

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
функционального анализа
и операторных уравнений



Каменский М.И.

подпись, расшифровка подписи

20.03.2025

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.В.13 Программирование для Интернет

- 1. Шифр и наименование направления подготовки:** 01.03.04 Прикладная математика
- 2. Профиль подготовки:** Применение математических методов к решению инженерных и экономических задач
- 3. Квалификация выпускника:** бакалавр
- 4. Форма образования:** очная
- 5. Кафедра, отвечающая за реализацию дисциплины:** кафедра функционального анализа и операторных уравнений математического факультета
- 6. Составители программы:** Груздев Денис Владиславович, преподаватель математического факультета, кафедры функционального анализа и операторных уравнений
- 7. Рекомендована:** научно-методическим советом математического факультета, 18.03.2025 Протокол №0500-03
- 8. Учебный год:** 2028-2029 **Семестр:** 8

9. Цели и задачи учебной дисциплины:

Целью изучения дисциплины «Программирование для Интернет» является изучение современных методов программирования приложений, использующих в своей работе среду Internet, а также создания интернет сайтов, наполненных актуальным и динамически изменяющимся контентом.

Задачи курса:

- освоить основные понятия компьютерных сетей и систем телекоммуникации, основы объектно-ориентированного подхода к разработке программного обеспечения;
- научиться ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий, программировать на одном из алгоритмических языков;
- познакомиться с основами алгоритмизации и разработками программного обеспечения.

10. Место учебной дисциплины в структуре ООП:

Блок 1, вариативная часть. Данная дисциплина базируется на компетенциях, полученных при изучении дисциплин «Технология программирования», «Практикум на ЭВМ», «Объектно-ориентированное программирование».

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников): ПК-1.1; ПК-1.2; ПК-1.3

| Код | Название компетенции | Код(ы) | Индикатор(ы) | Планируемые результаты обучения |
|------|--|--------|--|---|
| ПК-1 | Способен выявлять естественно-научную сущность проблем, возникающих при решении инженерных и экономических задач | ПК-1.1 | Обладает базовыми знаниями в области математических наук, программирования и информационных технологий | <p>Знать: – методы анализа и описания предметной области в терминах модели сущность-связь, выбора исходных данных для проектирования, методы и средства построения физической реляционной схемы базы данных, языковые средства описания и манипулирования данными;</p> <p>– общие механизмы обеспечения целостности и безопасности, связанные с управлением информацией в базах данных, эффективного использования этих механизмов.</p> <p>Уметь: – описывать предметную область в понятиях модели сущность-связь, применять методы и средства построения физической реляционной схемы базы данных, практически использовать языковые средства описания и манипулирования данными;</p> <p>– применять механизмы обеспечения целостности и безопасности инфор-</p> |

| | | | |
|--|--------|---|--|
| | | | <p>магии в базах данных.</p> <p>Владеть: – практическими навыками предпроектного обследования произвольной предметной области, навыками построения физической реляционной схемы базы данных и использования языка SQL для создания спецификации базы данных;</p> <p>– навыками использования систем управления базами данных при решении стандартных задач профессиональной деятельности.</p> |
| | ПК-1.2 | <p>Умеет собирать, обрабатывать, анализировать результаты исследований, полученных при решении инженерных и экономических задач</p> | <p>знать: современные прикладные программные средства, современные языки программирования</p> <p>уметь: осваивать современные технологии программирования, демонстрировать знания современных языков программирования, операционных систем</p> <p>владеть (иметь навык(и)): современными прикладными программными средствами, информационно-телекоммуникационной сети "Интернет", способов и механизмов управления данными, принципов организации, состава и схемы работы операционных систем.</p> |
| | ПК-1.3 | <p>Имеет практических опыт научно-исследовательской деятельности в математике и информатике</p> | <p>знать: современные языки программирования и алгоритмы при решении практических задач</p> <p>уметь: применять языки программирования для решения задач в научно-исследовательской деятельности в математике и информатике</p> <p>владеть (иметь навык(и)): применять знания и навыки управления информацией</p> |

12. Объем дисциплины в зачетных единицах/часах в соответствии с учебным планом — 3/108.

Форма промежуточной аттестации — Зачет

13. Виды учебной работы

| Виды учебной работы | Трудоемкость | | |
|---------------------|--------------|--------------|--|
| | Всего | По семестрам | |
| | | № семестра 8 | |

| | | | |
|---|-----|-----|--|
| Аудиторные занятия | 52 | 52 | |
| В том числе: лекции | 26 | 26 | |
| практические | | | |
| лабораторные | 26 | 26 | |
| Самостоятельная работа | 56 | 56 | |
| Форма промежуточной аттестации (зачет – 0 час./экзамен – <u>36</u> час.) | | | |
| Итого: | 108 | 108 | |

13.1 Содержание разделов дисциплины

| п/п | Наименование раздела дисциплины | Содержание раздела дисциплины |
|--------------------------------|---|--|
| 1. Лекции | | |
| 1.1 | PHP. Вводная часть | Установка PHP, MySQL, Apache. Возможности, гибкость, история развития, примеры скриптов. Первая программа. PHP и HTML |
| 1.2 | Изучение языка PHP | Переменные в PHP, константы, типы данных, массивы, объекты, выражения в PHP, операторы, функции, логические операторы, строковые операторы |
| 1.3 | Конструкции языка PHP | Условия, циклы, выбор, конструкции включения (include, require), конструкции возврата значений, пользовательские функции |
| 1.4 | Основы ООП | Классы и объекты в PHP, доступ к классам и объектам, наследование, полиморфизм, |
| 1.5 | Методы передачи GET и POST. Формы HTML | Запрос GET, запрос POST, Использование форм HTML, Элементы форм, Обработка формы |
| 1.6 | База данных MySQL | Настройка базы данных. Денвер в phpMyAdmin. MySQL в Denwer. Настройка Денвера. Denwer phpMyAdmin. Подключение базы данных в PHP. |
| 1.7 | Основная информация о MySQL | Поля и типы данных, операторы и команды, INSERT, UPDATE, DELETE, SELECT. SQL запросы. |
| 1.8 | Функции PHP для работы с MySQL | Функции PHP для работы с MySQL. Настройка MySQL и функции |
| 1.9 | Манипуляция с данными при использовании PHP и MySQL | Поиск, выборки, вставка, функции работы с массивами, функции обработки строк, изменение данных, удаление данных |
| 1.10 | Изображение и хранение изображений | Вставка и создание изображений, |
| 1.11 | Работа с файлами и каталогами | Открытие, обработка файлов, функции работы с каталогами |
| 1.12 | Дата-время, корректность данных | Функции работы с датой и временем, корректный ввод данных, |
| 1.13 | Проектировка и создание сайта | Выбор информационного содержания сайта, проектировка, программирование, создание дизайна |
| 2. Практические занятия | | |
| 2.1 | | |
| 2.2 | | |
| 3. Лабораторные работы | | |
| 3.1 | PHP. Вводная часть | Установка PHP, MySQL, Apache. Возможности, гибкость, история развития, примеры скриптов. Первая программа. PHP и HTML |
| 3.2 | Изучение языка PHP | Переменные в PHP, константы, типы данных, массивы, объекты, выражения в PHP, операторы, функции, логические |

| | | |
|------|---|--|
| | | операторы, строковые операторы |
| 3.3 | Конструкции языка PHP | Условия, циклы, выбор, конструкции включения (include, require), конструкции возврата значений, пользовательские функции |
| 3.4 | Основы ООП | Классы и объекты в PHP, доступ к классам и объектам, наследование, полиморфизм, |
| 3.5 | Методы передачи GET и POST. Формы HTML | Запрос GET, запрос POST, Использование форм HTML, Элементы форм, Обработка формы |
| 3.6 | База данных MySql | Настройка базы данных. Денвер в phpMyAdmin. MySQL в Denwer. Настройка Денвера. Denwer phpMyAdmin. Подключение базы данных в PHP. |
| 3.7 | Основная информация о MySql | Поля и типы данных, операторы и команды, INSERT, UPDATE, DELETE, SELECT. SQL запросы. |
| 3.8 | Функции PHP для работы с MySql | Функции PHP для работы с MySql. Настройка MySQL и функции |
| 3.9 | Манипуляция с данными при использовании PHP и MySql | Поиск, выборки, вставка, функции работы с массивами, функции обработки строк, изменение данных, удаление данных |
| 3.10 | Изображение и хранение изображений | Вставка и создание изображений, |
| 3.11 | Работа с файлами и каталогами | Открытие, обработка файлов, функции работы с каталогами |
| 3.12 | Дата-время, корректность данных | Функции работы с датой и временем, корректный ввод данных, |
| 3.13 | Проектировка и создание сайта | Выбор информационного содержания сайта, проектировка, программирование, создание дизайна |
| 3.14 | Форма аттестации | Контрольная работа |
| 3.15 | Форма аттестации | Зачет |
| | | |
| | | |
| | | |

13.2 Разделы дисциплины и виды занятий

| № п/п | Наименование раздела дисциплины | Виды занятий (часов) | | | | |
|-------|---|----------------------|--------------|--------------|------------------------|-------|
| | | Лекции | Практические | Лабораторные | Самостоятельная работа | Всего |
| 1 | PHP. Вводная часть | 1 | | 1 | 4 | 6 |
| 2 | Изучение языка PHP | 2 | | 2 | 4 | 8 |
| 3 | Конструкции языка PHP | 2 | | 2 | 4 | 8 |
| 4 | Основы ООП | 1 | | 1 | 4 | 6 |
| 5 | Методы передачи GET и POST. Формы HTML | 2 | | 2 | 4 | 8 |
| 6 | База данных MySql | 2 | | 2 | 8 | 12 |
| 7 | Основная информация о MySql | 2 | | 2 | 4 | 8 |
| 8 | Функции PHP для работы с MySql | 2 | | 2 | 4 | 8 |
| 9 | Манипуляция с данными при использовании PHP и MySql | 2 | | 2 | 4 | 8 |
| 10 | Изображение и хранение изображений | 2 | | 2 | 4 | 8 |
| 11 | Работа с файлами и каталогами | 2 | | 2 | 4 | 8 |
| 12 | Дата-время, корректность данных | 2 | | 2 | 4 | 8 |
| 13 | Проектировка и создание сайта | 4 | | 4 | 4 | 12 |
| | | | | | | |

14. Методические указания для обучающихся по освоению дисциплины

Аудиторные занятия, лекции и лабораторные занятия, предполагают самостоятельную работу студентов по данному курсу. Работа с конспектами лекций. Ряд тем выносятся для самостоятельного изучения, предлагаются для самостоятельного построения некоторые алгоритмы. Предусмотрены практические домашние задания и оформление отчетов выполнения лабораторных заданий, а также дополнительные задания для сильных студентов.

В процессе преподавания дисциплины используются такие виды учебной работы, как лекции, практические занятия, а также различные виды самостоятельной работы обучающихся. На лекциях рассказывается теоретический материал, на лабораторных занятиях решаются примеры по теоретическому материалу, прочитанному на лекциях.

При изучении курса «Программирование для Интернет» обучающимся следует внимательно слушать и конспектировать материал, излагаемый на аудиторных занятиях. Для его понимания и качественного усвоения рекомендуется следующая последовательность действий.

1. После каждой лекции студентам рекомендуется подробно разобрать прочитанный теоретический материал, выучить все определения и формулировки теорем, разобрать примеры, решенные на лекции. Перед следующей лекцией обязательно повторить материал предыдущей лекции.

2. Перед практическим занятием обязательно повторить лекционный материал. После практического занятия еще раз разобрать решенные на этом занятии примеры, после чего приступить к выполнению домашнего задания. Если при решении примеров, заданных на дом, возникнут вопросы, обязательно задать на следующем практическом занятии или в присутственный час преподавателю.

3. При подготовке к практическим занятиям повторить основные понятия по темам, изучить примеры. Решая задачи, предварительно понять, какой теоретический материал нужно использовать. Наметить план решения, попробовать на его основе решить практические задачи.

3. Выбрать время для работы с литературой по дисциплине в библиотеке – каждый вторник, с 15:00 до 17:00.

Самостоятельная учебная деятельность студентов по дисциплине «**Программирование для интернет**» предполагает изучение рекомендуемой преподавателем литературы по вопросам лекционных и практических занятий (приведены выше), самостоятельное освоение понятийного аппарата и подготовку к текущим аттестациям (**выполнению практических заданий**) (примеры см. ниже).

Вопросы лекционных и практических занятий обсуждаются на занятиях в виде устного опроса – индивидуального и фронтального. При подготовке к лекционным и практическим занятиям, обучающимся важно помнить, что их задача, отвечая на основные вопросы плана занятия и дополнительные вопросы преподавателя, показать свои знания и кругозор, умение логически построить ответ, владение математическим аппаратом и иные коммуникативные навыки, умение отстаивать свою профессиональную позицию. В ходе устного опроса выявляются детали, которые по каким-то причинам оказались недостаточно осмысленными студентами в ходе учебных занятий. Тем самым опрос выполняет важнейшие обучающую, развивающую и корректирующую функции, позволяет студентам учесть недоработки и избежать их при подготовке к промежуточным аттестациям.

Все выполняемые студентами самостоятельно задания (выполнение контрольной работы и практических заданий) подлежат последующей проверке преподавателем.

давателем. Результаты текущих аттестаций учитываются преподавателем при проведении промежуточной аттестации (**8 семестр – зачет**).

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

| № п/п | Источник |
|-------|--|
| 1 | <i>Бенкен, Елена Сергеевна. PHP, MySQL, XML. Программирование для Интернета / Елена Бенкен .— СПб : БХВ-Петербург, 2007 .— 310 с.</i> |
| 2 | <i>Ли, Джеймс. Использование Linux, Apache, MySQL и PHP для разработки Web-приложений : пер. с англ. / Джеймс Ли, Брент Уэр .— М. : Вильямс, 2004 .— 429 с.</i> |
| 3 | <i>Дронов, Владимир Александрович. PHP, MySQL и Dreamweaver MX 2004 : разработка интерактивных Web-сайтов / Владимир Дронов .— СПб : БХВ-Петербург, 2005 .— 433 с.</i> |

б) дополнительная литература:

| № п/п | Источник |
|-------|---|
| 4 | <i>Костарев, Алексей Ф. PHP в Web-дизайне / Алексей Костарев .— СПб. и др. : БХВ-Петербург, 2002 .— 591 с.</i> |
| 5 | <i>Веллинг, Люк. Разработка Web-приложений с помощью PHP и MySQL : Пер. с англ. / Л. Веллинг, Л. Томсон .— 2-е изд. — М. и др. : Вильямс, 2003 .— 795 с.</i> |
| 6 | <i>Конверс, Тим. PHP 5 и MySQL : библия пользователя / Тим Конверс, Джойс Парк, Кларк Морган ; [пер. с англ. К.А. Птицын] .— М. [и др.] : Диалектика, 2007 .— 1207 с.</i> |

в) информационные электронно-образовательные ресурсы:

| № п/п | Источник |
|-------|---|
| 7 | Университетская библиотека ONLINE http://biblioclub.ru/ |
| 8 | Электронная библиотека ЗНБ ВГУ https://lib.vsu.ru/ |

16. Перечень учебно-методического обеспечения для самостоятельной работы

| № п/п | Источник |
|-------|--|
| 1 | Методическое пособие «PHP и MySQL» |
| 2 | Интернет материал «PHP в задачах» |
| 3 | Положение об организации самостоятельной работы обучающихся в Воронежском государственном университете |

17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение информационно-справочные системы

Ubuntu (бесплатное и/или свободное ПО, лицензия:

<https://ubuntu.com/download/desktop>);

LibreOffice (GNU Lesser General Public License (LGPL), бесплатное и/или свободное ПО, лицензия: <https://ru.libreoffice.org/about-us/license/>);

Denwer (бесплатное и/или свободное ПО, лицензия:

<http://www.denwer.ru/faq/other.html>);

Mozilla Firefox (Mozilla Public License (MPL), бесплатное и/или свободное ПО, лицензия: <https://www.mozilla.org/en-US/MPL/>);

Образовательный портал <https://edu.vsu.ru/course/view.php?id=12002>

18. Материально-техническое обеспечение дисциплины:

Специализированная мебель

Специализированная мебель, маркерная доска, персональные компьютеры

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

| Код и содержание компетенции (или ее части) | Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенции посредством формирования знаний, умений, навыков) | Этапы формирования компетенции (разделы (темы) дисциплины или модуля и их наименование) | ФОС* (средства оценивания) |
|--|--|---|----------------------------|
| ПК-1.2 Умеет собирать, обрабатывать, анализировать результаты исследований, полученных при решении инженерных и экономических задач | Знать: – методы анализа и описания предметной области в терминах модели сущность-связь, выбора исходных данных для проектирования, методы и средства построения физической реляционной схемы базы данных, языковые средства описания и манипулирования данными; – общие механизмы обеспечения целостности и безопасности, связанные с управлением информацией в базах данных, эффективного использования этих механизмов. Уметь: – описывать предметную область в понятиях модели сущность-связь, применять методы и средства построения физической реляционной схемы базы данных, практически использовать языковые средства описания и манипулирования данными; – применять механизмы обеспечения целостности и | Раздел 1-13. | Практическое задание |
| | | Раздел 1-13. | Практическое задание |

| | | | |
|---|---|--------------|----------------------|
| | <p>безопасности информации в базах данных.</p> <p>Владеть: – практическими навыками предпроектного обследования произвольной предметной области, навыками построения физической реляционной схемы базы данных и использования языка SQL для создания спецификации базы данных;</p> <p>– навыками использования систем управления базами данных при решении стандартных задач профессиональной деятельности.</p> | | |
| <p>ПК-1.1</p> <p>Обладает базовыми знаниями в области математических наук, программирования и информационных технологий</p> | <p>знать: современные прикладные программные средства, современные языки программирования</p> <p>уметь: осваивать современные технологии программирования, демонстрировать знания современных языков программирования, операционных систем</p> <p>владеть (иметь навык(и)): современными прикладными программными средствами, информационно-телекоммуникационной сети "Интернет", способов и механизмов управления данными, принципов организации, состава и схемы работы операционных систем.</p> | Раздел 1-13. | Практическое задание |
| | | Раздел 1-13. | Практическое задание |
| | | Раздел 1-13. | Практическое задание |
| <p>ПК-1.3</p> <p>готовностью применять знания и навыки управления информацией</p> <p>Имеет практический опыт научно-исследовательской деятельности в математике и информатике</p> | <p>знать: современные языки программирования и алгоритмы при решении практических задач</p> <p>уметь: применять языки программирования для решения задач в научно-исследовательской деятельности в математике и информатике</p> <p>владеть (иметь навык(и)): применять знания и навыки</p> | Раздел 1-13. | Практическое задание |
| | | Раздел 1-13. | Практическое задание |
| | | Раздел 1-13. | Контрольная работа |

| | | | |
|--|------------------------|--|--|
| | управления информацией | | |
|--|------------------------|--|--|

20. Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1. Текущий контроль успеваемости

Для оценивания результатов обучения на зачете используются следующие показатели (ЗУНы из 19.1):

- 1) знание учебного материала и владение понятийным аппаратом;
 - 2) умение связывать теорию с практикой;
 - 3) умение иллюстрировать ответ примерами, фактами, данными научных исследований;
 - 4) умение применять знания, решать поставленные задачи;
 - 5) владение способами программирования для интернет;
- владение понятийным аппаратом данной области науки (теоретическими основами дисциплины), способность иллюстрировать ответ примерами, фактами, данными научных исследований, применять теоретические знания для решения практических задач

Для оценивания результатов обучения на экзамене (зачете с оценкой) используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Для оценивания результатов обучения на зачете используется – зачтено, не зачтено
Соотношение показателей, критериев и шкалы оценивания результатов обучения.

| Критерии оценивания компетенций | Уровень сформированности компетенций | Шкала оценок |
|---|--------------------------------------|-------------------|
| Обучающийся знает основы работы в среде SQL, умеет решать предложенные задачи с использованием языка PHP. Успешно защищает проектное задание и дает правильные ответы на дополнительные вопросы | <i>Пороговый</i> | <i>Зачтено</i> |
| Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки при защите проектного задания на зачете и не дает правильные ответы на дополнительные вопросы. | – | <i>Не зачтено</i> |

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

Перечень вопросов к экзамену (зачету): (нужное выбрать)

Создать сайт со следующими возможностями:

- Регистрацией новых пользователей.
- Выводом данных пользователей в окне браузера
- Добавление новых пользователей
- Изменение данных уже существующих пользователей
- Удаление данных пользователей

Контрольные вопросы для оценки качества освоения учебной дисциплины

Большинство вопросов приведено на английском языке. В скобках в конце вопроса уточняется, к какой предметной области разработки, средству или языку относится вопрос.

1. Which of the following statements is NOT true? (PHP)
 - a) Class constants are public
Class constants are being inherited
 - a) Class constants can omit initialization (default to NULL)
 - b) Class constants can be initialized by consts

1. Type hinting in PHP allows the identification of the following variable types? (PHP)
 - a) String
 - Integer
 - Array
 - Any class or interface type

2. Identify the security vulnerability in the following example: (PHP)


```
echo "Welcome, {$_POST['name']}. ";
```

 - a) SQL Injection
 - Cross-Site Scripting
 - Remote Code Injection
 - None of the above

3. How many times will the function counter() be executed in the following code? (PHP)


```
function counter($start, &$stop)
{
    if ($stop > $start)
    {
        return;
    }
    counter($start--, ++$stop);
}
$start = 5;
$stop = 2;
counter($start, $stop);
```

 - a) 3
 - a) 4
 - a) 5
 - a) 6

4. When a class is defined as final it: (PHP)
 - a) Can no longer be extended by other classes.
Means methods in the class are not over-loadable.
Cannot be defined as such, final is only applicable to object methods.
Is no longer iterable.

5. What is the content of \$c after the following code has executed? (PHP)

```
$a = 2;
$b = 3;
$c = ($a++ * ++$b);
```

- a) 0
- a) 5
- a) 8
- a) 4

6. The function call returns "0". What does that mean? (PHP)

```
strcasecmp('hello my dear!', 'Hello my DEAR!');
```

- a) String 1 is less than string 2.
- a) The strings are considered equal.
- a) String 2 is less than string 1.
- a) The strings have equal length.

7. What is the output of the following script? (PHP)

```
class a
{
    public $val;
}
```

```
function renderVal (a $a)
{
    if ($a) {
        echo $a->val;
    }
}
```

```
renderVal (null);
```

- a) A syntax error in the function declaration line
- a) An error, because null is not an instance of 'a'
- a) Nothing, because a null value is being passed to renderVal()
- a) NULL

8. What is the output of the following code? (PHP)

```
echo 0x33, ' monkeys sit on ', 011, ' trees.';
```

- a) 33 monkeys sit on 11 trees.
- a) 51 monkeys sit on 9 trees.
- a) monkeys sit on trees.
- a) 0x33 monkeys sit on 011 trees.

9. When checking whether two English words are pronounced alike, which function should be used for the best possible result? (PHP)

- a) levenshtein()
- a) metaphone()
- a) similar_text()
- a) soundex()

10. Given the following code, what is correct? (PHP)

```
function f(stdClass &$x = NULL)
{
    $x = 42;
}
```

```
$z = new stdClass;
f($z);
var_dump($z);
```

- a) Error: Typehints cannot be NULL
- a) Error: Typehints cannot be references
- a) Result is NULL
- a) Result is object of type stdClass
- a) Result is 42

11. What is the maximum size of the VARCHAR column type? (MySQL)

- a) 255 Bytes
- a) 255 Characters
- a) 12 Bytes
- a) 512 Characters
- a) No Limit

12. Transactions can be used to: (Data Bases)

Recover from errors in case of a power outage or a failure in the SQL connection

- a) Ensure that the data is properly formatted
- a) Ensure that either all statements are performed properly, or that none of them are.
- a) Recover from user errors

13. Which piece of code will return the ASCII value of a character? (MySQL)

- a) (int)'t';
- a) ord('t');
- a) to_ascii('t');
- a) chr('t');

14. An HTML form contains this form element: (PHP)

```
<input type="image" name="myImage" src="image.png" />
```

The user clicks on the image to submit the form. How can you now access the relative coordinates of the mouse click?

- a) \$_IMAGE['myImage']['x'] and \$_IMAGE['myImage']['y']
- a) \$_POST['myImage']['x'] and \$_POST['myImage']['y']
- a) \$_POST['myImage.x'] and \$_POST['myImage.y']
- a) \$_POST['myImage_x'] and \$_POST['myImage_y']

15. You are creating an application that generates invoices in a variety of formats, including PDF, ODS and HTML. Each of these formats is represented as a PHP class in your application. While some of the operations can be performed on all of the different formats (such as saving and loading), other operations may be specific to one or two of the formats (such as setting as read only). Which design pattern should you use for this application? (System Design)

- a) Adapter
- a) Factory
- a) MVC
- a) Singleton

16. Как будет выглядеть код страницы после выполнения кода: (Smarty)

```
{assign var="groupID" value=$currentUser->getUserGroup()}
{if($groupID) > 1}{assign var="groupID" value=10}
```

```

    {if $groupID = 1}
        <div>You are an Admin</div>
    {else}
        <div>You are a Customer</div>
    {/if}
{/if}

```

- a) <div>You are an Admin</div> или <div>You are a Customer</div> в зависимости от переменной groupID
 Всегда <div>You are an Admin</div>
 Всегда <div>You are a Customer</div>

17. На каком языке написан следующий код:

```

function clearForm(name) {
    var f = document.forms[name];
    for(var i = 0; i < f.elements.length; ++i) {
        if(f.elements[i].checked)
            f.elements[i].checked = false;
    }
}

```

- a) PHP
 a) Java
 a) JavaScript

18. Какой результат выполнения скрипта? (JavaScript):

```

<script type="text/javascript" src="myscript.js">
    alert('I am pointless as I won't be executed');
</script>

```

- a) I am pointless as I won't be executed
 a) I am pointless as I won't be executed
 a) Результат выполнения скрипта из файла myscript.js
 a) Код содержит ошибки

20.2. Промежуточная аттестация

Вариант 1

- Ввести три числа. Если они могут быть длинами сторон прямоугольного треугольника, вывести их в порядке возрастания, вычислить площадь полученного треугольника.
- Запрограммировать математическую формулу
- Результат сохранить в базе данных

Вариант 2

- Ввести три числа. Если они могут быть длинами сторон остроугольного треугольника, вывести их в порядке убывания, вычислить площадь полученного треугольника.
- Запрограммировать математическую формулу
- Результат сохранить в базе данных

Вариант 3

- Ввести три числа. Если они могут быть длинами сторон тупоугольного треугольника, вывести их в порядке убывания, вычислить площадь полученного треугольника.
- Запрограммировать математическую формулу
- Результат сохранить в базе данных

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Текущий контроль представляет собой проверку усвоения учебного материала теоретического и практического характера, регулярно осуществляемую на занятиях.

Задание для текущего контроля и проведения промежуточной аттестации должны быть направлены *на оценивание*:

1. уровня освоения теоретических и практических понятий, научных основ профессиональной деятельности;
2. степени готовности обучающегося применять теоретические и практические знания и профессионально значимую информацию, сформированности когнитивных умений.
3. приобретенных умений, профессионально значимых для профессиональной деятельности.

Текущий контроль предназначен для проверки хода и качества формирования компетенций, стимулирования учебной работы обучаемых и совершенствования методики освоения новых знаний. Он обеспечивается проведением контрольных заданий и проведением коллоквиумов.

При текущем контроле уровень освоения учебной дисциплины и степень сформированности компетенции определяются оценками «зачтено» и «незачтено».

В ходе контрольной работы обучающемуся выдается КИМ с практическим перечнем заданий и предлагается решить данные задания.

Если текущая аттестация проводится в дистанционном формате, то у обучающийся обязательно должен иметь компьютер, микрофон, камеру. Если у обучающегося отсутствует необходимое оборудование, то он обязан сообщить преподавателю об этом за 3 суток. На контрольную работу в дистанционном режиме отводится ограничение по времени в 90 минут.

Промежуточная аттестация предназначена для определения уровня освоения всего объема учебной дисциплины. Промежуточная аттестация по дисциплине «Программирование для Интернет» проводится в форме зачета.

Промежуточная аттестация, как правило, осуществляется в конце семестра. Результаты текущей аттестации обучающегося по решению кафедры могут быть учтены при проведении промежуточной аттестации. При несогласии студента, ему дается возможность пройти промежуточную аттестацию (без учета его текущих аттестаций) на общих основаниях.

20.3 Фонд оценочных средств сформированности компетенций студентов, рекомендуемый для проведения диагностических работ

Программирование для интернет

Закрытые вопросы:**1. Дана программа на PHP:**

```
<?php
    $nums1 = [5, 10, 15];
    $nums2 = [20, 25, 30];
    $nums3 = [35, 40, 45];
    $result = array_merge($nums1, $nums2, $nums3);
?>
```

Определите, что делает функция **array_merge**?

- A. преобразовывает массивы
- B. объединяет массивы
- C. разъединяет массивы

Правильный ответ: B. объединяет массивы

2. Дана программа на PHP:

```
<?php
    $status = 1;
    switch($status)
    {
        case 1:
            echo 'Статус: Обрабатывается<br>';
        case 2:
            echo 'Статус: Передан в доставку<br>';
            break;
        case 3:
            echo 'Статус: Выполнен<br>';
            break;
    }
?>
```

Определите, что делает команда **break**?

- A. сокращает условие
- B. сравнивает значения
- C. прерывает текущее выполнение switch

Правильный ответ: C. прерывает текущее выполнение switch

3. Дана программа на PHP:

```
<?php
    $str = 'Это ';
    $str .= 'очень ';
    $str .= 'длинная ';
    $str .= 'строка.';
    echo $str;
?>
```

Какой здесь используется оператор?

- A. оператор конкатенации
- B. тернарный оператор

С. оператор сравнения

Правильный ответ: А. оператор конкатенации

4. Как называется этот оператор ++\$x:

А. пост-инкремент

В. пре-инкремент

С. пост-декремент

Правильный ответ: В. пре-инкремент

5. Что выведет следующая программа:

```
<?php
    $i = 0;
    do
    {
        $i++;
        echo $i;
    }
    while($i < 5);
?>
```

А. 12345;

В. 54321;

С. 1;

Д. 5.

Правильный ответ: А. 12345;

6. Что дает данный оператор \$a <=> \$b?

А. Истина, если \$a меньше или равна \$b

В. Истина, если \$a больше или равна \$b

С. Возвращает -1, 0 или 1 когда \$a соответственно меньше, больше или равно \$b

Правильный ответ: С. Возвращает -1, 0 или 1 когда \$a соответственно меньше, больше или равно \$b

7. Что обозначает \$a || \$b ?

А. Истина, если одна из \$a и \$b равна true, но не обе.

В. Истина, если \$a и \$b равны true.

С. Истина, если хотя бы одна из \$a и \$b равна true.

Правильный ответ: С. Истина, если хотя бы одна из \$a и \$b равна true.

8. Что такое оператор конкатенации?

А. более короткий аналог IF

В. оператор для склеивания нескольких значений в одну строку

- C. оператор увеличения значения переменной на единицу
- D. оператор уменьшения значения переменной на единицу

Правильный ответ: В. оператор для склеивания нескольких значений в одну строку

9. Что выведет данная программа?

```
<?php
    $price = 150;
    $currency = ' 3';
    echo $price . $currency;
?>
```

- A. 50
- B. 147
- C. 150 3
- D. 450

Правильный ответ: C. 150 3

10. Какой цикл используется в приведенном коде?

```
<?php
    $i = 0;
    do
    {
        $i++;
        echo $i;
    }
    while($i < 5);
?>
```

Варианты ответов:

- A. While
- B. Do while
- C. Echo

Правильный ответ: B. Do while

Открытые вопросы:**1. Дана программа на PHP:**

Какие значения будут выведены при выполнении этого кода?

```
<?php
    $var = 5;
    echo ++$var;
    echo $var;
    $var = 5;
    echo $var--;
    echo $var;
?>
```

Правильный ответ: 6 6 5 4

Правильный ответ: 6654

2. Дана программа на PHP:

Какие значения будут выведены при выполнении этого кода?

```
<?php
    $a = 5 > 1; echo $a;
    $a = 5 == 1; echo $a;
    $a = 5 != 9; echo $a;
    $a = 9 < 3; echo $a;
?>
```

Правильный ответ: true false true false

Правильный ответ: truefalsetruefalse

Правильный ответ: 1 0 1 0

Правильный ответ: 1010

3. Дана программа на PHP:

Какое значение будет выведено при выполнении этого кода?

```
<?php
    $price = 5000;
    if($price > 10000)
        $shipping = 0;
    elseif($price > 500)
        $shipping = 100;
    else
        $shipping = 200;
    echo $shipping;
?>
```

Правильный ответ: 100

4. Дана программа на PHP:

Чему равняется переменная \$shipping в следующей программе?

```
<?php
    $price = 100;
    $shipping = ($price < 1000) ? 200 : 100;
    echo "$shipping"
?>
```

Правильный ответ: 200

5. Как называется оператор '.', который используется в следующей программе?

```
<?php
    $price = 1500;
    $currency = 'руб';
    echo $price . $currency;
?>
```

Правильный ответ: Оператор конкатенации

Правильный ответ: Конкатенации

Правильный ответ: Конкатенация

6. Какое значение будет сохранено в переменную \$a после выполнения следующего оператора?

```
$a = 1 <= 2;
```

Правильный ответ: true

Правильный ответ: 1

7. Что выведет данная программа?

```
<?php
    $i = 2;
    do
        echo ++$i;
    while($i < 7);
?>
```

Правильный ответ: 34567

8. Что выведет приведенный ниже код?

```
<?php
    $x = 5;
```

```
$x = $x + 2;
```

```
$x += 2;
```

```
$x = $x - 2;
```

```
$x -= 2;
```

```
$x = $x * 2;
```

```
$x *= 2;
```

```
$x = $x / 2;
```

```
$x /= 2;
```

```
echo $x;
```

```
?>
```

Правильный ответ: 5

9. Что выведет приведенный ниже код?

```
<?php
$status = 1;
switch($status)
{
    case 1:
        echo 'Статус: Обрабатывается<br>';
        break;
    case 2:
        echo 'Статус: Передан в доставку<br>';
        break;
    case 3:
        echo 'Статус: Выполнен<br>';
        break;
}
?>
```

Правильный ответ: Статус: Обрабатывается

10. Дана программа на PHP:

```
<?php
$products[] = ['name' => 'Товар1', 'price' => 1000];
$products[] = ['name' => 'Товар2', 'price' => 2000];
$products[] = ['name' => 'Товар3', 'price' => 3000];
$products[] = ['name' => 'Товар4', 'price' => 4000];

foreach($products as $product)
    echo $product['name'] . '<br>';
?>
```

Что будет выведено на экран?

Правильный ответ: Товар1 Товар2 Товар3 Товар4

Правильный ответ: Товар1Товар2Товар3Товар4

Правильный ответ:

Товар1

Товар2

Товар3

Товар4

Критерии и шкалы оценивания заданий ФОС:

1) Задания закрытого типа (выбор одного варианта ответа, верно/неверно):

- 1 балл – указан верный ответ;
- 0 баллов – указан неверный ответ.

2) Задания закрытого типа (множественный выбор):

- 2 балла – указаны все верные ответы;
- 0 баллов — указан хотя бы один неверный ответ.

3) Задания закрытого типа (на соответствие):

- 2 балла – все соответствия определены верно;
- 0 баллов – хотя бы одно сопоставление определено неверно.

4) Задания открытого типа (короткий текст):

- 2 балла – указан верный ответ;
- 0 баллов – указан неверный ответ.

5) Задания открытого типа (число):

- 2 балла – указан верный ответ;
- 0 баллов – указан неверный ответ.

Задания раздела 20.3 рекомендуются к использованию при проведении диагностических работ с целью оценки остаточных результатов освоения данной дисциплины (знаний, умений, навыков).