

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»

Павловский филиал ННГУ

УТВЕРЖДЕНО

решением Ученого совета ННГУ
протокол от «31» мая 2023 г. № 6

**Рабочая программа дисциплины
РАЗРАБОТКА WEB-ПРИЛОЖЕНИЙ**

(наименование дисциплины (модуля))

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки / специальность

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

Направленность образовательной программы

ПРИКЛАДНАЯ ИНФОРМАТИКА В ЭКОНОМИКЕ И УПРАВЛЕНИИ

Форма обучения

ОЧНАЯ, ОЧНО-ЗАОЧНАЯ

Павлово
2023 год

1. Место и цели дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.07.02 «Разработка web-приложений» относится к дисциплинам по выбору части, формируемой участниками образовательных отношений, Блока 1. «Дисциплины (модули)» учебного плана ООП по направлению подготовки 09.03.03 «Прикладная информатика» (квалификация (степень) «бакалавр»).

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ПК-8. Способен разрабатывать лингвистическое, информационное и программное обеспечение ИС (ИИС) и сопровождающую его документацию	ПК-8.1. Способен использовать современные языки и системы программирования, формализмы описания знаний на концептуальном и инфологическом уровнях, требования к технической документации на все виды обеспечения ИС (ИИС).	Знать основы разработки web-сайтов	Практические задания, тесты
	ПК-8.2. Способен применять современные языки и системы программирования, формализмы описания знаний на концептуальном и инфологическом уровнях при разработке лингвистического, информационного и программного обеспечения ИИС и сопровождающей его документации.	Уметь проектировать сайты	Практические задания, тесты
	ПК-8.3. Способен осуществлять разработку лингвистического, информационного и программного обеспечения конкретной ИС (ИИС) и сопровождающей его документации.	Владеть технологиями проектирования, основами программирования сайтов различными программными средствами	Практические задания, тесты

3. Структура и содержание дисциплины

3.1. Трудоемкость дисциплины

Для очной формы обучения:

Общая трудоемкость	3 ЗЕТ
Часов по учебному плану	108
в том числе	
аудиторные занятия (контактная работа):	49
- занятия лекционного типа	16
- занятия лабораторного типа	32
самостоятельная работа	59
Промежуточная аттестация - зачет	

Для очно-заочной формы обучения:

Общая трудоемкость	3 ЗЕТ
Часов по учебному плану	108
в том числе	
аудиторные занятия (контактная работа):	25
- занятия лекционного типа	12
- занятия лабораторного типа	12
самостоятельная работа	83

3.2. Содержание дисциплины

Наименование и краткое содержание разделов и тем дисциплины	Всего (часы)			В том числе															Самостоятельная работа обучающегося, часы		
				Контактная работа (работа во взаимодействии с преподавателем), часы из них																	
				Занятия лекционного типа			Занятия лабораторного типа			Занятия семинарского типа			Консультации			Всего					
Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная				
Тема 1. Язык гипертекстовой разметки HTML	15	15		2	1		4	1							6	2		9	13		
Тема 2. Дизайн страниц с использованием стилевых таблиц CSS	15	15		2	1		4	1							6	2		9	13		
Тема 3. Объектная модель браузера	14	14		2	2		4	2							6	4		8	10		
Тема 4. Синтаксис языка сценариев Java Script.	15	15		2	2		4	2							6	4		9	11		
Тема 5. Объекты в Java Script	14	14		2	2		4	2							6	4		8	10		
Тема 6. Язык серверной активности PHP	14	14		2	2		4	2							6	4		8	10		
Тема 7. Управление данными СУБД MySQL средствами PHP	20	20		4	2		8	2							12	4		8	16		
КСР	1	1													1	1					
Контроль	36	36																			
ИТОГО	108	108		16	12		32	12							49	25		59	83		

Содержание дисциплины

Тема 1. Язык гипертекстовой разметки HTML

Структура языка гипертекстовой разметки HTML, основные принципы создания web-страниц. Способы компоновки элементов на странице.

Тема 2. Дизайн страниц с использованием стилевых таблиц CSS.

Принцип оформления сайта в едином стиле с помощью технологии каскадных стиле-

вых таблиц CSS.

Тема 3. Объектная модель браузера

Обзор объектной модели браузера Internet Explorer, различия в объектных моделях различных браузеров. Объект, его свойства, методы и события, способы реализации активности на странице.

Тема 4. Синтаксис языка сценариев Java Script.

Синтаксис, операторы и базовые конструкции языка JavaScript. Принципы программирования и реализация активности страницы на стороне клиента.

Тема 5. Объекты в Java Script

Обзор встроенных объектов в JavaScript, их свойства, события и методы. Технология использования этих объектов в сценариях JavaScript. Взаимодействие языка JavaScript и элементов страницы.

Тема 6. Язык серверной активности PHP.

Синтаксис, операторы и базовые конструкции языка серверных сценариев PHP. Разница между языками реализации активности на стороне клиента и языками, работающими на стороне сервера. Способы формирования web-страниц средствами PHP. Команды взаимодействия сценария и базы данных MySQL. Создание серверных сценариев.

Тема 7. Управление данными СУБД MySQL средствами PHP

Работа с базами данных из командной строки и с помощью специальных оболочек, создание и заполнение таблиц, организация выборки по критериям. Создание, изменение структуры, заполнение, удаление таблиц. Способы выборки информации по критериям.

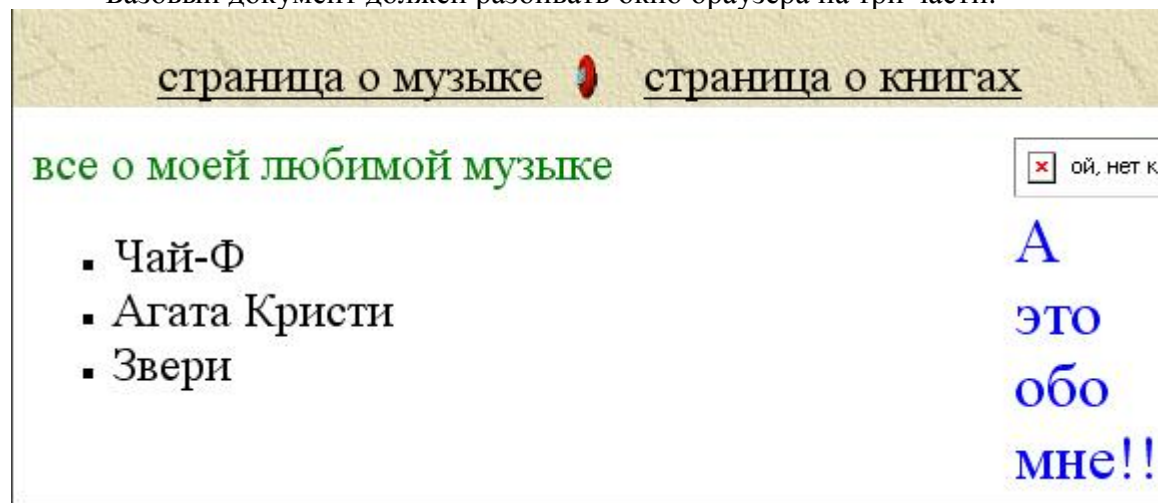
ПЛАНЫ СЕМИНАРСКИХ И ПРАКТИЧЕСКИХ ЗАДАНИЙ

Тема 1. Язык гипертекстовой разметки HTML

Задание 1

Используя в качестве инструментов блокнот и Internet Explorer, создайте набор страниц о чем угодно. В частности, это могут быть страницы о Ваших увлечениях.

Базовый документ должен разбивать окно браузера на три части:



В верхней части меню позиционировано с помощью таблицы и тега <center> (каждый элемент находится в отдельной ячейке). При выборе любого пункта этого меню в левом нижнем окне загружается соответствующая страница. Справа расположен небольшой фрейм с информацией об авторе, обязательно с небольшим изображением. Если файл изображения отсутствует, должна выводиться подпись.

На одной из страниц, загружаемых по выбору в меню, должна быть таблица (обязательно выполнить объединение ячеек!). Пример представлен на рисунке:

авторы

все о моих любимых авторах	Гарри Гаррисон	
	Корабельный врач	Специалист по этике или этический инженер
	Иоанна Хмелевская	
	Что сказал покойник	Все красное Лесь

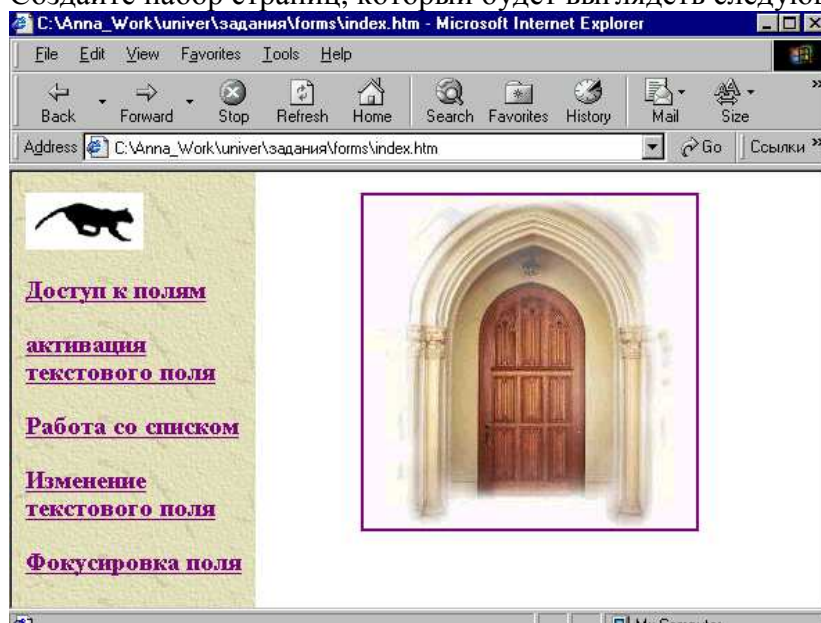
На второй странице представьте любую информацию в виде маркированного списка .

все о моей любимой музыке

- Чай-Ф
- Агата Кристи
- Звери

Задание 2

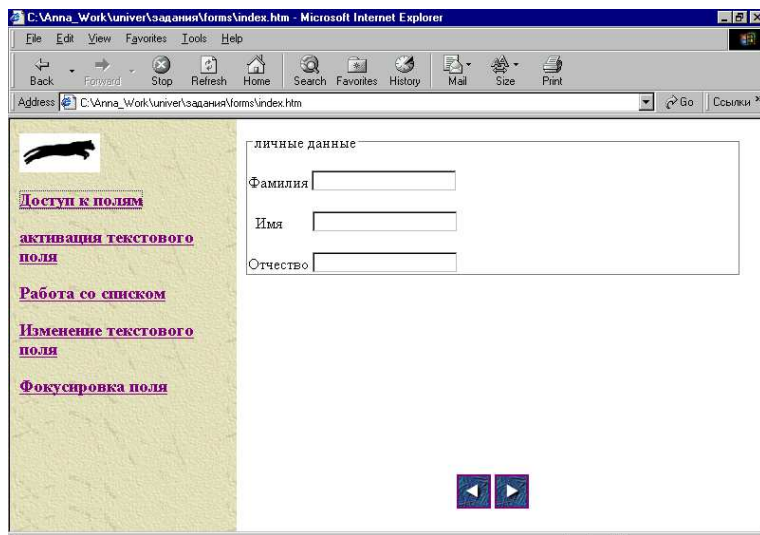
Создайте набор страниц, который будет выглядеть следующим образом:



Разбиение окна организуйте с помощью таблицы.

В левом окне пять ссылок на страницы, содержащие решение приведенных ниже задач. В правом окне картинка-ссылка, при нажатии на которую Вы попадаете на первую задачу.

Каждая страница-задача содержит две картинки-ссылки – вперед и назад, позволяя перемещаться по задачам по порядку (см. рис.)



Задачи:

1. В форму ввести три текстовых поля с метками. Организовать доступ к полям по нажатию клавиш Alt+1, Alt+2, Alt+3. Необходима всплывающая подсказка, на какие клавиши нажимать.
2. На экран вывести форму, содержащую элементы checkbox и text. При выборе checkbox текстовое поле должно быть активировано (свойство disabled).
3. Создать список Select. По выбору в списке отображать методом alert значение и номер выбранной позиции.
4. По нажатию кнопки текст в текстовом поле изменяется на любой другой.
5. Ввести форму с несколькими текстовыми полями. По нажатию на кнопку установить фокус в первое текстовое поле и выделить его содержимое.

Тема 2. Дизайн страниц с использованием стилевых таблиц CSS

Задание 1

Используя блокнот и браузер Internet Explorer, создать набор связанных страниц, оформленных с помощью одной таблицы стилей.

Шаблон страницы оформляется с помощью таблиц.

Таблица стилей должна содержать следующие описания:

Селектор BODY: цвет текста черный, размер шрифта 14 pt, фоновый рисунок aser3.jpg. Фоновый рисунок должен быть размещен по правому краю страницы, размножен по вертикали.

Класс bold: цвет текста оранжевый, размер шрифта 36pt, начертание жирное, выравнивание по правому краю.

ID-селектор over: текст подчеркнут сверху.

Класс red: цвет текста красный.

Классовые селекторы: цвет ссылки синий, цвет посещенной ссылки красный, подчеркивания нет, при наведении на ссылку цвет меняется на красный.

Создайте три связанных страницы.

Первая 1.htm включает текст, оформленный одновременно с помощью классического селектора bold и id-селектора, текст, оформленный только с помощью классического селектора bold и текст, оформленный только с помощью id-селектора. Кроме этого, разместите любую картинку и ссылку на документ 2.htm.

It's my page

Контрольная

Text



Text



Также добавьте на страницу любой текст, чтобы добиться появления полос прокрутки.

Файл 2.htm должен содержать форму следующего вида:

Данные1

Первое поле

Второе поле

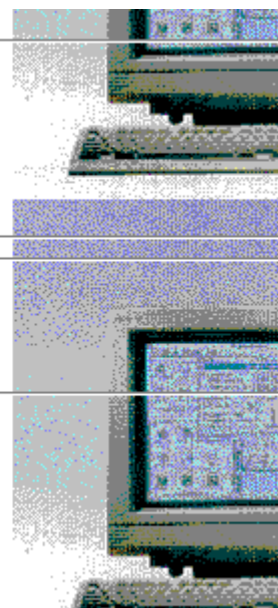
Данные2

first

second

third

Отправка данных



Далее

Назад

Список выбора должен содержать пять пунктов.

Третья страница должна содержать таблицу следующего вида:



Назад



Подписи в ячейках таблицы должны оформляться с помощью классового селектора red и id-селектора over. Рамку таблицы оформите с помощью внутреннего стиля.

Задание 2

Создайте набор страниц, который будет выглядеть следующим образом

<p>позиционирование</p> <p>z-index</p> <p>фильтры</p>	<p>Привет!</p> <p>С помощью меню Вы можете познакомиться с позиционированием, а также с фильтрами таблиц стилей.</p>
---	---

В левом окне меню с тремя ссылками:

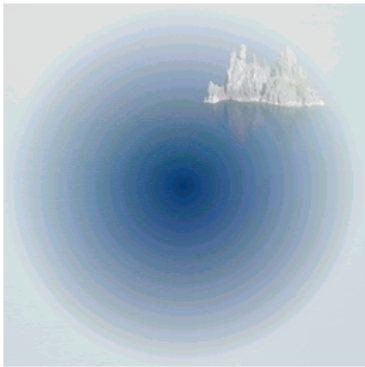
1. пример взаимного расположения текста при абсолютном и относительном позиционировании, например:

<p>позиционирование</p> <p>z-index</p> <p>фильтры</p>	<p>Этот текст позиционирован относительно</p> <p>Этот текст позиционирован абсолютно</p>
	<p>Этот текст позиционирован абсолютно</p> <p>Этот текст позиционирован относительно</p>

2. Пример создания трехмерных эффектов с помощью свойства z-index.

<p>позиционирование</p> <p>z-index</p> <p>фильтры</p>	<p>Привет!!!</p> <p>Это пример объемного изображения</p>
---	---

3. Пример трех любых фильтров.

позиционирование	blur
z-index	dropShadow
фильтры	
	alpha

При этом общий стиль всех страниц задан в связанной таблице стилей (файле style.css), а индивидуальный стиль каждой страницы дополнительно задается в каждом документе. Ссылки обязательно должны менять свой цвет, когда на них наводится указатель мыши!

Тема 3. Объектная модель браузера

Задание 1

Создайте одну страницу, содержащую четыре работающих ссылки: «обо мне», «О моих друзьях», «новости», «на главную».

По нажатию на каждую ссылку в центре страницы появляется соответствующая информация.

«Обо мне»



«О моих друзьях»

Много информации, а страница одна!!!



Это главное
меню

Обо мне

О моих друзьях

Новости

На главную

Мои друзья!!!



Это первый



Это второй

«Новости»

Много информации, а страница одна!!!



Это главное
меню

Обо мне

О моих друзьях

Новости

На главную



новость 1



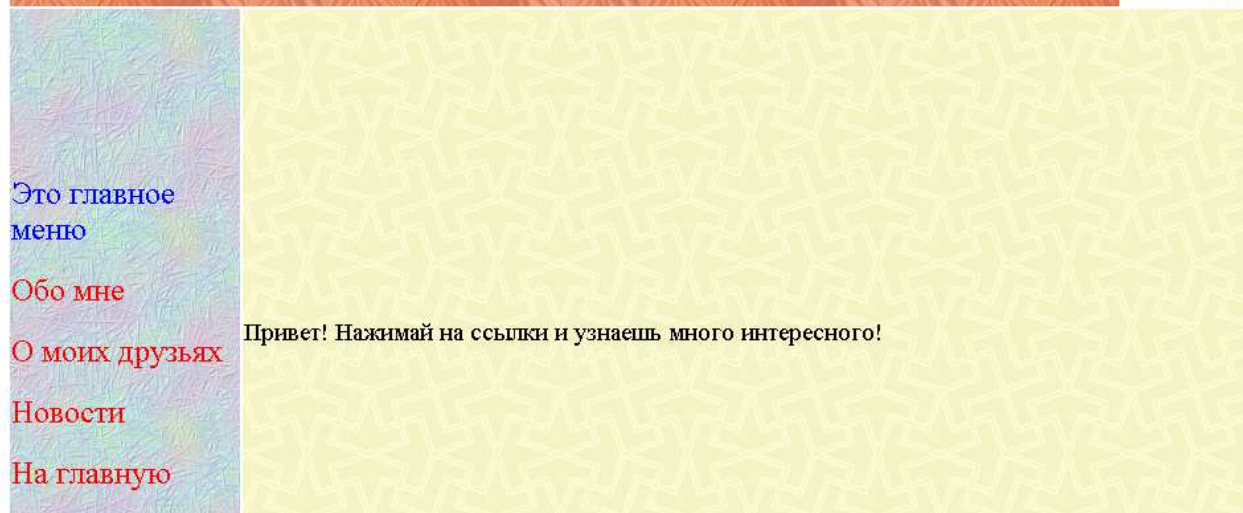
новость 2



новость 3

«На главную»

Много информации, а страница одна!!!



Первоначальный вид страницы такой же, как и при нажатии на ссылку «На главную».

Разбиение окна организуйте с помощью таблиц.

Оформление сделайте полностью с помощью стилевых таблиц.

Кроме этой страницы больше никаких других быть не должно!!!

Задание 2

Разместите на странице картинку и найдите координаты ее углов в клиентской области.

Задание 3

Выведите на экран таблицу из одной строки и двух ячеек. В первой ячейке содержится текст, по щелчку на который изменяется текст второй ячейки. Создайте две таких текстовых ссылки.

click me click ME	click me click ME	click me click ME
Change text	HELLO	Good Day
Первоначальный вид страницы	После щелчка по первому текстовому блоку	После щелчка по второму текстовому блоку

Задание 4

Создайте страницу, на которой расположите кнопку и картинку. Добейтесь, чтобы при нажатии на кнопку картинка менялась на другую.

Задание 5

Создайте страницу, на которой расположите кнопку. По нажатию на кнопку открыть окно шириной 100 и высотой 200. Загрузите в это окно произвольную страницу.

Задание 6

Создайте HTML-страницу с двумя текстовыми полями и одной кнопкой. В одно поле должен вводиться цвет текста, в другое – цвет фона. При нажатии на кнопку, цвета страницы устанавливаются в указанные в полях.

Тема 4. Синтаксис языка сценариев JavaScript

Задание 1

Создать HTML-страницу, которая при загрузке случайным образом выводит один из трех эпиграфов.

Задание 2

Написать функцию, которая проверяет, что в строке, переданной в качестве аргумента, соблюдается баланс открытых и закрытых скобок. При этом необходимо учесть, что закрывающая скобка не может появиться раньше соответствующей ей открывающей скобки. Функция возвращает строку "O'key" при соблюдении такого баланса, и "Error" в противном случае.

Тема 5. Объекты JavaScript

Задание 1

Добавьте к системному объекту Date метод isBusinessTime. Ваш метод, должен возвращать true, если экземпляр Date, в контексте которой он вызван, задает рабочее время и false, в противном случае.

Рабочим временем считайте время с 8 до 17 часов во все дни, кроме субботы и воскресенья. Напишите программу с тестами для демонстрации работы Вашего метода.

Задание 2

Возьмите несколько (не менее пяти) любых картинок одинакового размера. Сделайте так, чтобы на Вашей странице отображались одновременно все картинки в различных местах. Пользователь может нажать над одной из картинок кнопку мыши и, не отпуская кнопку, перевести мышь на другую картинку и там отпустить. Картинки, при этом, должны немедленно поменяться местами.

Следите, чтобы Ваша программа была "дуракоустойчивой". Например, если кнопку отпустили не над другой картинкой, а над пустым местом, или, наоборот, нажали кнопку над пустым местом, а отпустили над картинкой, Ваша программа должна вести себя предсказуемо.

Тема 6. Язык серверных сценариев PHP

На практических занятиях совместно с преподавателем разбираются следующие возможности:

- Использование файла функций для оформления сайта с помощью шаблона;
- Использование функций работы с датами для организации поиска;
- Использование численно индексированных и ассоциативных массивов.

Задание

Ситуация.

Необходимо создать имитацию приема заказов в электронном магазине.

Для этого необходимо организовать следующие файлы:

Форма заказа. Включает в себя информацию о ценах на товары и форму заказа, состоящую из текстовых полей и кнопки отправки данных. Каждое поле соответствует товару и в него вводится заказываемое количество. Добавьте ссылку на страницу, в которой сообщается текущая дата на русском языке. После просмотра даты, пользователь может вернуться назад. Результат заполнения формы выводится следующим файлом. Если форма заполнена верно (все данные числовые), то пользователю сообщается стоимость заказа по каждому товару и общая стоимость всего заказа. Можно заполнить не все поля, но если все поля пустые, то пользователь отсылается в форму заказа.

По ссылке пользователь переходит к форме регистрации. Файл включает форму для заполнения данными пользователя: имя, фамилия, электронный адрес и страна. Выпадающий список стран сформируйте с помощью массива.

Если форма заполнена правильно (все поля заполнены и электронный адрес соответствует стандарту xxx@x.x), то пользователю выдается сообщение об успешно принятом заказе на языке той страны, которая была выбрана в списке.

Ваш заказ принят, спасибо за покупку.

Your order is accepted. Thanks for purchase.

Votre commande est acceptée. Merci pour l'achat.

Ihre Bestellung ist übernommen. Danke für den Kauf.

Страница с формой регистрации пользователя внешне должна быть идентична по внешнему виду странице с формой заказа!!! Различие заключается в формах и в наличии ссылки на страницу заказа товаров со страницы регистрации. На странице с формой регистрации также содержится ссылка на просмотр даты и времени.

ОБРАТИТЕ ВНИМАНИЕ на обратную ссылку со страницы с датой. С какого бы места на нее не попал пользователь, ссылка «НАЗАД» должна работать корректно и возвращать его на ту страницу, с которой он пришел.

В итоге получается шесть файлов: форма заказа, форма регистрации, стилевая таблица, библиотека функций, просмотр даты и обработка заказа (подсчет суммы).

При организации проверки заполнения форм воспользуйтесь двумя методами:

Для формы заказа проверку организуйте в виде отдельной функции, которая вызывается в файле, куда пересылаются данные формы. Функцию проверки ввода поместите в библиотеку функций func.php.

Для формы регистрации проверку организуйте в этом же файле методом «if (\$submit)».

Посмотреть пример можно по адресу:

http://intranet:8008/bespalko/web/zadanie/form_work.htm.

Тема 7. Управление данными СУБД MySQL средствами PHP

На практических занятиях совместно с преподавателем разбираются следующие возможности:

- Вывод полученных по запросу к базе данных на страницу сайта;
- Создание отношения в базе и вывод списка отношений из базы;
- Заполнение отношения данными, введенными пользователем в форму;
- Возможности удаления данных;
- Сложный поиск.

В качестве задания по теме предлагается практическая ситуация, которая также реализуется и в качестве самостоятельной работы.

Ситуация.

Создайте систему работы с базой данных. В программе должны быть предусмотрены следующие действия: создание отношений, удаление отношений, заполнение отношений данными, удаление данных из отношений, просмотр данных и поиск.

Работающий пример по адресу: <http://intranet:8008/bespalko/web/data/index.php>

1. Вход в систему

Любая страница Вашей программы построена по одному шаблону: заголовок, блок меню и блок контента. Блок меню занимает 30% ширины экрана, блок контента – остальное.

Части шаблона прописываются в виде отдельных функций в файле, содержащем библиотеку функций. Разбиение страницы на блоке организуйте в виде таблицы. Отдельные части таблицы будут находиться в разных функциях. Пример такой организации шаблона находится в папке Cart (примеры PHP).

При загрузке файла index.php блок меню содержит главное меню, которое будет расположено в этом блоке на протяжении работы всей программы. Пункты меню:

- Создать отношения;
- Удалить отношения;
- Заполнить отношения;
- Очистить отношения;
- Автоматически заполнить отношения;
- Автоматически очистить отношения;
- Работа с данными.

В блоке контента расположена текстовая строка и кнопка. В строку пользователь вводит свое имя. При нажатии на кнопку пользователь остается в этом же окне – никаких видимых действий не происходит (можете добавить окно-приветствие). При выборе ЛЮБОГО пункта ме-

ню в строке заголовка ВСЕГДА отображается приветственное обращение к пользователю по тому имени, которое он ввел в строке. Это реализуется с помощью переменной сессии.

2. Создание отношений

При выборе пункта меню «Создание отношений» программа пытается создать отношения в базе данных. Про каждое успешно созданное отношение должно появиться соответствующее сообщение (в блоке контента в виде текста). Если хотя бы одно отношение не создано, об этом сообщается пользователю и для контроля на экран выводится список имеющихся в базе данных отношений (функции `mysql_list_tables` и `mysql_tablename`).

При создании отношений обратите внимание на работу с датами. MYSQL хранит дату в неудобном формате. Причем, попытка заполнить нормальной датой приведет к ее конвертации в формат MYSQL: «12.02.2006» преобразится в «2012-02-20». Поэтому для последующей организации поиска, сортировки и сравнения удобнее хранить дату в виде числа. Например, «12.02.2006» хранится как число 20060111. В случае сравнения двух дат мы получим корректный результат – сначала сравнение по году, потом по месяцу, потом по дню.

Для конвертации воспользуйтесь следующими функциями:

Конвертация в число

```
function convert_data($data)
```

```
{  
  $d=substr($data,6,4);  
  $d=$d.substr($data,3,2);  
  $d=$d.substr($data,0,2);  
  return $d;  
}
```

Конвертация в строку с вилом ДД.ММ.ГГГГ

```
function print_data($data)
```

```
{  
  $d=substr($data,6,2).".";  
  $d=$d.substr($data,4,2).".";  
  $d=$d.substr($data,0,4);  
  return $d;  
}
```

3. Удаление отношений

При выборе пункта меню «Удаление отношений» программа пытается удалить отношения из базы данных. Про каждое успешно удаленное отношение должно появиться соответствующее сообщение (в блоке контента в виде текста). Если хотя бы одно отношение не удалено, об этом сообщается пользователю и для контроля на экран выводится список имеющихся в базе данных отношений (функции `mysql_list_tables` и `mysql_tablename`).

4. Заполнить отношения

Все четыре отношения заполняются отдельно. При выборе этого пункта меню в блоке контента отображается набор ссылок – по одной на каждое отношение. Нажатие на ссылку открывает страницу, у которой в блоке контента расположена форма заполнения соответствующего отношения. Пример организации заполнения находится в папке `databases` (файлы `form_ins.htm` и `insert.php`). У Вас не будет файла `htm`, а форма будет формироваться и обрабатываться в `rnr`-файле.

5. Очистить отношения

В этой части программы предполагается предоставить пользователю удалять данные из отношений. Выбор удаляемого элемента организуйте в виде выпадающего списка. Лишь для отношения связи (в примере `OTPRAVKA`) выведите все данные и организуйте запрос на удаление с помощью `checkbox` – пользователь помечает те записи, которые надо удалить. Однако следует помнить, что как правило данные между собой связаны. Например, в отношении `TOVAR` хранится код типа товара, а сам тип описан в отношении `TYPE`. Удаление данных следует организовать двумя способами:

- для двух отношений (в примере это Товар и Type) нет возможности удалить тип товара из отношения TYPE, пока в отношении TOVAR есть хотя бы одна связанная запись. Также нет возможности удалить товар из отношения TOVAR, пока в отношении OTPRAVKA есть хотя бы одна связанная с этим товаром запись.

- Для одного отношения (в примере это отношение FIRMA) при попытке удаления фирмы, которой была совершена покупка (записи в отношении OTPRAVKA) пользователю предоставляется возможность выбрать: вместе с фирмой удалить также связанные записи из OTPRAVKA или не удалять записи вообще.

Для проверки наличия связанных записей организуйте запрос типа:

```
select * from otpravka where idTovar=$idTovar
```

Если в результате выполнения запроса была возвращена хотя бы одна строка, тогда или запрет на удаление (первый случай) или запрос на удаление (второй случай).

Для организации удаления данных из отношения OTPRAVKA в примере реализован механизм помечания записей на удаление:

Код формирования таблицы:

```
$query="select idO, idOtp, dataOt,kolvo,price,naimTovar,naimFirm from otpravka,tovar,firma where tovar.idTovar=otpravka.idTovar and firma.idfirm=otpravka.idfirm";
```

```
$result=mysql_query($query);
```

```
$num=mysql_num_rows($result);
```

```
echo "<form action=show_del_otpravka.php method=post>";
```

```
echo "<table border=5>";
```

```
echo "<tr><td>пометить на удаление</td><td>Id отправки</td>";
```

```
echo "<td>Дата отправки</td>";
```

```
echo "<td>Наименование товара</td>";
```

```
echo "<td>Количество товара</td>";
```

```
echo "<td>Цена единицы товара</td>";
```

```
echo "<td>Фирма-покупатель</td>";
```

```
for ($i=0; $i<$num; $i++)
```

```
{
```

```
    $row=mysql_fetch_array($result);
```

```
    $idO=$row['idO'];
```

```
    $dO=print_data($row['dataOt']);
```

//вывод элемента checkbox и формирование его имени в виде массива

```
echo "<tr><td><input type=checkbox name=myar[\".$i.\"]></td>";
```

```
echo "<td>".$row['idOtpr']."</td>";
```

```
echo "<td>".$dO."</td>";
```

```
echo "<td>".$row['naimTovar']."</td>";
```

```
echo "<td>".$row['kolvo']."</td>";
```

```
echo "<td>".$row['price']."</td>";
```

```
echo "<td>".$row['naimFirm']."</td>";
```

//сохраняем массив с кодами записей, помеченных на удаление. Массив формируется как имя скрытого текстового поля, значение каждого элемента – код помеченной записи.

```
echo "<input type=hidden name=mar[$i] value=$idO>";
```

```
}
```

```
echo "</table>";
```

```
echo "<input type=submit value='Показать' name='submit'>";
```

Код обработки помеченных записей:

```
if (isset($myar))
```

```
{
```

```
    $query="select idO from otpravka";
```



```

$result=mysql_query($query);
$num=mysql_num_rows($result);
for ($i=0;$i<=$num;$i++)
{
//проверка, помечена ли хотя бы одна запись – выбран ли хотя бы один checkbox
if (isset($myar[$i]))
//если пометки на удаление есть – удаляем запись
$query="delete from otpravka where idO=$mar[$i]";
$result=mysql_query($query);
if ($result)
{
?>
<script language="javascript">alert("Запись удалена!")
document.location="http://localhost/anna/data/del_otpravka.php"</script>
<?
}
}
}

```

6. Автоматически заполнить отношения

Для удобства отладки вставьте этот пункт меню. Код выложен вместе с заданием: autoinsert.php и func.php. Это сократит время отладки – после удаления отношений можно будет автоматически их заполнить.

7. Автоматически очистить отношения

Для удобства отладки вставьте этот пункт меню. Код выложен вместе с заданием: auto-delete.php и func.php.

8. Работа с данными

В блоке контента появляются ссылки на просмотр всех отношений, которые у вас имеются. Для трех отношений (в примере это TOVAR, FIRMA и TYPE) выбор пункта просмотра приводит к выводу таблицы с содержимым отношения. Не забудьте выводить связанные записи из других отношений не в виде кодов. Например, в отношении TOVAR название типа товара берется из отношения TYPE по коду.

Для отношения связи (в примере это OTPRAVKA) организуем поиск. Запомните, что поиск организовывается для отношения, которое связывает между собой остальные отношения (даже если и не все, то основные – «товар продается фирме» или «пользователь покупает услугу» и т.д.)

Поля формы поиска организуйте в зависимости от данных, которые хранятся в Вашем отношении. Рассмотрим на примере OTPRAVKA.

Отношение OTPRAVKA содержит в себе следующие данные:

idO – код записи, первичный ключ.

idOtp – код отправки (в случае отправки нескольких товаров; для соблюдения избыточности как правило подобное отношение разбивают на два).

dataOt - дата отправки (целое).

kolvo - количество отправленного товара.

Price - цена, по которой была продана единица товара.

idFirm - код фирмы, купившей товар.

idTovar - код отправленного товара.

Форма поиска будет содержать два выпадающих списка: наименование товара и название фирмы. Причем в каждом из списков первая строка пустая – на случай, если пользователь не сделает своего выбора. Тестовое поле «Идентификатор отправки» - это код отправки. В поля «Начальная дата» и «Конечная дата» пользователь вводит даты, между которыми осуществляется поиск.

Поиск должен работать корректно независимо от того, какие поля заполнены:

- Если заполнена одна дата из двух, то поиск ведется именно по этой дате;
- Если форма пустая, то выводятся все данные;

В поле «Идентификатор отправки» должно содержаться числовое значение (проверка – функция is_int). В полях «Начальная дата» и «Конечная дата» должны быть введены корректные даты: dd.mm.yyyy.

Код формирования запроса поиска:

```
function searh_form($idtovar,$idFirm,$idOtp,$databegin,$dataend)
```

```
{
```

```
//начало строки запроса
```

```
$query="select idOtp,price,kolvo, dataOt,naimTovar,sebestoim, naimFirm from otpravka,tovar,firma where tovar.idTovar=otpravka.idTovar and firma.idFirm=otpravka.idFirm";
```

```
//формирование условий в зависимости от заполненных полей
```

```
//Если указан товар
```

```
if ($idtovar<>0)
```

```
{
```

```
    $query=$query." and otpravka.idTovar=$idtovar";
```

```
}
```

```
//Если указана фирма
```

```
if ($idFirm<>0)
```

```
{
```

```
    $query=$query." and otpravka.idFirm=$idFirm";
```

```
}
```

```
//Если введены обе даты
```

```
if (!empty($databegin) && !empty($dataend))
```

```
{
```

```
    $databegin=convert_data($databegin);
```

```
    $dataend=convert_data($dataend);
```

```
    $query=$query." and dataOt>=$databegin and dataOt<=$dataend";
```

```
}
```

```
//Если введена только первая дата
```

```
if (!empty($databegin) && empty($dataend))
```

```
{
```

```
    $databegin=convert_data($databegin);
```

```
    $query=$query." and dataOt=$databegin";
```

```
}
```

```
//Если введена только вторая дата
```

```
if (empty($databegin) && !empty($dataend))
```

```
{
```

```
    $dataend=convert_data($dataend);
```

```
    $query=$query." and dataOt=$dataend";
```

```
}
```

```
//Если указан код отправки
```

```
if (!empty($idOtp))
```

```
{
```

```
    $query=$query." and idOtp=$idOtp";
```

```
}
```

```
//выполнение запроса
```

```
$result=mysql_query($query);
```

```
if ($result)
```

```
{
```

```
    $num=mysql_num_rows($result);
```

```
//Вывод результата, если таковой имеется
```

```

if ($num>0)
{
echo "<table border=5>";
echo "<tr><td style='font-weight:bold'>Id отправки</td>";
echo "<td style='font-weight:bold'>Дата отправки</td>";
echo "<td style='font-weight:bold'>Наименование товара</td>";
echo "<td style='font-weight:bold'>Количество товара</td>";
echo "<td style='font-weight:bold'>Цена единицы товара</td>";
echo "<td style='font-weight:bold'>Фирма-покупатель</td>";
echo "<td style='font-weight:bold'>Общая стоимость</td>";
echo "<td style='font-weight:bold'>Себестоимость единицы товара</td>";
echo "<td style='font-weight:bold'>Прибыль</td></tr>";

for ($i=0; $i<$num; $i++)
{
$row=mysql_fetch_array($result);
$dO=print_data($row['dataOt']);
$stoim=$row['price']*$row['kolvo'];
$prib=$stoim-$row['sebestoim']*$row['kolvo'];
echo "<tr><td>".$row['idOtp']. "</td>";
echo "<td>".$row['naimTovar']. "</td>";
echo "<td>".$dO. "</td>";
echo "<td>".$row['kolvo']. "</td>";
echo "<td>".$row['price']. "</td>";
echo "<td>".$row['naimFirm']. "</td>";
echo "<td>".$stoim. "</td>";
echo "<td>".$row['sebestoim'];
echo "<td>".$prib. "</td></tr>";
}
echo "</table>";
echo "<p><span style='color:red;font-size:18 pt'>
<center><a href=select_otpravka.php>Назад</a></span>";
}
else
{
echo "<span style='color:blue;font-size:18 pt'><center>
По вашему запросу ничего не найдено";
echo "<p><span style='color:red;font-size:18 pt'>
<center><a href=select_otpravka.php>Назад</a></span>";
}
}
}

```

Функции print_data и convert_data описаны в пункте «Создание отношений».

Занятия по дисциплине организуются в том числе в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

Практическая подготовка предусматривает решение прикладных задач. Часы практической подготовки выделяются из часов занятий лекционного типа и лабораторных занятий в объеме, равном 50% от общего объема часов, отведенных на перечисленные виды занятий.

Практическая подготовка направлена на формирование и развитие:

- ✓ практических навыков в соответствии с профилем ОП:
 - формирование и анализ требований к информатизации и автоматизации прикладных процессов, формализация предметной области проекта;
 - программирование приложений, создание прототипа информационной системы;
- ✓ компетенций - ПК-8.

Текущий контроль успеваемости реализуется в рамках занятий лабораторного типа.

Промежуточная аттестация проходит в традиционной форме - зачет, включающий ответы на вопросы по программе дисциплины.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа является важнейшей составной частью учебного процесса и обязанностью каждого студента. Качество усвоения учебного материала находится в прямой зависимости от способности студента самостоятельно и творчески учиться. Самостоятельная работа способствует формированию аналитического и творческого мышления, совершенствует способы организации исследовательской деятельности, воспитывает целеустремленность, системность и последовательность в работе студентов, развивает у них навык завершать начатую работу.

Цель самостоятельной работы - подготовка современного компетентного специалиста и формирование способностей и навыков к непрерывному самообразованию и профессиональному совершенствованию.

Основные виды самостоятельной работы студентов:

- работа с основной и дополнительной литературой;
- изучение понятийного аппарата дисциплины;
- самостоятельное изучение отдельных вопросов по темам;
- подготовка докладов-презентаций;
- работа в библиотеке;
- посещение консультаций преподавателя данной дисциплины при затруднениях освоения материала;
- подготовка к зачету.

Работа с основной и дополнительной литературой

Изучение литературы следует начинать с нормативных и правовых актов по бухгалтерскому учету, учебников и учебных пособий, затем переходить к научным монографиям и материалам периодических изданий. Работа с литературой предусматривает конспектирование наиболее актуальных и познавательных материалов. Это не только мобилизует внимание, но и способствует более глубокому осмыслению материала, его лучшему запоминанию, а также позволяет студентам проводить систематизацию и сравнительный анализ изучаемой информации. Таким образом, конспектирование – одна из основных форм самостоятельного труда, которая требует от студента активно работать с нормативно-правовой, учебной литературой и не ограничиваться конспектом лекций.

Студент должен уметь самостоятельно ориентироваться в законодательстве РФ по бухгалтерскому учету, подбирать необходимую литературу для учебной и научной работы, уметь обращаться с предметными каталогами и библиографическим справочником библиотеки.

Изучение понятийного аппарата дисциплины

Изучение и осмысление экономических категорий требует проработки основной и дополнительной литературой, лекционного материала, выполнения практических заданий, изучение словарей, энциклопедий, справочников.

Индивидуальная самостоятельная работа студента направлена на овладение и грамотное применение экономической терминологии в области бухгалтерского учета.

Самостоятельное изучение тем дисциплины

Особое место отводится самостоятельной проработке студентами отдельных тем изучаемой дисциплины. Такой подход вырабатывает у студентов инициативу, стремление к увеличению объема знаний, умений и навыков, всестороннего овладения способами и приемами профессиональной деятельности.

Изучение вопросов определенной темы направлено на более глубокое усвоение основных составляющих бухгалтерского учета, понимание экономических процессов, происходящих в деятельности отдельных хозяйствующих субъектов и обществе, в целом.

Самостоятельная работа в библиотеке

Важным аспектом самостоятельной подготовки студентов является работа с библиотечным фондом.

Эта работа предполагает различные варианты повышения профессионального уровня студентов:

- а) получение книг для подробного изучения в течение семестра на научном абонементе;
- б) изучение книг, журналов, газет - в читальном зале;
- в) возможность поиска необходимого материала посредством электронного каталога;
- г) получение необходимых сведений об источниках информации у сотрудников библиотеки.

Посещение консультаций преподавателя дисциплины при затруднениях освоения материала

Перед консультацией по предмету следует составить список вопросов, требующих дополнительного разъяснения преподавателем на консультации. Далее выдается задание на самостоятельную работу по разъясненному вопросу.

Подготовка к зачету

Промежуточная аттестация студентов по дисциплине проходит в виде зачета. Условием успешного прохождения промежуточной аттестации является посещение занятий и систематическая работа студента в течение семестра. В этом случае подготовка к зачету является систематизацией всех полученных знаний по данной дисциплине.

Рекомендуется внимательно изучить перечень вопросов к зачету, а также использовать в процессе обучения программу, учебно-методический комплекс, другие методические материалы.

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

5. Фонд оценочных средств для промежуточной аттестации по дисциплине, включающий:

5.1. Описание шкал оценивания результатов обучения по дисциплине

Уровень сформированности компетенций (индикаторы достижения компетенций)	Шкала оценивания сформированности компетенций						
	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	Не зачтено		Зачтено				
Знания	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки.	Минимально допустимый уровень знаний. Допущено много негрубых ошибок.	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько незначительных ошибок	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок.	Уровень знаний в объеме, превышающем программу подготовки.
Умения	Отсутствие минимальных	При решении стандартных	Продемонстрированы	Продемонстрированы	Продемонстрированы	Продемонстрированы	Продемонстрированы

	умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	задач не продемонстрированы основные умения. Имели место грубые ошибки.	основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.	все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.	все основные умения, решены все основные задачи с отдельными несущественными недочётами, выполнены все задания в полном объеме.	все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
Навыки	Отсутствие владения материалом. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки.	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов.	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов.	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценки при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	Превосходно	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «превосходно»
	Отлично	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
	Очень хорошо	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
	Хорошо	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
	Удовлетворительно	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
Не зачтено	Неудовлетворительно	Хотя бы одна часть компетенции сформирована на уровне «неудовлетворительно», ни одна из компетенций не сформирована на уровне «плохо»
	Плохо	Хотя бы одна часть компетенции сформирована на уровне «плохо»

5.2. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения.

5.2.1 Контрольные вопросы к зачету

Вопрос	Код формируемой компетенции
1. Взаимодействие браузера и сервера Web	ПК-8
2. Типы данных в PHP. Константы PHP	ПК-8
3. Активные и пассивные серверы Web. Серверная активность в приложениях Web	ПК-8
4. Передача переменных по ссылке в PHP	ПК-8
5. Приложения WEB	ПК-8
6. Операции PHP. Проверка и установка типов переменных в	ПК-8

PHP. Проверка состояния переменных PHP	
7. Создание таблицы базы данных MySQL	ПК-8
8. Условный оператор в PHP	ПК-8
9. Типы данных MySQL	ПК-8
10. Цикл WHILE в PHP	ПК-8
11. Типы даты и времени в MySQL	ПК-8
12. Цикл FOR в PHP	ПК-8
13. Строковые Типы данных MySQL	ПК-8
14. Цикл FOREACH в PHP	ПК-8
15. Просмотр Данных в MySQL	ПК-8
16. Оператор BREAK в PHP. Оператор CONTINUE в PHP	ПК-8
17. Вставка данных в MySQL	ПК-8
18. Численно индексированные массивы в PHP. Многомерные массивы в PHP	ПК-8
19. Обновление записей в MySQL	ПК-8
20. Ассоциативные массивы в PHP	ПК-8
21. Изменение таблиц после создания в MySQL	ПК-8
22. Функции работы с массивами в PHP	ПК-8
23. Удаление записей из базы данных в MySQL	ПК-8
24. Работа со строками PHP: объединение, разделение, подготовка к хранению, сравнение, поиск и замена	ПК-8
25. Удаление таблиц в MySQL	ПК-8
26. Регулярные выражения в PHP	ПК-8
27. Удаление базы данных в MySQL	ПК-8
28. Директивы INCLUDE() и REQUIRE() в PHP	ПК-8
29. Извлечение данных из таблиц в MySQL	ПК-8
30. Сессии в PHP	ПК-8
31. Язык серверных сценариев PHP. Дескрипторы. Переменные	ПК-8
32. Команды работы с базой данных в PHP: подключение, выполнение запроса, получения результатов	ПК-8

5.2.2. Типовые задания для оценки компетенции ПК-8

Задания распределяются индивидуально, по вариантам.

Задание 1.

1. Создать HTML-страницу, которая при загрузке случайным образом выводит один из трех эпиграфов.

2. Написать функцию, которая проверяет, что в строке, переданной в качестве аргумента, соблюдается баланс открытых и закрытых скобок. При этом необходимо учесть, что закрывающая скобка не может появиться раньше соответствующей ей открывающей скобки.

Функция возвращает строку "O'key" при соблюдении такого баланса, и "Error" в противном случае.

3. Добавьте к системному объекту **Date** метод **isBusinessTime**. Ваш метод, должен возвращать true, если экземпляр Date, в контексте которой он вызван, задает рабочее время и false, в противном случае.

Рабочим временем считайте время с 8 до 17 часов во все дни, кроме субботы и воскресенья.

Напишите программу с тестами для демонстрации работы Вашего метода.

4. Продублируйте ссылку кнопкой. Т.е. у вас должна быть пара "ссылка-кнопка". При нажатии на кнопку должно происходить то же самое, как если бы пользователь нажал на ссылку.

Кнопка при этом должна быть универсальной. Т.е., если изменить ссылку (*например, поставить в ней другой URL*), все должно работать правильно. В описании кнопки менять ничего не нужно.

Разместите на своей странице несколько (не менее трех) пар ссылка-кнопка.

Задание 2.

Сверстайте три цветовые схемы web-страницы, используя CSS, различные цветовые стили дизайна и композиции.

Воспользовавшись поисковыми машинами, найдите в сети шаблон сайта в формате *.psd, и сверстайте его с применением нарезки.

Создать web-страницу с использованием CSS. На странице необходимо разместить изображения, текст, список, таблицу, форму.

Задание 3.

Создайте генератор простейших web-страниц средствами PHP. Подготовьте форму, в которую необходимо ввести заголовок страницы, цвет шрифта, цвет фона и текст. Сгенерируйте HTML-страницу с помощью этой формы.

Задание 4.

Напишите игру «Угадай число» средствами PHP. Пусть компьютер генерирует случайное число и предоставляет пользователю возможность его угадать, подсказывая, является ли его ответ слишком большим, маленьким или правильным. После того, как пользователь угадает значение, сообщить, сколько попыток для этого потребовалось.

Задание 5.

Написать сценарий на языке Javascript, позволяющий для изображения на web-странице менять ширину и высоту изображения, создавать рамку вокруг изображения, менять ее толщину и цвет, задавать альтернативный текст.

Задание 6.

Создайте базу данных, содержащую следующую информацию: ФИО работника, должность, подразделение, з/п. Разработайте web-страницу, позволяющую добавлять данные о новых работниках, а также web-страницу выводющую содержимое таблицы с сортировкой по зарплате по убыванию, при условии, что зарплата выше 10 т.р.

Задание 7.

Создайте базу данных, содержащую следующую информацию: дата, заголовок новости, текст новости. Разработайте: web-страницу, позволяющую добавлять новости; web-страницу на которую выводится заголовки 5 последних новостей.

Задание 8.

Создайте базу данных, содержащую следующую информацию: ФИО пользователя, его возраст, место работы или учебы, логин, пароль. Напишите программу, которая запрашивает логин и пароль ищет в базе данных этого пользователя. При положительном результате на web-страницу должна выводиться следующая информация: фамилия и имя пользователя. При несовпадении логина/пароля выводится сообщение, что пользователь не найден либо пароль не правильный.

Задание 9.

Найдите в Интернет два web-сайта для анализа функций и информационной архитектуры. Сайт должен содержать не менее десяти разделов.

В документе Word сформулируйте:

цели каждого из web-сайтов;

функции каждого из web-сайтов;

предполагаемую аудиторию каждого из web-сайтов.

Создайте логическую структуру для каждого из web-сайтов в виде блок-схемы. Для этого воспользуйтесь графическим редактором, рекомендуется CorelDRAW. Поместите созданные схемы в документе Word.

Создайте предполагаемую физическую структуру web-сайта в виде блок-схемы. Присваивайте названия папкам и файлам согласно правилам, изученным в теоретическом блоке.

Задание 10.

Составить три блочные цветовые схемы, используя различные цветовые стили дизайна и различные композиции web-страницы.

Задание 11.

Найдите в интернете понравившийся Вам шаблон сайта в формате psd, и сверстайте его с помощью нарезки.

Задание 12.

Создать flash-баннер для web-страницы на свободную тему.

Задание 13.

Создать web-страницу в Dreamweaver с использованием CSS. На странице необходимо разместить изображения, текст, список, таблицу, форму, flash-баннер.

Задание 14.

Написать сценарий на языке Javascript, позволяющий для изображения на web-странице менять ширину и высоту изображения, создавать рамку вокруг изображения, менять ее толщину и цвет, задавать альтернативный текст.

Задание 15.

Написать сценарий на JavaScript, который реализует обмен рисунков на web-странице. Пусть на web-странице расположено четыре изображения, пронумерованных от 1 до 4. В текстовых полях указываются номера рисунков, которые необходимо поменять местами. Требуется, чтобы после нажатия на кнопку «Поменять местами» изображения переместились на нужные места.

Задание 16.

Написать сценарий на JavaScript, который рассчитывает нагрузку преподавателя в часах. В анкете задать поля, в которые вводятся количество часов, отведенных на чтение лекций и проведение практических занятий, а также число студентов. Если по предмету читаются лекции, дополнительно планируется нагрузка: 10% времени от лекционных часов отводится на консультации, для приема экзамена планируется по 30 минут на человека. Если по предмету проводятся практические занятия, предусмотрена контрольная работа из расчета 15 минут на человека, зачет – из расчета 20 минут на человека.

Задание 17.

Написать сценарий на JavaScript, который позволяет продемонстрировать, как будет меняться таблица и ее ячейки при изменении значений параметров border, cellpadding, cellspacing.

Задание 18.

Приводятся данные о закупках пяти наименований товаров: цена за единицу и количество приобретаемых экземпляров. Напишите сценарий на JavaScript, определяющий сумму, затраченную на приобретенные товары. Определите, имеются ли товары, на которые потра-

чена одинаковая сумма, и сколько их. Постройте диаграмму, отражающую суммы, затраченные на приобретение разных товаров.

Задание 19.

Средствами РНР создайте генератор web-страниц. Подготовьте форму, в которую бы вводился заголовок страницы, цвет фона, цвет шрифта и текст страницы. Создайте HTML-страницу с помощью этой формы.

5.2.2. Тестовые задания для оценки компетенций ПК-8

Задание № 1

Вопрос:

Текст, заключенный в теги отобразится

Выберите один из 3 вариантов ответа:

- 1) Жирным шрифтом
- 2) Подчеркнутым
- 3) Курсивом

Задание № 2

Вопрос:

Установите соответствие между атрибутами тега <table> их значениями:

- 1 height
- 2 cellpadding
- 3 valign
- 4 bgcolor
- 5 width

Укажите порядок следования всех 5 вариантов ответа:

- ___ определяет ширину таблицы
- ___ определяет расстояние (в пикселах) между границами соседних ячеек.
- ___ должен определять способ вертикального выравнивания для содержимого таблицы.
- ___ определяет высоту таблицы
- ___ определяет цвет фона ячеек таблицы

Задание № 3

Вопрос:

Определите вид гиперссылки?

<http://videouroki.net/informatika.php>

Выберите один из 3 вариантов ответа:

- 1) Прямая
- 2) Относительная
- 3) Абсолютная

Задание № 4

Вопрос:

Какие атрибуты мы можем использовать для тега

Выберите несколько из 5 вариантов ответа:

- 1) height
- 2) align
- 3) border
- 4) color
- 5) width

Задание № 5

Вопрос:

Какое расширение нужно использовать для сохранения интернет-страницы, набранной в программе Блокнот?

Выберите один из 5 вариантов ответа:

- 1) .doc
- 2) .gif
- 3) .php
- 4) .css
- 5) .html

Задание № 6

Вопрос:

Какой тип списка вы видите перед собой?

``

`элемент 1`

`элемент 2`

`элемент 3`

``

Выберите один из 3 вариантов ответа:

- 1) Маркированный
- 2) Нумерованный
- 3) Список определений

Задание № 7

Вопрос:

Атрибуты тега ...

Выберите один из 4 вариантов ответа:

- 1) Нужно писать сразу после закрытия тега
- 2) Нужно писать в закрывающем теге
- 3) Можно использовать вместо самого тега
- 4) Изменяют и уточняют его действия

Задание № 8

Вопрос:

Для форматирования текста в html обычно рекомендуют использовать ...

Выберите один из 3 вариантов ответа:

- 1) тег ``
- 2) программы Microsoft Word
- 3) CSS стили

Задание № 9

Вопрос:

Какое основное требование предъявляется к изображениям размещаемым на веб-странице?

Выберите один из 4 вариантов ответа:

- 1) Изображение должно быть оптимизировано
- 2) Изображение должно быть ярким
- 3) Высокое качество изображения
- 4) Изображение должно быть красивым

Задание № 10

Вопрос:

Установите соответствие

- 1 ``
- 2 ``

- 3 font-family
- 4 font-size
- 5 color

Укажите порядок следования всех 5 вариантов ответа:

- ☐ Текст, заключенный в эти теги будет отображен жирным
- ☐ Определяет шрифт
- ☐ Определяет размер шрифта
- ☐ Текст заключенный в эти теги будет отображен курсивом
- ☐ Определяет цвет текста

Задание № 11

Вопрос:

Установите соответствие

- 1 высота изображения
- 2 ширина изображения
- 3 текст который будет виден если отображение изображений отключено в браузере
- 4 смещение относительно текста
- 5 рамка и её толщина

Укажите порядок следования всех 5 вариантов ответа:

- ☐ height="70"
- ☐ alt="Текст"
- ☐ border="5"
- ☐ width="100"
- ☐ align="left"

Задание № 12

Вопрос:

Что связывают между собой гиперссылки?

Выберите один из 4 вариантов ответа:

- 1) Теги
- 2) Атрибут и его значение
- 3) Веб-страницы
- 4) Теги и атрибуты

Задание № 13

Вопрос:

Сколько строк и столбцов будет в этой таблице?

```
<table>
<tr>
<td>
Привет
</td>
</tr>
</table>
```

Выберите один из 4 вариантов ответа:

- 1) 2 строка 1 столбец
- 2) 2 строки 2 столбца
- 3) 1 строка 1 столбец
- 4) 1 строка 2 столбца

Задание № 14

Вопрос:

Укажите неправильный синтаксис ссылки

Выберите один из 3 вариантов ответа:

- 1) Видеоуроки в сети
- 2) Видеоуроки в сети
- 3) Видеоуроки в сети

Задание № 15

Вопрос:

Какие виды гиперссылок существуют в html?

Выберите несколько из 5 вариантов ответа:

- 1) Синтаксические
- 2) Относительные
- 3) Уникальные
- 4) Абсолютные
- 5) Логические

Задание № 16

Вопрос:

Значение атрибутов тега заключается в ...

Выберите один из 4 вариантов ответа:

- 1) скобки
- 2) кавычки
- 3) теги
- 4) атрибуты

Задание № 17

Вопрос:

Какой тип списка определяет тег ?

Выберите один из 3 вариантов ответа:

- 1) Нумерованный
- 2) Маркированный
- 3) Список определений

Задание № 18

Вопрос:

Установите соответствие

- 1
- 2
- 3

Укажите порядок следования всех 3 вариантов ответа:

- ___ Определяет неупорядоченный список
- ___ Определяет упорядоченный список
- ___ Определяет элемент списка

Задание № 19

Вопрос:

Текст, заключенный в теги отобразится

Выберите один из 3 вариантов ответа:

- 1) Курсивом
- 2) Подчеркнутым
- 3) Жирным шрифтом

Задание № 20

Вопрос:

Укажите правильный формат задания атрибутов

Выберите один из 4 вариантов ответа:

- 1) <имя_атрибута имя_тега="значение">
- 2) <значение имя_атрибута="имя_тега">
- 3) <имя_тега значение="имя_атрибута">
- 4) <имя_тега имя_атрибута="значение">

6. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебное пособие для академического бакалавриата / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2018. — 218 с. — (Университеты России). — ISBN 978-5-534-00515-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: (доступен в ЭБС «Юрайт», режим доступа: <https://urait.ru/bcode/413954>)

2. Мартишин С.А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для применения проектирования информационных систем: Учебное пособие / Мартишин С.А., Симонов В.Л., Храпченко М.В. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2017. - 368 с. URL (доступно в ЭБС «Znanium.com», режим доступа: <http://znanium.com/catalog/product/556449>)

3. Лисьев, Г. А. Программное обеспечение компьютерных сетей и web-серверов : учеб. пособие / Г.А. Лисьев, П.Ю. Романов, Ю.И. Аскерко. — Москва : ИНФРА-М, 2019. — 145 с. — (Высшее образование: Бакалавриат). — www.dx.doi.org/10.12737/textbook_5a93ba6860adc5.11807424. - ISBN 978-5-16-013565-6. - Текст : электронный. - URL: (доступно в ЭБС «Знаниум», режим доступа: <https://znanium.com/catalog/product/1002586>

б) дополнительная литература:

1. Волкова, В. Н. Теория информационных процессов и систем : учебник и практикум для академического бакалавриата / В. Н. Волкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2018. — 432 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-05621-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: (доступно в ЭБС «Юрайт», режим доступа: <https://urait.ru/bcode/413050>)

2. Гвоздева, В. А. Информатика, автоматизированные информационные технологии и системы : учебник / В. А. Гвоздева. — Москва : ФОРУМ : ИНФРА-М, 2020. — 542 с. - ISBN 978-5-8199-0877-8. - Текст : электронный. - URL: (доступно в ЭБС «Знаниум», режим доступа: <https://znanium.com/catalog/product/1066785>)

3. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 432 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-07604-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: (доступно в ЭБС «Юрайт», режим доступа: <https://biblio-online.ru/bcode/436514>

4. Стасышин, В. М. Базы данных: технологии доступа : учебное пособие для академического бакалавриата / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2018. — 178 с. — (Высшее образование). — ISBN 978-5-534-03405-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: (доступно в ЭБС «Юрайт», режим доступа: <https://biblio-online.ru/bcode/415342>)

5. Стружкин, Н. П. Базы данных: проектирование : учебник для академического бакалавриата / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2019. — 477 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-00229-4. — Текст : электронный // ЭБС

Юрайт [сайт]. — URL: (доступно в ЭБС «Юрайт», режим доступа <https://bibli-online.ru/bcode/432177>)

в) программное обеспечение и Интернет-ресурсы

1. Литература для студента <http://www.libsib.ru/etika/etika-delovogo-obscheniya/vse-stranitsi>
2. Федеральный портал «Российское образование» <http://www.edu.ru/>
3. Федеральное хранилище «Единая коллекция цифровых образовательных ресурсов» <http://school-collection.edu.ru/>
4. <http://dev.mysql.com/>
5. <http://html.manual.ru/>
6. <http://php.ru/>
7. <http://ru.html.net/>
8. <http://ru.php.net/manual/ru/>
9. <http://webmascon.com/>
10. <http://www.citforum.ru/database/mysql.shtml>
11. <http://www.citforum.ru/internet/php.shtml>
12. <http://www.mysql.ru/>
13. <http://www.php.net/>
14. <http://www.phpclub.ru/>
15. <http://www.phpinside.ru/>
16. Операционная система Microsoft Windows
17. Пакет прикладных программ Microsoft Office
18. Правовая система «Консультант плюс»
19. Правовая система «Гарант».
20. Интернет браузеры (Mozilla Firefox, Google Chrome)
21. Комплект разработчика приложений Java Development Kit

7. Материально-техническое обеспечение дисциплины

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения: компьютерная техника с подключением к сети «Интернет», экран, проектор для вывода мультимедиа материалов на экран, динамики для воспроизведения звука, доска.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Специальные условия организации обучения по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья

Организация обучения по дисциплине инвалидов и лиц с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья при наличии таких обучающихся путем создания специальных условий для получения образования.

Профессорско-преподавательский состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья, индивидуальными программами реабилитации инвалидов (при наличии).

В соответствии с Методическими рекомендациями по организации образовательного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса, утв. Минобрнауки РФ 08.04.2014 АК-44/05вн при изучении дисциплины предполагается использование социально-активных и рефлексивных методов обучения, техноло-

гий социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе.

При освоении дисциплины используются различные сочетания видов учебной работы с методами и формами активизации познавательной деятельности обучающихся для достижения запланированных результатов обучения и формирования компетенций. Форма проведения промежуточной аттестации для обучающихся-инвалидов и лиц с ограниченными возможностями здоровья устанавливается с учетом индивидуальных психофизиологических особенностей. По личной просьбе обучающегося с ограниченными возможностями здоровья, изложенной в форме письменного заявления, по дисциплине предусматриваются:

- замена устного ответа на письменный ответ при сдаче зачёта;
- увеличение продолжительности времени на подготовку к ответу на зачёте;
- при подведении результатов промежуточной аттестации студентов выставляется максимальное количество баллов за посещаемость аудиторных занятий.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки 09.03.03 «Прикладная информатика».

Программа одобрена на заседании учебно-методической комиссии Павловского филиала ННГУ протокол № 3 от 24.05.2023.