

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ

И.о. декана факультета информационных
технологий и компьютерной безопасности
_____/А.В. Бредихин/

2024 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Web-технологии в жизненном цикле»

Направление подготовки 09.04.01 Информатика и вычислительная техника

Профиль Жизненный цикл изделий в едином информационном пространстве цифрового производства

Квалификация выпускника магистр

Нормативный период обучения 2 года / 2 года и 4 м.

Форма обучения очная / заочная

Год начала подготовки 2024

Автор программы _____ Д.Е. Пачевский

И.о. заведующего кафедрой
Компьютерных
интеллектуальных
технологий проектирования _____ М.И. Чижов

Руководитель ОПОП _____ М.И. Чижов

Воронеж 2024

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

получение умений и навыков в backend-разработке интернет систем

1.2. Задачи освоения дисциплины

Изучение основ языка веб-программирования PHP, получение практических и теоретических навыков по проектированию и программной реализации сложных веб-порталов и управлению ими.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Web-технологии в жизненном цикле» относится к дисциплинам блока ФТД.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Web-технологии в жизненном цикле» направлен на формирование следующих компетенций:

ПК-1 - Способен формализовать задачи по разработке модулей компонентов программных средств поддержки жизненного цикла изделия

ПК-3 - Способен применять методы моделирования в профессиональной деятельности

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-1	Знать основы языка PHP
	Уметь создавать веб-приложения на основе методов языка PHP
	Владеть основными средами web-программирования
ПК-3	Знать базовые команды MySQL
	Уметь работать с базой данных на MySQL
	Владеть основными методами управления phpMyAdmin

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Web-технологии в жизненном цикле» составляет 2 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		1
Аудиторные занятия (всего)	36	36
В том числе:		

Лекции	18	18
Лабораторные работы (ЛР)	18	18
Самостоятельная работа	36	36
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость:		
академические часы	72	72
зач.ед.	2	2

заочная форма обучения

Виды учебной работы	Всего часов	Семестры
		1
Аудиторные занятия (всего)	8	8
В том числе:		
Лекции	4	4
Лабораторные работы (ЛР)	4	4
Самостоятельная работа	60	60
Часы на контроль	4	4
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость:		
академические часы	72	72
зач.ед.	2	2

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Введение в backend разработку	Отличие back от front. Функции разработчика.	4	4	6	14
2	Язык PHP. Типы данных	Переменные. Операции присваивания. Математические операции. Специальные значения.	4	4	6	14
3	Циклы в PHP	Цикл foreach. Цикл for. Цикл while.	4	4	6	14
4	Массивы в PHP	Элементы массива. Ассоциативный массив. Ключи. Многомерные массивы.	2	2	6	10
5	Условия в PHP	Сокращенные типы записи. Операции сравнения. Равенства. Логические переменные.	2	2	6	10
6	Функции в PHP	Пользовательские функции. Функция echo. Функция var_dump	2	2	6	10
Итого			18	18	36	72

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Введение в backend разработку	Отличие back от front. Функции разработчика.	2	2	10	14
2	Язык PHP. Типы данных	Переменные. Операции присваивания. Математические операции. Специальные	2	2	10	14

		значения.				
3	Циклы в PHP	Цикл foreach. Цикл for. Цикл while.	-	-	10	10
4	Массивы в PHP	Элементы массива. Ассоциативный массив. Ключи. Многомерные массивы.	-	-	10	10
5	Условия в PHP	Сокращенные типы записи. Операции сравнения. Равенства. Логические переменные.	-	-	10	10
6	Функции в PHP	Пользовательские функции. Функция echo. Функция var_dump	-	-	10	10
Итого			4	4	60	68

5.2 Перечень лабораторных работ

1. Язык PHP. Типы данных:

- Изучение переменных. Операции присваивания
- Математические операции. Специальные значения.

2. Циклы в PHP:

- Цикл foreach. Цикл for. Цикл while.
- Вложения.

3. Массивы в PHP:

- Элементы массива. Ассоциативный массив.
- Ключи. Многомерные массивы.

4. Условия в PHP:

- Сокращенные типы записи. Операции сравнения.
- Равенства. Логические переменные.

5. Функции в PHP:

- Пользовательские функции. Функция echo.
- Функция var_dump

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины не предусматривает выполнение курсового проекта (работы) или контрольной работы.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

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

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность	Критерии оценивания	Аттестован	Не аттестован
-------------	---	---------------------	------------	---------------

	компетенции			
ПК-1	Знать основы языка PHP	укажите критерий	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь создавать веб-приложения на основе методов языка PHP	укажите критерий	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть основными средами web-программирования	укажите критерий	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-3	Знать базовые команды MySQL	укажите критерий	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь работать с базой данных на MySQL	укажите критерий	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть основными методами управления phpMyAdmin	укажите критерий	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 1 семестре для очной формы обучения, 1 семестре для заочной формы обучения по двухбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ПК-1	Знать основы языка PHP	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь создавать веб-приложения на основе методов языка PHP	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть основными средами web-программирования	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-3	Знать базовые команды MySQL	Тест	Выполнение теста на 70-100%	Выполнение менее 70%

	Уметь работать с базой данных на MySQL	Решение стандартных практических задач	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены
	Владеть основными методами управления phpMyAdmin	Решение прикладных задач в конкретной предметной области	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. В чем разница между конструкциями include() и require()?

Конструкция include, в отличие от require, позволяет включать файлы в код PHP скрипта во время выполнения сценария. Поясним это на примере, попробовав подключить 5 файлов 1.txt, 2.txt... 5.txt. Содержимое файлов - десятичные цифры от 1 до 5 (по одной цифре в каждом файле). Создадим следующий PHP-сценарий:

```
<?php
for($i=1; $i<=5; $i++) {
include "$i.txt";
}
// Выведет на экран 12345
?>
```

Т.е. каждый из файлов был включен по одному разу прямо во время выполнения цикла! Если же мы напишем require вместо include, то возникнет критическая ошибка (fatal error).

Отличие include от require также заключается в том, что require выдает Fatal error при невозможности подключения файла по любой причине. include выдаст Warning и продолжит работу.

2. Как мы можем получить IP-адрес клиента?

Этот вопрос может показать интервьюеру, насколько изобретателен и креативен кандидат, т.к. ответов может быть множество. Код \$_SERVER["REMOTE_ADDR"]; является самым простым решением, но вы можете написать настоящий сценарий, отвечая на этот вопрос.

3. В чем разница между unset() и unlink()?

unset() устанавливает переменную в "undefined", в то время как unlink() удаляет файл, который отправляется из файловой системы.

4. Что выведет на экран следующий код:

```
$a = '1';
$b = &$a;
$b = "2$b";
echo $a.", ".$b;
Ответ: 21, 21
```

5. Каковы основные типы ошибок в PHP и чем они отличаются?

В PHP существует три основных типа ошибок:

Notices (замечания)

Простые, некритические ошибки, которые произошли во время выполнения сценария. Примером Notices возникновения будет обращение к неопределенной переменной.

Warnings (предупреждения)

Более серьезные ошибки, чем Notices, однако выполнение сценария не прервется. Примером может быть подключение не существующего файла с помощью include().

Fatal (критические)

Этот тип ошибки вызывает прекращение выполнения сценария. Примером фатальной ошибки будет доступ к свойству несуществующего объекта или require() несуществующего файла.

6. В чем разница между GET and POST?

1. GET отправляет данные как часть URL, в то время как при POST эта информация не показывается, поскольку кодируется в запросе.

2. GET может обрабатывать максимум 2048 символов, POST не имеет таких ограничений.

3. GET работает только с ASCII данными, POST не имеет таких ограничений, двоичные данные также допускаются.

4. Обычно GET используется для получения данных, а POST для добавления и обновления.

7. Как вы включить сообщения об ошибках в PHP?

Установите display_errors = on в php.ini или объявите ini_set('display_errors', 1) в вашем сценарии. Затем, добавьте error_reporting(E_ALL) в ваш код, чтобы отобразить все типы сообщений об ошибках во время выполнения скрипта.

Включение сообщений об ошибках очень важно, особенно в процессе отладки, так как вы можете мгновенно узнать строку, которая производит ошибку, а также убедиться, что сценарий работает правильно.

8. Что такое Трейты (Traits)?

Трейты представляют собой механизм, который позволяет создавать повторно используемый код в таких языках, как PHP, где множественное наследование не поддерживается. Трейт не может быть создан сам по себе.

10. Может ли значение константы измениться во время выполнения скрипта?

Нет, значение константы не может быть изменено, если она уже была объявлена в ходе выполнения PHP-кода.

Можно ли вы расширить определенный класс Final?

Нет, это невозможно. Класс Final или объявление метода не позволяет дочернему классу или методу переопределять настройки.

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Что делают методы __construct() и __destruct() в PHP-классе?

Все объекты в PHP имеют встроенные методы конструктор и деструктор. Метод Конструктор вызывается сразу после того, как создается новый экземпляр класса и он используется для инициализации свойств класса. Метод Деструктор не принимает никаких параметров.

2. Как мы можем получить количество элементов в массиве?

Функция count() используется для возвращения количества элементов в массиве.

3. Как бы вы объявили функцию, которая принимает один параметр hello?

Если hello = true, то функция должна напечатать hello, но если функция не получает hello или hello = false, то функция должна напечатать bye.

```
<?php
function showMessage($hello = false){
    echo ($hello) ? 'hello' : 'bye';
}
?>
```

4. Значение переменной \$input представляет собой строку вида 1,2,3,4,5,6,7. Как подсчитать сумму чисел в этой строке?

```
<?php
    echo array_sum(explode(',',$input));
?>
```

Функция explode является одной из наиболее часто используемых функций в PHP, поэтому очень важно понять знает ли разработчик эту функцию. На этот вопрос нет однозначного ответа, но ответ должен быть похож на представленный выше.

5. Предположим, что у вас есть форма подписки на рассылку.

Эта форма имеет только одно текстовое поле input name="email". Как проверить является ли поле пустым и вывести сообщение "Электронная почта не может быть пустой" в этом случае.

```
<?php
if(empty($_POST['email'])){
    echo "Электронная почта не может быть пустой";
}
?>
```

6. Реализуйте класс с именем Dragonball с атрибутом ballCount, равным нулю, и метод iFoundaBall. Когда вызывается iFoundaBall, то ballCount увеличивается на единицу. Если значение ballCount равно семи, то должно появиться сообщение "Загадайте желание", а ballCount снова станет равен нулю.

```
<?php
class dragonBall{
    private $ballCount;

    public function __construct(){
        $this->ballCount=0;
    }

    public function iFoundaBall(){
        $this->ballCount++;
        if($this->ballCount===7){
            echo "Загадайте желание";
            $this->ballCount=0;
        }
    }
}
?>
```

7. Какие есть 3 области видимости в PHP ?

Private

Видна только в своем собственном классе

Public

Видна для любого другого кода, осуществляющего доступ к классу

Protected

Видна только родительских классов и классов, которые расширяют текущий класс

8. Что такое геттеры и сеттеры и для чего они нужны?

Геттеры и сеттеры – это методы, используемые для объявления или получения значений переменных, и, как правило, приватных. Они важны, т.к. позволяют

центральному расположению обрабатывать данные, объявляя и выводя их. Внутри геттера или сеттера вы можете последовательно обрабатывать данные, которые в конечном счете будут переданы в переменную или дополнительные функции. Примером этого может быть имя пользователя. Если вы не используете сеттера и просто объявите переменную \$userName вручную, чтобы у вас появился результат, например: "kevin", "KEVIN", "KeViN", "" и т.д. С сеттером вы можете не только изменить значение, например, ucfirst(\$userName), но вы также можете обрабатывать ситуации, когда данные не валидны, например, когда "" пропущено. То же самое относится и к геттерам - когда данные возвращаются, вы можете изменить результаты, чтобы включить strtoupper(\$userName) для правильного форматирования в дальнейшем.

9. Что означает MVC и что делает каждый компонент?

MVC расшифровывается как Model (Модель) View (Представление) Controller (Контроллер).

Контроллер обрабатывает данные, передаваемые в него Представлением, а также передает данные в Представление. Он отвечает за интерпретацию данных, отправленных Представлением и распространение этих данных к подходящим моделям для того, чтобы передать их обратно к Представлению.

Задача Модели состоит в том, чтобы обрабатывать отдельные задания, относящиеся к отдельным областям приложения или функционалу. Модели напрямую взаимодействуют с базой данных или другой системой хранения данных и обрабатывают бизнес-логику, относящуюся к результатам.

Представление – это данные, переданные контроллером и отображаемые пользователю.

10. Каким образом можно предотвратить появление следующего предупреждения 'Warning: Cannot modify header information – headers already sent' (Внимание: Не удастся изменить информацию в заголовке - заголовок уже отправлен) и почему оно появляется на первом месте?

Ничего не выводите в браузер до использования кода, который преобразует HTTP-заголовки. После того, как вы вызовете echo или любой другой код, который очищает буфер, вы больше не сможете присваивать «куки» или заголовки. Это также подходит к сообщениям об ошибках, поэтому если ошибка появляется до того, как вы используете команду заголовка и INI-директива display_errors установлена, это также выведет сообщение об ошибке.

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Что такое SQL-инъекции и каковы наиболее эффективные методы их предотвращения?

SQL-инъекции являются методом are a method to изменения а запроса в SQL-операторе, отправленном на сервер базы данных. Этот измененный запрос затем может пропускать такую информацию, как логин\пароль, и может поспособствовать хакерам подвергнуть сервер угрозе.

Чтобы предотвратить инъекции SQL необходимо постоянно проверять и удалять все вводимые пользователем данные. В PHP это часто забывается из-за простоты доступа к \$_GET и \$_POST, а также из-за невнимательности начинающих разработчиков. Но также существует множество методов, с помощью которых пользователи могут манипулировать переменными, используемыми в SQL, через файлы «cookies» и даже загруженные файлы. Единственный настоящий способ защиты – постоянное последовательное использование заготовленных операторов.

2. Что делает следующий код?

```
$i = 016;
```

echo \$i / 2;

Выведет 7. Ведущий нуль в PHP указывает на восьмеричное число, поэтому \$i вычисляется как десятичное число, равное 14, а не 16.

3. Почему используют === вместо ==?

Если вы хотите проверить какой-то определенный тип, например, целое число или логическое значение, то === будет делать именно то, что ожидается от конкретно набранной команды, в то время как == будет конвертировать данные лишь на время и попытается сопоставить оба типа операндов. Тожественный оператор (===) работает быстрее, поскольку не приходится иметь дело с преобразованием типов. Особенно при проверке переменных на истину\ложь лучше избегать == т.к. 0/1 или другая похожая форма также будет принята в расчет.

4. Что такое PSR?

PSR является набором рекомендаций по стандартам PHP, которые направлены на стандартизацию общих аспектов PHP-разработок. Примером PSR является PSR-2, который является своеобразным гидом по стилям кодирования.

5. Какому PSR стандарты вы следуете? Почему нужно следовать стандарту PSR?

Лучше следовать PSR, т.к. стандарты программирования часто варьируются между разработчиками и компаниями. Это может создать некоторые проблемы у разработчика при проверке или исправлении кода, написанного другим человеком. Стандарт PSR позволяет рационализировать ожидания от стиля написанного кода, и, соответственно, уменьшить вероятность недопонимания, а в некоторых случаях и синтаксических ошибок.

6. Используете ли вы Composer? Если да, то какие преимущества вы нашли в нем?

Использование Composer'a – это инструмент для управления зависимостями. Вы можете объявлять библиотеки, с которыми работаете, а Composer займется их установкой и обновлением. Еще одним преимуществом является последовательный метод управления библиотеками, с которыми вы работаете, так что вы потратите меньше времени на это самое управление, занимаясь своим проектом.

7. Логический оператор ИЛИ на языке php выражается символом:

- |
- + ||
- &&

8. Какие нововведения появились в php 5.6?

- + Выражения в константах, нефиксированные параметры в функциях
- Короткая запись массивов, константы классов
- API хеширование паролей, трейты, замыкания

9. В чем разница между print и echo?

- print – это конструкция, echo – это функция
- В объеме задействованной оперативной памяти при выполнении кода.

29. Обращаясь к функции print, интерпретатор задействует больше оперативной памяти, чем при обращении к функции echo

- + print – это функция, echo - это конструкция

10. Какие из перечисленных переменных являются суперглобальными?

- + \$_REQUEST, \$_SERVER, \$_FILES, \$_COOKIE, \$_GET
- \$_AMOUNT, \$_NULL, \$_DELETE, \$_VOID
- \$_TRAIT, \$_REQUEST, \$_GET, \$_SESSION

7.2.4 Примерный перечень вопросов для подготовки к зачету

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Язык программирования php нашел наиболее широкое применение в:

- Автоматизированном проектировании
- + Разработке web-приложений
- Создании и управлении базами данных

2. Какими разделителями окружены скрипты php скрипта?

- + <? php ... ?>
- <php> ... </php>
- </> php </>

3. Используя синтаксис языка php, напишите "Hello, World!":

- printf "Hello, World!"
- <p> Hello, World! </p>
- + echo "Hello, World!"

4. С какого символа начинаются все переменные в php?

- <>
- + \$
- !

5. К какому языку программирования синтаксически наиболее близок php?

- + C и Perl
- .Net
- VBScript

6. Перечислите основные типы данных в php:

- Boolean, float, logic, text, subject, NULL
- String, boolean, text, massive, logic, NULL
- + Boolean, integer, float, string, array, object, resource, NULL

7. В php используется ... типизация данных.

- + Динамическая
- Синтаксическая
- Коммутационная

8. Как правильно включить файл "time.inc"?

- + <? php include "time.inc"; ?>
- </> php include 'time.inc' <>

- <!-- include file="time.inc" -->

9. Укажите правильный способ создания функции в php:

- create newFunction()

- new_function newFunction()

+ function newFunction()

10. Укажите неверно заданное имя:

+ \$my-Var

- \$myVar

- \$my_Var

11. Укажите верно заданный массив с использованием синтаксиса языка php:

- \$months = array["September", "October", "November"]

- \$months = "September", "October", "November"

+ \$months = array("September", "October", "November")

12. В php файлы cookie задаются следующим образом:

+ setcookie()

- makecookie()

- createcookie()

13. Для написания комментариев в пределах работающего кода на php используется запись вида:

+ /*...*/

- <!--...-->

- <comment>...</comment>

14. Как правильно записать добавление 1 (единицы) к переменной \$count?

- \$count+1

+ \$count++

- \$count=+1

15. В php суперглобальная переменная, содержащая информацию о местоположении скриптов, путях, заголовках – это:

+ \$_SERVER

- \$_GLOBALS

- \$_GET

16. Укажите оператор, написанный на языке php, который делал бы файл "filename.txt" доступным для чтения.

+ fopen("filename.txt", "r")

- open("filename.txt", "read")

- fopen("filename.txt", "r+")

17. Каждый php-оператор должен заканчиваться:

- Двоеточием (:)

+ Точкой с запятой (;)

- Знаки препинания в php не используются

18.Какой способ использования тегов рекомендован для языка программирования php?

- <?

- <php

+ <? php

19.Укажите код, результатом выполнения которого является вывод на экран имени (переменная "name").

- ?php

```
$name++'Mike';
```

```
$age++'37';
```

```
echo $age;
```

?

- //?php

```
?name = Mike;
```

```
?age = 37;
```

```
echo $name;
```

?//

+ <? php

```
$name = 'Mike';
```

```
$age = 37;
```

```
echo $name;
```

?>

20.Вызов функции из самой себя – это:

+ Рекурсия

- Регрессия

- Массив

21.Выберите верное утверждение, характеризующее множественное наследование в php.

- Множественное наследование задается через подключение родственных классов. Для этого используется функция _multi()

+ В php не существует понятия «множественное наследование», т.е. у класса может быть лишь один родитель. Эмуляция множественного наследования задается функцией _call () или трейтами

- Организовать наследование позволяет метод _constructor(), создающий родственные экземпляры базового класса

22.Имеются следующие варианты написания кода: (1) <?php echo 'Выполнить документ';?> и (2) <?= 'Выполнить документ' ?>. Какое из приведенных ниже утверждений верное?

+ Записи (1) и (2) представляют собой одинаковый код. Разница в том, что (1) – это

стандартный синтаксис, (2) – краткий. В php приемлемы оба варианта

- Запись (1) представляет собой код, который выведет на экран фразу ‘Выполнить документ’, (2) – это комментарий к коду, который никак не отобразится у конечных пользователей

- Запись (1) – это программа, написанная на языке php, (2) – на языке Java. Результирующий вывод при их исполнении одинаковый

Какой тип данных следует использовать для выражения “-42”?

- boolean

- string

+ int

23. Для чего используется ключевое слово global?

+ Для получения доступа к глобальной переменной, которая находится внутри функции

- Для подключения библиотек из глобального каталога

- Для указания того, что переменная будет многократно использоваться

24. Расставьте слова (1) echo (2) “Robert” (3) \$name= (4) \$name в таком порядке, чтобы на экран выводилось имя, переданное переменной name.

+ 3, 2, 1, 4

- 1, 3, 2, 4

- 3, 1, 4, 2

25. Какой результат выведет следующий код: \$x = 'y'; \$y = 'x'; echo \$\$x; ?

- x+y

- x*y

+ x

7.2.5 Примерный перечень заданий для подготовки к экзамену

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Оценка «Не зачтено» ставится в случае, если студент набрал менее 10 баллов.

2. Оценка «Зачтено» ставится в случае, если студент набрал от 11 до 20 баллов

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Введение в backend разработку	ПК-1, ПК-3	Тест, защита лабораторных работ

2	Язык PHP. Типы данных	ПК-1, ПК-3	Тест, защита лабораторных работ
3	Циклы в PHP	ПК-1, ПК-3	Тест, защита лабораторных работ
4	Массивы в PHP	ПК-1, ПК-3	Тест, защита лабораторных работ
5	Условия в PHP	ПК-1, ПК-3	Тест, защита лабораторных работ
6	Функции в PHP	ПК-1, ПК-3	Тест, защита лабораторных работ

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Основная литература

Эл. Web-технологии: Сычев А.В. Web-технологии [Электронный ресурс]: учебное пособие/ Сычев А.В.— Электрон. текстовые данные.— Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.— 407 с.— Режим доступа: <http://www.iprbookshop.ru/89412.html>.— ЭБС «IPRbooks»

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

Лицензионное программное обеспечение

1. Microsoft Office 2013/2017

Ресурсы информационно-телекоммуникационной сети «Интернет»

1. <http://edu.ru>

2. Образовательный портал ВГТУ

Информационные справочные системы

1. <http://windows.edu.ru>

2. <http://wiki.cchgeu.ru>

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Лаборатории 208/2 или 210/2, расположенные по адресу г. Воронеж,
ул. Плехановская, д. 11

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Web-технологии в жизненном цикле» читаются лекции, проводятся лабораторные работы.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоения учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов;

	- работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом три дня эффективнее всего использовать для повторения и систематизации материала.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	----------------------------	--