный API для доступа к SQL-базам данных). Научиться использовать библиотеку PEAR DB для разработки информационной системы, независимой от типа выбираемого SQL-сервера.

### Задание

Разработать программное обеспечение для работы в заданной предметной области. Программное обеспечение является независимым от типа SQL-сервера (для обеспечения независимости используется библиотека PEAR DB). Работоспособность системы будет проверяться на MySQL и PostgreSQL.

На выбор предлагаются следующие предметные области:

- деканат института
- школа
- больница
- трамвайно-троллейбусное депо
- магазин
- склад
- еще что-нибудь того же уровня (для желающих быть оригинальными)

*Программное обеспечение состоит из следующих частей:*

1. **Блок инициализации** – создает начальные структуры таблиц на SQL-сервере и производит заполнение базовых справочников. Возможны два варианта реализации:
  - Два текстовых файла с набранными командами на SQL (два – из-за отличий в синтаксисе MySQL и PostgreSQL, хотя можно и одним обойтись, если придерживаться стандартов). Файл(ы) через FTP записываются на сервер (Avalon), затем запускаются с консолей соответствующих SQL-серверов (mysql и psql).
  - Скрипт на PHP, который присоединяется к БД и создает все сам через метод `exec`.
2. **Блок корректировки** - заносит новую информацию предметной области в БД и справочники, корректирует имеющиеся данные.
3. **Блок генерации отчетов** – предоставляет возможность генерации отчетов по предметной области (естественно, в виде HTML).

*Предметная область должна содержать (как минимум):*

1. 3 таблицы (1 таблица + 2 ассоциированных справочника)
2. механизм работы с данными в этих таблицах
3. 2 любых отчета по этим таблицам на тематику предметной области

Описание библиотеки PEAR DB и другая необходимая информация по PHP и SQL находится на Web-сервере (Avalon) в разделе «Документация» на русском и английском языках.

### **Примерная реализация предметной области «Деканат»**

1. Необходимы следующие справочники и таблицы:
  - Справочник «Учебные курсы» имеет поля (код предмета, наименование)
  - Справочник «Студенты» имеет поля (код студента, ФИО, группа, номер зачетки)
  - Таблица «Ведомость» имеет поля (код студента, код предмета, оценка)
2. Необходимо реализовать следующие механизмы доступа и работы с данными:
  - Добавление/изменение/удаление информации из справочника «Учебные курсы»
  - Добавление/изменение/удаление информации из справочника «Студенты»
  - Добавление/изменение/удаление информации из таблицы «Ведомость»
3. Необходимо реализовать подготовку и вывод следующих отчетов:
  - «Зачетная книжка» студента – по выбранной фамилии выводятся все предметы и оценки по ним для конкретного студента
  - «Зачетный лист» группы – для выбранной группы и предмета формируется список студентов группы и оценок по предмету

### **Тест**

1. Укажите правильный вариант определения изображения в качестве гиперссылки.

- a. <a HREF="адрес файла"> IMG SRC="image.gif">
- b. <a HREF="image.gif">
- c. <a HREF="адрес файла"> <IMG="image.gif">

Ответ: а

2. Найдите ошибочное определение гиперссылки.

- a. <a HREF="alexfine.htm" TARGET="left"> alexfine
- b. <a TARGET="alexfine.htm" HREF="new"> alexfine
- c. <a HREF="alexfine.htm"> alexfine

Ответ: b

3. В какой таблице ширина промежутков между ячейками составит 20 пикселей?

- a. <table cellpadding="20">
- b. <table gridspacing="20">
- c. <table cellspacing="20">

Ответ: а

4. Как указать выравнивание текста в ячейке таблицы?

- a. с помощью атрибута CELLPADDING
- b. с помощью атрибута VALIGN
- c. с помощью атрибута ALIGN

Ответ: b, c

5. Какой атрибут элемента FORM определяет список кодировок для вводимых данных?

- a. altb. accept-charset
- c. enctype-charset

Ответ: b

6. Что определяет атрибут CELLSPACING у элемента разметки TABLE?

- a. расстояние от содержания до границы ячейки
- b. расстояние между ячейкамис. ширину границы
- d. ширину ячейки

Ответ: b

7. Какой атрибут тэга BODY позволяет задать цвет фона страницы?

- a. color
- b. background
- c. set
- d. bgcolor

Ответ: d

8. Какой атрибут тега <img> задает горизонтальное расстояние между вертикальной границей страницы и изображением?

- a. BORDER
- b. HSPACE
- c. VSPACE

Ответ: b

9. Какой из приведенных тегов позволяет создавать нумерованные списки?

- a. OL
- b. DL
- c. UL



d. DT

Ответ: a

10. Какой полный URL будет сформирован для ссылки в приведенном фрагменте? `<base href=""><a>http://alexfine.ru"> <BODY> <A HREF="doc1.html">Документ 1`

a. `http://alexfine.ru/docs/doc1.html`

b. `http://alexfine.ru/doc1.html`

c. правильный URL не может быть сформирован

Ответ: b

11. В каких случаях атрибут выравнивания `align` имеет более высокий приоритет?

a. `<TH align="left">`

b. `<COL align="left">`

c. `<TABLE align="left">`

Ответ: a

12. Какой атрибут принадлежит тегу `<AREA>`?

a. SRC

b. SHAPE

c. CIRCLE

Ответ: b

13. Какой тэг определяет заголовок документа HTML?

a. HTML

b. ISINDEX

c. BODY

d. HEAD

Ответ: d

14. Какой из приведенных примеров задает гипертекстовую ссылку из документа 1.html на другой документ?

a. `<A HREF="#m1">ссылка`

b. `<A HREF=m1>ссылка`

c. `<A HREF="2.html#m1">ссылка`

Ответ: c

15. Выберите вариант корректного описания синтаксиса тега `SCRIPT`.

a. `<scripT Type="тип_языка_программирования">текст программы`

b. `<scripT nAME="язык_программирования">текст программы`

c. `<scripT TYPE="тип_документа">текст программы`

Ответ: a

16. Какой из приведенных фрагментов кода создает переключатель?

a. `<input Type="checkbox" nAME="a1" vAlue="1"><input TYPE="checkbox" nAME="a1" vAlue="2"><input TYPE="text" nAME="a1" vAlue="2">`

b. `<input TYPE="radiobutton" nAME="a1" vAlue="1"><input TYPE="radiobutton" nAME="a1" vAlue="2">`

c. `<input TYPE="radio" nAME="a1" vAlue="1"><input TYPE="radio" nAME="a1" vAlue="2">`

Ответ: c

17. Какие значения атрибута `ALIGN` используются для определения положения изображения относительно окружающего текста?

- a. left
- b. bottom
- c. baseline
- d. right
- e. top

Ответ: a, b, c, d, e

18. В какой таблице текст выровнен по центру ячеек?

- a. `<table align=""center"" width=""300"">`
- b. `<table align=""left"">`
- c. нет правильного ответа
- d. `<table align=""left"">`

Ответ: c

19. Какие из приведенных тегов неверно описывают активное изображение?

- a.
- b.
- c.

Ответ: a, c

20. Какой тэг определяет тело документа HTML?

- a. META
- b. BODY
- c. HTML
- d. HEAD

Ответ: b

21. В каких примерах правильно организован синтаксис тега BASE?

- a. `<bAse href=""<a">http://www.alexfine.ru/intro.html" TARGET=new>`
- b. `<bAse A="" href="alexfine.ru/intro.html">`
- c. `<bAse href=""<a">http://www.alexfine.ru/intro.html">`

Ответ: a, c

22. В каком примере корректно описан элемент TR?

- a. `<TR> <TD>ячейка1`
- b. `<TD> <TR>ячейка1ячейка2<TD>`
- c. `<TR> <TD>ячейка1`

Ответ: a

23. Какой атрибут тега `<img>` указывает файл изображения и путь к нему?

- a. SRC
- b. ALT
- c. ALIGN

Ответ: a

24. Укажите неверные варианты описания синтаксиса тега SCRIPT.

- a. `<sCriP t nAME="язык_программирования">текст программы<scripT>`
- b. `<sCriP t TYPE="тип_документа">текст программы`
- c. `<sCriP t TYPE="тип_языка" программирования="">текст программы`

Ответ: a, b

25. В каком случае форма будет отправлена методом "post"?

- a. <fOrM method=""post"" action=""http://www.alexfine.ru/shop/"">
- b. <fOrM method=""post"" action=""http://www.alexfine.ru/shop/shop.pl"">
- c. <fOrM method=""default"" action=""http://www.alexfine.ru/shop/sp.pl"">
- e. <fOrM method=""get"" action=""http://www.alexfine.ru/"">

Ответ: a, b

26. Какой атрибут тега BODY позволяет изменять цвет "активных" гиперссылок?

- a. COLOR
- b. VLINK
- c. ALINK
- d. TEXT

Ответ: c

27. В каких примерах данные формы будут переданы обработчику как часть URL?

- a. <fOrM method=""get"" action=""http://www.alexfine.ru/"">
- b. <fOrM method=""post"" action=""http://www.alexfine.ru/help/first.pl"">
- c. <fOrh1 method=""try"" action=""http://www.alexfine.ru/help/script.php?param=test"">
- d. <fOrh1 method=""get"" action=""http://www.alexfine.ru/cgi"">
- e. <fOrh1 method=""post"" action=""mailto:info@alexfine.ru"">

Ответ: a, d

28. HTML - это:

- a. язык редактирования
- b. язык структурной разметки
- c. язык программирования
- d. язык гипертекстовой разметки

Ответ: d

29. С помощью какого элемента можно создавать прокручивающиеся списки в формах?

- a. TEXTAREA
- b. TR
- c. SELECT
- d. INPUT

Ответ: c

30. Какие методы можно применять для отправки формы?

- a. POST
- b. TRY
- c. PUT
- d. HEAD
- e. GET
- f. MAILTO

Ответ: a, e

### КРИТЕРИИ ОЦЕНКИ

Тест: за каждый правильный ответ – 1 балл.

оценка «5» - 90-100%;

оценка «4» - 70-89%;

оценка «3» - 69-50%;

оценка «2» - менее 49%.

### Критерии оценки внеаудиторной (самостоятельной) работы

Процент результат ивности	Балл (оцен ка)	Критерии оценивания
90-100%	5	<ul style="list-style-type: none"> <li>– глубокое изучение учебного материала, литературы и нормативных актов по вопросу;</li> <li>– правильность формулировок, точность определения понятий;</li> <li>– последовательность изложения материала;</li> <li>– обоснованность и аргументированность выводов;</li> <li>– правильность ответов на дополнительные вопросы;</li> <li>– своевременность выполнения задания.</li> </ul>
70-89%	4	<ul style="list-style-type: none"> <li>– полнота и правильность изложения материала;</li> <li>– незначительные нарушения последовательности изложения;</li> <li>– неточности в определении понятий;</li> <li>– обоснованность выводов приводимыми примерами;</li> <li>– правильность ответов на дополнительные вопросы;</li> <li>– своевременность выполнения задания.</li> </ul>
50-69%	3	<ul style="list-style-type: none"> <li>– знание и понимание основных положений учебного материала;</li> <li>– наличие ошибок при изложении материала;</li> <li>– непоследовательность изложения материала;</li> <li>– наличие ошибок в определении понятий, искажающих их смысл;</li> <li>– несвоевременность выполнения задания.</li> </ul>
0-49%	2	<ul style="list-style-type: none"> <li>– незнание, невыполнение или неправильное выполнение большей части учебного материала;</li> <li>– ошибки в формулировке определений, искажающие их смысл;</li> <li>– беспорядочное и неуверенное изложение материала;</li> <li>– отсутствие ответов на дополнительные вопросы;</li> <li>– отсутствие выводов и неспособность их сформулировать;</li> <li>– невыполнение задания.</li> </ul>