

АНО ВО «Межрегиональный открытый социальный институт»

УТВЕРЖДЕНО

на заседании Совета факультета  
экономики и информационной безопасности  
Протокол заседания Совета факультета  
№ 6 «18» июль 2018г.  
Декан факультета экономики и  
информационной безопасности  
Т.А. Сафина

ОДОБРЕНО

на заседании кафедры информационной  
безопасности  
Протокол заседания кафедры  
№ 10 «30» май 2018г.  
Зав. кафедрой информационной  
безопасности Гусак Т.М. Гусакова

РАБОЧАЯ ПРОГРАММА

по дисциплине Программирование РНР  
(наименование)  
образовательная программа 38.03.05 Бизнес-информатика, «Электронный бизнес»  
форма обучения очная, заочная

ПРОГРАММА РАЗРАБОТАНА

кр доцент, канд. техн. наук, доцент  
Кречетов А.А.  
(должность, Ф. И. О., ученая  
степень, звание автора(ов)  
программы)

Йошкар-Ола, 2018

## Содержание

1. Пояснительная записка.....	3
2. Структура и содержания дисциплины .....	8
3. Оценочные средства и методические рекомендации по проведению промежуточной аттестации .....	22
4. Учебно-методическое и информационное обеспечение дисциплины.....	41
5. Материально-техническое обеспечение дисциплины.....	42
6. Методические указания для обучающихся по освоению дисциплины.....	44

## 1. Пояснительная записка

**Цель изучения дисциплины:** обеспечение знаниями основ программирования по курсу «Программирование PHP» на основе как теоретического, так и практического материала.

**Место дисциплины в учебном плане:**

Предлагаемый курс относится к обязательным дисциплинам вариативной части образовательной программы 38.03.05 Бизнес-информатика. Электронный бизнес.

**Дисциплина «Программирование PHP» обеспечивает овладение следующими компетенциями:**

продолжает формирование профессиональной компетенции:

управление контентом предприятия и Интернет-ресурсов, процессами создания и использования информационных сервисов (контент-сервисов) (ПК-6) – 2/2 этап;

умение разрабатывать контент и ИТ-сервисы предприятия и интернет-ресурсов (ПК-16) – 3/2 этап.

### Этапы формирования компетенции (очная форма обучения)

Код компетенции	Формулировка компетенции	Учебная дисциплина	Семестр	Этап
ПК-6	управление контентом предприятия и Интернет-ресурсов, процессами создания и использования информационных сервисов (контент-сервисов)	Мультимедиа технологии	5	1
		Разработка и создание Web-ресурсов		
		Web-дизайн		
		Компьютерная графика		
		Учебная практика (практика по получению первичных профессиональных умений и навыков)		
		Оптимизация и продвижение Web-сайтов	6	2
		Программирование PHP		
		Профессиональная html-верстка		
		Введение в технологии создания интернет-узлов		
		Введение в интернет-рекламу	7	3
		Ремаркетинг		
		Разработка IT-продуктов для портативных устройств	8	4
Разработка				

		приложений для операционной системы Android		
		Разработка облачных бизнес-приложений		
		Разработка бизнес-приложений на основе открытых систем управления базами данных		
		Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты		
ПК-16	умение разрабатывать контент и ИТ-сервисы предприятия и интернет-ресурсов	Объектно-ориентированный анализ и программирование	3,4	1
		Мультимедиа технологии	5	2
		Разработка и создание Web-ресурсов		
		Web-дизайн		
		Компьютерная графика		
		Оптимизация и продвижение Web-сайтов	6	3
		Программирование PHP		
		Профессиональная html-верстка		
		Введение в технологии создания интернет-узлов		
		Производственная практика (практика по получению профессиональных умений и опыта профессиональной деятельности)	7	4
		Программирование 1С		
		Производственная практика (преддипломная)	8	5
		Защита выпускной квалификационной		

		работы, включая подготовку к процедуре защиты и процедуру защиты		
--	--	--	--	--

**Этапы формирования компетенции  
(заочная форма обучения)**

Код компетенции	Формулировка компетенции	Учебная дисциплина	Семестр	Этап
ПК-6	управление контентом предприятия и Интернет-ресурсов, процессами создания и использования информационных сервисов (контент-сервисов)	Мультимедиа технологии	5	1
		Разработка и создание Web-ресурсов	6	2
		Учебная практика (практика по получению первичных профессиональных умений и навыков)		
		Программирование PHP		
		Оптимизация и продвижение Web-сайтов	7	3
		Web-дизайн	8	4
		Компьютерная графика		
		Профессиональная html-верстка		
		Введение в технологии создания интернет-узлов		
		Введение в интернет-рекламу		
		Ремаркетинг		
		Разработка IT-продуктов для портативных устройств		
		Разработка приложений для операционной системы Android		
		Разработка облачных бизнес-приложений		
Разработка бизнес-приложений на основе открытых систем управления базами данных				

		Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты	10	5
ПК-16	умение разрабатывать контент и ИТ-сервисы предприятия и интернет-ресурсов	Мультимедиа технологии	5	1
		Объектно-ориентированный анализ и программирование	6	2
		Разработка и создание Web-ресурсов		
		Программирование PHP	7	3
		Оптимизация и продвижение Web-сайтов		
		Web-дизайн		
		Компьютерная графика		
		Профессиональная html-верстка		
		Введение в технологии создания интернет-узлов		
		Производственная практика (практика по получению профессиональных умений и опыта профессиональной деятельности)		
		Программирование 1С	9	5
		Производственная практика (преддипломная)	10	6
		Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты		

**В результате освоения дисциплины обучающийся должен:**

ПК-6	<b>Знать:</b> <ul style="list-style-type: none"> <li>– синтаксис языка программирования PHP;</li> <li>– методы и функции работы с файлами на языке PHP;</li> <li>– базовые средства управляемого языка программирования PHP и технологию</li> </ul>
------	---

	<p>создания программ для решения стандартных задач профессиональной деятельности.</p> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>– делать простейшие автоматизированные арифметические расчеты;</li> <li>– обрабатывать файлы и работать с информацией в файлах;</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>– основами программирования на языке PHP (составление, отладка и тестирование программ; разработка и создание веб-сайта)</li> </ul>
ПК-16	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>– основные операторы и конструкции;</li> <li>– принципы построения реляционных баз данных для электронного бизнеса.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>– строить структуру реляционной базы данных для электронного бизнеса;</li> <li>– работать с таблицами базы данных;</li> <li>– создавать приложения на управляемом языке программирования PHP на основе информационной и библиографической культуры</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>– теоретическими основами программирования и практическими навыками создания приложений на языке программирования PHP с учетом основных требований информационной безопасности.</li> <li>– практическими приемами программирования на языке PHP.</li> </ul>

**Формы текущего контроля успеваемости студентов:** устный опрос, практические задания.

**Формы промежуточной аттестации:** экзамен.

## 2. Структура и содержания дисциплины

Трудоемкость 4 зачетные единицы, 144 часа, из них:

очная форма обучения: 18 лекционных, 36 практических занятий, 72 часа самостоятельной работы, 18 часов контроль.

заочная форма обучения: 8 лекционных, 8 практических, 119 часов самостоятельной работы, 9 часов контроль

### 2.1. Тематический план учебной дисциплины (очная форма обучения)

№ п/п раздела	Наименование разделов и тем	Количество часов по учебному плану				
		Всего	Виды учебной работы			
			Аудиторная работа			Самостоятельная работа
		Лекции	Практические (семинарские) занятия	Лабораторные занятия		
1	2	3	4	5	6	7
Раздел 1. Основы php						
1	Введение в php	19	4	6	-	9
2	Строковые функции	17	2	6	-	9
3	Массивы	15	2	4	-	9
4	Функции	15	2	4	-	9
Раздел 2. Работы с файлами в php						
5	Функции работы с файлами	15	2	4	-	9
Раздел 3. Основы работы с базами данными						
6	Реляционные базы данных	15	2	4	-	9
7	Введение в mysql.	15	2	4	-	9
8	Команды SQL.	15	2	4	-	9
	<b>Итого по курсу:</b>	<b>126</b>	<b>18</b>	<b>36</b>	<b>-</b>	<b>72</b>



(заочная форма обучения)

№ п/п раздела	Наименование разделов и тем	Количество часов по учебному плану					
		Всего	Виды учебной работы				
			Аудиторная работа				Самостоятельная работа
			Лекции	Практические (семинарские) занятия	Лабораторные занятия		
1	2	3	4	5			
Раздел 1. Основы php							
1	Введение в php	20	2	2	-	16	
2	Строковые функции	20	2	2	-	16	
3	Массивы	20	2	2	-	16	
4	Функции	19	2	2	-	15	
Раздел 2. Работы с файлами в php							
5	Функции работы с файлами	14	-	-	-	14	
Раздел 3. Основы работы с базами данными							
6	Реляционные базы данных	14	-	-	-	14	
7	Введение в mysql.	14	-	-	-	14	
8	Команды SQL.	14	-	-	-	14	
	<b>Итого по курсу:</b>	<b>135</b>	<b>8</b>	<b>8</b>	<b>-</b>	<b>119</b>	

## 2.2. Тематический план лекций:

№ п/п раздела	Наименование разделов и тем	Количество часов
1	2	3
Раздел 1. Основы php		
1	Введение в php	4/2
2	Строковые функции	2/2
3	Массивы	2/2
4	Функции	2/2
Раздел 2. Работы с файлами в php		
5	Функции работы с файлами	2/-
Раздел 3. Основы работы с базами данными		
6	Реляционные базы данных	2/-
7	Введение в mysql.	2/-
8	Команды SQL.	2/-
	<b>Итого по курсу</b>	<b>18/8</b>

### Содержание лекционных занятий

#### Раздел 1. Основы php

##### Тема 1. Введение в php.

**План:**

1. Основы php. Операторы выбора: if...else, Переключатель switch.
2. Операторы цикла: цикл с предусловием, цикл с постусловием, итерационный цикл, итерационный цикл foreach.

##### Тема 2. Строковые функции.

**План:**

1. Функции поиска в тексте.
2. Функции форматного вывода.
3. Функции объединения, разъединения строк.
4. Функции работы с URL. Функции преобразования регистра.

##### Тема 3. Массивы.

**План:**

1. Инициализация массивов.

2. Цикл foreach для обхода массивов.
3. Многомерные массивы.
4. Сортировка массивов.

#### **Тема 4. Функции.**

##### **План:**

1. Определения и вызовы функций.
2. Область видимости переменных.
3. Рекурсия.

#### **Раздел 2. работы с файлами в php Тема 5. Функции работы с файлами.**

##### **План:**

1. Открытие файлов.
2. Отображение файлов.
3. Закрытие файлов.
4. Чтение из файлов и запись в файлы.
5. Атрибуты файлов.

#### **Раздел 3. Основы работы с базами данными Тема 6. Реляционные базы данных.**

##### **План:**

1. Модели данных.
2. Основы реляционных баз данных.

#### **Тема 7. Введение в mysql.**

##### **План:**

1. История sql.
2. Обзор sql.

#### **Тема 8. Команды SQL.**

##### **План:**

1. Выборка данных.
2. Добавление записи.
3. Редактирование записи.
4. Удаление записи.
5. Группировка.

#### **Основная литература**

Савельева, Н.В. Язык программирования PHP / Н.В. Савельева. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 330 с. : схем., ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428975> (Электронная библиотечная система «Университетская библиотека ONLINE»)

#### **Дополнительная литература**

Малашкевич, В.Б. Интернет-программирование: лабораторный практикум / В.Б. Малашкевич; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2017. - 96 с.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=476400> (Электронная библиотечная система «Университетская библиотека ONLINE»)

## **Информационно-справочные системы, профессиональные базы данных и интернет-ресурсы**

1. Сайт Security Lab [Электронный ресурс]. – Режим доступа: <http://www.securitylab.ru/> (новости, экспертные статьи, софты, форум, раздел, где оперативно публикуется информация об уязвимостях, а также даются конкретные рекомендации по их устранению)

2. Сайт Threatpost [Электронный ресурс]. – Режим доступа: <https://threatpost.com/> (содержит новости об информационной безопасности от Kaspersky Lab)

3. Сайт Anti-Malware [Электронный ресурс]. – Режим доступа: <https://www.anti-malware.ru/> (содержит сравнительные тесты антивирусов, аналитические статьи)

4. Сайт Научного журнал «Вопросы кибербезопасности» [Электронный ресурс]. – Режим доступа: <http://cybertrust.com/> (содержит статьи российских и иностранных ученых по кибербезопасности, безопасности приложений, технической защите информации, аудиту безопасности систем и программного кода, тестированию, анализу защищенности и оценке соответствия ПО требованиям безопасности информации)

5. Профессиональная база данных по бизнес-информатике [Электронный ресурс]. - Режим доступа: [http://dorlov.blogspot.ru/p/blog-page\\_3151.html](http://dorlov.blogspot.ru/p/blog-page_3151.html)

6. СПС «Консультант Плюс», СПС «Гарант» (договор о сотрудничестве от 23.09.2013 г. с ЗАО «Компьютерные технологии» (ПС Гарант)), регистрационный лист зарегистрированного пользователя ЭПС «Система ГАРАНТ» от 16.02.2012 г. №12-40272-000944; договоры с ООО «КонсультантПлюс Марий Эл» №2017-СВ-4 от 28.12.2016 г

### 2.3. Тематический план практических (семинарских) занятий

№ п/п раздела	Наименование разделов и тем	Количество часов
1	2	3
Раздел 1. Основы php		
1	Введение в php	6/2
2	Строковые функции	6/2
3	Массивы	4/2
4	Функции	4/2
Раздел 2. Работы с файлами в php		
5	Функции работы с файлами	4/-
Раздел 3. Основы работы с базами данными		
6	Реляционные базы данных	4/-
7	Введение в mysql.	4/-
8	Команды SQL.	4/-
	<b>Итого по курсу</b>	<b>36/8</b>

#### Содержание практических занятий

##### Раздел 1. Основы php

##### Тема 1. Введение в php.

###### План

1. Типы данных.
2. Переменные.
3. Типы данных.
4. Константы.
5. Комментарии.
6. Операторы.

###### Практические задания

1. Написать скрипт, который выполнял бы элементарные арифметические действия (сложение, умножение, вычитание, деление) и вывод результата на экран.

###### Решение:

```
<form method="post" name="form1">
```

```
Действие:<input type="radio" name="arifm" value="1" checked> +
```

```

<input type="radio" name="arifm" value="2"> -
<input type="radio" name="arifm" value="3"> *
<input type="radio" name="arifm" value="4"> / <br>
Введите a:<input type="text" name="a"><br>
Введите b:<input type="text" name="b"><br>
<input type="submit" name="button" value="Расчет">
</form>
<?
if (isset($_POST['button']))
{
switch($_POST['number'])
{
case 1:
{
$result=$_POST['a']+$_POST['b'];
break;
}
case 2:
{
$result=$_POST['a']-$_POST['b'];
break;
}
case 3:
{
$result=$_POST['a']*$_POST['b'];
break;
}
case 4:
{
if ($_POST['b']!=0) {$result=$_POST['a']/$_POST['b'];}
else {$result="Знаменатель не должен равняться нулю!";}
break;
}
}
}
echo 'Результат: '.$result;
}
?>

```

2. Организовать проверку двух введенных чисел. Если a меньше b, то вывести число a меньше b, и если больше, то соответственно. Пример результата "Число 4 меньше 7"

3. Написать скрипт вычисления корней квадратного уравнения

## Тема 2. Строковые функции.

### План:

2. Строковый тип данных.
3. Основные операции со строками
4. Функции поиска в тексте.
5. Функции форматного вывода.
6. Функции работы с URL.
7. Функции преобразования регистра.

## Практические задания

1. Найти количество вхождений фрагмента в строку.

**Решение.**

```
<?
$str = "dfhd@ffs@dfskfk@asas";
$substr_count = substr_count($str,"@");
echo ($substr_count);
?>
```

2. Организовать поиск. Слова должны обрезаться на 2 символа, чтобы искалось к примеру не «Тюменский», а «Тюменск», т.к. в тексте может встречать Тюменскую область, Тюменский район.

### Тема 3. Массивы.

**План:**

1. Понятие массив.
2. Способы объявления массива.
3. Обход элементов массив.
4. Реализация многомерного массива.
5. Поиск значения в массиве.

#### Практические задания

1. Обойти все элементы массива и вывести их на экран.

**Решение.**

```
<?
$array = array("Мышь", "Клавиатура", "Монитор","ИБП");
foreach($array as $index => $val)
{
    echo("$index -> $val <br>");
}
?>
```

#### Практические задания

1. Организовать поиск заданного значения в массиве. Вывести весь массив, причем искомый элемент должен быть выделен жирным.

### Тема 4. Функции.

**План:**

1. Понятие функции.
2. Входные параметры функции.
3. Вызов функции.
4. Рекурсия.

#### Практические задания

1. Написать функцию вычисления корней квадратного уравнения. Соответственно с входными параметрами **a**, **b**, **c**.

### Раздел 2. работы с файлами в php Тема 5. Функции работы с фалами.

**План:**

1. Открытие файлов.
2. Отображение файлов.
3. Закрытие файлов.
4. Чтение из файлов и запись в файлы.
5. Копирование, переименование и удаление файлов.

## 6. Атрибуты файлов.

### Практические задания

1. Вывести содержимое файла на экран.

**Решение.**

```
<?
$file = fopen("c:/www/html/pavlovo.jpg","rb");
if(!file)
{ echo("Ошибка открытия файла"); }
else
{ fpassthru($file); }
?>
```

2. Записать в файл текстовую строку.

**Решение.**

```
<?
$file = fopen ("file.txt", "r+");
$str = "Hello, world!";
if( !$file )
{ echo("Ошибка открытия файла");}
else
{ fputs ( $file, $str); }
fclose ($file);
?>
```

3. Написать скрипт ведения адресной книги, где хранится ФИО человека, номер телефона и его адрес. Все данные хранить в текстовом файле по шаблону:

Иванов И. И||33-33-33||Моторостроителей 33 кв .4

Петров П. П.||35-35-35||Энергетиков 123 кв .77

и т.д.

## Раздел 3. Основы работы с базами данными

### Тема 6. Реляционные базы данных

**План:**

1. Понятие модели базы данных.
2. Понятие базы данных.
3. Перечислить модели БД и краткое описание каждой.
4. Реляционная модель базы данных. Основные понятия, описывающие реляционную модель базы данных.
5. Понятие ключа в базах данных.

### Тема 7. Введение в mysql.

**План:**

1. История появления sql.
2. Основные возможности sql.
3. Преимущества и недостатки sql.
4. Типы данных в sql.

### Тема 8. Команды SQL.

**План:**

1. Выборка данных. Выборка данных с условием. Что является результатом выборки.



2. Группировка.
3. Сортировка результата выборки.
4. Добавление записи.
5. Редактирование записи.
6. Удаление записи.

### Практические задания

1. Вывести содержимое таблицы clients с сортировкой по зарплате по убыванию, при условии, что зарплата выше 10 т.р.

**Решение.**

```
<?
$result=mysql_query("SELECT * FROM `clients` WHERE `zp`>'10000' ORDER BY
`zp` DESC ");
$count=mysql_num_rows($result);
$value=mysql_fetch_array($result);
for ($i=1; $i<=$count; $i++)
{
echo $value['fio'].'| | '.$value['zp'].'<br>';
$value=mysql_fetch_array($result);
}
mysql_free_result($result);
?>
```

2. Написать скрипт новостей. Сортировка новостей по дате. Вывод последних пяти новостей (SELECT.....LIMIT 5), Добавление новости, редактирование новости, удаление новости через web интерфейс. Поиск новости.

### Основная литература

Савельева, Н.В. Язык программирования PHP / Н.В. Савельева. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 330 с. : схем., ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428975> (Электронная библиотечная система «Университетская библиотека ONLINE»)

### Дополнительная литература

Малашкевич, В.Б. Интернет-программирование: лабораторный практикум / В.Б. Малашкевич; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2017. - 96 с.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=476400> (Электронная библиотечная система «Университетская библиотека ONLINE»)

### Информационно-справочные системы, профессиональные базы данных и интернет-ресурсы

1. Сайт Security Lab [Электронный ресурс]. – Режим доступа: <http://www.securitylab.ru/> (новости, экспертные статьи, софты, форум, раздел, где оперативно публикуется информация об уязвимостях, а также даются конкретные рекомендации по их устранению)
2. Сайт Threatpost [Электронный ресурс]. – Режим доступа: <https://threatpost.com/> (содержит новости об информационной безопасности от Kaspersky Lab)
3. Сайт Anti-Malware [Электронный ресурс]. – Режим доступа: <https://www.anti-malware.ru/> (содержит сравнительные тесты антивирусов, аналитические статьи)

4. Сайт Научного журнал «Вопросы кибербезопасности» [Электронный ресурс]. – Режим доступа: <http://cybergnus.com/> (содержит статьи российских и иностранных ученых по кибербезопасности, безопасности приложений, технической защите информации, аудиту безопасности систем и программного кода, тестированию, анализу защищенности и оценке соответствия ПО требованиям безопасности информации)

5. Профессиональная база данных по бизнес-информатике [Электронный ресурс]. - Режим доступа: [http://dorlov.blogspot.ru/p/blog-page\\_3151.html](http://dorlov.blogspot.ru/p/blog-page_3151.html)

6. СПС «Консультант Плюс», СПС «Гарант» (договор о сотрудничестве от 23.09.2013 г. с ЗАО «Компьютерные технологии» (ПС Гарант)), регистрационный лист зарегистрированного пользователя ЭПС «Система ГАРАНТ» от 16.02.2012 г. №12-40272-000944; договоры с ООО «КонсультантПлюс Марий Эл» №2017-СВ-4 от 28.12.2016 г

## 2.4. Тематический план самостоятельной работы

№ п/п раздела	Наименование разделов и тем	Количество часов
1	2	3
Раздел 1. Основы php		
1	Введение в php	9/16
2	Строковые функции	9/16
3	Массивы	9/16
4	Функции	9/15
Раздел 2. Работы с файлами в php		
5	Функции работы с файлами	9/14
Раздел 3. Основы работы с базами данными		
6	Реляционные базы данных	9/14
7	Введение в mysql.	9/14
8	Команды SQL.	9/14
	<b>Итого по курсу</b>	<b>72/119</b>

### Содержание самостоятельной работы

#### Тема 1. Введение в php

1. Реализация шаблонов средствами PHP.
2. Безопасность сайта электронной коммерции.

#### Тема 2. Строковые функции

1. Реализация аутентификации средствами PHP и MySQL.
2. Реализация безопасных транзакций средствами PHP и MySQL.

#### Тема 3. Массивы

1. Генерация изображений средствами PHP.
2. Разработка покупательской тележки средствами PHP и MySQL.

#### Тема 4. Функции

1. Разработка системы управления контентом.
2. Разработка почтовой web-службы.
3. Разработка диспетчера списков рассылки.

### **Тема 5. Функции работы с файлами**

1. Разработка приложений поддержки web-форумов.
2. Генерация персонифицированных документов в PDF-формате.
3. JavaScript и DHTML: визуальные эффекты, меню и навигация, слои, позиционирование элементов.

### **Тема 6. Реляционные базы данных**

1. SEO-оптимизация и продвижение web-сайта в сети Интернет.
2. Композиция web-сайта.
3. Цветовое оформление web-сайтов.

### **Тема 7. Введение в mysql.**

1. Создание анимации для web-сайтов.
2. Работа с видео и звуком в web.

### **Тема 8. Команды SQL.**

1. Юзабилити. Организация навигации с точки зрения удобства пользователя.
2. Роль графики в web-дизайне.
3. Технология размещения сайта в сети Internet.

### **Основная литература**

Савельева, Н.В. Язык программирования PHP / Н.В. Савельева. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 330 с. : схем., ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428975> (Электронная библиотечная система «Университетская библиотека ONLINE»)

### **Дополнительная литература**

Малашкевич, В.Б. Интернет-программирование: лабораторный практикум / В.Б. Малашкевич; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2017. - 96 с.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=476400> (Электронная библиотечная система «Университетская библиотека ONLINE»)

### **Информационно-справочные системы, профессиональные базы данных и интернет-ресурсы**

1. Сайт Security Lab [Электронный ресурс]. – Режим доступа: <http://www.securitylab.ru/> (новости, экспертные статьи, софты, форум, раздел, где оперативно публикуется информация об уязвимостях, а также даются конкретные рекомендации по их устранению)

2. Сайт Threatpost [Электронный ресурс]. – Режим доступа: <https://threatpost.com/> (содержит новости об информационной безопасности от Kaspersky Lab)

3. Сайт Anti-Malware [Электронный ресурс]. – Режим доступа: <https://www.anti-malware.ru/> (содержит сравнительные тесты антивирусов, аналитические статьи)

4. Сайт Научного журнал «Вопросы кибербезопасности» [Электронный ресурс]. – Режим доступа: <http://cybergnus.com/> (содержит статьи российских и иностранных ученых по кибербезопасности, безопасности приложений, технической защите информации, аудиту безопасности систем и программного кода, тестированию, анализу защищенности и оценке соответствия ПО требованиям безопасности информации)

5. Профессиональная база данных по бизнес-информатике [Электронный ресурс]. - Режим доступа: [http://dorlov.blogspot.ru/p/blog-page\\_3151.html](http://dorlov.blogspot.ru/p/blog-page_3151.html)

6. СПС «Консультант Плюс», СПС «Гарант» (договор о сотрудничестве от 23.09.2013 г. с ЗАО «Компьютерные технологии» (ПС Гарант)), регистрационный лист зарегистрированного пользователя ЭПС «Система ГАРАНТ» от 16.02.2012 г. №12-40272-000944; договоры с ООО «КонсультантПлюс Марий Эл» №2017-СВ-4 от 28.12.2016 г

**Распределение трудоемкости СРС при изучении дисциплины**

Вид самостоятельной работы	Трудоемкость (час)
Подготовка к экзамену	10/32
Проработка конспекта лекций	15/22
Подготовка к практическим (семинарским) занятиям	15/23
Проработка учебного материала	17/22
Решение отдельных задач	15/20

### 3. Оценочные средства и методические рекомендации по проведению промежуточной аттестации

При проведении экзамена по дисциплине «Программирование РНР» может использоваться устная или письменная форма проведения.

#### Примерная структура экзамена по дисциплине «Программирование РНР»:

##### 1. устный ответ на вопросы

Студенту на экзамене дается время на подготовку вопросов теоретического характера.

##### 2. выполнение тестовых заданий

Тестовые задания выполняются в течение 30 минут и состоят из 25 вопросов разных типов. Преподаватель готовит несколько вариантов тестовых заданий.

##### 3. выполнение практических заданий

Практических задания выполняются в течение 30 минут. Бланки с задачами готовит и выдает преподаватель.

**Устный ответ студента на экзамене должен отвечать следующим требованиям:**

- научность, знание и умение пользоваться понятийным аппаратом;
- изложение вопросов в методологическом аспектах, аргументация основных положений ответа примерами из современной практики, а также из личного опыта работы;
- осведомленность в важнейших современных проблемах программирования РНР, знание классической и современной литературы.

**Выполнение практического задания должно отвечать следующим требованиям:**

- Владение профессиональной терминологией;
- Последовательное и аргументированное изложение решения.

#### Критерии оценивания ответов

	Устный ответ	Практическое задание	Тестовые задания
<i>Отлично</i>	знание учебного материала в пределах программы; логическое, последовательное изложение вопроса с опорой на разнообразные источники, с использованием знаний других наук; определение своей позиции в раскрытии различных подходов к рассматриваемой проблеме; показ значения разработки данного теоретического вопроса для практики	свободное владение профессиональной терминологией; умение высказывать и обосновать свои суждения; студент дает четкий, полный анализ ситуации.	90–100 % правильно выполненных заданий
<i>Хорошо</i>	знание учебного материала в пределах программы; раскрытие различных подходов к рассматриваемой	студент владеет профессиональной терминологией, осознанно применяет	70–90 % правильно выполненных заданий

	проблеме; опора при рассмотрении вопроса на обязательную литературу, включение соответствующих примеров из практики	теоретические знания для решения практического задания, но содержание и форма ответа имеют отдельные неточности; ответ правильный, полный, с незначительными неточностями или недостаточно полный.	
<b>Удовлетворительно</b>	знание учебного материала в пределах программы на основе изучения какого-либо одного подхода к рассматриваемой проблеме	студент допускает неточности в определении понятий, в применении знаний для решения практического задания, не может доказательно обосновать свои суждения; обнаруживается недостаточно глубокое понимание материала.	50–70 % правильно выполненных заданий
<b>Неудовлетворительно</b>	пробелы в знаниях основного учебно-программного материала, принципиальные ошибки в выполнении предусмотренных программой заданий	допущены ошибки в определении понятий, искажен их смысл; студент не может применять знания для решения практического задания.	менее 50% правильно выполненных заданий

**Итоговая отметка** за экзамен по предмету выставляется с учетом полученных отметок в соответствии с правилами математического округления.

#### **Рекомендации по проведению экзамена**

1. Студенты должны быть заранее ознакомлены с требованиями к экзамену, критериями оценивания. В результате экзамена студент должен обязательно четко понять, почему он получил именно ту экзаменационную отметку, которая была ему поставлена за его ответ, а не другую.

2. Необходимо выяснить на экзамене, формально или нет владеет студент знаниями по данному предмету. Вопросы при ответе по билету помогут выяснить степень понимания студентом материала, знание им связей излагаемого вопроса с другими изучавшимися им понятиями, а практические задания – умения применять знания на практике.

3. На экзамене следует выяснить, как студент знает программный материал, как он им овладел к моменту экзамена, как он продумал его в процессе обучения и подготовки к экзамену.

4. При устном опросе целесообразно начинать с легких, простых вопросов, ответы на которые помогут подготовить студента к спокойному размышлению над дальнейшими более трудными вопросами и практическими заданиями.

5. Тестирование по дисциплине проводится либо в компьютерном классе, либо в аудитории на бланке с тестовыми заданиями.

Во время тестирования обучающиеся могут пользоваться калькулятором. Результат каждого обучающегося оценивается в соответствии с оценочной шкалой, приведенной в пункте 3.

6. Выполнение практических заданий осуществляется в учебной аудитории. Результат каждого обучающегося оценивается в соответствии с оценочной шкалой, приведенной в пункте 3

### **Перечень вопросов к экзамену по курсу «Программирование PHP»**

1. Язык php представляет собой.
2. Область применения php.
3. Функциональные возможности php.
4. Типы данных php. Преобразование типов данных php.
5. Строковые функции. Основные операции со строками.
6. Константы и переменные. Обработка данных форм, при помощи php.
7. Функции. Область видимости переменных.
8. Входные параметры функции.
9. Возможности использования рекурсии.
10. Массивы. Основные операции с массивами.
11. Способы обхода элементов массива.
12. Функции для чтения и записи в файл. Вывод содержимого файла.
13. Приемы работы с файлами.
14. Копирование, удаление и перемещение файлов.
15. Способы передачи значений переменных между скриптами.
16. Сессии в php.
17. Cookies в php.
18. Модели баз данных
19. Реляционная модель
20. Принципы организации структуры таблицы, построенной на основе реляционной модели базы данных.
21. Что представляет собой mysql.
22. История появления sql.
23. Основные возможности sql.
24. Типы данных в sql.
25. Группировка в sql.
26. Древовидная структура базы данных. Способы организации древовидной структуры.

### **Примерный перечень практических заданий**

1. Даны два файла со словами, разделенными пробелами. Создать новый файл, содержащий: а) строки, которые встречаются только в первом файле; б) строки, которые встречаются в обоих файлах; в) строки, которые встречаются в каждом файле более двух раз.



2. Даны два файла, состоящие из предложений. Создать третий файл, содержащий все предложения, которые есть хотя бы в одном из файлов. Повторы не добавлять в третий файл.

3. Дан файл со словами. Упорядочить слова по алфавиту.

4. Дан файл. Каждая строка содержит имя, пароль и email, разделенные символами ':' (двоеточие). Удалить строки с повторами email. Удалить строки, в которых имена совпадают.

5. Написать функцию, которая будет показывать список всех файлов в данной папке (поиск файлов происходит и во всех вложенных уровнях).

6. Пользователю предлагается ввести имя каталога в локальной файловой системе сервера. Сценарий PHP выводит содержимое этого каталога в следующем формате: пиктограмма, указывающая на тип файла, имя файла, размер (или специальная пометка, если файл является каталогом), дата и время последней модификации.

## Тестовые задания по дисциплине «Программирование PHP»

### 0 вариант

**Какие парадигмы программирования поддерживает PHP?**

логическую

✓ процедурную

✓ объектно-ориентированную

**Написать программу, которая выводит «жирными» буквами (тег <b>) строку «Добро пожаловать!» с использованием языка PHP.**

✓

```
<?php
echo
"<b> Добро пожаловать!</b> ";
?>
<?php
<b>
echo "Добро пожаловать!"
</b>
?>
<?
echo "<bold>Добро пожаловать!</bold>"
?>
```

**Известно, что настройки PHP можно сохранять/изменять не только в php.ini, но и в самих PHP скриптах, в файлах .htaccess и в файле настроек сервера httpd.conf. Где можно установить значение опции session.auto\_start?**

✓ в php.ini или httpd.conf

✓ в скрипте пользователя

✓ в файле .htaccess

**Какой оператор обозначает равенство значений в языке PHP?**

:=

=

==

✓ ==

**Как можно задать массив в языке PHP?**

```
$arr[«a»,»b»,»c»] = «q»;
```

```
$arr («0"=> «a»);
```

✓ `$arr = array(«a»,»b»,»c»);`

✓ `$arr[0] = «a»;`

**Какие из утверждений относительно оператора require верны?**

require выполняет код указанного в нем файла только один раз

✓ require используется для включения в программу содержимого другого файла  
при использовании внутри условных блоков require не нужно заключать в фигурные скобки

**Каким будет результат выполнения следующей программы**

```
<?php
for ($j=1; ☺
{
$i = round (9/$j);
switch ($i)
{
case 5: echo "+";
break;

case 9: echo "-";
break;

case 3: echo "!";
break 2;

default: echo $i;
break;
}
$j++;
}
?>
```

ошибка синтаксиса

бесконечный цикл

✓ `-+!`

**Дана команда: if(!\$var) echo «Hello»; В каком случае на экран будет выведено слово «Hello»?**

если `$var == true`

✓ если `$var = 0`

✓ если `$var` преобразуется к логическому `false`

**Какие из операторов switch записаны правильно с точки зрения синтаксиса?**

```
switch ($par){
"1": echo "1";
"2": echo 2;
}
```

✓

```
switch ($par){
case "1": echo "1";
case 2: echo 2;
default: echo 3;
}
```

✓

```
switch ($par):
case "1": echo "1";
break;
case 2: echo 2;
break;
endswitch;
Файл vars.php:
```

```
<?php
$a = 1;
?>
Файл index.php:
```

```
<?php
for ($I=0; $I < 3; $I++)
{
    include("vars.php");
    echo $a++;
}
?>
```

**Что мы получим в результате обработки интерпретатором файла index.php?**

12  
1234  
123

✓ 111

**В каком случае выполняется блок действий цикла**

```
for (expr1; expr2; expr3) {
// блок действий
}
```

✓ если второе выражение (expr2) вычисляется как true

если третье выражение (expr3) вычисляется как true

если первое выражение (expr1) вычисляется как true

**Какое из утверждений относительно оператора include верно?**

файл, включаемый с помощью include, должен быть правильным HTML-файлом

✓ include используется для включения в программу содержимого другого

файла

файл, включаемый с помощью include, может быть любым файлом

**Как можно узнать виртуальный путь до выполняющегося в данный момент скрипта?**

- ✓ getenv('SCRIPT\_NAME')
- ✓ \$\_SERVER['SCRIPT\_NAME']
- ✓ \$\_SERVER['PHP\_SELF']

с помощью константы SCRIPT\_NAME

#### **Чем отличается клиент от сервера?**

- ✓ клиент посылает запросы, а сервер обрабатывает их
- ✓ клиент отображает данные на экране компьютера пользователя, а сервер предоставляет данные

✓ сервер выполняет запросы клиента, специализируясь на эффективном решении задач определенного класса

сервер есть часть компьютерной архитектуры сервер-сервер, а клиент – архитектуры клиент-клиент

сервер создает соединение, а клиент, пользуясь им, передает свой запрос

#### **Как можно получить адрес страницы, с которой был послан запрос?**

- ✓ \$\_SERVER['HTTP\_REFERER']

getenv('REMOTE\_HOST')

- ✓ getenv('HTTP\_REFERER')

с помощью константы HTTP\_REFERER

#### **В чем состоят функции сервера?**

- ✓ обрабатывать запросы клиента
- ✓ запускать процессы, запрошенные клиентом, и возвращать клиенту результаты

отображать данные на экране компьютера пользователя

инициировать соединение с клиентом

#### **Дана следующая html форма:**

```
<form name="my_form" method="get">
Field 1 <input type="text" name="f1"><br>
Select field
<select name="f2">
  <option value="1">first
  <option value="2">second
</select><br>
<input type="submit">
</form>
```

#### **Как можно получить переданные клиентом значения из этой формы?**

\$f1 и \$f2, если register\_globals=off

\$\_POST[«f1»] и \$\_POST[«f2»]

- ✓ \$\_GET[«f1»] и \$\_GET[«f2»]

\$POST[«f1»] и \$POST[«f2»]

- ✓ \$\_REQUEST[«f1»] и \$\_REQUEST[«f2»]

**Создать класс А, расширяющий класс В с помощью метода add() и**

**переменной \$a.**

```
class B{
  var $b;
  function B(){
  }
}
class B extends A {
  var $a;
  function add($b){
    $this->a += $b;
  }
}
✓
```

```
class B{
  var $b;
  function B(){
  }
}
class A extends B {
  var $a;
  function add($b){
    $this->a += $b;
  }
}
class A extends B {
  var $a;
  function add($b){
    $this-> += $b;
  }
}
```

**Как можно получить и вывести на экран список всех методов класса, представителем которого является объект?**

```
get_methods(get_class(объект));
✓ get_class_methods(get_class(объект));
get_class_methods(get_class -> объект);
```

**Класс MyClass задан следующим образом:**

```
<?php
class MyClass{
  var $a;
  function MyClass() {
    $this->a = "hello";
  }
  function MyFunc($b) {
    return $b . " " . $this->a;
  }
}
?>
```

**Создать представителя класса MyClass. Получить и вывести значения всех свойств этого класса. Вызвать методы данного класса.**

✓

```
$myObj = new MyClass();  
echo $myObj->a;  
$str = $myObj->MyFunc("people");
```

**Конструктор MyClass() вызывается автоматически при создании представителя класса.**

```
new MyClass();  
echo $MyClass->a;  
$str = $MyClass->MyFunc("people");
```

**Конструктор MyClass() вызывается автоматически при создании представителя класса.**

```
$myObj = new MyClass("test");  
$b = "a";  
echo $myObj->$b;  
MyFunc($b);
```

**Имеется объект (\$obj) какого-то класса. Требуется получить список всех свойств класса, которому принадлежит объект.**

```
$get_class_vars(get_class($obj);  
get_class_var(get_class($obj));
```

✓ \$vars = get\_class\_vars(get\_class(\$obj));

**Как можно программно узнать имя класса, представителем которого является объект?**

```
$(get_class->объект);
```

✓ get\_class(объект);

```
class(объект);
```

**Имеется следующая форма:**

```
<form action=task3.php>  
Запись номер 1 <input type=checkbox name=id[] value=10> <br>  
Запись номер 2 <input type=checkbox name=id[] value=20> <br>  
Запись номер 3 <input type=checkbox name=id[] value=30> <br>  
<input type=submit value="Отправить">  
</form>
```

**Изменить значения переданных элементов, увеличив их в 10 раз. Использовать функцию array\_walk(). Вывести значения элементов до и после применения array\_walk**

✓

```
<?  
print_r($_GET);  
array_walk($_GET,"test");  
function test(&$val,$key){  
foreach ($val as $k=> $v)  
$val[$k] = $v*10;  
}  
print_r($_GET);
```

```

?>
<?
print_r($_GET);
array_walk($_GET,"test");
function test(&$val,$key){
$val = $val*10;
}
print_r($_GET)
?>
<?
print_r($_POST);
array_walk($_POST,"test");
function test(&$val,$key){
foreach ($val as $k=> $v)
$val[$k] = $v*10;
}
print_r($_POST);
?>

```

**Дана строка «<h2>Наука – <font color=red>двигатель</font> прогресса!</h2>». Как можно вывести ее в браузер именно в таком виде, без форматирования согласно стандарту HTML?**

✓

```

echo htmlspecialchars("<h2>Наука - <font color=red>двигатель</font>
прогресса!</h2>");
echo htmlentities("<h2>Наука – <font color=red>двигатель</font> прогресса!</h2>");

```

✓

```

echo htmlspecialchars("<h2>Наука - <font color=red>двигатель</font>
прогресса!</h2>", ENT_COMPAT);

```

**Дана строка текста: «PHP – очень простой язык. Я знаю PHP!!!». Заменить в ней все вхождения слова «PHP» на слово «Perl».**

```

<?php
$str = "PHP – очень простой язык. Я знаю PHP!!!";
echo str_replace($str, "PHP", "Perl");
?>

```

✓

```

<?php
$str = "PHP – очень простой язык. Я знаю PHP!!!";
echo str_replace("PHP", "Perl", $str);
?>

```

```

<?php
$str = "PHP – очень простой язык. Я знаю PHP!!!";
echo substr_replace("PHP", "Perl", $str);
?>

```

**Что делает функция move\_uploaded\_file (временное имя файла, место назначения)?**

**удаляет загруженный файл**

✓ проверяет, загружен ли файл, и если да, то перемещает его из временной директории в указанную директорию для хранения  
перемещает файл из временной директории в указанную директорию для хранения

### Имеется следующая html-форма

```
<form enctype="multipart/form-data" action="parse.php" method="post">
<input type="hidden" name="MAX_FILE_SIZE" value="30000" />
Загрузить файл: <input type="file" name="my" /><br>
<input type="submit" value="Отправить файл" />
</form>
```

**Как узнать размер файла, который клиент хочет загрузить на сервер с помощью этой формы.**

✓ filesize(\$\_FILES['my']['tmp\_name'])

filesize(\$\_FILES['my']['name'])

\$\_FILES['size']

filesize(\$\_FILES['name']['my'])

**С помощью какой функции можно проверить, существует ли файл?**

✓ file\_exists()

is\_writable()

is\_readable()

**Какие из перечисленных функций считывают данные из файла, ссылка на который установлена функцией fopen?**

✓ fgetc()

✓ fgets()

file\_read()

✓ fread()

### Имеется следующая html-форма

```
<form enctype="multipart/form-data" action="parse.php" method="post">
<input type="hidden" name="MAX_FILE_SIZE" value="30000" />
Загрузить файл: <input type="file" name="my" /><br>
<input type="submit" value="Отправить файл" />
</form>
```

**Как узнать, что файл, загружаемый на сервер с помощью этой формы, был загружен успешно?**

if (isset(\$\_FILES['my'])) echo «файл загружен успешно»;

✓ if (\$\_FILES['my']['error'] == 0) echo «файл загружен успешно»;

if (\$\_FILES['my']['error'] == 1) echo «файл загружен успешно»;

**Каков синтаксис оператора удаления таблицы?**

REMOVE TABLE [IF EXISTS] имя\_таблицы [, имя\_таблицы,...] [RESTRICT | CASCADE]

✓ DROP TABLE [IF EXISTS] имя\_таблицы [, имя\_таблицы,...] [RESTRICT | CASCADE]

DELETE TABLE [IF EXISTS] имя\_таблицы [, имя\_таблицы,...] [RESTRICT | CASCADE]

**Составить запрос к таблице Articles на получение всех значений таких ее полей, как название статьи (title), автор (author) и краткое содержание (abstract). Упорядочить результат запроса по имени автора.**



```
SELECT * FROM Articles WHERE author="$author", title="$title",
abstract="$abstract" ORDER BY author;
SELECT (title, author, abstract) FROM Articles WHERE ORDER BY author;
```

✓

```
SELECT title, author, abstract FROM Articles ORDER BY author;
```

**Оператор update предназначен для обновления значений существующих столбцов таблицы в соответствии с введенными значениями. Каков синтаксис этого оператора?**

```
UPDATE имя_таблицы
SET имя_столбца1=выражение1 [, имя_столбца2=выражение2, ...]
UPDATE      имя_таблицы      SET      имя_столбца1=выражение1      [,
имя_столбца2=выражение2, ...]
```

✓

```
UPDATE [LOW_PRIORITY] [IGNORE] имя_таблицы
SET имя_столбца1=выражение1 [, имя_столбца2=выражение2, ...]
[WHERE where_definition] [LIMIT число]
```

**Составить запрос к таблице описаний статей (Articles): изменить название статьи (title), автор (author) которой «Петров» на название «Второе название».**

✓

```
UPDATE Articles SET title="Второе название" WHERE author="Петров";
UPDATE Articles SET title="Второе название";
UPDATE Articles SET title="Второе название" WHERE title="Первое название";
```

**Составить запрос к таблице Articles на добавление описания статьи с названием (title) «Новая статья» и автором (author) «Сидоров С. С.».**

✓

```
INSERT INTO Articles SET title='Новая статья', author='Сидоров С. С.';
INSERT INTO Articles title='Новая статья', author='Сидоров С. С.';
```

✓

```
INSERT INTO Articles (title, author) VALUES('Новая статья', 'Сидоров С. С.);
```

**Получить имена всех полей таблицы persons базы данных book**

```
<?php
$fld = mysql_list_fields("book", "persons");
$n = mysql_num_fields($fld);
```

```
for($i=0;$i<$n; $i++){
    $name_f = mysql_field_name ($i);
    echo "<br>Имя поля: ". $name_f;
}
?>
```

✓

```
<?php
$conn = mysql_connect("localhost", "nina", "123");
$fld = mysql_list_fields("book", "persons", $conn);
$n = mysql_num_fields($fld);
```

```
for($i=0;$i<$n; $i++){
    $name_f = mysql_field_name ($fld, $i);
    echo "<br>Имя поля: ". $name_f;
```

```

}
?>
<?php
$conn = mysql_connect("localhost", "nina", "123");
$fields = mysql_list_fields("book", "persons", $conn);

for($i=0;$i<count($fields); $i++){
    echo "<br>Имя поля: ". $fields[$i];
}
?>

```

**Функция mysql\_connect, устанавливающая соединение с базой данных MySQL, имеет следующий синтаксис:**

```

mysql_connect ( server, username, password,
               new_link, client_flags)

```

**Какие значения будут установлены для параметров server, username, password, если они не были заданы при вызове функции mysql\_connect?**

✓

```

server = 'localhost:3306'
username = имя пользователя владельца процесса сервера
password = пустой пароль
эти параметры обязательны для функции mysql_connect
server = 'localhost:8080'
username = имя пользователя владельца процесса сервера
password = пароль пользователя владельца процесса сервера

```

**Каковы синтаксис и семантика функции explode?**

массив explode (строка string). Эта функция разбивает строку string на части с помощью разделителя » » и возвращает массив полученных строк  
эта функция без параметров. Ее семантика неизвестна

✓ массив explode(строка separator, строка string [, int limit]). Эта функция разбивает строку string на части с помощью разделителя separator и возвращает массив полученных строк

**Параметр session.cookie\_lifetime задает длительность жизни cookies в секундах. Какое значение имеет этот параметр по умолчанию?**

✓ по умолчанию это «0», т.е. данные в cookies считаются правильными до закрытия окна браузера

по умолчанию это «10», т.е. данные в cookies считаются правильными ровно 10 секунд

по умолчанию это «60»

**Уничтожить текущую сессию целиком можно командой session\_destroy(); К чему приведет уничтожение сессии?**

после уничтожения сессии очищается массив \$\_SESSION и больше ничего не происходит

✓ после уничтожения сессии уничтожается ее идентификатор, мы больше не можем ни регистрировать переменные, ни вообще производить какие-либо действия с сессией

после уничтожения сессии массив \$\_SESSION заполняется нулями

**С помощью какой функции можно получить идентификатор сессии?**

id\_session  
session\_identificator  
✓ session\_id

**В одном из скриптов программы имеется переменная \$user\_name = «Иван Петров» . Как сделать так, чтобы эта переменная была доступна во всех скриптах программы.**

✓ передавать эту переменную при переходе от скрипта к скрипту в качестве скрытого элемента HTML-формы

✓ в этом скрипте нужно создать сессию (или восстановить текущую) с помощью команды session\_start(). Потом нужно зарегистрировать переменную в качестве переменной сессии таким образом:

```
$_SESSION["user_name"] = "Иван Петров";
```

Все остальные скрипты программы начинать с команды session\_start();

в этом скрипте нужно создать сессию (или восстановить текущую) с помощью команды session\_start(). Потом нужно зарегистрировать переменную в качестве переменной сессии таким образом:

```
$_SESSION["user_name"] = "Иван Петров";
```

нужно зарегистрировать переменную в качестве переменной сессии:

```
$_SESSION["user_name"] = "Иван Петров";
```

**Какова структура регулярного выражения?**

✓ общая структура регулярного выражения: шаблон, выделенный с помощью специального символа разделителя, модификатор, влияющий на способ обработки регулярного выражения

общая структура регулярного выражения: шаблон, заключенный в круглые скобки, и функция для его обработки

общая структура регулярного выражения: шаблон, выделенный с помощью специального символа разделителя

**Назначение метасимвола «-«?»**

вычисляет символьный класс

отрицание класса, но только если это первый символ

✓ задает диапазон символов

**Выделить из URL адреса ресурса имя хоста**

✓

```
<?php
```

```
preg_match ("/^(http:\V)?([^\V]+)/i", "http://www.php.net/test/index.html", $matches);  
echo "Host:", $matches[2];
```

```
?>
```

```
<?php
```

```
preg_match ("/^(http:\/\/)(\w+[\V]+)/i", "http://www.php.net/test/index.html", $matches);  
echo "Host:", $matches[2];
```

```
?>
```

```
<?php
```

```
preg_match ("/^(http:\V)?([^\V]+)/i", "http://www.php.net/test/index.html", $matches);  
echo "Host:", $matches[0];
```

```
?>
```

**Что делает следующая функция?**

```
function Test($str){
$pattern = "\d{3}-\d{2}-\d{2}/m";
$num_match = preg_match_all ($pattern, $str, $result);
return $num_match;
}
```

возвращает 0 или 1 в зависимости от того, встречена ли в строке семизначная комбинация цифр, записанных в виде: три цифры, тире, две цифры, тире, две цифры

✓ возвращает число встреченных в строке семизначных комбинаций цифр, записанных в виде: три цифры, тире, две цифры, тире, две цифры

возвращает число встреченных в строке семизначных комбинаций цифр, состоящих из цифр 3 и 2

**Функция domxml\_new\_doc в качестве результата возвращает пустой XML-документ. Что передается этой функции в качестве параметра?**

в качестве параметра передается строка, в которой содержатся первоначальные данные XML-документа

в качестве параметра передается строка, содержащая путь, где будет храниться XML-документ

✓ в качестве параметра передается версия создаваемого XML-документа

у этой функции параметров нет

**Что возвращает метод document\_element класса DomDocument?**

содержимое элемента

✓ корневой элемент

значение атрибута

**Какая комбинация технологий позволяет получить HTML-документы?**

✓ XML + XSLT

XML + DTD

XSLT + DTD

**Что представляет собой шаблон Smarty?**

это набор переменных Smarty и html-тегов

это набор html-тегов

✓ это набор переменных, циклов, условных операторов, операторов вставки файлов и т.д

**Для чего используется метод parse() при работе с шаблонами FastTemplate?**

обрабатывает шаблон и выводит его содержимое на экран

✓ этот метод инициализирует обработку шаблона и сохраняет обработанный шаблон в переменную

этот метод присваивает переменной указанное значение

**Как задается переменная в шаблоне Smarty?**

{имя\_переменной}

✓ {\$имя\_переменной}

✓ {#bodyBgColor#}

**Дан массив \$b = array(«23aaa»,»4",»qww»,»с», 3). Найти в массиве число 3, не перебирая все элементы массива. Если элемент найден, вывести значение его ключа.**

```
<?php
```

```
$b = array("23aaa", "4", "qww", "c", 3);
```

```
$index = array_search($b, "3");
```

```

if ($index === false) echo"Нет такого числа в массиве";
else { echo"Число найдено с ключом ";
echo $index;
}
?>
<?php
$b = array("23aaa","4","qww","c", 3);
if (!array_search("3",$b)) echo"Нет такого числа в массиве";
else { echo"Число найдено с ключом ";
echo array_search("3",$b);
}
?>
✓

```

```

<?php
$b = array("23aaa","4","qww","c", 3);
$index = array_search(3,$b,true);
if ($index === false) echo"Нет такого числа в массиве";
else { echo"Число найдено с ключом ";
echo $index;
}
?>

```

**Что делает приведенная ниже программа.**

```

<?php
$f = fopen("file.html", "r");
echo fread($f, 1024);
fclose($f);
?>

```

открывает файл file.html только для записи. Дописывает в конец файла строку «1024». Если файла с именем file.html не существует в директории скрипта, то он будет создан и в него запишется слово «1024». Закрывает соединение с файлом

✓ открывает файл file.html только для чтения. Считывает 1024 байт текста из файла и выводит его на экран. Закрывает соединение с файлом

открывает файл file.html для записи и чтения. Считывает 1024 символа из файла и выводит его на экран. Закрывает соединение с файлом

Функция domxml\_open\_file в качестве результата возвращает объект класса DOMDocument. Что передается этой функции в качестве параметра?

в качестве параметра передается строка, содержащая XML-документ

✓ в качестве параметра передается строка, содержащая путь к XML-документу

в качестве параметра передается объект класса DOMNode

**Каким будет результат работы следующей программы**

```

<?php
function Test($a="q", $b)
{
    echo $a;
    return $a . $b;
}
Test(1);
?>

```

q1  
qq1

✓ предупреждение, что не хватает второго аргумента для функции, и значение

1

**Данные некоторой формы отправлены на сервер методом POST. Проверить, была ли передана в качестве значения какого-либо элемента формы строка «hello»**

```
<?php
if (array($_POST, "hello", true)) echo "Element found";
?>
<?php
if (array_search("hello", $_GET)) echo "Element found";
?>
✓
```

```
<?php
if (array_keys($_POST, "hello")) echo "Element found";
?>
```

**С какими параметрами может вызываться функция explode()?**

✓ максимальное количество возвращаемых строк

максимальная длина возвращаемых строк

✓ разделитель в виде строки

✓ строка для разделения

**Какие из циклов while записаны правильно с точки зрения синтаксиса?**

✓

```
while ($a < $b):
echo $a;
$a++;
endwhile;
```

✓

```
while ($a < $b){
echo $a;
$a++;
}
while ($a < $b){
echo $a;
$a++;
endwhile;
```

**Что такое пользовательская директория сервера и где она находится по умолчанию (для windows систем)?**

это директория, выделенная пользователю для личных нужд, файлы которой обрабатываются сервером. Находится в «C:/Program Files/Apache Group/Apache/htdocs/users/»

✓ это директория, выделенная пользователю для личных нужд, файлы которой обрабатываются сервером, если он получает запрос типа ~user. Находится в «C:/Program Files/Apache Group/Apache/users/»

это директория, выделенная пользователю для личных нужд, файлы которой никогда не обрабатываются сервером. Находится в «C:/Program Files/Apache

Group/Apache/users/»

это директория, выделенная пользователю для личных нужд, файлы которой обрабатываются сервером, если он получает запрос типа ~user. Находится в «C:/Program Files/Apache Group/Apache/htdocs/users/»

Дан массив \$arr. Вывести на экран четвертый элемент массива, если он эквивалентен строке «4».

```
<?php
if ($arr[3] === "3") echo $arr[4];
?>
```

```
<?php
if ($arr[4] === "4") echo $arr[4];
?>
```

✓

```
<?php
if ($arr[3] === "4") echo $arr[3];
?>
```

**С помощью каких функций можно записать данные в файл, соединение с которым открыто функцией fopen?**

file()

✓ fwrite()

✓ fputs()

Имеется следующая форма:

```
<form action=task3.php>
Запись номер 1 <input type=checkbox name=id[] value=10> <br>
Запись номер 2 <input type=checkbox name=id[] value=20> <br>
Запись номер 3 <input type=checkbox name=id[] value=30> <br>
<input type=submit value="Отправить">
</form>
```

Изменить значения переданных элементов, увеличив их в 10 раз. Использовать функцию array\_walk(). Вывести значения элементов до и после применения array\_walk

```
<?
print_r($_GET);
array_walk($_GET,"test");
function test(&$val,$key){
$val = $val*10;
}
print_r($_GET);
?>
```

✓

```
<?
print_r($_GET);
array_walk($_GET,"test");
function test(&$val,$key){
foreach ($val as $k=> $v)
$val[$k] = $v*10;
}
print_r($_GET);
```

```
?>
<?
print_r($_POST);
array_walk($_POST,"test");
function test(&$val,$key){
foreach ($val as $k=> $v)
$val[$k] = $v*10;
}
print_r($_POST);
?>
```



#### **4. Учебно-методическое и информационное обеспечение дисциплины**

##### **Основная литература**

Савельева, Н.В. Язык программирования PHP / Н.В. Савельева. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 330 с. : схем., ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428975> (Электронная библиотечная система «Университетская библиотека ONLINE»)

##### **Дополнительная литература**

Малашкевич, В.Б. Интернет-программирование: лабораторный практикум / В.Б. Малашкевич; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2017. - 96 с.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=476400> (Электронная библиотечная система «Университетская библиотека ONLINE»)

##### **Информационно-справочные системы, профессиональные базы данных и интернет-ресурсы**

4. Сайт Security Lab [Электронный ресурс]. – Режим доступа: <http://www.securitylab.ru/> (новости, экспертные статьи, софты, форум, раздел, где оперативно публикуется информация об уязвимостях, а также даются конкретные рекомендации по их устранению)

5. Сайт Threatpost [Электронный ресурс]. – Режим доступа: <https://threatpost.com/> (содержит новости об информационной безопасности от Kaspersky Lab)

6. Сайт Anti-Malware [Электронный ресурс]. – Режим доступа: <https://www.anti-malware.ru/> (содержит сравнительные тесты антивирусов, аналитические статьи)

4. Сайт Научного журнал «Вопросы кибербезопасности» [Электронный ресурс]. – Режим доступа: <http://cybergnus.com/> (содержит статьи российских и иностранных ученых по кибербезопасности, безопасности приложений, технической защите информации, аудиту безопасности систем и программного кода, тестированию, анализу защищенности и оценке соответствия ПО требованиям безопасности информации)

5. Профессиональная база данных по бизнес-информатике [Электронный ресурс]. - Режим доступа: [http://dorlov.blogspot.ru/p/blog-page\\_3151.html](http://dorlov.blogspot.ru/p/blog-page_3151.html)

6. СПС «Консультант Плюс», СПС «Гарант» (договор о сотрудничестве от 23.09.2013 г. с ЗАО «Компьютерные технологии» (ПС Гарант)), регистрационный лист зарегистрированного пользователя ЭПС «Система ГАРАНТ» от 16.02.2012 г. №12-40272-000944; договоры с ООО «КонсультантПлюс Марий Эл» №2017-СВ-4 от 28.12.2016 г

## 5. Материально-техническое обеспечение дисциплины

Наименование специальных* помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации (в соответствии с расписанием)	Специализированная мебель, технические средства обучения: переносной ноутбук, мультимедийный проектор, экран	СПС «Консультант Плюс», СПС «Гарант» (договор о сотрудничестве от 23.09.2013 г. с ЗАО «Компьютерные технологии» (ПС Гарант)), регистрационный лист зарегистрированного пользователя ЭПС «Система ГАРАНТ» от 16.02.2012 г. №12-40272-000944; договоры с ООО «КонсультантПлюс Марий Эл» №2017-СВ-4 от 28.12.2016 г., Windows 10 Education, Windows 8, Windows 7 Professional (Microsoft Open License), Office Standart 2007, 2010 (Microsoft Open License), Office Professional Plus 2016 (Microsoft Open License), Kaspersky Endpoint Security (Лицензия №17Е0-171117-092646-487-711, договор №Тг000171440 от 17.07.2017 г.).
Компьютерный класс, каб. 303	Специализированная мебель, технические средства обучения: Автоматизированные рабочие места (ASUSTeK Computer INC. P5KPL-AM SE/Pentium (R) Dual-Core CPU E5300 2.60GHz/512)	СПС «Консультант Плюс», СПС «Гарант» (договор о сотрудничестве от 23.09.2013 г. с ЗАО «Компьютерные технологии» (ПС Гарант)), регистрационный лист зарегистрированного пользователя ЭПС «Система ГАРАНТ» от 16.02.2012 г. №12-40272-000944; договоры с ООО «КонсультантПлюс Марий Эл» №2017-СВ-4 от 28.12.2016 г. Windows 7 Professional (Microsoft Open License). Sys Ctr Endpoint Protection ALNG Subscriptions VL OLVS E 1Month AcademicEdition Enterprise Per User (Сублиц. договор № Тг000171440 17.07.2017). Office Prosessional 2010 (Microsoft Open License). Архиватор 7-zip (GNU LGPL). Adobe Acrobat Reader DC (Бесплатное ПО). Adobe Flash Player (Бесплатное ПО). Deductor Academic (Бесплатное ПО). FreeCommander (Бесплатное ПО). Inkscape (GNU GPL 2). Notepad++ (GNU GPL 2). freePascal (Бесплатное ПО). Lazarus (Бесплатное ПО). Microsoft Visual Studio 2010 (Бесплатно в рамках подписки Imagine Premium Т89-00394 от 10.02.2017). Система виртуализации Oracle VM VirtualBox (GNU LGPL).
Помещение для самостоятельной работы, каб. 114	Специализированная мебель, технические средства обучения: автоматизированные рабочие места, с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную	СПС «Консультант Плюс», СПС «Гарант» (договор о сотрудничестве от 23.09.2013 г. с ЗАО «Компьютерные технологии» (ПС Гарант)), регистрационный лист зарегистрированного пользователя ЭПС «Система ГАРАНТ» от 16.02.2012 г.

	<p>информационную образовательную среду организации (AsusTeK COMPUTER INC H110M-R/ Itnel(R) Core(TM) i3-7100 CPU @ 3.90GHz/8192.00 Gb)</p>	<p>№12-40272-000944; договоры с ООО «КонсультантПлюс Марий Эл» №2017-СВ-4 от 28.12.2016 г. Windows 7 Professional (Microsoft Open License). Sys Ctr Endpoint Protection ALNG Subscriptions VL OLVS E 1Month AcademicEdition Enterprise Per User (Сублиц. договор № Tr000171440 17.07.2017). Office Standart 2010 (Microsoft Open License). Архиватор 7-zip (GNU LGPL). Adobe Acrobat Reader DC (Бесплатное ПО). Adobe Flash Player (Бесплатное ПО). 1С:Бухгалтерия государственного учреждения 8 ПРОФ (Лиценз. договор 011/216 от 01.09.2017). 1С:Предприятие 8. Комплект для обучения в высших и средних учебных заведениях (Лиценз. договор 011/216 от 01.09.2017)</p>
--	--	--

## **6. Методические указания для обучающихся по освоению дисциплины**

### **Методические указания для подготовки к лекционным занятиям**

В ходе лекций преподаватель излагает и разъясняет основные, наиболее сложные для понимания темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации на семинарское занятие и указания на самостоятельную работу.

В ходе лекционных занятий необходимо:

– вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

– задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

– дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой – в ходе подготовки к семинарам изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях журналах, газетах и т.д. При этом учесть рекомендации преподавателя и требования учебной программы.

– подготовить тезисы для выступлений по всем учебным вопросам, выносимым на семинар. Готовясь к докладу или реферативному сообщению, обращаться за методической помощью к преподавателю, составить план-конспект своего выступления, продумать примеры с целью обеспечения тесной связи изучаемой теории с реальной жизнью.

– своевременное и качественное выполнение самостоятельной работы базируется на соблюдении настоящих рекомендаций и изучении рекомендованной литературы. Студент может дополнить список использованной литературы современными источниками, не представленными в списке рекомендованной литературы, и в дальнейшем использовать собственные подготовленные учебные материалы при написании контрольных (РГР), курсовых и выпускных квалификационных работ.

### **Методические указания для подготовки к практическим (семинарским) занятиям**

Начиная подготовку к семинарскому занятию, необходимо, прежде всего, обратить внимание на конспект лекций, разделы учебников и учебных пособий, которые способствуют общему представлению о месте и значении темы в изучаемом курсе. Затем следует поработать с дополнительной литературой, сделать записи по рекомендованным источникам. Подготовка к семинарскому занятию включает 2 этапа:

- 1й этап - организационный;
- 2й этап - закрепление и углубление теоретических знаний. На первом этапе студент планирует свою самостоятельную работу, которая включает:
  - уяснение задания, выданного на самостоятельную работу;
  - подбор рекомендованной литературы;
  - составление плана работы, в котором определяются основные пункты предстоящей подготовки.

Составление плана дисциплинирует и повышает организованность в работе. Второй этап включает непосредственную подготовку студента к занятию. Начинать надо с изучения рекомендованной литературы. Необходимо помнить, что на лекции обычно рассматривается не весь материал, а только его часть. Остальная её часть восполняется в процессе самостоятельной работы. В связи с этим работа с рекомендованной литературой обязательна. Особое внимание при этом необходимо обратить на содержание основных

положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы студент должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале. Заканчивать подготовку следует составлением плана (конспекта) по изучаемому материалу (вопросу). Это позволяет составить концентрированное, сжатое представление по изучаемым вопросам. В процессе подготовки к занятиям рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретает практика в изложении и разъяснении полученных знаний, развивается речь. При необходимости следует обращаться за консультацией к преподавателю. Готовясь к консультации, необходимо хорошо продумать вопросы, которые требуют разъяснения.

В начале занятия студенты под руководством преподавателя более глубоко осмысливают теоретические положения по теме занятия, раскрывают и объясняют основные положения выступления.

Записи имеют первостепенное значение для самостоятельной работы обучающихся. Они помогают понять построение изучаемого материала, выделить основные положения и проследить их логику. Ведение записей способствует превращению чтения в активный процесс, мобилизует, наряду со зрительной, и моторную память. Следует помнить: у студента, систематически ведущего записи, создается свой индивидуальный фонд подсобных материалов для быстрого повторения прочитанного, для мобилизации накопленных знаний. Особенно важны и полезны записи тогда, когда в них находят отражение мысли, возникшие при самостоятельной работе. Важно развивать умение сопоставлять источники, продумывать изучаемый материал.

Большое значение имеет совершенствование навыков конспектирования. Преподаватель может рекомендовать студентам следующие основные формы записи план (простой и развернутый), выписки, тезисы. Результаты конспектирования могут быть представлены в различных формах.

План - это схема прочитанного материала, краткий (или подробный) перечень вопросов, отражающих структуру и последовательность материала. Подробно составленный план вполне заменяет конспект.

Конспект - это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов.

План-конспект - это развернутый детализированный план, в котором достаточно подробные записи приводятся по тем пунктам плана, которые нуждаются в пояснении.

Текстуальный конспект – это воспроизведение наиболее важных положений и фактов источника.

Свободный конспект - это четко и кратко сформулированные (изложенные) основные положения в результате глубокого осмысливания материала. В нем могут присутствовать выписки, цитаты, тезисы; часть материала может быть представлена планом.

Тематический конспект составляется на основе изучения ряда источников и дает более или менее исчерпывающий ответ по какой-то схеме (вопросу).

Ввиду трудоемкости подготовки к семинару следует продумать алгоритм действий, еще раз внимательно прочитать записи лекций и уже готовый конспект по теме семинара, тщательно продумать свое устное выступление.

На семинаре каждый его участник должен быть готовым к выступлению по всем поставленным в плане вопросам, проявлять максимальную активность при их рассмотрении. Выступление должно строиться свободно, убедительно и аргументировано. Необходимо следить, чтобы выступление не сводилось к репродуктивному уровню (простому воспроизведению текста), не допускать и простое чтение конспекта. Необходимо, чтобы выступающий проявлял собственное отношение к тому, о чем он говорит, высказывал свое личное мнение, понимание, обосновывал его и мог сделать

правильные выводы из сказанного.

Выступления других обучающихся необходимо внимательно и критически слушать, подмечать особенное в суждениях обучающихся, улавливать недостатки и ошибки. При этом обратить внимание на то, что еще не было сказано, или поддержать и развить интересную мысль, высказанную выступающим студентом. Изучение студентами фактического материала по теме практического занятия должно осуществляться заблаговременно. Под фактическим материалом следует понимать специальную литературу по теме занятия, систему нормативных правовых актов, а также арбитражную практику по рассматриваемым проблемам. Особое внимание следует обратить на дискуссионные теоретические вопросы в системе изучаемого вопроса: изучить различные точки зрения ведущих ученых, обозначить противоречия современного законодательства. Для систематизации основных положений по теме занятия рекомендуется составление конспектов.

Обратить внимание на:

- составление списка нормативных правовых актов и учебной и научной литературы по изучаемой теме;
- изучение и анализ выбранных источников;
- изучение и анализ арбитражной практики по данной теме, представленной в информационно- справочных правовых электронных системах и др.;
- выполнение предусмотренных программой заданий в соответствии с тематическим планом;
- выделение наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями кафедры на их еженедельных консультациях;
- проведение самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов, написания рефератов и эссе по отдельным вопросам изучаемой темы;

Семинарские занятия завершают изучение наиболее важных тем учебной дисциплины. Они служат для закрепления изученного материала, развития умений и навыков подготовки докладов, сообщений, приобретения опыта устных публичных выступлений, ведения дискуссии, аргументации и защиты выдвигаемых положений, а также для контроля преподавателем степени подготовленности обучающихся по изучаемой дисциплине.

#### **Методические указания для обучающихся по освоению дисциплины для самостоятельной работы**

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных особенностей студентов и условий учебной деятельности.

При этом преподаватель назначает студентам варианты выполнения самостоятельной работы, осуществляет систематический контроль выполнения студентами графика самостоятельной работы, проводит анализ и дает оценку выполненной работы.

Самостоятельная работа обучающихся осуществляется в аудиторной и внеаудиторной формах. Самостоятельная работа обучающихся в аудиторное время может включать:

- конспектирование (составление тезисов) лекций, выполнение контрольных работ;
- решение задач;
- работу со справочной и методической литературой;
- работу с нормативными правовыми актами;
- выступления с докладами, сообщениями на семинарских занятиях;

- защиту выполненных работ;
  - участие в оперативном (текущем) опросе по отдельным темам изучаемой дисциплины;
  - участие в собеседованиях, деловых (ролевых) играх, дискуссиях, круглых столах, конференциях;
  - участие в тестировании и др.
- Самостоятельная работа обучающихся во внеаудиторное время может состоять из:
- повторение лекционного материала;
  - подготовки к семинарам (практическим занятиям);
  - изучения учебной и научной литературы;
  - изучения нормативных правовых актов (в т.ч. в электронных базах данных);
  - решения задач, выданных на практических занятиях;
  - подготовки к контрольным работам, тестированию и т.д.;
  - подготовки к семинарам устных докладов (сообщений);
  - подготовки рефератов, эссе и иных индивидуальных письменных работ по заданию преподавателя;
  - выполнения курсовых работ, предусмотренных учебным планом;
  - выделение наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями кафедры на их еженедельных консультациях;
  - проведение самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов;
  - написания рефератов и эссе по отдельным вопросам изучаемой темы.
  - подготовки к семинарам устных докладов (сообщений);
  - подготовки рефератов, эссе и иных индивидуальных письменных работ по заданию преподавателя;
  - выполнения курсовых работ, предусмотренных учебным планом;
  - выполнения выпускных квалификационных работ и др.
  - выделение наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями кафедры на их еженедельных консультациях;
  - проведение самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов;
  - написания рефератов и эссе по отдельным вопросам изучаемой темы.