

Сплошная практика. Создаем динамический сайт

Структура будущего сайта

Создадим полностью динамический сайт с использованием всех изученных технологий, например, каталог ресурсов Интернета. Сайт будет управляться из одного файла и состоять из трех разделов — общедоступного, раздела для зарегистрированных пользователей и раздела для администратора сайта.

В общедоступном разделе создадим следующие файлы:

- ☐ index.php — центральная страница сайта, на которой выводится рубрикатор каталога;
- ☐ catalog.php — вывод сайтов по рубрике с разбиением на страницы;
- ☐ search.php — для поиска по каталогу с разбиением на страницы;
- ☐ contact.php — страница с формой обратной связи;
- ☐ gbook.php — гостевая книга с разбиением на страницы;
- ☐ err401.php — файл с сообщением об ошибке 401 (не авторизован);
- ☐ err403.php — файл с сообщением об ошибке 403 (нет доступа);
- ☐ err404.php — файл с сообщением об ошибке 404 (ресурс не найден);
- ☐ err401.txt — для регистрации ошибки 401;
- ☐ err403.txt — для регистрации ошибки 403;
- ☐ err404.txt — для регистрации ошибки 404.

В разделе для зарегистрированных пользователей (каталог user) создадим файлы:

- ☐ index.php — страница с формой входа в систему, с формой регистрации нового пользователя и формой восстановления пароля;
- ☐ add.php — для регистрации нового сайта;
- ☐ exit.php — для выхода из системы.

В разделе для администратора сайта (каталог admin) создадим файлы:

- ☐ index.php — для вывода статистики и сообщений об ошибках;
- ☐ rubr.php — для добавления новой рубрики, удаления или переименования существующей;

- ❑ `moder.php` — для вывода сайтов, находящихся на модерации, и результатов поиска по URL-адресу с возможностью одобрения или удаления выделенных сайтов;
- ❑ `catalog.php` — для редактирования описания сайта;
- ❑ `gbook.php` — для просмотра содержимого гостевой книги с выделением новых сообщений и с возможностью одобрения или удаления выделенных сообщений.

Дополнительно создадим файл `allscript.php`, который будет содержать все функции. Редактируя этот файл, можно изменить весь дизайн сайта. В файле опишем постоянные величины (константы), которые будем использовать при создании страниц (листинг 1).

Листинг 1. Описание констант

```
// URL-адрес сайта (например, http://www.site.ru/)
define('URL_SITE', '');

define('WIDTH_TABLE_1', '760'); // Ширина таблиц 1-го уровня
define('WIDTH_TABLE_2', '600'); // Ширина таблиц 2-го уровня
define('WIDTH_TABLE_3', '100%'); // Ширина таблиц 3-го уровня

// Данные для подключения к базе данных
define('HOST_CONNECT', ''); // Сервер
define('LOGIN_CONNECT', ''); // Логин
define('PASSW_CONNECT', ''); // Пароль
define('DB_CONNECT', ''); // База данных

// Данные для писем (например, support <unicross@mail.ru>)
define('MAIL_POST', '');
// Для формы обратной связи (например, unicross@mail.ru)
define('MAIL_ADRES', '');

define('TIME_SESS', 3600); // Время жизни сессии в секундах

// Количество сообщений в гостевой книге
define('COUNT_POS_PAGE_GBOOK', 10);
// Количество сайтов на странице
define('COUNT_POS_PAGE', 10);
```

Создание шаблона сайта

Создание любого сайта начинается с разработки дизайна, так как все страницы должны быть одинаково оформлены. Шаблон сайта будет состоять из трех частей:

- ❑ верхний колонтитул — в этой части мы разместим логотип сайта, рекламный баннер, панель навигации и поисковую форму, а также таблицу стилей и функции на JavaScript;
- ❑ основная часть страницы — здесь будет располагаться основное содержимое, которое будет меняться в зависимости от назначения страницы;
- ❑ нижний колонтитул — тут будет выводиться различная информация для пользователя и администратора. Для пользователей раздел будет содержать только панель навигации, а для администратора — поисковую форму и специальную панель навигации. В дальнейшем в этом разделе можно разместить счетчики и логотипы различных каталогов.

Все части сайта опишем в виде отдельных функций, которые разместим в файле `allscript.php`. Этот файл мы будем подключать во всех страницах с помощью оператора `require_once`.

Шаблон страниц сайта для пользователей будет выглядеть, как показано в листинге 2.

Листинг 2. Шаблон страниц сайта для пользователей

```
<?php
define('IN_CATALOG', true);
require_once('allscript.php');
// Заголовок
$title = "";
// Описание страницы
$description = "";
// Ключевые слова для поисковых машин
$keywords = "";
// Выводим верхний колонтитул
f_header_all($title, $description, $keywords);
?>

<!-- Основное содержание страницы -->

<?php
f_footer_user(); // Выводим нижний колонтитул
?>
```

Для администратора сайта шаблон будет немного другим (листинг 3).

Листинг 3. Шаблон страниц сайта для администратора

```

<?php
define('IN_CATALOG', true);
require_once('allscript.php');
// Заголовок
$title = "";
// Описание страницы
$description = "";
// Ключевые слова для поисковых машин
$keywords = "";
// Выводим верхний колонтитул
f_header_all($title, $description, $keywords, 2);
?>

<!-- Основное содержание страницы -->

<?php
f_footer_admin(); // Выводим нижний колонтитул
?>

```

Заголовок страницы, описание содержимого страницы и ключевые слова должны отличаться. Для этого мы передаем эти параметры в функцию `f_header_all()`, создающую верхний колонтитул.

Создание верхнего колонтитула

Верхний колонтитул выводится с помощью функции `f_header_all()`. Функция имеет следующий формат:

```
f_header_all(<1>, <2>, <3>, <4>, <5>)
```

- ❑ <1> — заголовок страницы (тег <TITLE>).
- ❑ <2> — описание страницы (тег <META>, параметр `description`).
- ❑ <3> — ключевые слова для поисковых машин (тег <META>, параметр `keywords`).
- ❑ <4> — если параметр равен 1, то страница разрешена для индексации поисковыми роботами (значение по умолчанию). В страницу будет вставлена строка

```
<meta name="robots" content="index, follow">
```

Если любое другое значение, то индексация запрещена. В страницу будет вставлена строка

```
<meta name="robots" content="noindex, nofollow">
```

- ❑ <5> — поисковая фраза, которая отобразится в строке поисковой формы. По умолчанию пустая строка.

На самом деле `f_header_all()` лишь промежуточная функция, которая запускает на выполнение несколько других функций (листинг 4).

Листинг 4. Функция `f_header_all()`

```
function f_header_all($t, $d, $k, $r=1, $s="") {  
    // Заголовок для всех страниц  
    f_header($t, $d, $k, $r); // Выводим верхний колонтитул  
    f_logo(); // Выводим логотип и баннер  
    f_menu(); // Выводим панель навигации  
    f_table(); // Выводим таблицу-разделитель  
    f_search($s); // Выводим форму поиска  
    f_table(); // Выводим таблицу-разделитель  
    f_table_center_start(); // Начало таблицы для основного содержания  
}
```

Рассмотрим каждую функцию по отдельности.

Вывод верхнего колонтитула

Для вывода верхнего колонтитула используется функция `f_header()` (листинг 5).

Листинг 5. Функция `f_header()`

```
function f_header($t, $d, $k, $r=1) {  
    // Верхний колонтитул  
    ?>  
    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
        "http://www.w3.org/TR/html4/loose.dtd">  
  
    <html>  
    <head>  
  
    <title><?php echo $t; ?></title>  
  
    <meta name="description" content="<?php echo $d; ?>">  
    <meta name="keywords" content="<?php echo $k; ?>">  
    <meta http-equiv="content-type" content="text/html; charset=windows-1251">  
  
    <?php  
        if ($r==1) {
```

```

        echo '<meta name="robots" content="index, follow">' . "\n";
    }
    else {
        echo '<meta name="robots" content="noindex, nofollow">' . "\n";
    }
    f_style(); // Выводим таблицу стилей
?>
<script type="text/javascript">
<!--
window.status="<?php echo $t; ?>";
function f_add_favor() {
    external.addFavorite("<?php echo URL_SITE; ?>", "<?php echo $t; ?>");
    return false;
}
function f_HomePage(obj) {
    obj.style.behavior="url(#default#homepage)";
    obj.setHomePage("<?php echo URL_SITE; ?>");
    return false;
}
//-->
</script>
</head>
<body>
<?php
}

```

Итак, функция `f_header()` выводит полностью раздел `HEAD` HTML-документа, включая заголовок, описание страницы и ключевые слова для поиска, информацию о кодировке страницы и возможности индексации. Кроме того, с помощью функций JavaScript заголовок выводится в статусной строке Web-браузера, функция `f_add_favor()` позволяет добавить сайт в избранное, а функция `f_HomePage()` позволяет сделать главную страницу сайта стартовой (первой страницей, которую увидит пользователь при запуске Web-браузера).

Внутри функции `f_header()` используется константа `URL_SITE`, которая содержит URL-адрес сайта. Обратите внимание: константа, в отличие от переменной, видна внутри функции.

Создание таблицы стилей

Внутри функции `f_header()` вызывается функция `f_style()`, которая выводит таблицу стилей (листинг 6).

Листинг 6. Функция `f_style()`

```
function f_style() {
?>

<!-- Таблица стилей Начало -->
<style type="text/css">
  A:link { text-decoration: none; font-weight: bold; color: #000000 }
  A:visited { text-decoration: none; font-weight: bold; color: #000000 }
  A:hover { text-decoration: underline; font-weight: bold;
color: #000000 }
  BODY { font-family: "Verdana", "Tahoma", sans-serif; font-size: 11px }
  BODY { margin-top: 0; background-color: #FFFFFF }
  TABLE {
    font-family: "Verdana", "Tahoma", sans-serif;
    font-size: 12px;
    background-color: #FFFFFF
  }
  H1 {
    font-family: "Tahoma", "Verdana", sans-serif;
    font-size: 16px;
    font-weight: bold;
    text-align: center
  }
  .err {
    font-size:12px;
    font-weight: bold;
    text-align: center;
    color: #FF0000
  }
  .ok {
    font-size:12px;
    font-weight: bold;
    text-align: center;
```

```
        color: #008000
    }
    .bold { font-weight: bold }
    .color_table { background-color: #E8EAF1 }
    .logo-table { text-align: center; vertical-align: middle }
    .menu-table {
        text-align: center;
        vertical-align: middle;
        background-color: #43568E
    }
    A.menu:link {
        text-decoration: none;
        font-weight: bold;
        color: #FFFFFF;
        font-size: 11px;
        font-family: "Verdana", "Tahoma", sans-serif
    }
    A.menu:visited {
        text-decoration: none;
        font-weight: bold;
        color: #FFFFFF;
        font-size: 11px;
        font-family: "Verdana", "Tahoma", sans-serif
    }
    A.menu:hover {
        text-decoration: underline;
        font-weight: bold;
        color: #FFFFFF;
        font-size: 11px;
        font-family: "Verdana", "Tahoma", sans-serif
    }
    .search-table {
        text-align: center;
        vertical-align: middle;
        background-color: #FFFFFF
    }
    .txt_frm {
        background-color: #FFFFFF;
```



```
        font-size: 8pt;
        color: #000000;
        font-weight: bold;
        border-bottom: 1px solid;
        border-right: 1px solid;
        border-left: 1px solid;
        border-top: 1px solid;
        font-family: "Verdana", "Tahoma", sans-serif
    }
    .textarea_frm {
        background-color: #FFFFFF;
        font-weight: bold;
        font-size: 8pt;
        color: #000000;
        border-bottom: 1px solid;
        border-right: 1px solid;
        border-left: 1px solid;
        border-top: 1px solid;
        font-family: "Verdana", "Tahoma", sans-serif
    }
    .select_frm {
        margin-top: 3px;
        background-color: #FFFFFF;
        font-family: "Verdana", "Tahoma";
        font-size: 8pt;
        font-weight: bold
    }
    .text {
        font-family: "Verdana", "Tahoma", sans-serif;
        font-size: 12px;
        color: #000000;
        vertical-align: top;
        border: 1px solid silver;
    }
    .search_frm { margin-top: 0; margin-bottom: 0 }
</style>
<!-- Таблица стилей Конец -->
```

```
<?php
}
```

Обратите внимание, как мы выводим определения стилей внутри функции:

```
function f_style() {
?>

<!-- Таблица стилей Начало -->
<style type="text/css">
Определения стилей
</style>
<!-- Таблица стилей Конец -->

<?php
}
```

Такой метод вывода является вполне допустимым и не вызовет ошибки. Более того, этот метод удобен для дальнейшего редактирования таблицы стилей.

Итак, с помощью псевдостилей гиперссылок мы задаем внешний вид всех ссылок в документе. Для "тела" документа задаем размер и тип шрифта и цвет фона. Определяем стиль по умолчанию для таблиц и заголовков первого уровня. Для сообщений об ошибках создаем стиль `.err`, а для сообщений об удачной операции — стиль `.ok`. Кроме того, для выделения фрагментов текста определим стиль `.bold` и будем использовать его вместо тега ``.

Для таблицы, используемой при форматировании панели навигации, определяем стиль `.menu-table`, а внешний вид ссылок задаем стилем `.menu`.

С помощью стиля `.txt_frm` будем описывать текстовое поле, а с помощью `.textarea_frm` — поле для ввода многострочного текста. Для списков определен стиль `.select_frm`.

Для ячейки таблицы, содержащей логотип и баннер, предназначен стиль `.logo-table`. Ячейку, в которой находится поисковая форма, описывает стиль `.search-table`, а для поисковой формы предназначен стиль `.search_frm`. И, наконец, ячейка таблицы, в которой находится основное содержимое страницы, описывается с помощью стиля `.text`.

Вывод логотипа и рекламного баннера

Для размещения логотипа и баннера на странице мы используем таблицу высотой 70 пикселей. Ширина таблицы задается константой `WIDTH_TABLE_1`. Так как пока у нас нет баннера, то вместо кода запишем неразрывный пробел. Для размещения логотипа

необходимо создать еще одну ячейку справа или слева от баннера. Для вывода используется функция `f_logo()` (листинг 7).

Листинг 7. Функция `f_logo()`

```
function f_logo() {
?>

<!-- Логотип и баннер Начало -->
<table width="<?php echo WIDTH_TABLE_1; ?>" align="center" border="0"
cellspacing="0" cellpadding="0">
<tr><td class="logo-table" height="70">
&ampnbsp
</td></tr></table>
<!-- Логотип и баннер Конец -->
<?php
}
```

Вывод панели навигации

Панель навигации выводится с помощью функции `f_menu()` (листинг 8). Внутри функции мы используем константу `URL_SITE`.

Листинг 8. Функция `f_menu()`

```
function f_menu() {
?>

<!-- Панель навигации Начало -->
<table align="center" width="<?php echo WIDTH_TABLE_1; ?>" border="0"
cellpadding="0" cellspacing="0">
<tr class="menu-table"><td height="21">
<a href="<?php echo URL_SITE; ?>" class="menu">На главную</a>
</td><td>
<a href="<?php echo URL_SITE; ?>user/add.php" class="menu">Добавить
сайт</a></td><td>
<a href="<?php echo URL_SITE; ?>gbook.php" class="menu">Гостевая
книга</a></td><td>
<a href="<?php echo URL_SITE; ?>contact.php" class="menu">Обратная
связь</a></td><td>
<a href="#" onclick="return f_add_favor();" class="menu">Добавить в
Избранное</a></td><td>
<a href="#" onclick="return f_HomePage(this);" class="menu">Сделать
стартовой</a>
```

```
</td></tr>
</table>
<!-- Панель навигации Конец -->
<?php
}
```

Вывод таблицы-разделителя

Функция `f_table()` демонстрирует возможность применения таблицы для задания промежутка между элементами страницы (листинг 9). Вместо таблицы можно также использовать тег `<DIV>` или управлять внешними отступами в других элементах с помощью атрибута `margin`.

Листинг 9. Функция `f_table()`

```
function f_table() {
?>
<!-- Таблица-разделитель Начало -->
<table width="<?php echo WIDTH_TABLE_1; ?>" align="center" border="0"
cellspacing="0" cellpadding="0">
<tr><td height="5">&nbsp;</td></tr></table>
<!-- Таблица-разделитель Конец -->
<?php
}
```

Вывод поисковой формы

Поисковая форма является неотъемлемой частью любой страницы. Выводится форма с помощью функции `f_search()` (листинг 10).

Листинг 10. Функция `f_search()`

```
function f_search($txt='') {
?>
<!-- Поисковая форма Начало -->
<script type="text/javascript">
<!--
function f_submit_src() {
    var src = document.getElementById("search");
    if (src.value=="") {
        window.alert("Поле не заполнено!");
    }
}
```

```
        src.focus();
        return false;
    }
    if (src.value.length<3) {
        window.alert("В поле допустимо не менее 3 символов");
        src.focus();
        return false;
    }
    if (src.value.length>50) {
        window.alert("В поле допустимо не более 50 символов");
        src.focus();
        return false;
    }
    return true;
}
//-->
</script>
<table width="<?php echo WIDTH_TABLE_1; ?>" align="center" border="0"
cellspacing="0" cellpadding="0">
<tr><td class="search-table">
<form action="<?php echo URL_SITE; ?>search.php" class="search_frm"
onsubmit="return f_submit_src();">
<span class="bold">Поиск по каталогу: </span>
<input type="text" name="search" id="search" value="<?php echo $txt; ?>"
size="70">
<input type="submit" value="Найти">
</form>
</td></tr></table>
<!-- Поисковая форма Конец -->

<?php
}
```

При нажатии кнопки **Найти** проверяется значение, введенное в строку поиска, при помощи функции `f_submit_src()`. Если поле не заполнено, а также если длина меньше 3-х или больше 50-ти символов, то отправка данных формы прерывается, и пользователю выводится соответствующее сообщение.

Создание элементов основной части страницы

Вся основная часть страницы будет находиться в ячейке таблицы. Эту таблицу мы будем выводить по частям. Начало таблицы входит в состав верхнего колонтитула (листинг 11), а конец таблицы — в состав нижнего колонтитула (листинг 12).

Листинг 11. Функция `f_table_center_start()`

```
function f_table_center_start() {  
    ?>  
    <!-- Таблица для основного содержания страницы Начало -->  
    <table width="<?php echo WIDTH_TABLE_1; ?>" align="center" border="0"  
    cellpadding="0" cellspacing="10">  
    <tr><td height="300" class="text">  
    <?php  
    }
```

Листинг 12. Функция `f_table_center_end()`

```
function f_table_center_end() {  
    ?>  
    </td></tr></table>  
    <!-- Таблица для основного содержания страницы Конец -->  
    <?php  
    }
```

Кроме того, внутри ячейки основной таблицы мы создадим еще одну таблицу, которая в будущем может быть использована для вывода баннеров 120×240 или 120×600, а также для вывода новостей сайта или другой информации. Таблицу будем выводить также по частям. Начало таблицы выведем в функции `f_table_2_start()` (листинг 13), а конец таблицы — в функции `f_table_2_end()` (листинг 14).

Листинг 13. Функция `f_table_2_start()`

```
function f_table_2_start() {  
    ?>  
    <!-- Таблица второго уровня Начало -->  
    <table align="center" width="<?php echo WIDTH_TABLE_2; ?>"  
    bgcolor="#FFFFFF" border="0" cellpadding="2" cellspacing="0">  
    <tr><td valign="top">  
    <?php
```

```
}
```

Листинг 14. Функция `f_table_2_end()`

```
function f_table_2_end() {
?>
</td></tr></table>
<!-- Таблица второго уровня Конец -->
<?php
}
```

Для вывода описания конкретного сайта создадим функцию `f_table_site()` (листинг 15). Функция имеет следующий формат:

`f_table_site(<URL-адрес>, <Название сайта>, <Описание сайта>)`

Листинг 15. Функция `f_table_site()`

```
function f_table_site($url_site, $title_site, $descr_site) {
// Описание ресурса
?>
<table width="<?php echo WIDTH_TABLE_3; ?>" align="center" border="0"
cellspacing="0">
<tr><td class="color_table">
<a href="<?php echo $url_site; ?>" target="_blank"><?php echo
$title_site; ?></a></td></tr>
<tr><td><?php echo $descr_site; ?></td></tr></table><br>
<?php
}
```

Вывод сообщений в гостевой книге оформим с помощью функции `f_table_gbook()` (листинг 16). Функция имеет следующий формат:

`f_table_gbook(<Дата сообщения>, <Автор>, <Сообщение>);`

Листинг 16. Функция `f_table_gbook()`

```
function f_table_gbook($msg_date, $author, $msg) {
// Сообщение в гостевой книге
?>
<table width="<?php echo WIDTH_TABLE_3; ?>" align="center" border="0"
cellspacing="0">
<tr><td class="color_table">
<span class="bold"><?php echo $msg_date . ' ' . $author; ?></span>
</td></tr>
```

```
<tr><td><?php echo $msg; ?></td></tr></table><br>
<?php
}
```

Для вывода количества страниц создадим две функции `f_table_page_start()` (листинг 17) и `f_table_page_end()` (листинг 18). Первая функция выводит начало таблицы, а вторая — конец.

Листинг 17. Функция `f_table_page_start()`

```
function f_table_page_start() {
// Таблица для количества страниц Начало
?>

<table width="<?php echo WIDTH_TABLE_3; ?>" align="center" border="0"
cellspacing="0"><tr><td>
<?php
}
```

Листинг 18. Функция `f_table_page_end()`

```
function f_table_page_end() {
// Таблица для количества страниц Конец
?>

</td></tr></table><br>
<?php
}
```

Создание нижнего колонтитула

Нижние колонтитулы для пользователя и администратора будут отличаться. Для пользователей нижний колонтитул будет содержать только панель навигации, а для администратора — поисковую форму и специальную панель навигации.

Вывод нижнего колонтитула для пользователей

Для пользователей нижний колонтитул выводится с помощью функции `f_footer_user()` (листинг 19).

Листинг 19. Функция `f_footer_user()`

```
function f_footer_user() {
// Выводим нижний колонтитул для пользователей
```



```
f_table_center_end();  
// Выводим конец таблицы для основного содержания  
f_table(); // Выводим таблицу-разделитель  
f_menu(); // Выводим панель навигации  
f_footer (); // Выводим нижний колонтитул  
}
```

Практически все функции мы уже рассматривали за исключением одной — `f_footer()` (листинг 20).

Листинг 20. Функция `f_footer()`

```
function f_footer() {  
?>  
<!-- Нижний колонтитул -->  
</body></html>  
<?php  
}
```

В дальнейшем в этой функции можно разместить счетчики и логотипы различных каталогов.

Вывод нижнего колонтитула для администратора

Для администратора нижний колонтитул выводится с помощью функции `f_footer_admin()` (листинг 21).

Листинг 21. Функция `f_footer_admin()`

```
function f_footer_admin() {  
    // Выводим нижний колонтитул для администратора  
    f_table_center_end();  
    // Выводим конец таблицы для основного содержания  
    f_table(); // Выводим таблицу-разделитель  
    f_search_admin(); // Поисковая форма для админа  
    f_table(); // Выводим таблицу-разделитель  
    f_menu_admin(); // Выводим панель навигации  
    f_footer (); // Выводим нижний колонтитул  
}
```

Для администратора нижний колонтитул содержит две новые функции — `f_search_admin()` и `f_menu_admin()`. Функция `f_search_admin()` выводит поисковую форму для администратора (листинг 22).

Листинг 22. Функция `f_search_admin()`

```
function f_search_admin() {
?>

<!-- Поисковая форма для админа Начало -->
<table width="<?php echo WIDTH_TABLE_1; ?>" align="center" border="0"
cellspacing="0" cellpadding="0">
<tr><td height="25" class="search-table">
<form action="moder.php" class="search_frm">
<span class="bold">Поиск сайта по URL: </span>
<input type="text" name="search" size="70">
<input type="submit" value="Найти">
</form></td></tr></table>
<!-- Поисковая форма для админа Конец -->
<?php
}
```

Функция `f_menu_admin()` выводит панель навигации для администратора (листинг 23).

Листинг 23. Функция `f_menu_admin()`

```
function f_menu_admin() {
?>

<!-- Панель навигации для администратора Начало -->
<table align="center" width="<?php echo WIDTH_TABLE_1; ?>" border="0"
cellpadding="0" cellspacing="0">
<tr class="menu-table"><td height="21">
<a href="<?php echo URL_SITE; ?>admin/" class="menu">На
главную</a></td><td>
<a href="<?php echo URL_SITE; ?>admin/rubr.php"
class="menu">Рубрикатор</a></td><td>
<a href="<?php echo URL_SITE; ?>admin/gbook.php" class="menu">Гостевая
книга</a></td><td>
<a href="<?php echo URL_SITE; ?>admin/moder.php" class="menu">Сайты на
модерации</a></td>
</tr></table>
<!-- Панель навигации для администратора Конец -->
```

```
<?php  
}
```

Структура файла конфигурации

В итоге содержимое файла allscript.php должно выглядеть, как показано в листинге 24.

Листинг 24. Структура файла allscript.php

```
<?php  
// Если файл запускается из Web-браузера, то выходим  
if (!defined('IN_CATALOG')) exit;  
  
// Константы, определяющие конфигурацию системы  
// URL-адрес сайта (например, http://www.site.ru/  
define('URL_SITE', '');  
define('WIDTH_TABLE_1', '760'); // Ширина таблиц 1-го уровня  
define('WIDTH_TABLE_2', '600'); // Ширина таблиц 2-го уровня  
define('WIDTH_TABLE_3', '100%'); // Ширина таблиц 3-го уровня  
  
// Данные для подключения к базе данных  
define('HOST_CONNECT', ''); // Сервер  
define('LOGIN_CONNECT', ''); // Логин  
define('PASSW_CONNECT', ''); // Пароль  
define('DB_CONNECT', ''); // База данных  
  
// Данные для писем (например, support <unicross@mail.ru>  
define('MAIL_POST', '');  
// Для формы обратной связи (например, unicross@mail.ru)  
define('MAIL_ADRES', '');  
  
define('TIME_SESS', 3600); // Время жизни сессии в секундах  
  
// Количество сообщений в гостевой книге  
define('COUNT_POS_PAGE_GBOOK', 10);  
// Количество сайтов на странице  
define('COUNT_POS_PAGE', 10);
```

```
function f_db_connect() { // Листинг 25
    // Подключаемся к базе данных
}

function f_header_all($t, $d, $k, $r=1, $s="") { // Листинг 4
    // Заголовок для всех страниц
}

function f_header($t, $d, $k, $r=1) { // Листинг 5
    // Выводим заголовок
}

function f_search($txt="") { // Листинг 10
    // Вывод поисковой формы
}

function f_search_admin() { // Листинг 22
    // Поисковая форма для администратора
}

function f_style() { // Листинг 6
    // Таблица стилей
}

function f_logo() { // Листинг 7
    // Логотип и баннер
}

function f_menu() { // Листинг 8
    // Панель навигации
}

function f_menu_admin() { // Листинг 23
    // Панель навигации для администратора
}

function f_table() { // Листинг 9
    // Таблица-разделитель
}

function f_table_center_start() { // Листинг 11
    // Таблица для основного содержания страницы. Начало
}

function f_table_center_end() { // Листинг 12
    // Таблица для основного содержания страницы. Конец
}
```

```
function f_table_2_start() { // Листинг 13
// Таблица второго уровня. Начало
}
function f_table_2_end() { // Листинг 14
// Таблица второго уровня. Конец
}
function f_footer() { // Листинг 20
// Нижний колонтитул
}
function f_footer_user() { // Листинг 19
// Выводим нижний колонтитул для пользователей
}
function f_footer_admin() { // Листинг 21
// Выводим нижний колонтитул для администратора
}
function f_table_site($url_site, $title_site, $descr_site) {
// Листинг 15
// Описание ресурса
}
function f_table_gbook($msg_date, $author, $msg) { // Листинг 16
// Сообщение в гостевой книге
}
function f_table_page_start() { // Листинг 17
// Таблица для количества страниц. Начало
}
function f_table_page_end() { // Листинг 18
// Таблица для количества страниц. Конец
}
function f_passw_generator($count_char=8) { // Листинг 25
// Генератор паролей
}
?>
```

Все функции из этого файла мы уже рассмотрели, за исключением функций `f_passw_generator()` и `f_db_connect()` (листинг 25). Функцию `f_passw_generator()`, предназначенную для генерации паролей, мы рассматривали при изучении языка программирования PHP. Функция `f_db_connect()` предназначена для подключения к базе данных.

Листинг 25. Функции `f_passw_generator()` и `f_db_connect()`

```

function f_passw_generator($count_char=8) {
    // Генератор паролей
    $mass = array('a','b','c','d','e','f','g','h','i','j','k','l',
        'm','n','o','p','q','r','s','t','u','v','w','x','y','z',
        'A','B','C','D','E','F','G','H','I','J','K','L',
        'M','N','O','P','Q','R','S','T','U','V','W',
        'X','Y','Z','1','2','3','4','5','6','7','8','9','0');
    $passw = '';
    $count = count($mass)-1;
    for ($i=0; $i<$count_char; $i++) {
        $passw .= $mass[mt_rand(0, $count)];
    }
    return $passw;
}

function f_db_connect() {
    // Подключаемся к базе данных
    if ($db = @mysql_connect(HOST_CONNECT, LOGIN_CONNECT, PASSW_CONNECT))
    {
        mysql_select_db(DB_CONNECT);
        mysql_query('SET NAMES cp1251'); // Установка кодировки соединения
        return $db;
    }
    else {
        $err = '<div style="font-size:12px; font-weight: bold; ';
        $err .= 'text-align:center; color:#FF0000">';
        $err .= 'Не удалось установить соединение с базой данных</div>';
        exit($err);
    }
}

```

Файл `allscript.php` должен быть расположен в каталоге, заданном директивой `include_path` файла `php.ini`. Изменить значение директивы можно с помощью файла конфигурации `.htaccess`:

```

php_value include_path ".;C:\Apache2\config"

```

В ОС UNIX для разделения путей используется символ двоеточие, вместо точки с запятой:

```

php_value include_path ".:~/home/login/config/"

```

SQL-запросы для создания таблиц

Для доступа к Личному кабинету необходима таблица, в которой будут храниться регистрационные данные пользователей — логин (совпадающий с адресом электронной почты) и зашифрованный пароль. В таблице дополнительно создадим поле, с помощью которого можно запретить доступ в систему любому зарегистрированному пользователю. Необходимо также поле с автоматической индексацией и уникальным значением. Индекс пользователя будет сохраняться в таблице `site` при регистрации сайта.

SQL-запрос для создания таблицы приведен в листинге 26.

Листинг 26. SQL-запрос для таблицы с данными пользователей

```
CREATE TABLE `user` (  
  `id_user` mediumint(9) auto_increment,  
  `email` char(50),  
  `passw` char(32),  
  `status_user` enum('y','n'),  
  PRIMARY KEY (`id_user`),  
  UNIQUE KEY `email` (`email`)  
) ENGINE=MyISAM DEFAULT CHARSET=cp1251;
```

Все сайты мы будем распределять по тематике. Для этого необходима таблица с рубрикаторм. В таблице создадим два поля — поле с уникальным значением и поле с названием рубрики.

SQL-запрос для создания таблицы рубрикатора приведен в листинге 27.

Листинг 27. SQL-запрос для таблицы рубрикатора

```
CREATE TABLE `rubr` (  
  `id_rubr` smallint(6) auto_increment,  
  `name_rubr` char(150),  
  PRIMARY KEY (`id_rubr`)  
) ENGINE=MyISAM DEFAULT CHARSET=cp1251;  
  
INSERT INTO `rubr` (`id_rubr`, `name_rubr`) VALUES  
(NULL, 'Автомобили'),  
(NULL, 'Безопасность'),  
(NULL, 'Бытовая техника'),  
(NULL, 'Города и регионы'),
```

```
(NULL, 'Гостиницы'),  
(NULL, 'Знакомства'),  
(NULL, 'Игры'),  
(NULL, 'Интернет'),  
(NULL, 'Искусство'),  
(NULL, 'Каталоги'),  
(NULL, 'Кино'),  
(NULL, 'Компании'),  
(NULL, 'Компьютеры'),  
(NULL, 'Кулинария'),  
(NULL, 'Литература'),  
(NULL, 'Магазины On-Line'),  
(NULL, 'Мебель'),  
(NULL, 'Медицина'),  
(NULL, 'Мобильные телефоны'),  
(NULL, 'Мода'),  
(NULL, 'Музыка'),  
(NULL, 'Наука'),  
(NULL, 'Недвижимость'),  
(NULL, 'Непознанное'),  
(NULL, 'Образование'),  
(NULL, 'Окна и двери'),  
(NULL, 'Подарки и праздники'),  
(NULL, 'Политика'),  
(NULL, 'Природа'),  
(NULL, 'Провайдеры'),  
(NULL, 'Продукты питания'),  
(NULL, 'Работа'),  
(NULL, 'Развлечения'),  
(NULL, 'Разное'),  
(NULL, 'Реклама'),  
(NULL, 'Сервисы'),  
(NULL, 'СМИ'),  
(NULL, 'Спорт'),  
(NULL, 'Страхование'),  
(NULL, 'Строительство'),  
(NULL, 'Театр'),  
(NULL, 'Товары и услуги'),  
(NULL, 'Транспорт'),
```



```
(NULL, 'Туризм'),  
(NULL, 'Финансовые услуги'),  
(NULL, 'Фото'),  
(NULL, 'Хобби'),  
(NULL, 'Юридические услуги');
```

SQL-запрос для создания таблицы с описаниями сайтов приведен в листинге 28.

Листинг 28. SQL-запрос для создания таблицы с описаниями сайтов

```
CREATE TABLE `site` (  
  `id_site` mediumint(9) auto_increment,  
  `id_rubr` smallint(6),  
  `id_user` mediumint(9),  
  `url_site` char(255),  
  `url_site_dop` char(255),  
  `title` char(80),  
  `descr` text,  
  `status_site` enum('y','n','s'),  
  `iq_site` smallint(6),  
  `add_date` date,  
  PRIMARY KEY (`id_site`),  
  KEY id_rubr (`id_rubr`),  
  KEY url_site_dop (`url_site_dop`)  
) ENGINE=MyISAM DEFAULT CHARSET=cp1251;
```

Таблица содержит следующие поля:

- ☐ `id_site` — поле с уникальным значением и автоматической индексацией;
- ☐ `id_rubr` — идентификатор рубрики. Должен совпадать с одноименным полем в таблице `rubr`;
- ☐ `id_user` — идентификатор пользователя. Должен совпадать с одноименным полем в таблице `user`;
- ☐ `url_site` — полный URL-адрес, например, **http://www.site.ru**;
- ☐ `url_site_dop` — доменное имя, например, `site.ru`;
- ☐ `title` — название сайта;
- ☐ `descr` — описание сайта;
- ☐ `status_site` — текущий статус. Может содержать следующие значения:
 - `y` — активен;

- n — на модерации;
- s — временно не работает;
- ❑ iq_site — показатель значимости ресурса в виде числа. По этому полю будет осуществляться сортировка. Чем выше показатель, тем выше сайт в результатах;
- ❑ add_date — дата регистрации.

Все сообщения пользователей в гостевой книге будут храниться также в базе данных. SQL-запрос для создания таблицы приведен в листинге 29.

Листинг 29. SQL-запрос для таблицы, используемой гостевой книгой

```
CREATE TABLE `gbook` (
  `id_msg` smallint(6) auto_increment,
  `author` char(55),
  `msg` text,
  `msg_date` datetime,
  `msg_new` enum('y','n'),
  PRIMARY KEY (`id_msg`)
) ENGINE=MyISAM DEFAULT CHARSET=cp1251;
```

Поле `msg_new` предназначено для определения статуса сообщения (прочтено или нет).

Вывод оглавления и десяти последних добавленных сайтов

Прежде чем создать любую программу, вначале необходимо описать алгоритм ее работы. При выводе рубрикатора алгоритм следующий.

- ❑ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение и завершаем работу сценария.
- ❑ Делаем запрос, возвращающий общее количество зарегистрированных сайтов, и сохраняем значение в переменной `$count_site`.
- ❑ Делаем запрос, чтобы узнать количество сайтов, зарегистрированных за текущий день, и сохраняем значение в переменной `$count_site_evr`.
- ❑ Выводим названия рубрик в алфавитном порядке. Все рубрики распределяем по трем равным по размеру столбцам.
- ❑ Выводим значения переменных `$count_site` и `$count_site_evr`.
- ❑ Выводим десять последних добавленных сайтов.

❑ Закрываем соединение с базой данных.

Исходный код программы приведен в листинге 30.

Листинг 30. Содержимое файла index.php

```
<?php
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
// Заголовок
$title = 'Каталог ресурсов интернет';
// Описание страницы
$description = '';
// Ключевые слова для поисковых машин
$keywords = 'Добавить сайт, каталог сайтов';
// Выводим верхний колонтитул
f_header_all($title, $description, $keywords);

// Получаем количество сайтов в каталоге и количество добавленных за
// сегодня
$count_site = 0;
$count_site_evr = 0;
if ($res = @mysql_query('SELECT COUNT(`id_site`) FROM `site`')) {
    $pole = mysql_fetch_row($res);
    $count_site = $pole[0];
}
$query = 'SELECT COUNT(`id_site`) FROM `site` WHERE
`add_date`=CURDATE()';
if ($res = @mysql_query($query)) {
    $pole = mysql_fetch_row($res);
    $count_site_evr = $pole[0];
}
?>

<h1>Каталог ресурсов интернет</h1><br>
<table align="center" width="600" bgcolor="#FFFFFF" border="0"
    cellpadding="2" cellspacing="0">
<tr><td valign="top">
```

```

$query = 'SELECT `id_rubr`, `name_rubr` FROM `rubr` ORDER BY
`name_rubr`';
$res = @mysql_query($query) or die('Ошибка ' . mysql_errno());
$count = mysql_num_rows($res);
if ($count>0) {
    $c1 = ceil($count/3);
    $c2 = $c1*2;
    $i = 0;
    while ($row = mysql_fetch_assoc($res)) {
        if ($i==$c1 || $i==$c2) echo '</td><td valign="top">';
        echo '<a href="' . URL_SITE . 'catalog.php?rubr=' .
$row['id_rubr'];
        echo '">' . htmlspecialchars($row['name_rubr']) . "</a><br>\n";
        $i++;
    }
}
?>
</td></tr>
</table>
<?php
echo '<br><div align="center" class="bold">';
echo 'Количество сайтов в каталоге - ' . $count_site . '<br>';
echo 'Сегодня добавлено - ' . $count_site_evr . '</div><br>';
echo "<h1>Новые сайты в каталоге</h1><br>\n";
f_table_2_start(); // Выводим таблицу второго уровня
// Выводим новые сайты
$query = "SELECT * FROM `site` WHERE `status_site`='y' ";
$query .= 'ORDER BY `id_site` DESC LIMIT 0, 10';
$res = @mysql_query($query) or die('Ошибка ' . mysql_errno());
$count = mysql_num_rows($res);
if ($count>0) {
    while ($r = mysql_fetch_assoc($res)) {
        // Выводим описание
        $title = htmlspecialchars($r['title']);
        $descr = htmlspecialchars($r['descr']);
        f_table_site($r['url_site'], $title, $descr);
    }
}
else echo '<div align="center" class="bold">Сайтов нет</div><br>';

```

```
f_table_2_end(); // Конец таблицы второго уровня
// Закрываем соединение с базой данных
mysql_close($db);
f_footer_user(); // Выводим нижний колонтитул
?>
```

Для разделения рубрик на 3 равных столбца делим общее количество рубрик на 3 и округляем к целому числу:

```
$c1 = ceil($count/3);
```

Для получения второго числа умножаем на 2:

```
$c2 = $c1*2;
```

Далее проверяем условие. Если текущее значение счетчика равняется первому или второму числу, то выводим теги для разделения ячеек таблицы.

```
if ($i==$c1 || $i==$c2) echo '</td><td valign="top">';
```

Каждое название рубрики является ссылкой, в которой передается идентификатор рубрики методом GET. По этому идентификатору будут выводиться сайты, зарегистрированные в указанной рубрике.

```
echo '<a href="' . URL_SITE . 'catalog.php?rubr=' . $row['id_rubr'];
echo '>' . htmlspecialchars($row['name_rubr']) . "</a><br>\n";
```

Вывод каталога по параметрам

При выводе каталога алгоритм работы программы следующий.

- ☐ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение и завершаем работу сценария.
- ☐ Создаем короткие имена переменных `$rubr` (идентификатор рубрики) и `$page` (номер страницы).
- ☐ С помощью регулярных выражений проверяем, чтобы идентификатор рубрики был числом. Если это не так, то идентификатору присваивается значение 1.
- ☐ Делаем запрос для определения названия рубрики и выводим ее название в заголовок страницы.
- ☐ Делаем запрос на общее количество сайтов в рубрике.
- ☐ Проверяем формат номера страницы и рассчитываем начальную позицию.
- ☐ Делаем запрос на выборку заданного количества сайтов, начиная с начальной позиции.
- ☐ Выводим описания сайтов.
- ☐ Если число сайтов больше заданного максимального количества сайтов на странице, то выводим номера страниц.

❑ Закрываем соединение с базой данных.

Исходный код программы приведен в листинге 31.

Листинг 31. Содержимое файла catalog.php

```
<?php
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
if (isset($_GET['rubr'])) $rubr = (int)$_GET['rubr'];
else $rubr = 1;
if (isset($_GET['page'])) $page = (int)$_GET['page'];
else $page = 1;
if (!preg_match('/^[0-9]+$/', $rubr) || $rubr==0) $rubr = 1;
$query = 'SELECT `name_rubr` FROM `rubr` WHERE `id_rubr`=' . $rubr;
$res_r = @mysql_query($query) or die('Ошибка ' . mysql_errno());
if (mysql_num_rows($res_r)==1) {
    $row = mysql_fetch_assoc($res_r);
    $rubrika = $row['name_rubr'];
}
else $rubrika = '';
// Заголовок
$title = 'Каталог сайтов &gt;&gt; Рубрика - ' .
htmlspecialchars($rubrika);
// Описание страницы
$description = '';
// Ключевые слова для поисковых машин
$keywords = $rubrika;
// Выводим верхний колонтитул
f_header_all($title, $description, $keywords);
f_table_2_start(); // Выводим таблицу второго уровня
echo '<h1>Каталог сайтов &gt;&gt; ' . htmlspecialchars($rubrika) .
"</h1><br>\n";
// Выводим сайты
$query_count = 'SELECT COUNT(`id_site`) FROM `site` ' .
$query_count .= "WHERE `id_rubr`=$rubr AND `status_site`='y'";
$res_count = @mysql_query($query_count) or die('Ошибка ' .
mysql_errno());
```

```

if (mysql_num_rows($res_count)==1) {
    $pole = mysql_fetch_row($res_count);
    $count = $pole[0];
    if (preg_match('/^[0-9]+/s', $count) && $count>0) {
        if (!isset($page) || $page<1) $page = 1;
        if (!preg_match('/^[0-9]+$/s', $page)) $page = 1;
        $count_page = ceil($count/COUNT_POS_PAGE);
        if ($page>$count_page) $page = $count_page;
        $pos_start = ($page-1)*COUNT_POS_PAGE;
        $query = 'SELECT * FROM `site` WHERE `id_rubr`=' . $rubr;
        $query .= " AND `status_site`='y' ";
        $query .= 'ORDER BY `iq_site` DESC ';
        $query .= 'LIMIT ' . $pos_start . ', ' . COUNT_POS_PAGE;
        $res = @mysql_query($query) or die('Ошибка ' . mysql_errno());
        while ($r = mysql_fetch_assoc($res)) {
            // Выводим описание
            f_table_site($r['url_site'], htmlspecialchars($r['title']),
            htmlspecialchars($r['descr']));
        }
        if ($count>COUNT_POS_PAGE) { // Выводим количество страниц
            f_table_page_start(); // Таблица для количества страниц Начало
            echo '<span class="bold">Страницы:</span>' . "\n";
            for ($j=1; $j<$count_page+1; $j++) {
                if ($j==$page) echo '[' . $j . "]\n";
                else echo "<a href=\""?rubr=$rubr&page=$j\">$j</a> \n";
            }
            f_table_page_end(); // Таблица для количества страниц Конец
        }
    }
    else {
        echo '<div align="center" class="bold">Сайтов нет</div><br>';
    }
}
else echo '<div class="err">Ошибка</div><br>';
f_table_2_end(); // Конец таблицы второго уровня
mysql_close($db); // Закрываем соединение с базой данных
f_footer_user(); // Выводим нижний колонтитул
?>

```

Организация поиска по каталогу

При поиске по каталогу алгоритм работы программы следующий.

- ☐ Запрещаем кеширование страницы.
- ☐ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение и завершаем работу сценария.
- ☐ Создаем короткие имена переменных `$text` (поисковая фраза) и `$page` (номер страницы).
- ☐ Если директива `magic_quotes_gpc` включена, то удаляем защитные слешы.
- ☐ Удаляем спецсимволы из поисковой фразы и проверяем ее на допустимость.
- ☐ Проверяем корректность номера страницы.
- ☐ Экранируем все спецсимволы с помощью функции `mysql_real_escape_string()`.
- ☐ При поиске по шаблону символы "_" и "%" являются специальными. По этой причине добавляем для их защиты слешы при помощи функции `addslashes()`.
- ☐ Выполняем запрос, выводящий результаты поиска.
- ☐ Получаем общее количество сайтов в результате поиска с помощью MySQL-функции `FOUND_ROWS()`. Чтобы получить это количество мы указали параметр `SQL_CALC_FOUND_ROWS` в запросе `SELECT`.
- ☐ Выводим описания сайтов.
- ☐ Выводим ссылки для перемещения к следующей и предыдущей страницам.
- ☐ Если существуют ошибки, то выводим их описание.
- ☐ Закрываем соединение с базой данных.

Исходный код программы приведен в листинге 32.

Листинг 32. Содержимое файла `search.php`

```
<?php
header('Expires: Sun, 27 May 2007 01:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Pragma: no-cache');
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
```



```
if (isset($_GET['search'])) $text = $_GET['search'];
else $text = '';
if (isset($_GET['page'])) $page = (int)$_GET['page'];
else $page = 1;
// Если директива magic_quotes_gpc включена,
// то удаляем защитные слэши перед спецсимволами
if (get_magic_quotes_gpc()) {
    $text = stripslashes($text);
}
$_err_text = '';
$text = str_replace("\n", ' ', $text);
$text = str_replace("\t", ' ', $text);
$text = str_replace("\r", '', $text);
$text = trim($text);
if ($text == '') {
    $_err_text = 'Не задана строка поиска<br>';
}
if (strlen($text)<3) {
    $_err_text .= 'В поле допустимо не менее 3 символов<br>';
}
if (strlen($text)>50) {
    $_err_text .= 'В поле допустимо не более 50 символов<br>';
}
// Заголовок
$title = 'Результаты поиска - ' . htmlspecialchars($text);
// Описание страницы
$_description = '';
// Ключевые слова для поисковых машин
$_keywords = '';
// Выводим верхний колонтитул
f_header_all($title, $_description, $_keywords, 2,
    htmlspecialchars($text));
f_table_2_start(); // Выводим таблицу второго уровня
echo '<h1>Результаты поиска</h1><br>' . "\n";
if ($_err_text == '') {
    $t = urlencode($text);
    if (!isset($page) || $page<1) $page = 1;
    if (!preg_match('/^[0-9]+$/', $page)) $page = 1;
    $start = ($page-1)*COUNT_POS_PAGE;
```

```

$text = mysql_real_escape_string($text);
$text = addslashes($text, '_%');
// Выводим результаты поиска
$query = 'SELECT SQL_CALC_FOUND_ROWS * FROM `site` ';
$query .= "WHERE `status_site`='y' AND ";
$query .= "(`title` LIKE '%" . $text . "%' ";
$query .= "OR `descr` LIKE '%" . $text . "%' ";
$query .= "OR `url_site` LIKE '%" . $text . "%') ";
$query .= 'ORDER BY `iq_site` DESC ';
$query .= 'LIMIT ' . $start . ', ' . COUNT_POS_PAGE;
$res = @mysql_query($query);
if ($res) {
    $count = mysql_num_rows($res);
    $r = @mysql_query('SELECT FOUND_ROWS()') or die('Ошибка');
    $pole = mysql_fetch_row($r);
    $allcount = $pole[0];
    if ($count>0) {
        while ($p = mysql_fetch_assoc($res)) {
            // Выводим описание
            f_table_site($p['url_site'], htmlspecialchars($p['title'],
htmlspecialchars($p['descr'])));
        }
    }
    elseif ($allcount == 0) {
        echo '<div align="center" class="bold">';
        echo 'По вашему запросу ничего не найдено</div><br>';
    }
    else {
        echo '<div align="center">';
        echo '<a href="?search=' . $t . '&page=1">';
        echo 'На первую страницу результатов</a></div>';
        $allcount = 0;
    }
    if ($allcount>COUNT_POS_PAGE) {
        echo '<div align="center">';
        if (($start-COUNT_POS_PAGE)>=0) {
            echo '<a href="?search=' . $t . '&page=' . ($page-1);
            echo '>&lt;&lt; Предыдущая страница</a> ';
        }
    }
}

```

```

        if (($start+COUNT_POS_PAGE)<$allcount) {
            echo '<a href="?search=' . $t . '&page=' . ($page+1);
            echo '>>Следующая страница &gt;&gt;</a> ' ;
        }
        echo '</div>';
    }
}
else {
    echo '<div class="err">Ошибка в запросе</div><br>';
}
}
if ($err_text != '') {
    echo '<div class="err">' . $err_text . "</div><br>\n";
}
f_table_2_end(); // Конец таблицы второго уровня
mysql_close($db); // Закрываем соединение с базой данных
f_footer_user(); // Выводим нижний колонтитул
?>

```

Создание гостевой книги

Гостевая книга должна обладать следующими возможностями:

- ☐ проверка корректности введенных данных с использованием JavaScript на стороне клиента;
- ☐ проверка корректности введенных данных на стороне сервера с помощью PHP;
- ☐ на одной странице должно быть не больше заданного количества сообщений, а снизу должны быть ссылки на все остальные страницы;
- ☐ при нажатии кнопки **Обновить** сообщение не должно повторно добавляться в базу;
- ☐ не должно быть более 25-ти символов подряд без пробела;
- ☐ при выводе сообщений все спецсимволы должны быть заменены на HTML-эквиваленты. В базе данных мы будем хранить сообщения как их ввел пользователь;
- ☐ в случае ошибки должно быть выведено соответствующее сообщение.

Алгоритм работы программы следующий.

- ☐ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение.
- ☐ Если форма отправлена (переменная \$_POST['go'] будет существовать):

- создаем короткие имена переменных;
 - удаляем пробельные символы;
 - если директива `magic_quotes_gpc` включена, то удаляем защитные слешы;
 - проверяем значения переменных на допустимость;
 - экранируем все спецсимволы с помощью функции `mysql_real_escape_string()`;
 - проверяем на повтор, сравнивая новое сообщение с последним добавленным;
 - если все нормально, добавляем новое сообщение в базу данных и делаем авторедирект для сброса данных формы.
- ☐ Запрещаем кеширование страницы.
 - ☐ Создаем короткое имя для номера страницы.
 - ☐ Делаем запрос на выборку.
 - ☐ Проверяем корректность номера страницы и рассчитываем начальную и конечную позиции.
 - ☐ Выводим сообщения при этом заменяем все спецсимволы на HTML-эквиваленты с помощью функции `htmlspecialchars()`.
 - ☐ Если число сообщений больше заданного максимального количества сообщений на странице, то выводим номера страниц.
 - ☐ Закрываем соединение с базой данных.
 - ☐ Если существуют ошибки, то выводим их описание.
 - ☐ Выводим форму. Если возникли ошибки при добавлении сообщения, то заполняем все поля для редактирования.

Исходный код программы приведен в листинге 33.

Листинг 33. Содержимое файла `gbook.php`

```
<?php
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
$author = '';
$msg = '';
$err_msg = '';
if (isset($_POST['go'])) {
```

```
// Форма отправлена
if (isset($_POST['author'])) $author = $_POST['author'];
else $author = '';
if (isset($_POST['msg'])) $msg = $_POST['msg'];
else $msg = '';
// Удаляем пробельные символы
$author = trim($author);
$msg = trim($msg);
// Если директива magic_quotes_gpc включена,
// то удаляем защитные слэши перед спецсимволами
if (get_magic_quotes_gpc()) {
    $author = stripslashes($author);
    $msg = stripslashes($msg);
}
$msg = str_replace("\r", '', $msg);
if (strlen($author)>50 || strlen($author) == 0) {
    $err_msg .= 'Поле Имя не заполнено или длина более ';
    $err_msg .= '50 символов!<br>';
}
if (strlen($msg)>2000 || strlen($msg) == 0) {
    $err_msg .= 'Поле сообщение не заполнено или длина ';
    $err_msg .= 'более 2000 символов<br>';
}
if (preg_match('/[a-zA-яё0-9,.\'"]{26}/is', $msg)) {
    $err_msg .= 'Поле сообщение содержит более 25 ';
    $err_msg .= 'символов без пробела<br>';
}
if ($err_msg == '') {
    // Если ошибок нет
    $author = mysql_real_escape_string($author);
    $msg = mysql_real_escape_string($msg);
    $query = 'INSERT INTO `gbook` VALUES ';
    $query .= "(NULL, '$author', '$msg', NOW(), 'n')";
    // Проверяем на повтор
    $query2 = 'SELECT * FROM `gbook` ORDER BY `id_msg` DESC LIMIT 0,1';
    if ($result = @mysql_query($query2)) {
        if (mysql_num_rows($result) == 0) {
            // Добавляем первое сообщение
            @mysql_query($query);
        }
    }
}
```

```

        // Делаем авторедирект
        header('Location: gbook.php');
        exit();
    }
    else {
        $r = mysql_fetch_assoc($result);
        if ($r['author']==stripslashes($author) &&
            $r['msg']==stripslashes($msg)) {
            $err_msg = 'Вы пытаетесь дважды добавить сообщение<br>';
            $author = $msg = '';
        }
        else {
            // Добавляем новое сообщение
            @mysql_query($query);
            // Делаем авторедирект
            header('Location: gbook.php');
            exit();
        }
    }
}

}

// Запрещаем кэширование
header('Expires: Sun, 27 May 2007 01:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Pragma: no-cache');
// Заголовок
$title = 'Гостевая книга';
// Описание страницы
$description = '';
// Ключевые слова для поисковых машин
$keywords = '';
// Выводим верхний колонтитул
f_header_all($title, $description, $keywords, 2);
?>
<script type="text/javascript">
<!--
function f_submit() {

```

```
var author = document.getElementById("author");
var msg = document.getElementById("msg");
if (author.value==" " || msg.value==" ") {
    window.alert("Не заполнено обязательное поле");
    return false;
}
if (author.value.length>50) {
    window.alert("В поле Имя допустимо не более 50 символов");
    author.focus();
    return false;
}
if (msg.value.length>2000) {
    window.alert("В поле Сообщение допустимо не более 2000 символов");
    msg.focus();
    return false;
}
return true;
}
function showForm() {
    var frm = document.getElementById("frm");
    if (frm.style.display=='none') {
        frm.style.display='block';
    }
    else {
        frm.style.display='none';
    }
}
//-->
</script>
<?php
f_table_2_start(); // Выводим таблицу второго уровня
echo '<h1>Гостевая книга</h1><br>' . "\n";
if (isset($_GET['page'])) $page = (int)$_GET['page'];
else $page = 1;
$query_count = 'SELECT COUNT(`id_msg`) FROM `gbook`';
$res_count = @mysql_query($query_count) or die('Ошибка ' .
mysql_errno());
if (mysql_num_rows($res_count)===1) {
    $pole = mysql_fetch_row($res_count);
```

```

$count = $pole[0];
if (preg_match('/^[0-9]+/s', $count) && $count>0) {
    if (!isset($page) || $page<1) $page = 1;
    if (!preg_match('/^[0-9]+$/s', $page)) $page = 1;
    $count_page = ceil($count/COUNT_POS_PAGE_GBOOK);
    if ($page>$count_page) $page = $count_page;
    $pos_start = ($page-1)*COUNT_POS_PAGE_GBOOK;
    $query = 'SELECT `author`, `msg`, '
    $query .= "DATE_FORMAT(`msg_date`, '%d.%m.%Y %H:%i') AS `d` ";
    $query .= 'FROM `gbook` ORDER BY `id_msg` DESC '
    $query .= 'LIMIT ' . $pos_start . ', ' . COUNT_POS_PAGE_GBOOK;
    $res = @mysql_query($query) or die('Ошибка ' . mysql_errno());
    while ($r = mysql_fetch_assoc($res)) {
        // Выводим сообщения
        $r['msg'] = htmlspecialchars($r['msg']);
        $r['msg'] = str_replace("\n", '<br>', $r['msg']);
        f_table_gbook($r['d'], htmlspecialchars($r['author']),
        $r['msg']);
    }
    if ($count>COUNT_POS_PAGE_GBOOK) { // Выводим количество страниц
        f_table_page_start(); // Таблица для количества страниц Начало
        echo '<span class="bold">Страницы:</span>' . "\n";
        for ($j=1; $j<$count_page+1; $j++) {
            if ($j==$page) echo '[' . $j . "] \n";
            else echo "<a href=\"?page=$j\">$j</a> \n";
        }
        f_table_page_end(); // Таблица для количества страниц Конец
    }
}
else {
    echo '<div align="center" class="bold">Сообщений нет</div><br>';
}
}
else echo '<div class="err">Ошибка</div><br>';
mysql_close($db);
f_table_2_end(); // Конец таблицы второго уровня
if ($err_msg == '') $display = 'none';
else $display = 'block';
?>

```



```

<div align="center">
<a href="#" onclick="showForm(); return false;">Добавить новое
сообщение</a>
</div>
<div id="frm" style="display:<?php echo $display; ?>" align="center">
<hr width="70%">
<h1>Новое сообщение</h1>
<?php
if ($err_msg != '') {
    // Если возникли ошибки, то выводим их описание
    $author = stripslashes($author);
    $msg = stripslashes($msg);
    $author = htmlspecialchars($author);
    $msg = htmlspecialchars($msg);
    echo '<div class="err">' . $err_msg . "</div><br>\n";
}
?>
<form action="gbook.php" method="POST" onsubmit="return f_submit();">
<span class="bold">Ваше имя:</span>
<input type="text" class="txt_frm" style="width:260px" name="author"
id="author" value="<?php echo $author; ?>">
<span class="bold">Сообщение:</span><br>
<textarea name="msg" id="msg" cols="15" rows="20" class="textarea_frm"
style="width: 500px; height: 150px"><?php echo $msg; ?>
</textarea><br>
<input type="submit" name="go" value="Добавить сообщение">
</form></div>
<?php
f_footer_user(); // Выводим нижний колонтитул
?>

```

Создание формы обратной связи

С помощью формы обратной связи можно отправить письмо создателю сайта, просто заполнив ее и нажав кнопку **Отправить**. Алгоритм работы программы следующий.

❑ Если форма отправлена (переменная \$_POST['go'] будет существовать):

- создаем короткие имена переменных;
- удаляем все теги и специальные символы;

- проверяем значения переменных на допустимость;
 - если ошибок нет, то отправляем письмо.
- ☐ Если существуют ошибки, то выводим их описание.
- ☐ Если существует сообщение об удачной отправке письма, то выводим его.
- ☐ Отображаем форму. Если возникли ошибки при отправке сообщения, то заполняем все поля для редактирования.

Исходный код программы приведен в листинге 34.

Листинг 34. Содержимое файла contact.php

```
<?php
define('IN_CATALOG', true);
require_once('allscript.php');
// Заголовок
$title = 'Форма обратной связи';
// Описание страницы
$description = '';
// Ключевые слова для поисковых машин
$keywords = '';
// Выводим верхний колонтитул
f_header_all($title, $description, $keywords, 2);
?>
<script type="text/javascript">
<!--
function f_test_mail(email) {
    var p = /^[a-z0-9_\. \-]+@[a-z0-9\-\.\.]+\.[a-z]{2,6}$/i;
    if (!p.test(email)) return false;
    return true;
}
function f_submit() {
    var author = document.getElementById("author");
    var msg = document.getElementById("msg");
    var tema = document.getElementById("tema");
    var emails = document.getElementById("emails");
    if (author.value=="" || msg.value=="" ||
        tema.value=="") {
        window.alert("Не заполнено обязательное поле");
        return false;
    }
}
```

```
}
if (author.value.length>20) {
    window.alert("В поле Имя допустимо не более 20 символов");
    author.focus();
    return false;
}
if (tema.value.length>50) {
    window.alert("В поле Тема допустимо не более 50 символов");
    tema.focus();
    return false;
}
if (emails.value.length>50 || !f_test_mail(emails.value)) {
    window.alert("Недопустимое значение поля E-mail");
    emails.focus();
    return false;
}
return true;
}
//-->
</script>
<?php
f_table_2_start(); // Выводим таблицу второго уровня
echo "<h1>Обратная связь</h1><br>\n";
$emails = '';
$author = '';
$tema = '';
$msg = '';
$err_msg = '';
$ok_add = '';
if (isset($_POST['go'])) {
    // Форма отправлена
    if (isset($_POST['emails'])) $emails = $_POST['emails'];
    else $emails = '';
    if (isset($_POST['author'])) $author = $_POST['author'];
    else $author = '';
    if (isset($_POST['tema'])) $tema = $_POST['tema'];
    else $tema = '';
    if (isset($_POST['msg'])) $msg = $_POST['msg'];
    else $msg = '';
```

```

// Удаляем теги
$author = strip_tags($author);
$tema = strip_tags($tema);
// Удаляем пробельные символы
$author = trim($author);
$tema = trim($tema);
$msg = trim($msg);
// Если директива magic_quotes_gpc включена,
// то удаляем защитные слэши перед спецсимволами
if (get_magic_quotes_gpc()) {
    $author = stripslashes($author);
    $tema = stripslashes($tema);
    $msg = stripslashes($msg);
}
$author = str_replace('"', ' ', $author);
$tema = str_replace('"', ' ', $tema);
$author = str_replace("'", ' ', $author);
$tema = str_replace("'", ' ', $tema);
$pattern = '/^[a-z0-9_.-]+@([a-z0-9-]+\.)+[a-z]{2,6}$/is';
if (!preg_match($pattern, $emails)) {
    $err_msg .= 'Недопустимый адрес E-mail !!!<br>';
}
if (strlen($emails)>50) {
    $err_msg .= 'Длина E-mail больше допустимой!!!<br>';
}
if (strlen($author)>20 || strlen($author) == 0) {
    $err_msg .= 'Поле имя не заполнено или длина более ';
    $err_msg .= '20 символов!<br>';
}
if (strlen($tema)>50 || strlen($tema) == 0) {
    $err_msg .= 'Поле тема не заполнено или длина более 50 ';
    $err_msg .= 'символов!<br>';
}
if (strlen($msg) == 0) {
    $err_msg .= 'Поле Сообщение не заполнено!<br>';
}
if ($err_msg == '') {
    // Если ошибок нет отправляем письмо
    $author = "=?windows-1251?B?" . base64_encode($author) . "=?";
}

```

```

        $tema = "=?windows-1251?B?" . base64_encode($tema) . "?=";
        $header = "Content-Type: text/plain; charset=windows-1251\n";
        $header .= "From: $author <$emails>";
        @mail(MAIL_ADRES, $tema, $msg, $header);
        $author = $emails = $tema = $msg = '';
        $ok_add = 'Ваше сообщение успешно отправлено<br>';
    }
}

if ($err_msg != '') {
    // Если возникли ошибки, то выводим их описание
    $author = htmlspecialchars($author);
    $tema = htmlspecialchars($tema);
    $msg = htmlspecialchars($msg);
    echo '<div class="err">' . $err_msg . '</div><br>';
}

if ($ok_add != '') {
    echo '<div class="ok">' . $ok_add . '</div><br>';
}

?>

<form action="contact.php" method="POST" onsubmit="return f_submit();">
<table width="100%" border="0" cellpadding="1" align="center">
<tr><td align="right" width="40%">
<span class="bold">Имя: </span>
</td><td>
<input type="text" name="author" id="author" size="40" maxlength="20"
class="txt_frm" value="<?php echo $author; ?>">
</td></tr>
<tr><td align="right" width="40%">
<span class="bold">E-mail: </span>
</td><td>
<input type="text" name="emails" id="emails" maxlength="50" size="40"
value="<?php echo $emails; ?>" class="txt_frm">
</td></tr>
<tr><td align="right" width="40%">
<span class="bold">Тема: </span>
</td><td>
<input type="text" name="tema" id="tema" size="40" maxlength="50"
class="txt_frm" value="<?php echo $tema; ?>">
</td></tr>

```

```

<tr><td align="right" width="40%"><span class="bold">Сообщение:</span>
</td><td>
<textarea name="msg" id="msg" cols="15" rows="20" class="textarea_frm"
style="width: 300px; height: 150px"><?php echo $msg; ?></textarea>
</td></tr>
<tr><td align="right" width="40%">&nbsp;</td><td>
<input type="submit" name="go" value="Отправить" class="txt_frm">
</td></tr>
</table>
</form><br>
<?php
f_table_2_end(); // Конец таблицы второго уровня
f_footer_user(); // Выводим нижний колонтитул
?>

```

Следует заметить, что очень часто письма не доходят до почтового ящика. Например, если письмо будет расценено почтовым сервером как спам, то оно будет удалено. По этой причине сообщения пользователей лучше сохранять, например, в базе данных, а на адрес администратора отправлять только уведомление о новом письме. В этом случае вы всегда получите сообщение, даже если уведомление о нем не дошло до адресата.

Создание страниц регистрации ошибок

Для регистрации и вывода сообщений об ошибках 401, 403 и 404 создадим отдельные страницы. В случае возникновения ошибки сообщение о ней будет записываться в файл, а пользователь получит сообщение в едином дизайне сайта.

Содержимое файла `err401.php` приведено в листинге 35.

Листинг 35. Содержимое файла `err401.php`

```

<?php
header('Expires: Sun, 27 May 2007 01:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Pragma: no-cache');
define('IN_CATALOG', true);
require_once('allscript.php');
// Заголовок
$title = 'Ошибка 401';
// Описание страницы

```

```
$description = '';
// Ключевые слова для поисковых машин
$keywords = '';
// Выводим верхний колонтитул
f_header_all($title, $description, $keywords, 2);
?>
<h1>Ошибка 401</h1><br>
<?php
@$file = fopen('err401.txt', 'a+') or die('Ошибка');
flock($file, 2);
$date = date('H:i:s d-m-Y');
if (isset($_SERVER['REMOTE_USER'])) {
    $remoteUser = $_SERVER['REMOTE_USER'];
}
else $remoteUser = '';
if (isset($_SERVER['REQUEST_URI'])) {
    $requestUri = $_SERVER['REQUEST_URI'];
}
else $requestUri = '';
if (isset($_SERVER['HTTP_REFERER'])) {
    $httpReferer = $_SERVER['HTTP_REFERER'];
}
else $httpReferer = '';
if (isset($_SERVER['HTTP_USER_AGENT'])) {
    $userAgent = $_SERVER['HTTP_USER_AGENT'];
}
else $userAgent = '';
if (isset($_SERVER['REMOTE_ADDR'])) {
    $remoteAddr = $_SERVER['REMOTE_ADDR'];
}
else $remoteAddr = '';
$msg = "$date\nПользователь: " . $remoteUser . "\n";
$msg .= 'Текст ошибки: ' . $requestUri . "\n";
$msg .= 'Ссылка: ' . $httpReferer . "\n";
$msg .= 'Браузер: ' . $userAgent . "\n";
$msg .= 'IP-адрес: ' . $remoteAddr . "\n\n";
fwrite($file, $msg);
flock($file, 3);
fclose($file);
```

```
f_footer_user(); // Выводим нижний колонтитул
?>
```

Содержимое файла err403.php приведено в листинге 36.

Листинг 36. Содержимое файла err403.php

```
<?php
header('Expires: Sun, 27 May 2007 01:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Pragma: no-cache');
define('IN_CATALOG', true);
require_once('allscript.php');
// Заголовок
$title = 'Ошибка 403';
// Описание страницы
$description = '';
// Ключевые слова для поисковых машин
$keywords = '';
// Выводим верхний колонтитул
f_header_all($title, $description, $keywords, 2);
?>

<h1>Ошибка 403</h1><br>

<?php
@$file = fopen('err403.txt', 'a+') or die('Ошибка');
flock($file, 2);
$date = date('H:i:s d-m-Y');
if (isset($_SERVER['REMOTE_USER'])) {
    $remoteUser = $_SERVER['REMOTE_USER'];
}
else $remoteUser = '';
if (isset($_SERVER['REQUEST_URI'])) {
    $requestUri = $_SERVER['REQUEST_URI'];
}
else $requestUri = '';
if (isset($_SERVER['HTTP_REFERER'])) {
    $httpReferer = $_SERVER['HTTP_REFERER'];
}
else $httpReferer = '';
```



```
if (isset($_SERVER['HTTP_USER_AGENT'])) {
    $userAgent = $_SERVER['HTTP_USER_AGENT'];
}
else $userAgent = '';
if (isset($_SERVER['REMOTE_ADDR'])) {
    $remoteAddr = $_SERVER['REMOTE_ADDR'];
}
else $remoteAddr = '';
$msg = "$date\nПользователь: " . $remoteUser. "\n";
$msg .= 'Текст ошибки: ' . $requestUri. "\n";
$msg .= 'Ссылка: ' . $httpReferer . "\n";
$msg .= 'Браузер: ' . $userAgent . "\n";
$msg .= 'IP-адрес: ' . $remoteAddr . "\n\n";
fwrite($file, $msg);
flock($file, 3);
fclose($file);
f_footer_user(); // Выводим нижний колонтитул
?>
```

Содержимое файла `err404.php` приведено в листинге 37.

Листинг 37. Содержимое файла `err404.php`

```
<?php
header('Expires: Sun, 27 May 2007 01:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Pragma: no-cache');
define('IN_CATALOG', true);
require_once('allscript.php');
// Заголовок
$title = 'Ошибка 404';
// Описание страницы
$description = '';
// Ключевые слова для поисковых машин
$keywords = '';
// Выводим верхний колонтитул
f_header_all($title, $description, $keywords, 2);
?>
<h1>Ошибка 404</h1><br>
```

```

<?php
@$file = fopen('err404.txt', 'a+') or die('Ошибка');
flock($file, 2);
$date = date('H:i:s d-m-Y');
if (isset($_SERVER['REMOTE_USER'])) {
    $remoteUser = $_SERVER['REMOTE_USER'];
}
else $remoteUser = '';
if (isset($_SERVER['REQUEST_URI'])) {
    $requestUri = $_SERVER['REQUEST_URI'];
}
else $requestUri = '';
if (isset($_SERVER['HTTP_REFERER'])) {
    $httpReferer = $_SERVER['HTTP_REFERER'];
}
else $httpReferer = '';
if (isset($_SERVER['HTTP_USER_AGENT'])) {
    $userAgent = $_SERVER['HTTP_USER_AGENT'];
}
else $userAgent = '';
if (isset($_SERVER['REMOTE_ADDR'])) {
    $remoteAddr = $_SERVER['REMOTE_ADDR'];
}
else $remoteAddr = '';
$msg = "$date\nПользователь: " . $remoteUser . "\n";
$msg .= 'Текст ошибки: ' . $requestUri . "\n";
$msg .= 'Ссылка: ' . $httpReferer . "\n";
$msg .= 'Браузер: ' . $userAgent . "\n";
$msg .= 'IP-адрес: ' . $remoteAddr . "\n\n";
fwrite($file, $msg);
flock($file, 3);
fclose($file);
f_footer_user(); // Выводим нижний колонтитул
?>

```

Для того чтобы эти страницы отображались при соответствующих ошибках, необходимо создать в папке файл `.htaccess` и добавить строки, приведенные в листинге 38.

Листинг 38. Содержимое файла `.htaccess`

```
ErrorDocument 401 /err401.php
ErrorDocument 403 /err403.php
ErrorDocument 404 /err404.php
php_value include_path ".;C:\Apache2\config"
php_flag register_globals Off
php_flag magic_quotes_gpc Off
php_flag magic_quotes_runtime Off
<Files err*.txt>
    Deny from all
</Files>
```

Создание Личного кабинета для пользователей при помощи PHP

Для добавления сайтов в каталог создадим Личный кабинет. Это позволит защититься от автоматического добавления сайтов. В дальнейшем можно добавить возможность редактирования описания сайтов пользователями и другие возможности.

Для реализации Личного кабинета создадим каталог user, в котором разместим три файла:

- ☐ index.php — для входа в систему, регистрации новых пользователей и восстановления пароля;
- ☐ add.php — для регистрации новых сайтов;
- ☐ exit.php — для выхода из системы.

Создание страницы с формами "Регистрация", "Вход в систему", "Восстановить пароль"

Данный пример демонстрирует возможность обработки нескольких форм с помощью одного файла.

Алгоритм работы программы следующий.

- ☐ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение.
- ☐ Если форма для входа в систему отправлена (переменная \$_POST['enter'] будет существовать):
 - создаем короткие имена переменных;
 - проверяем корректность введенных данных;
 - если ошибки не найдены:

- шифруем введенный пароль с помощью функции `md5()`;
 - делаем запрос. Если пользователь найден, то запускаем сессию и регистрируем в ней переменные;
 - закрываем соединение с базой данных;
 - перебрасываем пользователя на страницу добавления сайта.
- ☐ Если форма для регистрации нового пользователя отправлена (переменная `$_POST['add']` будет существовать):
- создаем короткие имена переменных;
 - проверяем корректность введенных данных;
 - если ошибки не найдены:
 - проверяем, не зарегистрирован ли E-mail раньше;
 - шифруем введенный пароль с помощью функции `md5()`;
 - регистрируем пользователя;
 - запускаем сессию и регистрируем в ней переменные;
 - закрываем соединение с базой данных;
 - перебрасываем пользователя на страницу добавления сайта.
- ☐ Если форма для восстановления пароля отправлена (переменная `$_POST['mail_pass']` будет существовать):
- создаем короткое имя переменной;
 - проверяем корректность введенных данных;
 - если ошибки не найдены:
 - проверяем, зарегистрирован ли E-mail раньше;
 - генерируем новый пароль с помощью функции `f_passw_generator()`;
 - шифруем пароль с помощью PHP-функции `md5()`;
 - добавляем новый пароль в базу данных;
 - отправляем новый пароль на E-mail.
- ☐ Если возникли ошибки при входе в систему, то выводим их описание.
- ☐ Выводим форму "Вход в систему".
- ☐ Если возникли ошибки при регистрации, то выводим их описание.
- ☐ Выводим форму "Регистрация нового пользователя".
- ☐ Если возникли ошибки при восстановлении пароля, то выводим их описание.
- ☐ В случае успешной отправки нового пароля на E-mail выводим подтверждение.

❑ Выводим форму "Восстановить пароль".

❑ Закрываем соединение с базой данных.

Исходный код программы приведен в листинге 39.

Листинг 39. Содержимое файла index.php

```
<?php
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
// Заголовок
$title = 'Вход в Личный кабинет';
// Описание страницы
$description = '';
// Ключевые слова для поисковых машин
$keywords = '';

// Вход в Личный кабинет
$error_enter = '';
if (isset($_POST['enter'])) {
    if (isset($_POST['login'])) $login = $_POST['login'];
    else $login = '';
    if (isset($_POST['passw'])) $passw = $_POST['passw'];
    else $passw = '';
    $login = trim($login);
    $passw = trim($passw);
    $pattern = '/^[a-z0-9_.-]+@[a-z0-9-]+\.[a-z]{2,6}$/is';
    if (!preg_match($pattern, $login)) {
        $error_enter = 'Недопустимый адрес E-mail<br>';
    }
    if (strlen($login)>50) {
        $error_enter .= 'Длина E-mail больше допустимой<br>';
    }
    if (strlen($passw)>16 || strlen($passw)<6) {
        $error_enter .= 'Длина пароля должна быть в ';
        $error_enter .= 'пределах от 6 до 16 символов<br>';
    }
}
```

```

if (!preg_match('/^[a-z0-9]{6,16}$/is', $passwd)) {
    $err_enter .= 'В пароле допустимы только буквы ';
    $err_enter .= 'A-Z (a-z) или цифры 0-9<br>';
}
if ($err_enter == '') {
    $passwd = md5($passwd);
    $query_enter = 'SELECT * FROM `user` ';
    $query_enter .= "WHERE `email`='$login' AND `passwd`='$passwd'";
    $res_enter = @mysql_query($query_enter) or die('Ошибка ' .
mysql_errno());
    if (mysql_num_rows($res_enter)===1) {
        session_start();
        $_SESSION['sess_login'] = $login;
        $_SESSION['sess_pass'] = $passwd;
        $_SESSION['sess_data'] = date('U');
        mysql_close($db);
        header('Location: add.php');
        exit();
    }
    else {
        $err_enter .= 'Логин/пароль не найден<br>';
    }
}
}

// Регистрация нового пользователя
$emails_tmp = '';
$err_add = '';
if (isset($_POST['add'])) {
    if (isset($_POST['emails'])) $emails = $_POST['emails'];
    else $emails = '';
    if (isset($_POST['pass'])) $pass = $_POST['pass'];
    else $pass = '';
    $pattern = '/^[a-z0-9_.-]+@[a-z0-9-]+\.[a-z]{2,6}$/is';
    if (!preg_match($pattern, $emails)) {
        $err_add = 'Недопустимый адрес E-mail<br>';
    }
    if (strlen($emails)>50) {
        $err_add .= 'Длина E-mail больше допустимой<br>';
    }
}

```

```
}
if (strlen($pass)>16 || strlen($pass)<6) {
    $err_add .= 'Длина пароля должна быть в пределах ';
    $err_add .= 'от 6 до 16 символов<br>';
}
if (!preg_match('/^[a-z0-9]{6,16}$/is', $pass)) {
    $err_add .= 'В пароле допустимы только буквы ';
    $err_add .= 'A-Z (a-z) или цифры 0-9<br>';
}
if ($err_add == '') {
    $q = "SELECT * FROM `user` WHERE `email`='$emails'";
    $res_add = @mysql_query($q) or die('Ошибка ' . mysql_errno());
    if (mysql_num_rows($res_add)===0) {
        $pass = md5($pass);
        $pass = mysql_real_escape_string($pass);
        $query_add = 'INSERT INTO `user` VALUES ';
        $query_add .= "(NULL, '$emails', '$pass', 'Y')";
        if (@mysql_query($query_add)) {
            session_start();
            $_SESSION['sess_login'] = $emails;
            $_SESSION['sess_pass'] = $pass;
            $_SESSION["sess_data"] = date('U');
            mysql_close($db);
            header('Location: add.php');
            exit();
        }
        else {
            $err_add .= 'Попробуйте сделать запрос через некоторое
время<br>';
        }
    }
    else {
        $err_add .= "E-mail $emails уже зарегистрирован ранее!!!<br>";
        $err_add .= 'Восстановить пароль можно заполнив форму ниже<br>';
    }
}
else $emails_tmp = htmlspecialchars($emails);
}
```

```

// Восстанавливаем пароль
$serr_mail = '';
$msg = '';
if (isset($_POST['mail_pass'])) {
    if (isset($_POST['mails'])) $mails = $_POST['mails'];
    else $mails = '';
    $mails = trim($mails);
    $pattern = '/^[a-z0-9_.-]+@[a-z0-9-]+\.[a-z]{2,6}$/is';
    if (!preg_match($pattern, $mails)) {
        $serr_mail = 'Недопустимый адрес E-mail<br>';
    }
    if (strlen($mails)>50) {
        $serr_mail .= 'Длина E-mail больше допустимой<br>';
    }
    if ($serr_mail == '') {
        $q = "SELECT `id_user` FROM `user` WHERE `email`='$mails'";
        $res_mail = @mysql_query($q) or die('Ошибка ' . mysql_errno());
        if (mysql_num_rows($res_mail)===1) {
            $row = mysql_fetch_assoc($res_mail);
            $id_user = $row['id_user'];
            // Генерируем новый пароль
            $test_passw = f_passw_generator();
            $pass2 = md5($test_passw);
            $query = "UPDATE `user` SET `passw`='$pass2' ";
            $query .= "WHERE `id_user`='$id_user' LIMIT 1";
            @mysql_query($query) or die('Ошибка ' . mysql_errno());
            $msg = "Добрый день!\n\n";
            $msg .= "Пароль для доступа $test_passw \n\n";
            $msg .= URL_SITE . "\n";
            $header = "Content-Type: text/plain; charset=windows-1251\r\n";
            $header .= 'From: ' . MAIL_POST;
            $tema = 'Регистрационные данные';
            @mail($mails, $tema, $msg, $header);
            $msg = "Пароль отправлен на E-mail $mails";
        }
    }
    else {
        $serr_mail .= 'E-mail не найден<br>';
    }
}

```



```
}

// Выводим верхний колонтитул
f_header_all($title, $description, $keywords, 2);
?>
<script type="text/javascript">
<!--
function f_test_mail(email) {
    var c_Reg = /^[a-z0-9_\. \-]+@([a-z0-9\-\-]+\.)+[a-z]{2,6}$/i;
    if (!c_Reg.test(email)) return false;
    return true;
}

function f_test_passw(passw) {
    var c_Reg = /^[a-z0-9]{6,16}$/i;
    if (!c_Reg.test(passw)) return false;
    return true;
}

function f_submit_enter() {
    var login = document.getElementById("login");
    var passw = document.getElementById("passw");
    if (login.value=="" || passw.value=="") {
        window.alert("Не заполнено обязательное поле");
        login.focus();
        return false;
    }
    if (login.value.length>50 || !f_test_mail(login.value)) {
        window.alert("Недопустимое значение поля E-mail");
        login.focus();
        return false;
    }
    if (!f_test_passw(passw.value)) {
        var msg = "Недопустимое значение поля Пароль\n";
        msg += "Допустимо от 6 до 16 символов\n";
        msg += "Русские буквы использовать нельзя";
        window.alert(msg);
        passw.focus();
        return false;
    }
    return true;
}
```

```
}  
function f_submit_add() {  
    var emails = document.getElementById("emails");  
    var pass = document.getElementById("pass");  
    if (emails.value==" " || pass.value==" ") {  
        window.alert("Не заполнено обязательное поле");  
        emails.focus();  
        return false;  
    }  
    if (emails.value.length>50 || !f_test_mail(emails.value)) {  
        window.alert("Недопустимое значение поля E-mail");  
        emails.focus();  
        return false;  
    }  
    if (!f_test_passw(pass.value)) {  
        var msg = "Недопустимое значение поля Пароль\n";  
        msg += "Допустимо от 6 до 16 символов\n";  
        msg += "Русские буквы использовать нельзя";  
        window.alert(msg);  
        pass.focus();  
        return false;  
    }  
    return true;  
}  
function f_submit_mail() {  
    var mails = document.getElementById("mails");  
    if (mails.value==" ") {  
        window.alert("Поле не заполнено");  
        mails.focus();  
        return false;  
    }  
    if (mails.value.length>50 || !f_test_mail(mails.value)) {  
        window.alert("Недопустимое значение поля E-mail");  
        mails.focus();  
        return false;  
    }  
    return true;  
}  
//-->
```

```
</script>
<?php
f_table_2_start(); // Выводим таблицу второго уровня
echo '<h1>Личный кабинет</h1><br>' . "\n";
if ($err_enter != '') {
    echo '<div class="err">' . $err_enter . "</div><br>\n";
}
?>

<form action="index.php" method="POST" onsubmit="return
f_submit_enter();">
<table width="100%" border="0" cellpadding="1" align="center">
<tr><td align="right" width="50%">
<span class="bold">E-mail: </span></td><td>
<input type="text" name="login" id="login" size="23" class="txt_frm">
</td></tr><tr><td align="right" width="50%">
<span class="bold">Пароль: </span></td><td>
<input type="password" name="passw" id="passw" size="23" class="txt_frm">
</td></tr><tr><td align="right" width="50%">&nbsp;</td><td>
<input type="submit" name="enter" value="Вход" class="txt_frm">
</td></tr>
</table></form><br>
<?php
if ($err_add != '') {
    echo '<div class="err">' . $err_add . "</div><br>\n";
}
?>

<form action="index.php" method="POST" onsubmit="return f_submit_add();">
<table width="100%" border="0" cellpadding="1" align="center">
<tr><td align="right" width="50%">
<span class="bold">E-mail: </span></td><td>
<input type="text" name="emails" id="emails" size="23" class="txt_frm"
value ="<?php echo $emails_tmp; ?>">
</td></tr><tr><td align="right" width="50%">
<span class="bold">Пароль: </span></td><td>
<input type="password" name="pass" id="pass" size="23" class="txt_frm">
</td></tr>
<tr><td align="right" width="50%">&nbsp;</td><td>
<input type="submit" name="add" value="Регистрация" class="txt_frm">
</td></tr>
```

```

</table></form><br>
<?php
if ($err_mail != '') {
    echo '<div class="err">' . $err_mail . "</div><br>\n";
}
if ($msg != '') {
    echo '<div class="ok">' . $msg . "</div><br>\n";
}
?>

<form action="index.php" method="POST" onsubmit="return
f_submit_mail();">
<table width="100%" border="0" cellpadding="1" align="center">
<tr><td align="right" width="50%">
<span class="bold">E-mail: </span></td><td>
<input type="text" name="mails" id="mails" size="23" class="txt_frm">
</td></tr><tr><td align="right" width="50%">&nbsp;</td><td>
<input type="submit" name="mail_pass" value="Восстановить пароль"
class="txt_frm">
</td></tr>
</table></form><br>
<?php
// Закрываем соединение с базой данных
mysql_close($db);
f_table_2_end(); // Конец таблицы второго уровня
f_footer_user(); // Выводим нижний колонтитул
?>

```

Внимание!

Так как мы устанавливаем заголовки ответа сервера, то перед функцией `session_start()` не должно быть никаких операторов вывода. Даже пустая строка перед открывающим PHP-дескриптором (`<?php`) вызовет ошибку. Пустые строки внутри PHP-дескрипторов ошибку не генерируют, так как вывод информации осуществляется только с помощью операторов `echo` или `print`.

Почему мы восстанавливаем пароль, а не просто высылаем сохраненный в базе данных? Все дело в том, что при сохранении в базе данных мы шифруем пароль с помощью функции `md5()`. Способа расшифровать данный пароль нет. Поэтому для восстановления необходимо генерировать новый пароль. Для сравнения введенного пароля с сохраненным в базе достаточно зашифровать введенный пароль, а затем сравнить.

В ряде случаев для восстановления пароля можно использовать другой метод: при запросе на восстановление высылать на E-mail не новый пароль, а ссылку с уникальным идентификатором, и лишь после перехода по ссылке генерировать новый пароль.

Некоторые пользователи оставляют на форумах ссылки с идентификатором сессии. Чтобы предотвратить переходы по таким ссылкам, мы регистрируем в сессии время входа в систему.

```
$_SESSION["sess_data"] = date('U');
```

С помощью константы `TIME_SESS` мы можем ограничить время действия сессии. Например, если пользователь не использовал Личный кабинет в течение часа, то идентификатор сессии уничтожается.

Добавление новых сайтов в базу

Алгоритм работы программы следующий.

- ☐ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение.
- ☐ Запускаем сессию.
- ☐ Если переменные зарегистрированы в сессии, то проверяем, не истекло ли время сессии. Если нет, то продлеваем время сессии.
- ☐ Делаем запрос для определения идентификатора пользователя.
- ☐ Если форма отправлена (переменная `$_POST['go']` будет существовать):
 - создаем короткие имена переменных;
 - если директива `magic_quotes_gpc` включена, то удаляем защитные слэши;
 - удаляем все специальные символы;
 - проверяем значения переменных на допустимость;
 - с помощью функции `parse_url()` вычлениаем адрес хоста из URL-адреса;
 - если ошибок нет, то пробуем добавить сайт;
 - проверяем на повтор;
 - экранируем все спецсимволы с помощью функции `mysql_real_escape_string()`;
 - если сайт не зарегистрирован ранее, то добавляем его в базу данных.
- ☐ Если существуют ошибки, то выводим их описание.
- ☐ Если сайт успешно добавлен в базу, то выводим подтверждение.
- ☐ Выводим первую часть формы.
- ☐ Заполняем список рубрик из базы данных.

- ☐ Выводим вторую часть формы.
- ☐ Размещаем ссылку для выхода из системы.
- ☐ Закрываем соединение с базой данных.

Исходный код программы приведен в листинге 40.

Листинг 40. Содержимое файла add.php

```
<?php
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
// Стартуем сессию
session_start();
if (isset($_SESSION['sess_login']) && isset($_SESSION['sess_pass'])) {
    if (isset($_SESSION['sess_data'])) {
        $d = date('U');
        $d2 = $d - $_SESSION['sess_data'];
        if ($d2 > TIME_SESS) { // Если превышено максимальное время сессии
            unset($_SESSION['sess_login']);
            unset($_SESSION['sess_pass']);
            unset($_SESSION['sess_data']);
            session_destroy();
            header('Location: index.php');
            exit();
        }
    }
    else {
        $_SESSION['sess_data'] = $d;
    }
}
else {
    header('Location: index.php');
    exit();
}
$sess_login = $_SESSION['sess_login'];
$sess_pass = $_SESSION['sess_pass'];
$sess_login = mysql_real_escape_string($sess_login);
$sess_pass = mysql_real_escape_string($sess_pass);
```

```
$query_enter = 'SELECT `id_user` FROM `user` ' ;
$query_enter .= "WHERE `email`='$sess_login' AND
`passw`='$sess_pass'";
$res_enter = @mysql_query($query_enter) or die('Ошибка ' .
mysql_errno());
if (mysql_num_rows($res_enter)===1) {
    $row = mysql_fetch_assoc($res_enter);
    $id_user = $row['id_user'];
}
else {
    header('Location: index.php');
    exit();
}
}
else {
    header('Location: index.php');
    exit();
}

// Заголовок
$title = 'Добавить сайт';
// Описание страницы
$description = '';
// Ключевые слова для поисковых машин
$keywords = '';
// Выводим верхний колонтитул
f_header_all($title, $description, $keywords, 2);
?>
<h1>Личный кабинет</h1><br>
<script type="text/javascript">
<!--
function f_submit() {
    var urls = document.getElementById("urls");
    var titles = document.getElementById("titles");
    var descr = document.getElementById("descr");
    var rubr = document.getElementById("rubr");
    var msg = '';
    if (urls.value.length<8 || titles.value=="" || descr.value=="") {
        window.alert("Поле не заполнено");
    }
}
```

```

        return false;
    }
    if (titles.value.length>70 || descr.value.length>500) {
        msg = "Длина Названия ресурса не должна превышать ";
        msg += "70 символов, а Описания - 500";
        window.alert(msg);
        return false;
    }
    var c_Reg = /^http:\/\/(www\.)?([a-z0-9\-\]+\.)+[a-z]{2,6}$/i;
    if (!c_Reg.test(urls.value)) {
        msg = "Недопустимый URL\nПравильно - \"http://www.mail.ru\"";
        msg += "Неправильно - \"http://www.mail.ru/\"";
        window.alert(msg);
        urls.focus();
        return false;
    }
    if (rubr.options[rubr.selectedIndex].value==0) {
        window.alert("Рубрика не выбрана");
        return false;
    }
    return true;
}
//-->
</script>
<?php
$url = '';
$rubr = '';
$title = '';
$descr = '';
$err_add = '';
$ok_add = '';
if (isset($_POST['go'])) { // Если форма отправлена
    if (isset($_POST['urls'])) $url = $_POST['urls'];
    else $url = '';
    if (isset($_POST['rubr'])) $rubr = (int)$_POST['rubr'];
    else $rubr = 0;
    if (isset($_POST['titles'])) $title = $_POST['titles'];
    else $title = '';
    if (isset($_POST['descr'])) $descr = $_POST['descr'];

```



```
else $descr = '';
// Если директива magic_quotes_gpc включена,
// то удаляем защитные слэши перед спецсимволами
if (get_magic_quotes_gpc()) {
    $titles = stripslashes($titles);
    $descr = stripslashes($descr);
}
$titles = str_replace("\n", ' ', $titles);
$titles = str_replace("\t", ' ', $titles);
$titles = str_replace("\r", '', $titles);
$descr = str_replace("\n", ' ', $descr);
$descr = str_replace("\t", ' ', $descr);
$descr = str_replace("\r", '', $descr);
$titles = trim($titles);
$descr = trim($descr);
if (strlen($titles)>70 || strlen($descr)>500) {
    $err_add .= 'Длина Названия ресурса не должна превышать ';
    $err_add .= '70 символов, а Описания - 500<br>';
}
if (preg_match('/[a-zA-яё0-9,.\'\"]{26}/is', $titles)) {
    $err_add .= 'Поле Названия ресурса содержит более 25 ';
    $err_add .= 'символов без пробела<br>';
}
if (preg_match('/[a-zA-яё0-9,.\'\"]{26}/is', $descr)) {
    $err_add .= 'Поле Описание содержит более 25 символов ';
    $err_add .= 'без пробела<br>';
}
if (strlen($titles)<2 || strlen($descr)<2) {
    $err_add .= 'Не заполнено обязательное поле<br>';
}
if (!preg_match('#^http://(www\.)?([a-z0-9-]+\.)+[a-z]{2,6}$#is',
$url) ||
    strlen($url)>200) {
    $err_add .= 'Недопустимый URL<br>';
}
else {
    $url = strtolower($url);
    $mass_url = parse_url($url);
    $host = $mass_url['host'];
```

```

        if (preg_match('/^www\./is', $host)) {
            $kol = strlen($host) - 4;
            $host = substr($host,4,$kol);
        }
    }
    if (!preg_match('/^[0-9]+$s', $rubr) || $rubr==0) {
        $err_add .= 'Рубрика не выбрана или имеет неправильный формат<br>';
    }
    if ($err_add == '') {
        //Если ошибок нет пробуем добавить сайт
        $q = "SELECT `id_site` FROM `site` WHERE `url_site_dop`='$host'";
        $res_add = @mysql_query($q) or die('Ошибка ' . mysql_errno());
        if (mysql_num_rows($res_add)==0) {
            // Если сайт не зарегистрирован ранее добавляем сайт
            $titles = mysql_real_escape_string($titles);
            $descr = mysql_real_escape_string($descr);
            $q = "INSERT INTO `site` VALUES (NULL, '$rubr', '$id_user', ";
            $q .= "'$urls', '$host', '$titles', '$descr', 'n', 0,
CURDATE())";
            if (@mysql_query($q)) {
                $ok_add = "Сайт $urls добавлен в базу данных<br>";
                $urls = 'http://';
                $rubr = 0;
                $titles = $descr = '';
            }
            else {
                $err_add .= 'Попробуйте сделать запрос через некоторое
время<br>';
            }
        }
        else {
            $err_add .= "Сайт $urls был зарегистрирован ранее<br>";
            $urls = 'http://';
            $rubr = 0;
            $titles = $descr = '';
        }
    }
}
else {

```

```
$urls = 'http://';
}
if ($err_add != '') {
    echo '<div class="err">' . $err_add . "</div><br>\n";
    $titles = stripslashes($titles);
    $descr = stripslashes($descr);
    $titles = htmlspecialchars($titles);
    $descr = htmlspecialchars($descr);
}
if ($ok_add != '') {
    echo '<div class="ok">' . $ok_add . "</div><br>\n";
}

?>
<form action="add.php" method="POST" name="frm" id="frm" onsubmit="return
f_submit();">
<table width="100%" border="0" cellpadding="1" align="center">
<tr><td align="right" width="40%">
<span class="bold">URL: </span>
</td><td>
<input type="text" name="urls" id="urls" size="40"
value="<?php echo $urls; ?>" class="txt_frm">
</td></tr>
<tr><td align="right" width="40%">
<span class="bold">Рубрика: </span>
</td><td>
<select name="rubr" id="rubr" class="select_frm">
<option value="0">-----Выберите рубрику-----</option>
<?php
$res = @mysql_query('SELECT * FROM `rubr` ORDER BY `name_rubr`');
while($row = mysql_fetch_row($res)) {
    echo '<option value="' . $row[0];
    if ($rubr==$row[0]) echo '" selected>';
    else echo '">';
    echo htmlspecialchars($row[1]);
    echo "</option>\n";
}
?>
</select>
```

```

</td></tr>
<tr><td align="right" width="40%">
<span class="bold">Название ресурса: </span>
</td><td>
<input type="text" name="titles" id="titles" size="40" maxlength="70"
class="txt_frm" value="<?php echo $titles; ?>">
</td></tr>
<tr><td align="right" width="40%">
<span class="bold">Описание ресурса:<BR> (до 500 символов)</span>
</td><td>
<textarea name="descr" id="descr" cols="15" rows="20"
class="textarea_frm"
style="width: 300px; height: 150px;"><?php echo $descr; ?></textarea>
</td></tr>
<tr><td align="right" width="40%">&nbsp;  </td><td>
<input type="submit" name="go" value="Добавить сайт" class="txt_frm">
</td></tr>
</table>
</form><br>
<div align="center"><a href="exit.php">Выйти из системы</a></div>
<?php
// Закрываем соединение с базой данных
mysql_close($db);
f_footer_user(); // Выводим нижний колонтитул
?>

```

Для выхода из системы создадим файл `exit.php` с содержимым листинга 41.

Листинг 41. Содержимое файла `exit.php`

```

<?php
session_start(); // Запускаем сессию
session_unset(); // Удаляем все переменные
session_destroy(); // Удаляем идентификатор
header('Location: ../index.php');
?>

```

Создание администраторской части сайта и защита с помощью сервера Apache

В качестве примера создадим Личный кабинет администратора сайта и защитим его не с помощью механизма сессий, а при помощи сервера Apache. Для этого в каталоге необходимо разместить файл `.htaccess` со следующим содержанием:

```
AuthType Basic
AuthName "Enter password"
AuthUserFile <Путь к файлу .htpasswd>
<Limit GET POST>
    require valid-user
</Limit>
```

Кроме того, необходимо создать файл `.htpasswd`, в котором должен быть расположен пароль для доступа.

Кабинет администратора разместим в каталоге `admin` и создадим следующие файлы:

- ☐ `index.php` — вывод статистики и сообщений об ошибках;
- ☐ `rubr.php` — для добавления новой рубрики, удаления или переименования существующей;
- ☐ `moder.php` — для вывода сайтов, находящихся на модерации, и результатов поиска по URL-адресу с возможностью одобрения или удаления выделенных сайтов;
- ☐ `catalog.php` — для редактирования описания сайта;
- ☐ `gbook.php` — вывод содержимого гостевой книги с выделением новых сообщений, а также возможностью одобрения или удаления выделенных сообщений.

Примечание

Каталог не обязательно должен называться `admin`. В целях безопасности лучше переименовать папку.

Добавление, изменение и удаление рубрик

Алгоритм работы программы следующий.

- ☐ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение.
- ☐ Если отправлена форма для добавления новой рубрики (переменная `$_POST['rubr_add']` будет существовать):
 - создаем короткое имя переменной;

- если директива `magic_quotes_gpc` включена, то удаляем защитные слешы;
- проверяем корректность введенных данных;
- если ошибки не найдены:
 - экранируем все спецсимволы с помощью функции `mysql_real_escape_string()`;
 - проверяем название новой рубрики на уникальность;
 - добавляем рубрику в базу данных.
- ☐ Если отправлена форма для удаления рубрики (переменная `$_POST['del_rubr']` будет существовать):
 - создаем короткое имя идентификатора рубрики;
 - проверяем корректность введенных данных;
 - если ошибки не найдены:
 - проверяем существование сайтов, зарегистрированных в данной рубрике;
 - если зарегистрированных сайтов нет, то удаляем рубрику.
- ☐ Если отправлена форма для изменения названия рубрики (переменная `$_POST['change']` будет существовать):
 - создаем короткие имена переменных;
 - если директива `magic_quotes_gpc` включена, то удаляем защитные слешы;
 - проверяем корректность введенных данных;
 - если ошибки не найдены:
 - экранируем все спецсимволы с помощью функции `mysql_real_escape_string()`;
 - проверяем название новой рубрики на уникальность;
 - проверяем существование идентификатора рубрики;
 - изменяем название рубрики.
- ☐ Если возникли ошибки при добавлении новой рубрики, то выводим их описание.
- ☐ В случае успешного добавления новой рубрики выводим подтверждение.
- ☐ Выводим форму для добавления новой рубрики.
- ☐ Если возникли ошибки при изменении или удалении рубрики, то выводим их описание.
- ☐ В случае успешного изменения или удаления рубрики выводим подтверждение.
- ☐ Выводим форму для изменения или удаления рубрики.
- ☐ Закрываем соединение с базой данных.

Исходный код программы приведен в листинге 42.

Листинг 42. Содержимое файла rubr.php

```
<?php
header('Expires: Sun, 27 May 2007 01:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Pragma: no-cache');
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
// Заголовок
$title = 'Администрирование';
// Выводим верхний колонтитул
f_header_all($title, '', '', 2);

// Добавление новой рубрики
$error_rubr = '';
$ok_rubr = '';
if (isset($_POST['rubr_add'])) {
    if (isset($_POST['rubrika'])) $rubrika = $_POST['rubrika'];
    else $rubrika = '';
    // Если директива magic_quotes_gpc включена,
    // то удаляем защитные слэши перед спецсимволами
    if (get_magic_quotes_gpc()) {
        $rubrika = stripslashes($rubrika);
    }
    if (strlen($rubrika)>150) {
        $error_rubr .= 'Длина Рубрики больше допустимой<br>';
    }
    if (strlen($rubrika)==0) {
        $error_rubr .= 'Поле не заполнено<br>';
    }
    if ($error_rubr == '') {
        $rubrika = mysql_real_escape_string($rubrika);
        $q = "SELECT * FROM `rubr` WHERE `name_rubr`='$rubrika'";
        $res_rubr = mysql_query($q);
```

```

        if (mysql_num_rows($res_rubr)===0) {
            $query_rubr = "INSERT INTO `rubr` VALUES (NULL, '$rubrika')";
            if (mysql_query($query_rubr)) {
                $ok_rubr = 'Рубрика успешно добавлена<br>';
            }
            else {
                $err_rubr .= 'Попробуйте сделать запрос через некоторое
время<br>';
            }
        }
        else {
            $err_rubr .= 'Рубрика уже зарегистрирована ранее<br>';
        }
    }
}

// Удаление рубрики
$ok_rubrikator = '';
$err_rubrikator = '';
if (isset($_POST['del_rubr'])) {
    if (isset($_POST['rubrikator'])) $rubrikator =
(int)$_POST['rubrikator'];
    else $rubrikator = 0;
    if (!preg_match('/^[0-9]+$/', $rubrikator) || $rubrikator == 0) {
        $err_rubrikator = 'Недопустимый формат номера рубрики!!!<br>';
    }
    if ($err_rubrikator == '') {
        $q = "SELECT * FROM `site` WHERE `id_rubr`=$rubrikator";
        $res_rubrikator = mysql_query($q);
        if (mysql_num_rows($res_rubrikator)===0) {
            $query_rubrikator = 'DELETE FROM `rubr` WHERE ' ;
            $query_rubrikator .= "`id_rubr`=$rubrikator LIMIT 1";
            if (@mysql_query($query_rubrikator)) {
                $ok_rubrikator = 'Рубрика успешно удалена<br>';
            }
            else {
                $err_rubrikator .= 'Попробуйте сделать запрос через некоторое
время<br>';
            }
        }
    }
}

```



```
    }
    else {
        $err_rubrikator .= 'Нельзя удалить рубрику, т.к. есть сайты в
указанной рубрике<br>';
    }
}

}

// Изменение названия рубрики
if (isset($_POST['change'])) {
    if (isset($_POST['rubrikator'])) $rubrikator =
(int)$_POST['rubrikator'];
    else $rubrikator = 0;
    if (isset($_POST['change_rubr'])) $change_rubr =
$_POST['change_rubr'];
    else $change_rubr = '';
    if (!preg_match('/^[0-9]+$/', $rubrikator) || $rubrikator == 0) {
        $err_rubrikator = 'Недопустимый формат номера рубрики<br>';
    }
    // Если директива magic_quotes_gpc включена,
    // то удаляем защитные слэши перед спецсимволами
    if (get_magic_quotes_gpc()) {
        $change_rubr = stripslashes($change_rubr);
    }
    if (strlen($change_rubr)>150) {
        $err_rubrikator .= 'Длина Рубрики больше допустимой!!!<br>';
    }
    if (strlen($change_rubr)==0) {
        $err_rubrikator .= 'Поле не заполнено!!<br>';
    }
    if ($err_rubrikator == '') {
        $change_rubr = mysql_real_escape_string($change_rubr);
        $q = "SELECT * FROM `rubr` WHERE `name_rubr`='$change_rubr'";
        $res_change_rubr = mysql_query($q);
        if (mysql_num_rows($res_change_rubr)===0) {
            $q = "SELECT * FROM `rubr` WHERE `id_rubr`=$rubrikator";
            $res_change_r = @mysql_query($q);
            if (mysql_num_rows($res_change_r)===1) {
                $query_change_rubr = 'UPDATE `rubr` SET ';
```

```

$query_change_rubr .= "`name_rubr`='$change_rubr' ";
$query_change_rubr .= "WHERE `id_rubr`=$rubrikator LIMIT 1";
if (@mysql_query($query_change_rubr)) {
    $ok_rubrikator = 'Рубрика успешно переименована<br>';
}
else {
    $err_rubrikator .= 'Попробуйте сделать запрос через
некоторое время<br>';
}
}
else {
    $err_rubrikator .= 'Рубрика не найдена<br>';
}
}
else {
    $err_rubrikator .= 'Рубрика с таким названием уже
существует<br>';
}
}
}
?>
<script type="text/javascript">
<!--
function f_submit_rubr() {
    var rubrika = document.getElementById("rubrika");
    if (rubrika.value=="") {
        window.alert("Поле не заполнено");
        rubrika.focus();
        return false;
    }
    if (rubrika.value.length>150) {
        window.alert("Недопустимое значение поля Рубрика");
        rubrika.focus();
        return false;
    }
    return true;
}

function f_change() {

```

```
var ch = document.getElementById("change_rubr");
var rubr = document.getElementById("rubrikator");
ch.value = rubr.options[rubr.selectedIndex].text;
}
function f_change_rubr(e) {
    e = e || window.event;
    var ch = document.getElementById("change_rubr");
    var rubr = document.getElementById("rubrikator");
    if (ch.value=="") {
        window.alert("Поле не заполнено");
        ch.focus();
        if (e.preventDefault) e.preventDefault();
        else e.returnValue = false;
    }
    else {
        var msg= "Вы действительно хотите переименовать\нрубрику \";
        msg += rubr.options[rubr.selectedIndex].text;
        msg += "\" в \"" + ch.value + "\"?";
        if (window.confirm(msg)) {}
        else {
            if (e.preventDefault) e.preventDefault();
            else e.returnValue = false;
        }
    }
}
function f_del_rubr(e) {
    e = e || window.event;
    var rubr = document.getElementById("rubrikator");
    var msg= "Вы действительно хотите УДАЛИТЬ\нрубрику \";
    msg += rubr.options[rubr.selectedIndex].text;
    msg += "\"?";
    if (window.confirm(msg)) {}
    else {
        if (e.preventDefault) e.preventDefault();
        else e.returnValue = false;
    }
}
//-->
</script>
```

```

<h1>Рубрикатор</h1><br><br>
<?php
if ($err_rubr != '') {
    echo '<div class="err">' . $err_rubr . '</div><br>';
}
if ($ok_rubr != '') {
    echo '<div class="ok">' . $ok_rubr . '</div><br>';
}
?>
<form action="rubr.php" method="POST" onsubmit="return f_submit_rubr();">
<table width="450" border="0" cellpadding="1" align="center">
<tr><td align="right" width="50%">
<span class="bold">Рубрика: </span></td><td>
<input type="text" name="rubrika" id="rubrika" maxlength="50" size="23"
class="txt_frm">
</td></tr><tr><td align="right" width="50%">&nbsp;</td><td>
<input type="submit" name="rubr_add" value="Добавить рубрику"
class="txt_frm">
</td></tr>
</table>
</form><br>
<?php
if ($err_rubrikator != '') {
    echo '<div class="err">' . $err_rubrikator . '</div><br>';
}
if ($ok_rubrikator != '') {
    echo '<div class="ok">' . $ok_rubrikator . '</div><br>';
}
?>
<form action="rubr.php" method="POST">
<table width="450" border="0" cellpadding="1" align="center">
<tr><td align="right" width="50%">
<span class="bold">Рубрика: </span></td><td>
<select name="rubrikator" id="rubrikator" class="select_frm"
style="width: 154px" onchange="f_change();">
<?php
$res_rubrikator = @mysql_query('SELECT * FROM `rubr` ORDER BY
`name_rubr`');
while($row = mysql_fetch_row($res_rubrikator)) {
    echo '<option value="' . $row[0] . '">';

```

```

        echo htmlspecialchars($row[1]);
        echo "</option>\n";
    }
    ?>
</select>
</td></tr><tr><td align="right" width="50%">
<span class="bold">Изменить на: </span></td><td>
<input type="text" name="change_rubr" id="change_rubr" maxlength="50"
size="23" class="txt_frm">
</td></tr><tr><td align="right" width="50%">
<input type="submit" name="del_rubr" value="Удалить рубрику"
class="txt_frm" style="background-color: #FF7F50"
onclick="f_del_rubr(event);">
</td><td>
<input type="submit" name="change" value="Изменить название"
class="txt_frm" onclick="f_change_rubr(event);">
</td></tr>
</table>
</form><br>
<?php
// Закрываем соединение с базой данных
mysql_close($db);
f_footer_admin(); // Выводим нижний колонтитул
?>

```

Обратите внимание, форма для изменения или удаления рубрики содержит две кнопки **Submit**. При нажатии на одну из них будет создана переменная `$_POST['change']`, а при нажатии на другую — `$_POST['del_rubr']`. Проверка на существование переменной позволяет обработать нажатие соответствующей кнопки.

Если в таблице `site` есть сайты, зарегистрированные на удаляемую рубрику, то рубрику удалять нельзя, так как это повредит целостность базы данных. Для предотвращения случайного нажатия кнопки **Удалить** мы выводим диалоговое окно для подтверждения.

Вывод сайтов, требующих проверки

Все добавляемые сайты должны пройти проверку модератором. Для этого при добавлении сайта мы указываем статус "на модерации". Сайты, имеющие такой статус, в каталоге не выводятся.

Для вывода сайтов, требующих проверки, создадим файл `moder.php`. Кроме того, этот файл будет использоваться для отображения результатов поиска по URL-адресу.

Алгоритм работы программы следующий.

- ☐ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение.
- ☐ Если форма для удаления сайта отправлена (переменные `$_GET['del_msg']` и `$_GET['check']` будут существовать):
 - в цикле проверяем идентификатор сайта, и если это число, то удаляем сайт.
- ☐ Если форма для одобрения сайта отправлена (переменные `$_GET['upd_msg']` и `$_GET['check']` будут существовать):
 - в цикле проверяем идентификатор сайта, и если это число, то изменяем статус сайта на "активен".
- ☐ Создаем короткое имя для номера страницы.
- ☐ Если форма для поиска по URL-адресу отправлена (переменная `$_GET['search']` будет существовать):
 - создаем короткое имя переменной;
 - если директива `magic_quotes_gpc` включена, то удаляем защитные слеш;
 - удаляем все специальные символы;
 - проверяем значения переменных на допустимость;
 - если ошибок нет:
 - экранируем все спецсимволы с помощью функции `mysql_real_escape_string()`;
 - создаем поисковый запрос.
- ☐ Если поисковый запрос не существует (поисковая форма не отправлена), создаем запрос для вывода сайтов, требующих проверки.
- ☐ Делаем запрос на выборку данных.
- ☐ Проверяем корректность номера страницы и рассчитываем начальную и конечную позицию.
- ☐ Выводим начало формы для удаления или одобрения сайта.
- ☐ Выводим описания сайтов. Рядом с каждым описанием добавляем поле для установки флажка. Название сайта служит ссылкой на страницу редактирования описания сайта (`catalog.php`).
- ☐ Выводим конец формы для удаления или одобрения сайта.
- ☐ Выводим ссылки для перемещения к следующей и предыдущей страницам.
- ☐ Если возникли ошибки, то выводим их описание.
- ☐ Закрываем соединение с базой данных.

Исходный код программы приведен в листинге 43.

Листинг 43. Содержимое файла moder.php

```
<?php
header('Expires: Sun, 27 May 2007 01:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Pragma: no-cache');
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
// Заголовок
$title = 'Администрирование';
// Выводим верхний колонтитул
f_header_all($title, '', '', 2);
?>
<h1>Сайты на модерации</h1><br>
<table align="center" width="600" bgcolor="#FFFFFF" border="0"
cellpadding="2" cellspacing="0">
<tr><td align="left" bgcolor="#FFFFFF">
<?php
// Удаляем сайт
if (isset($_GET['del_msg']) && isset($_GET['check']) &&
is_array($_GET['check'])) {
    foreach ($_GET['check'] as $value) {
        if (preg_match('/^[0-9]+$/', $value)) {
            $del = "DELETE FROM `site` WHERE `id_site`=$value LIMIT 1";
            @mysql_query($del);
        }
    }
}

// Ставим галочку "Одобрено"
if (isset($_GET['upd_msg']) && isset($_GET['check']) &&
is_array($_GET['check'])) {
    foreach ($_GET['check'] as $value) {
        if (preg_match('/^[0-9]+$/', $value)) {
            $upd = "UPDATE `site` SET `status_site`='y' ";
            $upd .= "WHERE `id_site`=$value LIMIT 1";
            @mysql_query($upd);
        }
    }
}
```

```

    }
}
if (isset($_GET['page'])) $page = (int)$_GET['page'];
else $page = 1;
if ($page < 1) $page = 1;
$start = ($page - 1) * COUNT_POS_PAGE;
$error_text = '';
if (isset($_GET['search'])) {
    // Выводим поисковый запрос
    $text = $_GET['search'];
    if (get_magic_quotes_gpc()) {
        $text = stripslashes($text);
    }
    $text = str_replace("\n", ' ', $text);
    $text = str_replace("\t", ' ', $text);
    $text = str_replace("\r", '', $text);
    $text = trim($text);
    if ($text == '') {
        $error_text = 'Не задана строка поиска<br>';
    }
    else {
        $text2 = mysql_real_escape_string($text);
        $query = 'SELECT SQL_CALC_FOUND_ROWS `site`.`id_site`, '
        $query .= '`site`.`url_site`, `site`.`title`, `site`.`descr`, '
        $query .= '`rubr`.`name_rubr` FROM `site`, `rubr` '
        $query .= 'WHERE `site`.`id_rubr`=`rubr`.`id_rubr` AND '
        $query .= "`url_site` LIKE '%$text2%' ";
        $query .= 'LIMIT ' . $start . ', ' . COUNT_POS_PAGE;
    }
}
if (!isset($query)) {
    // Выводим сайты для модерации
    $query = 'SELECT SQL_CALC_FOUND_ROWS `site`.`id_site`, '
    $query .= '`site`.`url_site`, `site`.`title`, `site`.`descr`, '
    $query .= '`rubr`.`name_rubr` FROM `site`, `rubr` '
    $query .= 'WHERE `site`.`id_rubr`=`rubr`.`id_rubr` AND '
    $query .= "`site`.`status_site`='n' ORDER BY `id_site` ";
    $query .= 'LIMIT ' . $start . ', ' . COUNT_POS_PAGE;
}

```



```

}
$res = @mysql_query($query) or die('Ошибка ' . mysql_errno());
$count = mysql_num_rows($res);
$r = @mysql_query('SELECT FOUND_ROWS()') or die('Ошибка');
$pole = mysql_fetch_row($r);
$allcount = $pole[0];
if ($count>0) {
    echo '<form action="moder.php">';
    echo '<div align="center">';
    if (isset($text)) {
        echo '<input type="hidden" name="search" value="' .
htmlspecialchars($text) . '">';
    }
    echo '<input type="hidden" name="page" value="' . $page . '">';
    echo '<input type="submit" name="upd_msg" value="Одобрить">';
    echo '<input type="submit" name="del_msg" value="Удалить">';
    echo "</div><br>\n\n";
    while ($p = mysql_fetch_row($res)) {
        // Выводим описание
        echo '<table width="100%" align="center" border="0"
cellspacing="0">';
        echo '<tr><td class="color_table">';
        echo '<input type="checkbox" name="check[]" ';
        echo 'value="' . $p[0] . '"> ';
        echo '<a href="catalog.php?id=' . $p[0] . '" ';
        echo 'target="_blank">' . htmlspecialchars($p[2]) .
'</a></td></tr>';
        echo '<tr><td>' . htmlspecialchars($p[3]) . '<br>';
        echo '<span class="bold">Рубрика: ' . $p[4] . '<br>';
        echo 'URL сайта: </span><a href="' . $p[1] . '" ';
        echo 'target="_blank">' . $p[1] . '</a>';
        echo "</td></tr></table><br>\n\n";
    }
    echo "</form>\n";
}
elseif ($allcount == 0) {
    echo '<div align="center" class="bold">';
    echo 'По вашему запросу ничего не найдено</div><br>';
}

```

```

else {
    echo '<div align="center">';
    if (isset($text)) {
        echo '<a href="?page=1&search=' . urlencode($text) . '">';
    }
    else {
        echo '<a href="?page=1">';
    }
    echo 'На первую страницу результатов</a></div>';
    $allcount = 0;
}

if ($allcount>COUNT_POS_PAGE) {
    if (isset($text)) {
        $src = '&search=' . urlencode($text);
    }
    else {
        $src = '';
    }
    echo '<div align="center">';
    if (($start-COUNT_POS_PAGE)>=0) {
        echo '<a href="?page=' . ($page-1) . $src;
        echo '">&lt;&lt; Предыдущая страница</a> ';
    }
    if (($start+COUNT_POS_PAGE)<$allcount) {
        echo '<a href="?page=' . ($page+1) . $src;
        echo '">Следующая страница &gt;&gt;</a> ';
    }
    echo '</div>';
}

if ($err_text != '') {
    echo '<div class="err">' . $err_text . '</div><br>';
}

?>
</td></tr>
</table>
<?php
// Закрываем соединение с базой данных
mysql_close($db);
f_footer_admin(); // Выводим нижний колонтитул

```

?>

Редактирование описания произвольного сайта

Алгоритм работы программы следующий.

- ☐ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение.
- ☐ Если форма для изменения описания сайта отправлена (переменная `$_POST["go"]` будет существовать):
 - создаем короткие имена переменных;
 - если директива `magic_quotes_gpc` включена, то удаляем защитные слешы;
 - удаляем все специальные символы;
 - проверяем корректность введенных данных;
 - если ошибок нет, то изменяем описание сайта.
- ☐ Если возникли ошибки, то выводим их описание.
- ☐ Если существует сообщение об удачном изменении описания сайта, то выводим его.
- ☐ Если существует идентификатор сайта (переменная `$_GET['id']` будет существовать):
 - создаем короткое имя переменной;
 - проверяем идентификатор сайта, и если это число, то делаем запрос;
 - выводим заполненную форму.
- ☐ Если возникли ошибки, то выводим их описание.
- ☐ Закрываем соединение с базой данных.

Исходный код программы приведен в листинге 44.

Листинг 44. Содержимое файла `catalog.php`

```
<?php
header('Expires: Sun, 27 May 2007 01:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Pragma: no-cache');
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
```

```
// Заголовок
$title = 'Администрирование';
// Выводим верхний колонтитул
f_header_all($title, '', '', 2);
?>
<h1>Редактирование описания сайта</h1><br>
<script type="text/javascript">
<!--
function f_submit() {
    var urls = document.getElementById("urls");
    var titles = document.getElementById("titles");
    var descr = document.getElementById("descr");
    var rubr = document.getElementById("rubr");
    var msg = '';
    if (urls.value.length<8 || titles.value=="" || descr.value=="") {
        window.alert("Поле не заполнено");
        return false;
    }
    if (titles.value.length>70) {
        msg = "Длина Названия не должна превышать 70 символов";
        window.alert(msg);
        return false;
    }
    c_Reg = /^http:\/\/(www\.)?([a-z0-9\-\]+\.)+[a-z]{2,6}$/i;
    if (!c_Reg.test(urls.value)) {
        msg = "Недопустимый URL\nПравильно - ";
        msg += "\"http://www.mail.ru\" \nНеправильно - ";
        msg += "\"http://www.mail.ru/\""
        window.alert(msg);
        urls.focus();
        return false;
    }
    if (rubr.options[rubr.selectedIndex].value==0) {
        window.alert("Рубрика не выбрана");
        return false;
    }
    return true;
}
//-->
```

```
</script>
<?php
$err_add = '';
$ok_add = '';
if (isset($_POST['go'])) { // Если форма отправлена
    if (isset($_POST['urls'])) $urls = $_POST['urls'];
    else $urls = '';
    if (isset($_POST['rubr'])) $rubr = (int)$_POST['rubr'];
    else $rubr = 0;
    if (isset($_POST['titles'])) $titles = $_POST['titles'];
    else $titles = '';
    if (isset($_POST['descr'])) $descr = $_POST['descr'];
    else $descr = '';
    if (isset($_POST['iq_site'])) $iq_site = (int)$_POST['iq_site'];
    else $iq_site = 0;
    if (isset($_POST['status_site'])) $status_site =
$_POST['status_site'];
    else $status_site = '';
    if (isset($_POST['id_site'])) $id_site = (int)$_POST['id_site'];
    else $id_site = 0;
    // Если директива magic_quotes_gpc включена,
    // то удаляем защитные слэши перед спецсимволами
    if (get_magic_quotes_gpc()) {
        $titles = stripslashes($titles);
        $descr = stripslashes($descr);
    }
    $titles = str_replace("\n", ' ', $titles);
    $titles = str_replace("\t", ' ', $titles);
    $titles = str_replace("\r", '', $titles);
    $descr = str_replace("\n", ' ', $descr);
    $descr = str_replace("\t", ' ', $descr);
    $descr = str_replace("\r", '', $descr);
    $titles = trim($titles);
    $descr = trim($descr);
    if (!preg_match('/^[0-9]+$/', $id_site) || $id_site==0) {
        $err_add .= 'Идентификатор имеет неправильный формат<br>';
    }
    if (!preg_match('/^[0-9]+$/', $iq_site)) {
        $err_add .= 'Параметр iq_site имеет неправильный формат<br>';
    }
}
```

```

}
if (strlen($titles)>70) {
    $err_add .= 'Длина Названия ресурса не должна превышать ';
    $err_add .= '70 символов<br>';
}
if (strlen($titles)<2 || strlen($descr)<2) {
    $err_add .= 'Не заполнено обязательное поле<br>';
}
if (!preg_match('#^http://(www\.)?([a-z0-9-]+\.)+[a-z]{2,6}$#is',
$urls) ||
    strlen($urls)>200) {
    $err_add .= 'Недопустимый URL !<br>';
}
else {
    $urls = strtolower($urls);
    $mass_url = parse_url($urls);
    $host = $mass_url['host'];
    if (preg_match('/^www\./is', $host)) {
        $kol = strlen($host) - 4;
        $host = substr($host, 4, $kol);
    }
}
if ($status_site==1) $status = 'y';
elseif ($status_site==2) $status = 'n';
elseif ($status_site==3) $status = 's';
else {
    $err_add .= 'Статус сайта имеет неправильный формат<br>';
}
if (!preg_match('/^[0-9]+$\/s', $rubr) || $rubr==0) {
    $err_add .= 'Рубрика не выбрана или имеет неправильный формат<br>';
}
if ($err_add == '') {
    //Если ошибок нет
    $res = @mysql_query("SELECT * FROM `site` WHERE
`id_site`=$id_site");
    if ($res && mysql_num_rows($res)==1) {
        // Если сайт найден изменяем описание
        $titles = mysql_real_escape_string($titles);
        $descr = mysql_real_escape_string($descr);
    }
}

```

```

$query = "UPDATE `site` SET `id_rubr`=$rubr, ";
$query .= "`url_site`='$urls', `url_site_dop`='$host', ";
$query .= "`title`='$titles', `descr`='$descr', ";
$query .= "`status_site`='$status', `iq_site`=$iq_site ";
$query .= "WHERE `id_site`=$id_site LIMIT 1";
if (@mysql_query($query)) {
    $ok_add = 'Информация о сайте ' . $urls . '
обновлена<br><br>';
    $ok_add .= '<a href="?id=' . $id_site . '">Назад</a><br>';
}
else {
    $err_add .= 'Попробуйте сделать запрос через некоторое
время<br>';
}
else {
    $err_add .= 'Сайт ' . $urls . ' не найден<br>';
}
}
}
if ($err_add != '') {
    echo '<div class="err">' . $err_add . '</div><br>';
}
if ($ok_add != '') {
    echo '<div class="ok">' . $ok_add . '</div><br>';
}
}
$serr_enter = '';
if (isset($_GET['id'])) {
    // Выводим заполненную форму
    $id = (int)$_GET['id'];
    if (!preg_match('/^[0-9]+$/', $id) || $id == 0) {
        $serr_enter .= 'Недопустимые символы в параметре id<br>';
    }
    else {
        $res = @mysql_query("SELECT * FROM `site` WHERE `id_site`=$id");
        if (mysql_num_rows($res)===1) {
            $r = mysql_fetch_assoc($res);
        }
    }
}
?>
<table width="500" align="center" border="0" cellspacing="0">

```

```

<tr><td class="color_table">
<a href="<?php echo $r['url_site']; ?>" target="_blank">
<?php echo htmlspecialchars($r['title']); ?></a></td></tr>
<tr><td><?php echo htmlspecialchars($r['descr']);
?></td></tr></table><br>
<form action="catalog.php" method="POST" onsubmit="return f_submit();">
<table width="100%" border="0" cellpadding="1" align="center">
<tr><td align="right" width="40%">
<span class="bold">URL: </span>
</td><td>
<input type="text" name="urls" id="urls" size="40"
value="<?php echo $r['url_site']; ?>" class="txt_frm">
<input type="hidden" name="id_site"
value="<?php echo $r['id_site']; ?>">
</td></tr>
<tr><td align="right" width="40%">
<span class="bold">Рубрика: </span>
</td><td>
<select name="rubr" id="rubr" class="select_frm">
<option value="0">-----Выберите рубрику-----</option>
<?php
$res_rubrikator = @mysql_query('SELECT * FROM `rubr` ORDER BY
`name_rubr`');
while($row = mysql_fetch_row($res_rubrikator)) {
    echo '<option value="' . $row[0];
    if ($r['id_rubr']==$row[0]) echo ' " selected>';
    else echo '>';
    echo htmlspecialchars($row[1]);
    echo "</option>\n";
}
?>

</select>
</td></tr>
<tr><td align="right" width="40%">
<span class="bold">Название ресурса: </span>
</td><td>
<input type="text" name="titles" id="titles" size="40" class="txt_frm"
value="<?php echo htmlspecialchars($r['title']); ?>">

```



```

</td></tr>
<tr><td align="right" width="40%">
<span class="bold">Описание ресурса: </span>
</td><td>
<textarea name="descr" id="descr" cols="15" rows="20"
class="textarea_frm" style="width:300px; height:150px">
<?php echo htmlspecialchars($r['descr']); ?></textarea>
</td></tr>
<tr><td align="right" width="40%">
<span class="bold">IQ-сайта: </span>
</td><td>
<input type="text" name="iq_site" id="iq_site" size="40" class="txt_frm"
value="<?php echo $r['iq_site']; ?>">
</td></tr>
<tr><td align="right" width="40%">
<span class="bold">Статус сайта: </span>
</td><td>
<input type="radio" name="status_site"
value="1"<?php echo ($r['status_site']=='y')?' checked':''; ?>>
<span class="bold">Активен</span><br>
<input type="radio" name="status_site"
value="2"<?php echo ($r['status_site']=='n')?' checked':''; ?>>
<span class="bold">На модерации</span><br>
<input type="radio" name="status_site"
value="3"<?php echo ($r['status_site']=='s')?' checked':''; ?>>
<span class="bold">Временно не работает</span>
</td></tr>
<tr><td align="right" width="40%">&nbsp;</td><td>
<input type="submit" name="go" value="Изменить" class="txt_frm">
</td></tr>
</table>
</form><br>
<?php
    }
    else $err_enter .= 'Сайт не найден<br>';
    }
}
if ($err_enter != '') {
    echo '<div class="err">' . $err_enter . '</div><br>';
}

```

```
}  
// Закрываем соединение с базой данных  
mysql_close($db);  
f_footer_admin(); // Выводим нижний колонтитул  
?>
```

Администрирование гостевой книги

Алгоритм работы программы следующий.

- ☐ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение.
- ☐ Если отправлена форма для удаления сообщения (переменные \$_GET['del_msg'] и \$_GET['check'] будут существовать):
 - в цикле проверяем идентификатор сообщения, и если это число, то удаляем сообщение.
- ☐ Если отправлена форма для одобрения сообщения (переменные \$_GET['upd_msg'] и \$_GET['check'] будут существовать):
 - в цикле проверяем идентификатор сообщения, и если это число, то изменяем статус сообщения на "прочитано".
- ☐ Делаем запрос на выборку данных.
- ☐ Проверяем корректность номера страницы и рассчитываем начальную и конечную позицию.
- ☐ Выводим начало формы для удаления или одобрения сообщения.
- ☐ Выводим сообщения. Рядом с каждым сообщением добавляем поле для установки флажка.
- ☐ Если сообщение имеет статус "не прочитано", то перед сообщением выводим "NEW !!!"
- ☐ Если число сообщений больше заданного максимального количества сообщений на странице, то выводим номера страниц.
- ☐ Выводим конец формы для удаления или одобрения сообщения.
- ☐ Закрываем соединение с базой данных.

Исходный код программы приведен в листинге 45.

Листинг 45. Содержимое файла gbook.php

```
<?php  
header('Expires: Sun, 27 May 2007 01:00:00 GMT');  
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
```

```
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Pragma: no-cache');
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
// Заголовок
$title = 'Администрирование';
// Выводим верхний колонтитул
f_header_all($title, '', '', 2);
?>
<h1>Гостевая книга</h1><br>
<table align="center" width="600" bgcolor="#FFFFFF"
border="0" cellpadding="2" cellspacing="0">
<tr><td align="left" bgcolor="#FFFFFF">

<?php
if (isset($_GET['page'])) $page = $_GET['page'];
else $page = 1;
// Удаляем сообщение
if (isset($_GET['del_msg']) && isset($_GET['check']) &&
is_array($_GET['check'])) {
    foreach ($_GET['check'] as $value) {
        if (preg_match('/^[0-9]+$/', $value)) {
            $del = "DELETE FROM `gbook` WHERE `id_msg`=$value LIMIT 1";
            @mysql_query($del);
        }
    }
}
// Ставим галочку "Прочитано"
if (isset($_GET['upd_msg']) && isset($_GET['check']) &&
is_array($_GET['check'])) {
    foreach ($_GET['check'] as $value) {
        if (preg_match('/^[0-9]+$/', $value)) {
            $upd = "UPDATE `gbook` SET `msg_new`='y' ";
            $upd .= "WHERE `id_msg`=$value LIMIT 1";
            @mysql_query($upd);
        }
    }
}
```

```

}
// Выводим сообщения
$query_count = 'SELECT COUNT(`id_msg`) FROM `gbook`';
$res_count = @mysql_query($query_count) or die('Ошибка ' .
mysql_errno());
if (mysql_num_rows($res_count)===1) {
    $pole = mysql_fetch_row($res_count);
    $count = $pole[0];
    if (preg_match('/^[0-9]+/s', $count) && $count>0) {
        if (!isset($page) || $page<1) $page = 1;
        if (!preg_match('/^[0-9]+$/s', $page)) $page = 1;
        $count_page = ceil($count/COUNT_POS_PAGE_GBOOK);
        if ($page>$count_page) $page = $count_page;
        $pos_start = ($page-1)*COUNT_POS_PAGE_GBOOK;
        $query = 'SELECT `id_msg`, `author`, `msg`, msg_new, '
        $query .= "DATE_FORMAT(`msg_date`, '%d.%m.%Y %H:%i') AS `d` ";
        $query .= 'FROM `gbook` ORDER BY `id_msg` DESC '
        $query .= 'LIMIT ' . $pos_start . ', ' . COUNT_POS_PAGE_GBOOK;
        $res = @mysql_query($query) or die('Ошибка ' . mysql_errno());
        echo '<form action="gbook.php">';
        echo '<div align="center"><input type="hidden" name="page" value="'
        . $page . '">';
        echo '<input type="submit" name="upd_msg" value="Прочитано">';
        echo '<input type="submit" name="del_msg"
value="Удалить"></div><br>';
        while ($r = mysql_fetch_assoc($res)) {
            $r['author'] = htmlspecialchars($r['author']);
            $r['msg'] = htmlspecialchars($r['msg']);
            $r['msg'] = str_replace("\n", '<br>', $r['msg']);
            echo '<table width="100%" align="center" border="0"
cellspacing="0">';
            echo '<tr><td class="color_table">';
            echo '<input type="checkbox" name="check[]" value="';
            echo $r['id_msg'] . '"> <b>' . $r['d'] . ' ' . $r['author'];
            echo '</b></td></tr><tr><td>';
            if ($r['msg_new']=='n') {
                echo '<span class="err">NEW !!! </span> '
            }
            echo $r['msg'] . "</td></tr></table><br>\n\n";
        }
    }
}

```

```

        if ($count > COUNT_POS_PAGE_GBOOK) { // Выводим количество страниц
            f_table_page_start(); // Таблица для количества страниц Начало
            echo '<span class="bold">Страницы:</span> ';
            for ($j=1; $j<$count_page+1; $j++) {
                if ($j==$page) echo '[' . $j . " ] \n";
                else echo "<a href=\"?page=$j\">$j</a> \n";
            }
            f_table_page_end(); // Таблица для количества страниц Конец
        }
        echo '</form>';
    }
    else {
        echo '<div align="center" class="bold">Сообщений нет</div><br>';
    }
}
else echo '<div class="err">Ошибка</div><br>';
mysql_close($db);
?>
</td></tr>
</table>
<?php
f_footer_admin(); // Выводим нижний колонтитул
?>

```

Каждое новое сообщение помечено надписью "NEW !!!" Для того чтобы сбросить флаг, необходимо установить флажки напротив нужных сообщений и нажать кнопку **Прочитано**. Надписи исчезнут. Для удаления сообщения нужно установить флажок напротив сообщения и нажать **Удалить**.

Создание страницы статистики

Данная страничка будет первой страницей, которую увидит администратор при входе в Личный кабинет. Она должна содержать информацию о количестве сайтов и сообщений, требующих проверки, а также все сообщения об ошибках.

Алгоритм работы программы следующий.

- ☐ Подключаемся к базе данных. Если подключиться не удалось, выводим сообщение.
- ☐ Если отправлена форма для очистки файла err401.txt (переменная \$_GET['go401'] будет существовать), то записываем в файл пробел.

- ☐ Если отправлена форма для очистки файла err403.txt (переменная \$_GET['go403'] будет существовать), то записываем в файл пробел.
- ☐ Если отправлена форма для очистки файла err404.txt (переменная \$_GET['go404'] будет существовать), то записываем в файл пробел.
- ☐ Делаем запрос на общее количество зарегистрированных сайтов.
- ☐ Делаем запрос на количество сайтов с группировкой по статусу.
- ☐ Делаем запрос на общее количество сообщений в гостевой книге.
- ☐ Делаем запрос на количество непрочитанных сообщений в гостевой книге.
- ☐ Выводим статистику.
- ☐ Считываем содержимое файлов err404.txt, err401.txt и err403.txt в соответствующие поля для многострочного текста и выводим кнопки для очистки этих файлов.
- ☐ Закрываем соединение с базой данных.

Исходный код программы приведен в листинге 46.

Листинг 46. Содержимое файла index.php

```
<?php
header('Expires: Sun, 27 May 2007 01:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Pragma: no-cache');
define('IN_CATALOG', true);
require_once('allscript.php');
// Подключаемся к базе данных
$db = f_db_connect();
// Заголовок
$title = 'Администрирование';
// Выводим верхний колонтитул
f_header_all($title, '', '', 2);

if (isset($_GET['go401'])) { // Очищаем файл err401.txt
    file_put_contents('../err401.txt', ' ');
}
if (isset($_GET['go403'])) { // Очищаем файл err403.txt
    file_put_contents('../err403.txt', ' ');
}
if (isset($_GET['go404'])) { // Очищаем файл err404.txt
```

```

    file_put_contents('../err404.txt', ' ');
}
$mass['s'] = 0;
$mass['y'] = 0;
$mass['n'] = 0;
$count_site = $msg_new = $count_msg = 0;
$res = @mysql_query('SELECT COUNT(`id_site`) FROM `site`) or die('Ошибка ' . mysql_errno());
if (mysql_num_rows($res)===1) {
    $pole = mysql_fetch_row($res);
    $count_site = $pole[0];
}
$query = 'SELECT `status_site`, COUNT(`id_site`) ';
$query .= 'FROM `site` GROUP BY `status_site`';
$res = @mysql_query($query) or die('Ошибка ' . mysql_errno());
while ($pole = mysql_fetch_row($res)) {
    $mass[$pole[0]] = $pole[1];
}
$res = @mysql_query('SELECT COUNT(`id_msg`) FROM `gbook`) or die('Ошибка ' . mysql_errno());
if (mysql_num_rows($res)===1) {
    $pole = mysql_fetch_row($res);
    $count_msg = $pole[0];
}
}
}
$q = "SELECT COUNT(`id_msg`) FROM `gbook` WHERE `msg_new`='n'";
$res = @mysql_query($q) or die('Ошибка ' . mysql_errno());
if (mysql_num_rows($res)===1) {
    $pole = mysql_fetch_row($res);
    $msg_new = $pole[0];
}
}
?>
<h1>Статистика</h1><br>
<br><div align="center" class="bold">
Количество сайтов в каталоге - <?php echo $count_site; ?><br>
- активных - <?php echo $mass['y']; ?><br>
<a href="moder.php">- на модерации - <?php echo $mass['n']; ?></a><br>
- нерабочие - <?php echo $mass['s']; ?><br></div>
<br><div align="center" class="bold">
Количество сообщений в гостевой книге - <?php echo $count_msg; ?><br>

```

```
<a href="gbook.php">- не прочитано - <?php echo $msg_new; ?></a>
</div>
<br><div align="center" class="bold">Ошибка 404<br>
<textarea name="msg404" id="msg404" cols="15" rows="20"
class="textarea_frm"
style="width: 500px; height: 150px">
<?php
if (file_exists('../err404.txt')) {
    echo htmlspecialchars(file_get_contents('../err404.txt'));
}
?>
</textarea><br>
<form action="index.php"><input type="submit" name="go404"
value="Очистить">
</form></div>
<br><div align="center" class="bold">Ошибка 401<br>
<textarea name="msg401" id="msg401" cols="15" rows="20"
class="textarea_frm"
style="width: 500px; height: 150px">
<?php
if (file_exists('../err401.txt')) {
    echo htmlspecialchars(file_get_contents('../err401.txt'));
}
?>
</textarea><br>
<form action="index.php"><input type="submit" name="go401"
value="Очистить">
</form></div>
<br><div align="center" class="bold">Ошибка 403<br>
<textarea name="msg403" id="msg403" cols="15" rows="20"
class="textarea_frm"
style="width: 500px; height: 150px">
<?php
if (file_exists('../err403.txt')) {
    echo htmlspecialchars(file_get_contents('../err403.txt'));
}
?>
</textarea><br>
<form action="index.php"><input type="submit" name="go403"
value="Очистить">
```



```
</form></div>
<?php
// Закрываем соединение с базой данных
mysql_close($db);
f_footer_admin(); // Выводим нижний колонтитул
?>
```

Домашнее задание

Итак, изучение основ закончено, теперь пора дать вам задания, которые позволят закрепить изученный материал.

- ❑ На центральной странице учебного сайта необходимо вывести количество зарегистрированных сайтов в каждой рубрике после ее названия. Рубрики, в которых нет зарегистрированных сайтов, также должны быть выведены. Например:

```
Автомобили (55)
Безопасность (12)
Бытовая техника (0)
и т.д
```

- ❑ Если в рубрике зарегистрировано большое количество сайтов, то на странице catalog.php будет очень много ссылок на другие страницы. Чтобы этого избежать разработайте функцию, которая будет выводить только указанное количество ссылок на другие страницы. Формат функции:

```
f_pager(<Номер текущей страницы>, <Всего страниц>, <Радиус>,
<URL-адрес>)
```

По умолчанию параметр <Радиус> должен иметь значение 2 (две ссылки справа и слева от текущей страницы), а параметр <URL-адрес> пустую строку. Внутри функции к указанному URL-адресу следует добавить параметр page со значением номера страницы:

```
<a href="test.php?page=20">20</a>
```

Вывод функции при таком цикле

```
for ($i = 1; $i < 51; $i++) {
    f_pager($i, 50, 5, '');
}
```

должен быть примерно следующим:

```
[1] 2 3 4 5 6...
1 [2] 3 4 5 6 7...
1 2 [3] 4 5 6 7 8 ...
1 2 3 [4] 5 6 7 8 9 ...
1 2 3 4 [5] 6 7 8 9 10 ...
```

```

1 2 3 4 5 [6] 7 8 9 10 11 ...
... 2 3 4 5 6 [7] 8 9 10 11 12 ...
... 3 4 5 6 7 [8] 9 10 11 12 13 ...
... 4 5 6 7 8 [9] 10 11 12 13 14 ...
... 5 6 7 8 9 [10] 11 12 13 14 15 ...

... 12 13 14 15 16 [17] 18 19 20 21 22 ...
... 19 20 21 22 23 [24] 25 26 27 28 29 ...
... 25 26 27 28 29 [30] 31 32 33 34 35 ...
... 36 37 38 39 40 [41] 42 43 44 45 46 ...

... 35 36 37 38 39 [40] 41 42 43 44 45 ...
... 36 37 38 39 40 [41] 42 43 44 45 46 ...
... 37 38 39 40 41 [42] 43 44 45 46 47 ...
... 38 39 40 41 42 [43] 44 45 46 47 48 ...
... 39 40 41 42 43 [44] 45 46 47 48 49 ...
... 40 41 42 43 44 [45] 46 47 48 49 50
... 41 42 43 44 45 [46] 47 48 49 50
... 42 43 44 45 46 [47] 48 49 50
... 43 44 45 46 47 [48] 49 50
... 44 45 46 47 48 [49] 50
... 45 46 47 48 49 [50]

```

Номера - это ссылки на соответствующие страницы. Номер внутри квадратных скобок - это текущая страница. Используйте эту функцию для вывода сайтов в рубрике, а также для вывода навигации в гостевой книге.

- ❑ С помощью JavaScript сделайте возможность предварительного просмотра добавляемого сообщения в Гостевой книге. Рядом с кнопкой **Добавить сообщение** разместите кнопку **Предварительный просмотр**. Щелчок на этой кнопке должен запускать какую-либо пользовательскую JavaScript-функцию, внутри которой введенные данные преобразуются и выводятся перед формой (или сразу после нее). Данные формы при этом отправляться на сервер не должны.
- ❑ Добавьте возможность администратору отвечать на сообщения, а также редактировать сообщения, добавляемые в Гостевую книгу.
- ❑ Такие формы как в Гостевой книге очень любят спам-роботы. Если не предусмотреть возможность защиты, то скоро Гостевая книга будет завалена спамом. Чистить ее каждый день очень утомительно. Добавьте картинку с уникальным кодом, который пользователь должен ввести в дополнительное поле.

Картинку формируйте с помощью библиотеки GD. Код для каждого пользователя должен быть разным.

- ❑ Мы уже говорили, что письма, отправленные через форму обратной связи, иногда не доходят до адресата из-за различных спам-фильтров. Сделайте возможность сохранения сообщения в базе данных. На E-mail должно отправляться только уведомление о новом сообщении. Внутри Админки добавьте возможность просматривать эти сообщения, а также удалять их.
- ❑ Если у посетителя отключены cookies, то он не сможет войти в Личный кабинет. Сделайте так, чтобы он мог это сделать, не включая cookies.
- ❑ При запросе восстановления пароля мы сразу генерируем новый пароль. Сделайте так, чтобы на E-mail отправлялась ссылка для подтверждения смены пароля с уникальным идентификатором. При переходе по этой ссылке генерируем новый пароль и сохраняем его в базе.
- ❑ В Личном кабинете выведите список сайтов, зарегистрированных пользователем с разбивкой на страницы. Добавьте возможность редактирования сайта пользователем. Причем пользователь может редактировать только свои сайты и не иметь доступа к чужим.
- ❑ В последнее время становятся популярными домены с русскими названиями. В 2010 году открывается зона с названием рф, например, сайт может иметь название <http://мойсайт.рф>. Измените регулярные выражения для проверки URL-адреса и E-mail адреса таким образом, чтобы предусмотреть возможность использования таких доменов.
- ❑ При переходе по ссылке для проверки работоспособности добавленного сайта браузер автоматически вставляет адрес ссылающейся страницы. Таким образом, любой владелец сайта может узнать местоположение Админки из отчетов статистики или логов сервера. Сделайте так, чтобы передавался адрес страницы, расположенной вне Админки.
- ❑ Ну и последнее задание. Очень часто добавленные сайты "умирают" через месяц другой. Напишите программу, которая проверяет работоспособность сайтов, путем получения исходного кода удаленной страницы или только заголовков ответа сервера.