АНО ВО «Межрегиональный открытый социальный институт»

УТВЕРЖДАЮ
Зав. кафедрой информационной безопасности
_____ Т.М. Гусакова
Протокол заседания кафедры
№ 1 «01» 09 2017 г.

Фонд оценочных средств
для проведения текущего контроля успеваемости и промежуточной аттестации

Учебная дисциплина Программирование PHP

Образовательная программа
38.03.05 Бизнес-информатика.
Электронный бизнес

Йошкар-Ола
2017

СОДЕРЖАНИЕ

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

В процессе освоения образовательной программы обучающиеся осваивают компетенции указанные в федеральных государственных образовательных стандартах высшего образования, сопоставленные с видами деятельности. Освоение компетенций происходит поэтапно через последовательное изучение учебных дисциплин, практик, подготовки ВКР и других видов работ предусмотренных учебным планом АНО ВО МОСИ.

№ п/п	Код компетенции	Формулировка компетенции	Номер этапа
1	ПК-6	управление контентом предприятия и Интернет-ресурсов, процессами создания и использования информационных сервисов (контент-сервисов)	3/3
2	ПК-16	умение разрабатывать контент и ИТ-сервисы предприятия и интернет-ресурсов	4/3

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Этапами формирования компетенций обучающихся при освоении дисциплины являются последовательное изучение содержательно связанных между собой разделов (тем) учебных занятий. Результаты текущего контроля и промежуточной аттестации позволяют определить уровень освоения компетенций обучающимися.

Перечень оценочных средств

№ п/п	Коды компетенций и планируемые результаты обучения		Оценочные средства	
			Наименование	Представление в ФОС
1	ПК-6	**Знать:** – синтаксис языка программирования php; – методы и функции для работы с файлами **Уметь:** – делать простейшие автоматизированные арифметические расчеты; – обрабатывать файлы и работать с информацией в файлах **Владеть:** – основами программирования на языке PHP (составление, отладка и тестирование программ; разработка и создание веб-сайта)	устный опрос, практические задания	вопросы для устного опроса, перечень практических заданий
2	ПК-16	**Знать:** – основные операторы и конструкции; – принципы построения реляционных баз данных. **Уметь:** – строить структуру реляционной базы данных; – работать с таблицами базы данных. **Владеть:** – практическими приемами программирования на языке PHP.	устный опрос, практические задания	вопросы для устного опроса, перечень практических заданий

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.

Текущая аттестация по дисциплине «Программирование PHP»

Студенты ОП 38.03.05 Бизнес-информатика. Электронный бизнес проходят текущую аттестацию по дисциплине «Программирование PHP» в 6/6 семестре.

Оценочные средства текущего контроля:

- *Устный опрос*
- *Практические задания*

Основные виды оценочных средств по темам представлены в таблице

№ п\п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Введение в php	ПК-6, ПК-16	устный опрос, практические задания
2	Строковые функции	ПК-6, ПК-16	устный опрос, практические задания
3	Массивы	ПК-6, ПК-16	устный опрос, практические задания
4	Функции	ПК-6, ПК-16	устный опрос, практические задания
5	Функции работы с файлами	ПК-6, ПК-16	устный опрос, практические задания
6	Реляционные базы данных	ПК-6, ПК-16	устный опрос,
7	Введение в mysql.	ПК-6, ПК-16	устный опрос,
8	Команды SQL.	ПК-6, ПК-16	устный опрос, практические задания

Вопросы для устного опроса

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.

Средство оценивания: устный опрос

Шкала оценивания:

– оценка «отлично» выставляется студенту, если студент не только глубоко и прочно усвоил весь программный материал, но и проявил знания, выходящие за его пределы, почерпнутые из дополнительных источников (учебная литература, научно-популярная литература, научные статьи и монографии, сборники научных трудов и интернет-ресурсы и т. п.); умеет самостоятельно обобщать программный материал, не допуская ошибок, проанализировать его с точки зрения различных школ и взглядов; увязывает знания с практикой; приводит примеры, демонстрирующие глубокое понимание материала или проблемы;

– оценка «хорошо» выставляется студенту, если студент твердо знает программный материал, грамотно и последовательно его излагает, увязывает с практикой, не допуская существенных неточностей в ответе на вопросы;

– оценка «удовлетворительно» выставляется студенту, если студент усвоил только основной программный материал, но не знает отдельных положений, в ответе допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала;

– оценка «неудовлетворительно» выставляется студенту, если студент не знает значительной части основного программного материала, в ответе допускает существенные ошибки, неправильные формулировки.

Перечень практических заданий

Тема 1. Введение в php.

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.

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.

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.

Практические задания

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

Тема 5. Функции работы с фалами.

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.

Тема 8. Команды SQL.
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.

Средство оценивания: Практические задания

Шкала оценивания:

Практическое задание оценивается по 5-балльной шкале. Баллы переводятся в оценки успеваемости следующим образом:

Оценка «отлично» выставляется обучающемуся, если практическое задание правильно решено, приведена подробная аргументация своего решение, показано хорошее знание теоретических аспектов решения кейса.

Оценка «хорошо» выставляется обучающемуся, если практическое задание правильно решено, приведена достаточная аргументация своего решение, показано определенное знание теоретических материала.

Оценка «удовлетворительно» выставляется обучающемуся, если практическое задание частично имеет правильное решение, аргументация не полная, не прослеживается знание теоретических материала.

Оценка «неудовлетворительно» выставляется обучающемуся, если практическое задание решено неверно, отсутствуют необходимые знания теоретического материала.

Промежуточная аттестация по дисциплине «Программирование PHP»

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

Примерная структура экзамена по дисциплине «Программирование PHP»:

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 PHP, 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**
Отлично	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
Хорошо	знание учебного материала в пределах программы; раскрытие различных подходов к	студент владеет профессиональной терминологией, осознанно	70–90 % правильно выполненных заданий

	рассматриваемой проблеме; опора при рассмотрении вопроса на обязательную литературу, включение соответствующих примеров из практики	применяет теоретические знания для решения практического задания, но содержание и форма ответа имеют отдельные неточности; ответ правильный, полный, с незначительными неточностями или недостаточно полный.	
Удовлетворительно	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
Неудовлетворительно	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

Критерии и шкала оценивания уровней освоения компетенций

Шкала оценивания	Шкала оценивания	Шкала оценивания
отлично	высокий	студент, овладел элементами компетенции «знать», «уметь» и «владеть», проявил всесторонние и глубокие знания программного материала по дисциплине, освоил основную и дополнительную литературу, обнаружил творческие способности в понимании, изложении и практическом

		использовании усвоенных знаний.
хорошо	продвинутый	студент овладел элементами компетенции «знать» и «уметь», проявил полное знание программного материала по дисциплине, освоил основную рекомендованную литературу, обнаружил стабильный характер знаний и умений и проявил способности к их самостоятельному применению и обновлению в ходе последующего обучения и практической деятельности.
удовлетвори тельно	базовый	студент овладел элементами компетенции «знать», проявил знания основного программного материала по дисциплине в объеме, необходимом для последующего обучения и предстоящей практической деятельности, изучил основную рекомендованную литературу, допустил неточности в ответе на экзамене, но в основном обладает необходимыми знаниями для их устранения при корректировке со стороны экзаменатора.
неудовлетво рительно	компетенции не сформированы	студент не овладел ни одним из элементов компетенции, обнаружил существенные пробелы в знании основного программного материала по дисциплине, допустил принципиальные ошибки при применении теоретических знаний, которые не позволяют ему продолжить обучение или приступить к практической деятельности без дополнительной подготовки по данной дисциплине.

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, приведённой в пункте 3.

6. Выполнение практических заданий осуществляется в учебной аудитории. Результат каждого обучающегося оценивается в соответствии с оценочной шкалой, приведённой 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 файлы 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 A, расширяющий класс B 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>
Запись номер 2 <input type=checkbox name=id[] value=20>
Запись номер 3 <input type=checkbox name=id[] value=30>
<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 – двигатель 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" />
<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()

✓ fgetss()

fileread()

✓ 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" />
<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

GET 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");
$fld = mysql_list_fields("book", "persons", $conn);

for($i=0;$i<count($fld); $i++){
    echo "<br>Имя поля: ". $fld[$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:\/\/)?([^\/]+)/i", "http://www.php.net/test/index.html", $matches);
echo "Host:", $matches[2];
?>
<?php
preg_match ("/^(http://)(\w+[^\/]+)/i", "http://www.php.net/test/index.html", $matches);
echo "Host:", $matches[2];
?>
<?php
```

```
preg_match ("/^(http:\/\/)?([^\/]+)/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»,»c», 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. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Средство оценивания: устный опрос

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

Экспресс - быстрый, безостановочный; удобная форма промежуточного контроля знаний. Главное преимущество – занимает мало времени от 5 до 7 мин., при этом в зависимости от количества вопросов (оптимальное 10), позволяет проверить большой объем и глубину знаний. Быстрая проверка, еще один плюс. Учащиеся сразу могут проверить правильность выполнения работы (правильные ответы могут быть просто открыты на об-ратной стороне доски). Экспресс-опрос проводится несколько раз за тему, что позволяет диагностировать, контролировать и своевременно корректировать усвоение материала в ходе его изучения, а не после, что значительно повышает эффективность обучения и закрепляет знания учащихся.

Средство оценивания: тест

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ТЕСТОВЫХ ЗАДАНИЙ

Непременной сопутствующей процедурой преподавания любой дисциплины являлся контроль уровня усвоения учебного материала. В настоящее время среди разнообразных форм контроля в учебном процессе стали активно применяться тестовые задания, которые позволяют относительно быстро определить уровень знаний студента. Тестовые задания является одной из наиболее научно обоснованных процедур для выявления реального качества знания у испытуемого студента. Впрочем, тестирование не может заменить собой другие педагогические средства контроля, используемые сегодня преподавателями. В их арсенале остаются устные экзамены, контрольные работы, опросы студентов и другие разнообразные средства. Они обладают своими преимуществами и недостатками и посему они наиболее эффективны при их комплексном применении в учебной практике.

По этой причине каждое из перечисленных средств применяется преподавателями на определенных этапах изучения дисциплины. Самое главное преимущество тестов – в том, что они позволяют преподавателю и самому студенту при самоконтроле провести объективную и независимую оценку уровня знаний в соответствии с общими образовательными требованиями. Наиболее важным положительным признаком тестового задания является однозначность интерпретации результатов его выполнения. Благодаря этому процедура проверки может быть доведена до высокого уровня автоматизма с минимальными временными затратами. При проведении тестирования степень сложности предлагаемых вопросов определяются преподавателем в зависимости от уровня подготовленности группы.