Бессонов Л. В.

ПРАКТИКУМ ПО ВЕБ-ПРОГРАММИРОВАНИЮ

Упражнения и практические задания

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Бессонов Л. В.

Практикум по веб-программированию

Упражнения и практические задания

Учебное-методическое пособие для студентов, обучающихся по направлению подготовки бакалавриата 09.03.03 «Прикладная информатика»

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Саратов
ООО Издательский Центр «Наука»
2016

УДК 004.432(076.5)
ББК 32.973.018я73
Б53

Бессонов Л.В.

Б53 **Практикум по веб-программированию. Упражнения и практические задания:** Учебно-методическое пособие для студентов, обучающихся по направлению подготовки бакалавриата 09.03.03 «Прикладная информатика» /Л.В. Бессонов. — Саратов: ООО Издательский Центр «Наука», 2016. — 40 с.

ISBN 978-5-9999-2716-3

Данное учебно-методическое пособие представляет собой методическую разработку для обучения основам веб-программирования на языке PHP. Пособие состоит из упражнений и практических работ. Упражнения направлены на быстрое освоение языка PHP на основе базовых знаний программирования на любом С-подобном языке. Практические работы содержат задания и методические рекомендации по их выполнению. Материалы практической работы №2 по вариантам могут быть использованы в качестве курсового задания.

Для студентов, обучающихся по направлению подготовки бакалавриата 09.03.03 «Прикладная информатика», также может быть полезно для начинающих веб-разработчиков.

Рецензенты:

Кафедра математического обеспечения вычислительных комплексов и информационных систем Саратовского национального исследовательского государственного университета имени Н.Г. Чернышевского

Зав. кафедрой, д.ф.-м.н., профессор ***Д.А. Андрейченко***

УДК 004.432(076.5)
ББК 32.973.018я73

Работа издана в авторской редакции

ISBN 978-5-9999-2716-3

© Л.В. Бессонов, 2016

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

Содержание

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Методические рекомендации

Структура практикума

Практикум состоит из набора упражнений и трех практических работ. Упражнения предназначены для адаптации имеющихся у студента знаний базового курса информатики и навыков программирования на любом Си-подобном языке к синтаксису языка PHP.

Первая практическая работа посвящена написанию простейшей CMS. Задача рафинирована от всех возможных сервисных функций CMS и позволяет на практике прочувствовать базовую составляющую любой CMS.

Вторая практическая работа подразумевает написание модуля сайта. Задание состоит из базовой части и развития. В процессе выполнения базовой части задания студент должен освоить базовые навыки написания веб-приложений. Работа имеет несколько вариантов заданий, при этом может выполняться как индивидуально, так группами из 2-4 студентов.

Третья практическая работа существенно проста по сути, но имеет нестандартную постановку задачи для студентов, имеющих лишь знание базового курса информатики и программирования. Предлагаемая для освоения технология является достаточно простым примером построения систем с встраиваемыми модулями, написанными сторонними программистами.

Условие получения допуска по практике

Чтобы практика была засчитана выполненной, студент должен выполнить первую практическую работу, базовую часть второй практической работы и третью практическую работу.

Выполнение развития задачи во второй практической работе является основанием для рекомендации на автоматический зачёт.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

Рекомендуется следующее распределение рабочего времени

Вид заданий	*Количество часов на освоение*
Упражнения	8 часов (4 пары)
Практическая работа №1	4 часа (2 пары)
Практическая работа №2	18 часов (9 пар)
Практическая работа №3	2 часа (1 пара)

Рекомендуемые параметры системы для выполнения практикума

Веб-сервер Apache. Предпочтительнее всего Apache 2.x, развёрнутый на базе ОС семейства Unix/Linux. Apache должен иметь в составе (и включенными) модули mod_rewrite и модуль интерпретации PHP. Также для корневой папки продукта должно быть разрешено переопределение параметров настройки инструкцией AllowOverride All. Для детальной настройки других веб-серверов рекомендуется связаться с разработчиками продукта.

Интерпретатор PHP 5 (желательно PHP 5.4 или выше). Рекомендуемые параметры настройки интерпретатора:

session.cache_limiter = nocache

session.auto_start = 0

expose_php = off

allow_url_fopen = off

magic_quotes_gpc = off

register_globals = off

display_errors = off

max_execution_time = 30

Объём выделяемой интерпретатору PHP памяти не должен быть меньше 32 Мб.

Для интерпретатора PHP должны быть установлены следующие расширения:

1. mysqli
2. xml

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

3. GD2 library
4. JSON
5. FileInfo
6. APC
7. PDO (PDO mysql)
8. hash
9. SimpleXML
10. DOM
11. date
12. tokenizer

СУБД MySQL 5 (MySQL 5.0.15 и выше, с поддержкой PDO). Учётной записи, которой будет пользоваться продукт должны быть предоставлены права на операции:

1. SELECT
2. INSERT
3. UPDATE
4. DELETE
5. CREATE
6. DROP
7. INDEX
8. ALTER

Также рекомендуется поставить параметру max_allowed_packet значение не менее 16Мб.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Упражнение 1. Простейшие скрипты без передачи параметров

Задача 1.

Базовая часть

Создайте программу, выводящую числа от 1 до 100.

Развитие

1. Выводите числа в таблице по возрастанию слева направо, по 10 чисел в строке. Например:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
...									

2. Введите управляющую переменную $m, 10<$m<100, значение которой будет задаваться в начале программы. Программа должна выводить такую же таблицу чисел от 1 до $m.
3. Введите управляющую переменную $p, 1<$p<20, значение которой будет задаваться в начале программы. Программа должна выводить ту же таблицу, по \$p чисел в строке. Для последней ячейки, при необходимости, используйте colspan.
4. Введите управляющую переменную $c, 1<$c<10, значение которой задаётся в начале программы. Программа должна выводить ту же таблицу, подкрашивая все числа кратные $c красным цветом.

Указания

При выводе ячейки таблицы тегом td, при указании значения атрибута colspan, происходит объединение колонок.

Например:

```
1.     <table border="1" cellpadding="2" cellspacing="2">
2.       <tr>
3.         <td>1</td>
```

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

```
4.         <td>2</td>
5.         <td>3</td>
6.       </tr>
7.       <tr>
8.         <td colspan="2">4</td>
9.         <td><span style="color: red;">5</span></td>
10.      </tr>
11.      <tr>
12.        <td>6</td>
13.        <td colspan="2">7</td>
14.      </tr>
15.    </table>
```

Создаст таблицу:

1	2	3
4		**5**
6	7	

Задание 2.

Базовая часть

Создайте программу, выводящую таблицу умножения. Выводится таблица 10x10, в каждой ячейке которой записано произведение номера столбца на номер строки.

Развитие

1. Введите две управляющие переменные $n_max, $m_max, значения которых задаются в начале программы. Таблица, выводимая программой, должна быть размерности n_max*m_max
2. Введите две управляющие переменные $n_min, $m_min, значения которых задаются в начале программы. Программа должна выводить строки таблицы умножения от $n_min до $n_max, и столбцы от $m_min до $m_max
3. Создайте двумерный ассоциативный массив $p. В выводимой программой таблице умножения произведение стоящее на месте ($x,$y) должно возводиться в степень $p[$x][$y]

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Подсказка "Вывод таблиц в одном цикле"

То с чего Вы начали при выполнении задания 1(1) выглядит примерно так:

```
1.     <?php
2.           print "<table border='1'>";
3.           print "<tr>";
4.           for ($i=1; $i<=100; $i++){
5.                 print "<td>$i</td>";
6.           }
7.           print "</tr>";
8.           print "</table>";
9.     ?>
```

При этом получили таблицу из одной строки, содержащей 100 ячеек. Так произошло, потому что, открыв строку тегом tr Вы в цикле выводили 100 ячеек.

Модифицируем скрипт следующим образом:

```
1.     <?php
2.           print "<table border='1'>";
3.           for ($i=1; $i<=100; $i++){
4.                 print "<tr>";
5.                 print "<td>$i</td>";
6.                 print "</tr>";
7.           }
8.           print "</table>";
9.     ?>
```

Теперь получилась таблица с одним столбцом, состоящим из 100 строк, потому что на каждом витке цикла открывается строка тегом tr, выводится ячейка с числом и затем строка закрывается закрывающим тегом tr.

Теперь наша задача — сделать так, чтобы открытие и закрытие строки таблицы происходило не на каждом шаге цикла, а лишь на некоторых (строка должна открываться на шагах 1, 11, 21, ...; и закрываться на шагах 10, 20, 30, ...). Для этого в строках 4 и 6 последнего листинга можно ввести условие, то есть чтобы вывод открывающих и закрывающих тегов tr происходил лишь при выполнении этих условий.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

```
1.    <?php
2.          print "<table border='1'>";
3.          for ($i=1; $i<=100; $i++){
4.                if (<некое условие>){
5.                      print "<tr>";
6.                }
7.                print "<td>$i</td>";
8.                if (<некое другое условие>){
9.                      print "</tr>";
10.               }
11.         }
12.         print "</table>";
13.   ?>
```

Чтобы построить условие, обратитесь к теме «Операции» методического пособия. Обратите внимание на то, что в php как и в C оператор сравнения на равенство имеет вид '=='!

Вы также заметите что Ваш скрипт выводит html-текст одной строкой (исходный текст страницы). Чтобы исходный html-текст стал более читаемым, используйте специальные символы перевода строки и табуляции. Например, так:

```
1.    <?php
2.          print "<table border='1'>\n";
3.          for ($i=1; $i<=100; $i++){
4.                if (<некое условие>){
5.                      print "\t<tr>\n";
6.                }
7.                print "\t\t<td>$i</td>\n";
8.                if (<некое другое условие>){
9.                      print "\t</tr>\n";
10.               }
11.         }
12.         print "</table>\n";
13.   ?>
```

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

Упражнение 2. Простейшие скрипты с передачей параметров

Задание 1. Калькулятор

Базовая часть

Напишите скрипт, в параметрах которому будет передаваться два действительных числа $a, $b и признак операции $op, принимающий значения: add, sub, mul, div. В результате скрипт должен выводить результат операции. Например, если передано a=20&b=10&op=div, выведется: '20 / 10 = 2'.

Развитие

1. Защитите свой скрипт от передачи некорректных параметров. Контролируйте ситуацию передачи чисел, правильность передачи признака операции и ошибку деления на 0.
2. Перепишите скрипт таким образом, чтобы ввод значений осуществлялся через html-форму ввода. Пример работы такого скрипта см. по адресу http://nto.immpu.sgu.ru/examples/calc.php

Задание 2. Ссылки

Базовая часть

Напишите скрипт выводящий ссылку с текстом '0'. При нажатии на ссылку происходит перезагрузка страницы, на вновь открывшейся странице отображается ссылка с текстом '1'. В общем случае, если открыта страница с текстом 'n', то при переходе по ссылке открывается страница с текстом 'm', где m=n+1.

Пример работы такого скрипта см. по адресу:
http://nto.immpu.sgu.ru/examples/link.php

Развитие

1. Защитите скрипт от передачи некоррекектных параметров. Параметр должен быть натуральным числом.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

2. Модифицируйте скрипт, чтобы выводился набор чисел от 1 до 10 и ссылка с текстом '>>', при нажатии на которую страница перезагружается и скрипт отображает ссылку с текстом '<<', числа от 11 до 20 и ссылку с текстом '>>'. При нажатии '<<' происходит переход на числа от 1 до 10, при нажатии '>>' переход на числа от 21 до 30, и так далее. (Пример см. по адресу http://nto.immpu.sgu.ru/examples/links.php)
3. Модифицируйте скрипт, чтобы каждое выводимое число было ссылкой (страница), причём открытая в настоящей момент страница ссылкой не являлась бы. При переходе по ссылка выводятся сообщения 'Это страница №n'. (Пример см. по адресу http://nto.immpu.sgu.ru/examples/pages.php)

Упражнение 3. Взаимодействие с СУБД MySQL

Для выполнения упражнений используйте *phpMyAdmin*. Имена таблиц начинайте с со своего номера группы и фамилии, например: 123_ivanov_tablename.

Задача 1. Отдел продаж

Создайте таблицу "Содержимое счёта" (назвать можно, например, "bill_content") со следующей структурой:

- id (int, auto_increment) — суррогатный ключ
- goods (varchar(255)) — товары
- price (float) — цена
- quantity (float) — количество

Заполните эту таблицу записями (не менее 10 записей).

Напишите программу выводящую таблицу с заголовком: товар, цена, количество, стоимость. Расчёт стоимости осуществляйте в запросе.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Развитие

1. Сделайте вывод итоговых сумм по закупленным товарам. Реализовать суммирование в запросе
2. Создайте дополнительную таблицу "Счета" ("bill"), содержащую: bid (int, auto_increment) — номер счёта, bdate (date) — дата выставления счёта, name (varchar(255)) — наименование организации-заказчика. Заполните эту таблицу тремя записями: два счёта от одной организации, один от другой. В таблицу "Содержимое счёта" добавьте колонку bid, которая для этой таблицы будет внешним ключом. Заполните эту колонку так, чтобы в каждом счёте было не менее 2 строк.

Сделайте вывод списка счетов с суммами (полная сумма счёта). Например, так:

№ счёта	Дата	Организация	Сумма
1	01.03.2010	ООО "Ромашка"	10000 рублей
2	03.03.2010	ЗАО "Газпром"	13 рублей
3	17.03.2010	ООО "Ромашка"	25000 рублей

3. Создайте таблицу "Оплаты по счетам" ("payment"): id (int, auto_increment), bid (int) — номер счёта, pdate (date) — дата оплаты, summa (float) — сумма оплаты. Заполните эту таблицу так, чтобы один счёт был полностью оплачен (несколькими частичными платежами), второй был частично оплачен (несколькими платежами), третий полностью не оплачен (ни одного платежа).

Выведите таблицу задолженностей по счетам.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Практическая работа №1. Разработка простейшего веб-приложения

Цель работы

Разработать простейшее веб-приложение, реализующее CMS (Content Management System).

Описание

Современные веб-разработчики всё реже обращаются к созданию статичных страниц. Веб-документ выполненный статичной вёрсткой — «штучный» товар. Всё чаще применяются скрипты, частично, либо полностью автоматизирующие процесс создания страниц (например, по шаблонам), а также автоматизирующие процесс обслуживания сайта.

Задание

Разработайте веб-приложение, отображающее страницы, хранящиеся в таблице базы данных. Структура таблицы:

1. page_id (int, auto_increment) — идентификатор страницы
2. page_name (varchar) — название страницы
3. page_text (text) — содержимое страницы
4. page_weight (int, значение по умолчанию 1) — «вес» страницы в меню сайта

Файлу приложения задайте имя «index.php».

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

При обращении приложение отображает страницу.

Заголовок и подвал — статичные части. Меню генерируется по запросу к таблице, причём пункты меню сортируются по весу. В части «Контент» отображается содержание страницы, идентификатор которой передаётся в строке запроса приложению, например: «index.php?page=15». Если не передан идентификатор, то приложение должно отобразить первую найденную в таблице страницу с весом 0.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Практическая работа №2. Разработка модуля сайта

Цель работы

Целью практической работы является разработка и написание модуля web-сайта, согласно индивидуальному заданию.

Теоретические материалы

Модулем web-сайта будем называть программу, хранящуюся и выполняющуюся на сервере, имеющую один или несколько вариантов представления хранимой на сервере информации и интерфейс управления этой информацией.

Общие требования к заданиям

Написанная по индивидуальному заданию система должна быть защищена от некорректной передачи параметров.

Для работы система должна подключать файл config.php, содержащий настройки для подключения к базе данных и параметр, управляющий детализацией протоколов работы.

Система должна вести протокол работы в файл <дата>.log со следующими возможными уровнями детализации (каждый больший номер включает в себя сообщения уровней с меньшими номерами):

1. Ошибки: ошибки при соединении с базой данных, неверный ввод пароля при доступе к административному интерфейсу.
2. Предупреждения: передача некорректных параметров.
3. Сообщения: доступ к административному интерфейсу, добавление/изменение/удаление информации в базу данных.
4. Отладка: все обращения к скрипту.

То есть, к примеру, отладка уровня 3 будет включать в себя ошибки, предупреждения и сообщения. А отладка уровня 2 — ошибки и предупреждения.

Записи протокола должны содержать дату-время события, наименование события (ошибка, предупреждение, сообщение, отладка), расшифровку

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

(например, какого типа ошибка или какая страница запрошена), логин пользователя (например, при попытке доступа и прочих манипуляциях с авторизацией).

Варианты индивидуальных заданий

Один вариант выполняется одним или *двумя* студентами. Варианты, помеченные знаком «звёздочка» выполняется *четырьмя* студентами.

Вариант 1

Базовая часть

Модуль "Система новостей". Информационная единица — новость — имеет структуру:

- Идентификатор новости;
- Дата-время размещения новости;
- Заголовок (текст не более 50 символов);
- Автор (выбор из списка авторов)
- Рубрика (выбор из списка рубрик)
- Анонс (краткий текст не более 250 символов);
- Текст новости (текст не более 65535 символов).

Модуль имеет представления:

1. Вывод блоков {Дата-время, Заголовок, Анонс} в ленту новостей, отсортированную в обратном хронологическом порядке и разбиваемую на страницы (по 10 новостей на страницу).
2. Вывод новостей по дате.
3. Вывод новостей по рубрике.
4. Вывод отдельной новости по идентификатору.

Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять отдельные новости.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

Расширение задачи

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные пользователей хранится в таблице:

- Идентификатор;
- Логин;
- Хеш пароля;
- Наличие прав суперпользователя;
- ФИО;
- Время последнего доступа к системе.

Вместо выбора из списка авторов для новостей реализуйте связь с таблицей пользователей. Система должна указывать автора автоматически, при добавлении новости аутентифицированным пользователем.

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять и удалять пользователей системы, а также изменять различные атрибуты существующих пользователей.

Вариант 2

Базовая часть

Модуль "Учёт пользователей". Информационная единица — пользователь — имеет структуру:

- Логин (текст не более 20 символов);
- Фамилия (текст не более 50 символов);
- Имя (текст не более 50 символов);
- Отчество (текст не более 50 символов);
- Должность (выбор из списка должностей);
- Дата-время регистрации;

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Модуль имеет представления:

1. Вывод списка пользователей с возможностью сортировки по колонкам, и разбиваемую на страницы (по 10 пользователей на страницу).
2. Вывод списка пользователей по заданной должности.
3. Вывод списка пользователей, зарегистрированных в заданном месяце.
4. Вывод карточки пользователя по заданному логину.

Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять записи о пользователях.

Расширение задачи

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные аутентификации хранится в отдельной таблице, связанной с таблицей пользователей и имеющей структуру:

- Логин;
- Хеш пароля;
- Наличие прав суперпользователя;
- Время последнего доступа к системе.

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет назначать пользователям права входа в интерфейс администрирования и права суперпользователя.

Вариант 3

Базовая часть

Модуль "Библиотека". Информационная единица — книга — имеет структуру:

- Идентификатор
- Название книги (текст не более 100 символов);

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

- Автор(ы) (текст не более 250 символов);
- Год издания;
- Аннотация (текст не более 65535 символов);
- Тематика (выбор из списка тематик);
- Выдана (NULL если книга не выдавалась, 0 если выдана);
- Имя графического файла, содержащего фото обложки.

Модуль имеет представления:

1. Вывод списка книг библиотеки с возможностью сортировки по колонкам. Список разбивается на страницы (по 10 книг на страницу).
2. Список доступных (не выданных) книг.
3. Вывод карточки книги.
4. Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять записи о книгах.

Расширение задачи

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные аутентификации хранится в отдельной таблице, имеющей структуру:

- Идентификатор;
- Логин;
- ФИО пользователя;
- Хеш пароля;
- Наличие прав суперпользователя;
- Время последнего доступа к системе.

В таблице записей о книгах модифицируйте атрибут "Выдана", чтобы он имел значение NULL, если книга никому не выдана в настоящий момент, и значение равное идентификатору пользователя, если книга выдана этому

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

пользователю. Администраторский интерфейс позволяет указывать факт выдачи книги пользователю и возврата книги.

Добавьте представление "Задолжники".

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять, изменять и удалять записи о пользователях, назначать права суперпользователя.

Вариант 4

Базовая часть

Модуль "Доска объявления". Информационная единица — объявление — имеет структуру:

- Идентификатор объявления;
- Дата-время размещения объявления;
- Заголовок (текст не более 100 символов);
- Автор (текст не более 50 символов);
- Email автора (текст не более 50 символов);
- Телефон (текст не более 20 символов);
- Текст объявления (текст не более 65535 символов).

Модуль имеет представления:

1. Вывод списка объявлений, отсортированный в обратном хронологическом порядке (по 20 объявлений на страницу).
2. Вывод объявления по идентификатору.
3. Вывод списка {Автор, Email, Телефон}.
4. Вывод объявлений по дате.

Модуль имеет открытый интерфейс, позволяющий добавлять, объявления. Также модуль имеет интерфейс администратора, позволяющий удалять объявления.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Расширение задачи

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные пользователей хранится в таблице:

- Идентификатор;
- Логин;
- Хеш пароля;
- ФИО;
- Время последнего доступа к системе.

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять и удалять пользователей системы, а также изменять различные атрибуты существующих пользователей.

Вариант 5

Базовая часть

Модуль "Геоинформационная база". Информационная единица — географический объект — имеет структуру:

- Идентификатор объекта;
- Координата X (вещественное число);
- Координата Y (вещественное число);
- Название (текст не более 50 символов);
- Описание (текст не более 65535 символов).
- Координаты будем считать декартовыми.

Модуль имеет представления:

1. Вывод списка географических объектов (по 10 объектов на страницу).
2. Вывод карточки объекта.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

3. Вывод списка объектов, удалённых друг от друга не дальше чем на указанное расстояние.
4. Вывод карточки объекта, ближайшего к указанной точке.

Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять отдельные объекты.

Расширение задачи

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные пользователей хранится в таблице:

- Идентификатор;
- Логин;
- Хеш пароля;
- Наличие прав суперпользователя;
- ФИО;
- Время последнего доступа к системе.

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять и удалять пользователей системы, а также изменять различные атрибуты существующих пользователей.

Вариант 6

Базовая часть

Модуль "Телефонный справочник организации". Информационная единица — телефонный номер — имеет структуру:

- Идентификатор;
- Сотрудник, ответственный за телефон (выбор из списка сотрудников);

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

- Отдел, которому принадлежит телефон (выбор из списка сотрудников);
- Телефонный номер (текст не более 20 символов);
- Описание (текст не более 65535 символов).

Модуль имеет представления:

1. Вывод списка всех телефонных номеров, сгруппированных по отделам (по 15 записей на страницу).
2. Вывод карточки телефона (полное описание).
3. Вывод списка телефонных номеров указанного отдела.
4. Вывод списка телефонных номеров указанного ответственного сотрудника.

Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять отдельные записи.

Расширение задачи

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные пользователей хранится в таблице:

- Идентификатор;
- Логин;
- Хеш пароля;
- Наличие прав суперпользователя;
- ФИО;
- Время последнего доступа к системе.

Вместо использования перечисления сотрудников, используйте связь с таблицей пользователей.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

Пользователь имеет право изменять или удалять запись о телефоне, если он является ответственным за эту запись.

При добавлении записи, ответственным сотрудником автоматически назначается создающий запись пользователь.

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять и удалять пользователей системы, а также изменять различные атрибуты существующих пользователей. Кроме того, суперпользователь имеет право создания, изменения и удаления любых записей о телефонных номерах.

*Вариант 7**

Базовая часть

Модуль "Распределение заданий". Информационная единица — задание — имеет структуру:

- Идентификатор задания;
- Дата-время добавления задания;
- Сотрудник, которому назначено задание (выбор из списка сотрудников)
- Планируемая дата завершения;
- Фактическая дата завершения;
- Заголовок (текст не более 50 символов);
- Текст задания(текст не более 65535 символов).

Сотрудники не могут сами отмечать задания как выполненные.

Модуль имеет представления:

1. Вывод заданий по сотруднику, сортируемый в обратном хронологическом порядке.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

2. Вывод статистики производительности сотрудников (количество заданий в месяц, при этом если задание растянуто на несколько месяцев, считать производительность пропорционально дням).
3. Вывод диаграммы Ганта занятости сотрудников в проектах (по завершённым проектам) на указанный интервал дат.

Для изображения диаграммы Ганта используйте таблицу с закрашенными ячейками. По горизонтальной оси диаграммы отложены дни. По вертикальной оси расположены группы {сотрудник → {проекты}}.

Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять отдельные задания.

Расширение задачи

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные пользователей хранится в таблице:

- Идентификатор;
- Логин;
- Хеш пароля;
- Наличие прав суперпользователя;
- ФИО;
- Время последнего доступа к системе.

Вместо перечисления сотрудников в базовой задаче используйте связь с таблицей пользователей.

Каждый пользователь имеет право пометить приписанную ему задачу как завершённую, при этом моментом завершения задачи считается момент выставления признака "Завершена". Добавлять, изменять и удалять задачи имеет право только суперпользователь.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять и удалять пользователей системы, а также изменять различные атрибуты существующих пользователей.

*Вариант 8**

Базовая часть

Модуль "Риелтор". Информационная единица — запись о квартире — имеет структуру:

- Идентификатор;
- Этажность дома;
- Этаж, на котором расположена квартира;
- Количество комнат;
- Цена в рублях;
- Строительный материал: кирпич, панель, дерево (выбор из списка материалов);
- Район города (выбор из списка районов);
- Имя риелтора (текст не более 50 символов);
- Email риелтора (текст не более 100 символов);
- Описание (текст не более 65535 символов).

Модуль имеет представления:

1. Вывод таблицы всех квартир с возможностью сортировки по колонкам (по 20 записей на страницу).
2. Вывод карточки квартиры (полное описание).
3. Вывод списка квартир, удовлетворяющих критерию отбора {Этаж, Этажность, Материал, Количество комнат, Минимальная цена, Максимальная цена, Район}. Часть параметров отбора могут быть не

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

определены. При отборе по цене должен выполняться критерий: Минимальная цена ≤ Цена ≤ Максимальная цена.

4. Вывод статистики по количеству предложений квартир по риелторам.

Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять отдельные записи.

Расширение задачи

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные пользователей хранится в таблице:

- Идентификатор;
- Логин;
- Хеш пароля;
- Наличие прав суперпользователя;
- ФИО;
- Email;
- Телефон;
- Время последнего доступа к системе.

Вместо использования текстовой строки «Имя риелтора», используйте связь с таблицей пользователей. При этом атрибут таблицы квартир «Email» становится не нужным. Переделайте все имеющиеся представления с учётом подбора ФИО и контактных данных риелтора из таблицы пользователей.

Пользователь имеет право изменять или удалять запись о квартире, если он является риелтором, выставляющим квартиру на продажу.

При добавлении записи, риелтором автоматически назначается создающий запись пользователь.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять и удалять пользователей системы, а также изменять различные атрибуты существующих пользователей. Кроме того, суперпользователь имеет право создания, изменения и удаления любых записей о телефонных номерах.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Практическая работа №3. Разработка веб-приложения с динамически подключаемыми модулями

Цель работы

Целью практической работы является освоение технологии программирования веб-приложения, подключающего и использующего внешние программы, если те написаны по заранее определённым правилам.

Теоретические материалы

1. Понятие обратного вызова
2. Пример написания системы подключающей плагины

Варианты индивидуальных заданий

Во всех заданиях подразумевается что файлы плагинов имеют имена, удовлетворяющие маске «*.plugin.php».

Во всех заданиях необходимо реализовать веб-приложение, подключающие плагины указанной структуры.

Встройте разработанную по индивидуальным заданиям систему в приложение, написанное при выполнении практической работы №1. Функции плагинов должны обрабатывать информацию, выводимую в поле «Контент».

Вариант 1

Система «Автозамена». Плагины содержат функции (одна функция в каждом плагине) «<имя модуля>_autochange($text)», получающие в аргументе текст и осуществляющие некоторую текстовую замену.

Сделайте плагины замены:

1. «о» на «а»;
2. «е» на «и»;
3. Текстовые смайлики на соответствующие изображения.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

Вариант 2

Система «Аналитик». Плагины содержат функции (две функция в каждом плагине):

1. '<имя модуля>_info()' — возвращает название анализируемого показателя;
2. '<имя модуля>_analysis($text)' — возвращает число-значение анализируемого показателя.

Сделайте плагины анализа:

1. Количество гласных букв в тексте;
2. Количество согласных букв в тексте;
3. Частоты употребления букв (возвращается гистрограмма с отложенными по оси абсцисс буквами, а по оси ординат частотами в процентах). Гистограмму реализовать таблицей с закрашиваемыми ячейками.

Вариант 3

Система «Подсветка». Плагины содержат функции (две функция в каждом плагине):

1. '<имя модуля>_info()' — возвращает название подсвечиваемых значений;
2. '<имя модуля>_analysis($text)' — возвращает вместо текст с подсвеченными частями.

Сделайте плагины подсветки:

1. Слова, заканчивающиеся на «ться» подкрашивать красным;
2. Слова, состоящие из одной и двух букв подсвечивать серым;
3. Знаки препинания под подсвечивать красным

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Пример реализации

Задача

Приложение «Линейная алгебра» должно вычислять нормы двумерных векторов (каждый элемент вектора — число). Правила вычисления нормы заведомо неизвестны.

Приложение находит и подключает файлы по маске *.plugin.php

Каждый файл плагина содержит две функции:

1. <имя плагина>_info() — возвращает строку с названием соответствующей нормы
2. <имя плагина>_norm($v=array()) — вычисляет норму вектора, переданного в аргументе

Приложение должно вычислять все «известные» ему варианты норм, выводя пары (Название нормы, Значение нормы).

Ход решения

Напишем саму систему, подключающую плагины. Алгоритм действия системы:

1. Прочитать все элементы каталога
2. Выделить среди них файлы, имя которых заканчивается на '.plugin.php'
3. Подключить эти файлы
4. Вызвать функции из плагинов для расчёта заданного вектора

Файл *algebra.php*

```
1.    <?php
2.    $dir = opendir("./" );
3.    $plugins = array();
4.    while($any_file = readdir($dir)){
5.       if (is_file($any_file)){
6.          if(strpos($any_file, '.plugin.php')!==FALSE){
7.           include("./".$any_file);
```

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

```
8.            $plugins[]=substr($any_file,0,strpos($any_file,
'.plugin.php'));
9.            }
10.       }
11.    }
12.    closedir($dir);
13.
14.    $vector = array(3, 4);
15.    print "<p>Вычисляем норму вектора</p><p>";
16.    print_r($vector);
17.    print "</p>";
18.    foreach ($plugins as $p){
19.       $info_func = $p.'_info';
20.       $calc_func = $p.'_norm';
21.       if (!function_exists($info_func) ||
!function_exists($calc_func)){
22.          print "<p>Плагин $p содержит ошибку</p>";
23.       }else{
24.          print "<p>";
25.          print call_user_func($info_func)." равна ";
26.          print call_user_func($calc_func, $vector);
27.          print "</p>";
28.       }
29.    }
30.    ?>
```

Построчный разбор:

1. Открылась инструкция php
2. Открыли текущую папку и сохранили указатель на неё в переменную $dir
3. Проинициализировали массив, в который будем складывать имена найденных плагинов
4. Начали цикл обхода всех элементов внутри каталога
5. Если рассматриваемый элемент является файлом, то
6. Если он к тому же ещё в имени имеет частицу '.plugin.php', то
7. Подключить этот файл
8. В массив имён плагинов добавить элемент, в который положить имя файла (из имени файла берётся всё, кроме '.plugin.php')
9. Закрыли ветвление
10. Закрыли ветвление

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

11. Закрыли цикл
12. Закрыли текущую папку
13. Создали переменную $vector, инициализировали её как массив из двух элементов: 3 и 4
14. Вывели текст
15. Вывели содержимое переменной (print_r — функция вывода структур данных)
16. Вывели текст
17. Открыли цикл по всем элементам массива имён плагинов, на каждом шаге цикла элемент перекладывается в переменную $p
18. В переменной определили имя информационной функции
19. В переменной определили имя рассчётной функции
20. Если хоть одна из этих функций не существует, то
21. Сообщаем об ошибке
22. В противном случае
23. Выводим текст
24. Выводим результат выполнения информационной функции
25. Выводим результат выполнения расчётной функции с переданным ей в аргументе нашим вектором
26. Выводим текст
27. Закрыли ветвление
28. Закрыли цикл по элементам массива имён плагинов
29. Закрыли php-инструкцию

Напишем плагины для этой системы

Файл *lp.plugin.php*

```
1.    <?php
2.
3.    function lp_info(){
4.          return "Евклидова норма";
```


САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

```
5.      }
6.
7.      function lp_norm($v=array()){
8.            $norm = 0;
9.            foreach ($v as $element){
10.                 $norm+=$element*$element;
11.           }
12.           return sqrt($norm);
13.     }
14.     ?>
15.     Файл max.plugin.php
16.     <?php
17.
18.     function max_info(){
19.           return "Чебышевская норма";
20.     }
21.
22.     function max_norm($v=array()){
23.             if (count($v)==0){
24.                     return 0;
25.             }
26.             $max = $v[0];
27.             foreach ($v as $element){
28.                     if ($element > $max){
29.                             $max = $element;
30.                     }
31.             }
32.             return $max;
33.     }
34.
35.     ?>
36.     Результатом вывода при запуске программы algebra.php будет:
37.
38.     Вычисляем норму вектора
39.     Array (
40.           [0] => 3
41.           [1] => 4
42.     )
43.     Евклидова норма равна 5
44.     Чебышевская норма равна 4
```

Файл *max.plugin.php*

```
1.      <?php
2.
3.      function max_info(){
4.            return "Чебышевская норма";
5.      }
6.
7.      function max_norm($v=array()){
8.             if (count($v)==0){
9.                     return 0;
```


САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

```
10.             }
11.             $max = $v[0];
12.             foreach ($v as $element){
13.                     if ($element > $max){
14.                             $max = $element;
15.                     }
16.             }
17.             return $max;
18.     }
19.
20.     ?>
```

Результатом вывода при запуске программы *algebra.php* будет:

```
1.
2.      Вычисляем норму вектора
3.      Array (
4.              [0] => 3
5.              [1] => 4
6.      )
7.      Евклидова норма равна 5
8.      Чебышевская норма равна 4
```

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Список литературы

1. Duckett J. Web Design with HTML, CSS, JavaScript and jQuery Set. – Wiley Publishing, 2014.
2. Дунаев В. В. Основы Web-дизайна. Самоучитель, 2 изд. – БХВ-Петербург, 2012.
3. Клименко Р. А. Веб-мастеринг на 100%. 2-е изд. – «Издательский дом ”Питер”», 2015.
4. Pomaska G. Grundkurs Web-Programmierung: Interaktion, Grafik und Dynamik—Mit XHTML und CSS, XML, JavaScript, Applets, SVG, PHP. – Springer-Verlag, 2015.
5. Квинт И. Создаем сайты с помощью HTML, XHTML и CSS на 100%. 3-е изд. – " Издательский дом"" Питер""", 2014.
6. Янк К. PHP и MySQL. От новичка к профессионалу. – Litres, 2014.
7. Warren T. PHP Programming For Beginners: The Simple Guide to Learning PHP Fast!. – 2015.
8. Колисниченко Д. Н. PHP и MySQL. Разработка Web-приложений. 4-е изд. – БХВ-Петербург, 2013.
9. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS. 2-е изд //СПб.: Питер. – 2013.
10. Котеров Д. В. PHP 5. 2 изд. – БХВ-Петербург, 2012.
11. Кузнецов М. В. Объектно-ориентированное программирование на PHP. – БХВ-Петербург, 2012.
12. Mikowski M. S., Powell J. C. Single Page Web Applications //B and W. – 2013.
13. Frain B. Responsive web design with HTML5 and CSS3. – Packt Publishing Ltd, 2012.
14. Лиза Г., Джейсон Г. Разработка веб-сайтов для мобильных устройств. – " Издательский дом"" Питер""", 2013.
15. Бессонов Л. В., Брагина И. Г. Операционные системы. Компьютерные сети: Пособие для студентов, обучающихся по дополнительной

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

специальности «Компьютерная графика и веб-дизайн» // Саратов: Изд-во «Научная книга», 2009. – 44 с. – ISBN 978-5-9758-1063-2

16. Бессонов Л. В., Брагина И. Г. Информационные технологии в профессиональной деятельности преподавателя: Учеб. пособие. – Саратов: Изд-во «Научная книга», 2014. – 28 с. – ISBN 978-5-9758-1524-8
17. Бессонов Л. В. Формирование требований к официальному сайту вуза на примере официального сайта СГУ // Наука и образование в XXI веке. Сборник научных трудов по материалам Международной научно-практической конференции 30 января 2015 г.: в 5 частях. Москва, 2015, С. 105–108.
18. Шаталина А.В., Бессонов Л.В. О подходе к задаче автоматизации проверки корректности и полноты входящих в ООП документов // Фундаментальные и прикладные исследования в современном мире. 2016. № 13-1. С. 68-71.
19. Бессонов Л.В., Сецинская Е.В., Кухарев В.В. Анализ требований к дизайну сайта вуза для лиц с ограниченными физическими возможностями // Фундаментальные и прикладные исследования в современном мире. 2015. №11-4. С. 85-87.
20. Бессонов Л.В., Тышкевич С.В., Панкратов Д.В. О подходе к формированию персональных страниц преподавателей на официальном сайте вуза // Фундаментальные и прикладные исследования в современном мире. 2015. №11-1. С. 120-123.
21. Дмитриев П.О., Никулин Б.Л. Моделирование системы представления олимпиадных задач на сайте вуза // Новые задачи технических наук и пути их решения: сборник статей Международной научно-практической конференции (10 апреля 2016 г., г. Пермь). Уфа: АЭТЕРНА. 2016. С. 26–28
22. Матершев И.В., Дмитриев П.О. О построении автоматизированной системы управления хостингом для обучающихся по IT-направлениям подготовки // Наука, образование и инновации: сборник статей Международной научно-практической конференции (25 июня 2016 г., г. Томск). В 4 ч. Ч.3. Уфа. 2016. С. 49–51.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

23. Бессонов Л.В., Дмитриев О.Ю. Подход к написанию технического задания на разработку сайта поддержки олимпиадного движения по предмету // Фундаментальные и прикладные исследования в современном мире. 2016. №15-1. С. 154-157.
24. Амелин Р. В. Информационные технологии: учебное пособие для студентов механико-математического факультета, обучающегося по специальности 351400 «Прикладная информатика» (по областям). – Саратов: Изд-во Сарат. ун-та, 2006. – 52 с.
25. Амелин Р. В. Мировые информационные ресурсы: учебное пособие для студентов механико-математического факультета, обучающегося по специальности 351400 «Прикладная информатика» (по областям). – Саратов: Изд-во Сарат. ун-та, 2006. – 88 с.
26. Амелин Р.В. Правовой режим государственных информационных систем: монография / под ред. С.Е. Чаннова. М.: ГроссМедиа, 2016. – 338 с.
27. Амелин Р.В. Обязанность по представлению информации в федеральные информационные системы // Административное и муниципальное право. – 2015. – № 10. – С. 1081–1089.
28. Дмитриев П.О. Практикум по веб-программированию. Теоретическое введение в язык PHP: Учебное пособие для студентов, обучающихся по направлению подготовки бакалаврита 09.03.03 «Прикладная информатика» — Саратов: ООО Издательский Центр «Наука», 2016. — 40 с.

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО

Учебное издание

Бессонов Л. В.

Практикум по веб-программированию

Упражнения и практические задания

Учебное-методическое пособие для студентов, обучающихся по направлению подготовки бакалавриата 09.03.03 «Прикладная информатика»

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Подписано в печать 18.11.2016. Формат 60x84 1/16. Бумага офсетная.
Гарнитура Times New Roman. Печать RISO. Объем 2,5 печ. л.
Тираж 100 экз. Заказ № 202.

ООО Издательский Центр «Наука»
410012, г. Саратов, ул. Пугачевская, 117, оф. 50

Отпечатано с готового оригинал-макета
Центр полиграфических и копировальных услуг
Предприниматель Серман Ю.Б. Свидетельство № 3117
410012, Саратов, ул. Московская, д.152, офис 311, тел. 26-18-19

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО