

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Силин Яков Петрович

Должность: Ректор

Дата подписания: 09.06.2026 15:45:39

Уникальный программный ключ

24f866be2aca16484036a8cbb3c509a9531e605f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Уральский государственный экономический университет»

09.12.2025 г.

протокол № 12

И.о. зав. кафедрой Кольева Н.С.

Утверждена

Советом по учебно-методическим
вопросам и качеству образования

16 декабря 2025 г.

протокол № 4

Председатель  Карх Д.А.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины	Разработка многоуровневых приложений
Направление подготовки	09.03.03 Прикладная информатика
Профиль	Инжиниринг предприятий и информационных систем
Форма обучения	очная
Год набора	2026

Разработана:
Профессор, д.п.н.
Плещев В.В.

Екатеринбург
2025 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП	3
3. ОБЪЕМ ДИСЦИПЛИНЫ	3
4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП	3
5. ТЕМАТИЧЕСКИЙ ПЛАН	7
6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ	8
7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	10
8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	15
9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	16
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	17
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	18

ВВЕДЕНИЕ

Рабочая программа дисциплины является частью основной профессиональной образовательной программы высшего образования - программы бакалавриата, разработанной в соответствии с ФГОС ВО

ФГОС ВО	Федеральный государственный образовательный стандарт высшего образования- бакалавриат по направлению подготовки 09.03.03 Прикладная информатика(приказ Минобрнауки России от 19.09.2017 г. № 922)
---------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Изучение современных методов и средств разработки многоуровневых приложений сайтов, мобильных и Web-приложений.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к части, формируемой участниками образовательных отношений.

3. ОБЪЕМ ДИСЦИПЛИНЫ

Промежуточная аттестация	Часов					З.е.
	Всего за семестр	Контактная работа (поуч.зан.)			Самостоятельная работа в том числе подготовка контрольных и курсовых	
		Всего	Лекции	Лабораторные		
Семестр 5						
Зачет	108	48	24	24	60	3
Семестр 6						
Экзамен	180	64	32	32	89	5
	288	112	56	56	149	8

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП

В результате освоения ОПОП у выпускника должны быть сформированы компетенции, установленные в соответствии ФГОС ВО.

Шифр и наименование компетенции	Индикаторы достижения компетенций
проектный	

<p>П К - 2 Способен разрабатывать и адаптировать прикладное программное обеспечение</p>	<p>ИД-1.ПК-2 Знать: Основы программирования Современные объектно-ориентированные языки программирования Современные структурные языки программирования Языки современных бизнес-приложений Современные методики тестирования разрабатываемых ИСУ Управление качеством: контрольные списки, верификация, валидация (приемо-сдаточные испытания) Инструменты и методы верификации структуры программного кода Современные методики тестирования разрабатываемых ИС: инструменты и методы модульного тестирования, инструменты и методы тестирования нефункциональных и функциональных характеристик ИС Инструменты и методы верификации структуры базы данных Инструменты и методы проектирования структур баз данных Регламенты кодирования на языках программирования Языки программирования и работы с базами данных Регламенты модульного тестирования Регламенты интеграционного тестирования Инструменты и методы интеграционного тестирования</p>
	<p>ИД-2.ПК-2 Уметь: Кодировать на языках программирования Тестировать результаты прототипирования Верифицировать структуру программного кода Верифицировать структуру баз данных Разрабатывать структуру баз данных Распределять работы и выделять ресурсы Контролировать исполнение поручений Анализировать исходные данные</p>
	<p>ИД-3.ПК-2 Иметь практический опыт: разработки структуры баз данных ИС в соответствии с архитектурной спецификацией; верификации структуры баз данных ИС относительно архитектуры ИС и требований заказчика к ИС; обеспечения соответствия разработанного кода и процесса кодирования на языках программирования принятым в организации или проекте стандартам и технологиям; назначения и распределение ресурсов; контроля соответствия разработанного кода и процесса кодирования на языках программирования принятым в организации или проекте стандартам и технологиям; обеспечения соответствия процессов модульного тестирования ИС принятым в организации или проекте стандартам и технологиям; контроля исполнения; обеспечения соответствия процессов интеграционного тестирования ИС принятым в организации или проекте стандартам и технологиям; анализа результатов тестирования с точки зрения организации процесса тестирования; разработки предложений по совершенствованию процесса тестирования</p>

<p>П К - 3 Способен проектировать ИС по видам обеспечения</p>	<p>ИД-1.ПК-3 Знать: Возможности (типовой) ИС Коммуникационное оборудование Сетевые протоколы Основы современных операционных систем Основы современных систем управления базами данных Устройство и функционирование современных ИС Системы классификации и кодирования информации, в том числе присвоение кодов документам и элементам справочников Основы бухгалтерского учета и отчетности организаций Основы налогового законодательства Российской Федерации Основы управленческого учета Основы финансового учета и бюджетирования Основы международных стандартов финансовой отчетности (МСФО) Основы управления торговлей, поставками и запасами Основы организации производства Основы управления взаимоотношениями с клиентами и заказчиками (CRM) Основы управления персоналом, включая вопросы оплаты труда Методология ведения документооборота в организациях Инструменты и методы определения финансовых и производственных показателей деятельности организаций Теория баз данных Системы хранения и анализа баз данных Инструменты и методы верификации архитектуры ИС Инструменты и методы проектирования архитектуры ИС Инструменты и методы модульного тестирования Инструменты и методы тестирования нефункциональных и функциональных характеристик ИС Инструменты и методы прототипирования пользовательского интерфейса Технологии подготовки и проведения презентаций Инструменты и методы проектирования и дизайна ИС Инструменты и методы проектирования архитектуры и дизайна ИС Инструменты и методы верификации архитектуры и дизайна ИС Инструменты и методы разработки пользовательской документации Основные принципы обучения Методики и типовые программы обучения пользователей, рекомендованные производителем ИС Основы системного администрирования Инструменты и методы интеграции ИС Инструменты и методы оптимизации ИС Инструменты и методы оценки качества и эффективности ИС Основы администрирования СУБД Рынок поставщиков товаров и услуг для создания (модификации) и ввода ИС в эксплуатацию Критерии выбора поставщиков Регламенты выбора поставщиков Методы управления несоответствующей продукцией Системы контроля версий и поддержки конфигурационного управления Архитектура, устройство и функционирование вычислительных систем Современные стандарты информационного взаимодействия систем</p>
	<p>ИД-2.ПК-3 Уметь: Проектировать архитектуру ИС Проверять (верифицировать) архитектуру ИС Проектировать архитектуру и дизайн ИС Проверять (верифицировать) архитектуру и дизайн ИС Работать с записями по качеству (в том числе с корректирующими действиями, предупреждающими действиями, запросами на исправление несоответствий) Разрабатывать пользовательскую документацию Устанавливать программное обеспечение Разрабатывать курсы обучения Разрабатывать технологии обмена данными Выполнять параметрическую настройку ИС Разрабатывать документацию</p>

<p>П К - 3 Способен проектировать ИС по видам обеспечения</p>	<p>ИД-3.ПК-3 Иметь практический опыт: разработки архитектурной спецификации ИС; согласования архитектурной спецификации ИС с заинтересованными сторонами; разработки прототипа ИС в соответствии с требованиями; тестирования прототипа ИС на проверку корректности архитектурных решений; анализа результатов тестов; принятия решения о пригодности архитектуры; согласования пользовательского интерфейса с заказчиком; разработки структуры программного кода ИС; верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС; устранения обнаруженных несоответствий; анализа зафиксированных в системе учета дефектов и несоответствий в архитектуре и дизайне ИС; установления причин возникновения дефектов и несоответствий; устранения дефектов и несоответствий; проверки результатов исправления дефектов и несоответствий в коде ИС и документации к ИС; фиксирования в системе учета факта внесения исправлений в код и документацию к ИС; разработки руководства администратора ИС; разработки руководства пользователя ИС; разработки руководства программиста ИС; разработки и выбор программ обучения пользователей ИС; проведения обучения пользователей ИС сложным программам обучения; осуществления выходного тестирования пользователей ИС; сбора замечаний и пожеланий пользователей для развития ИС; настройки ИС для оптимального решения задач заказчика; параметрической настройки ИС; экспертной оценки предложенных решений по реализации интерфейсов и форматов обмена данными; проведения технических советов по оценке предложенных решений по реализации интерфейсов и форматов обмена данными; выдачи экспертных заключений по предложенным решениям по реализации интерфейсов и форматов обмена данными;</p>
-------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>П К - 3 Способен проектировать ИС по видам обеспечения</p>	<p>ИД-4.ПК-3 Иметь практический опыт: предложения вариантов реализации интерфейсов и форматов обмена данными на основе накопленного опыта; разработки технологии обмена данными между ИС и существующими системами; количественного определения существующих параметров работы ИС; определения новых целевых показателей работы ИС; определения параметров, которые должны быть улучшены; осуществления оптимизации ИС для достижения новых целевых показателей; проверки фактического внесения изменений в ИС; изменения статуса проверенных запросов на изменение в системе учета; определения потребности приобретения товаров или услуг для создания (модификации) и ввода ИС в эксплуатацию; принятия решения "производить или покупать"; разработки описаний ИТ-продуктов или услуг для поставщиков; отбора поставщиков по установленным критериям на основании собранных предложений; контроля уровня качества поставленной продукции или услуг; управления несоответствующей продукцией; подтверждения (валидации) закупленной ИТ-продукции или услуг; определения версий программных базовых элементов конфигурации ИС, входящих в сборку; обеспечения сборки программных базовых элементов конфигурации ИС в соответствии с планом; верификации результатов сборки программных базовых элементов конфигурации ИС; подготовки технической информации для договоров на выполняемые работы; согласования договоров на выполняемые работы внутри организации; согласования договоров на выполняемые работы с контрагентами; организации подписания договоров на выполняемые работы; проведения переговоров об изменении условий договоров на выполняемые работы; подготовки дополнительных соглашений к договорам на выполняемые работы; согласования дополнительных соглашений к договорам на выполняемые работы внутри организации; согласования дополнительных соглашений к договорам на выполняемые работы с контрагентами; организации подписания дополнительных соглашений к договорам на выполняемые работы; подготовки технической информации для договоров сопровождения ИС; согласования договоров сопровождения ИС внутри организации; согласования договоров сопровождения ИС с контрагентами; организации подписания договоров сопровождения ИС</p>
-------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5. ТЕМАТИЧЕСКИЙ ПЛАН

Тема	Часов						
	Наименование темы	Всего часов	Контактная работа (по уч. зан.)			Самост. работа	Контроль самостоятельной работы
			Лекции	Лабораторные	Практические занятия		
Семестр 5		108					
Тема	Веб-технологии (ПК-2,	6	4			2	
Тема 2.	Разработка веб-интерфейсов.	8		2		6	
Тема 3.	Разработка веб-приложений на языке PHP	6		2		4	
Тема 4.	Организация сетевого взаимодействия с базой	6	4	2			

Тема 5.	Публикация данных из MySQL в Интернете (ПК-2, ПК-3)	18	6	4		8	
Тема 6.	Проектирование реляционных баз данных (ПК-2, ПК-3)	14	4	2		8	
Тема 7.	Структурное программирование (ПК-2, ПК-3)	14	2	4		8	
Тема 8.	Система управления содержимым (ПК-2, ПК-3)	14	2	4		8	
Тема 9.	Форматирование содержимого с помощью регулярных выражений (ПК-2, ПК-3)	12	2	2		8	
Тема 10.	Куки, сессии и контроль доступа (ПК-2, ПК-3)	10		2		8	
Семестр 6		153					
Тема 11.	Администрирование MySQL (ПК-2, ПК-3)	12	4	2		6	
Тема 12.	Расширенные SQL-запросы (ПК-2, ПК-3)	12	4	2		6	
Тема 13.	Бинарные данные (ПК-2, ПК-3)	14	6	4		4	
Тема 14.	JavaScript интерактивность приложений (ПК-2, ПК-3)	16	4	4		8	
Тема 15.	Разработка распределенных приложений на платформе .NET (ПК-2, ПК-3)	16	4	4		8	
Тема 16.	Система управления содержимым CMS (ПК-2, ПК-3)	20	4	6		10	
Тема 17.	Мобильный сайт и мобильное приложение (ПК-2, ПК-3)	17	2	4		11	
Тема 18.	Нативные приложения (ПК-2, ПК-3)	16	2	2		12	
Тема 19.	Гибридные приложения (ПК-2, ПК-3)	14		2		12	
Тема 20.	Создание мобильных приложений (ПК-2, ПК-3)	16	2	2		12	

6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ

Раздел/Тема	Вид оценочного средства	Описание оценочного средства	Критерии оценивания
Текущий контроль (Приложение 4)			
Темы 1-5	Практическая работа (приложение 4)	Работа состоит из 5-и задач	10 баллов
Темы 6-10	Практическая работа (приложение 4)	Работа состоит из 5-и задач	10 баллов
Тема 11-20	Практическая работа (приложение 4)	Работа состоит из 5-и задач	10 баллов
Промежуточная аттестация (Приложение 5)			

6 семестр(Эк)	Экзаменационный билет	Билет содержит 1 теоретический вопрос и 2 практических задания	100 баллов
5 семестр (За)	Билет для зачета	Билет содержит 1 теоретический вопрос и 2 практических задания	100 баллов

ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

Показатель оценки освоения ОПОП формируется на основе объединения текущего контроля и промежуточной аттестации обучающегося.

Показатель рейтинга по каждой дисциплине выражается в процентах, который показывает уровень подготовки студента.

Текущий контроль. Используется 100-балльная система оценивания. Оценка работы студента в течение семестра осуществляется преподавателем в соответствии с разработанной им системой оценки учебных достижений в процессе обучения по данной дисциплине.

В рабочих программах дисциплин и практик закреплены виды текущего контроля, планируемые результаты контрольных мероприятий и критерии оценки учебных достижений.

В течение семестра преподавателем проводится не менее 3-х контрольных мероприятий, по оценке деятельности студента. Если посещения занятий по дисциплине включены в рейтинг, то данный показатель составляет не более 20% от максимального количества баллов по дисциплине.

Промежуточная аттестация. Используется 5-балльная система оценивания. Оценка работы студента по окончании дисциплины (части дисциплины) осуществляется преподавателем в соответствии с разработанной им системой оценки достижений студента в процессе обучения по данной дисциплине. Промежуточная аттестация также проводится по окончании формирования компетенций.

Порядок перевода рейтинга, предусмотренных системой оценивания, по дисциплине, в пятибалльную систему.

Высокий уровень – 100% - 70% - отлично, хорошо.

Средний уровень – 69% - 50% - удовлетворительно.

Показатель оценки	По 5-балльной системе	Характеристика показателя
100% - 85%	отлично	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на высоком уровне
84% - 70%	хорошо	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Могут быть допущены недочеты, исправленные студентом самостоятельно в процессе работы (ответаи т.д.)
69% - 50%	удовлетворительно	обладают общими теоретическими знаниями, умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на среднем уровне. Допускаются ошибки, которые студент затрудняется исправить самостоятельно.
49 % и менее	неудовлетворительно	обладают не полным объемом общих теоретическими знаниями, не умеют самостоятельно применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Не сформированы умения и навыки для
100% - 50%	зачтено	характеристика показателя соответствует «отлично», «хорошо», «удовлетворительно»
49 % и менее	не зачтено	характеристика показателя соответствует «неудовлетворительно»

7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

7.1. Содержание лекций

<p>Тема 1. Веб-технологии (ПК-2, ПК-3) Веб–технологии - логическая составляющая Интернет. Эпохи Web. Наполнение и интерфейсресурсов в разные эпохи. Сематический Web.</p>
<p>Тема 4. Организация сетевого взаимодействия с базой данных (MySQL) (ПК-2, ПК-3) Введение в базы данных. Использование приложения phpMyAdmin для выполнения SQL-запросов.Структурированный язык запросов.</p>
<p>Тема 5. Публикация данных из MySQL в Интернете (ПК-2, ПК-3) Создание учетной записи пользователя в MySQL. Подключение к MySQL с помощью PHP.Настройка подключения.</p>
<p>Тема 6. Проектирование реляционных баз данных (ПК-2, ПК-3) Выборка из нескольких таблиц. Простые связи.</p>
<p>Тема 7. Структурное программирование (ПК-2, ПК-3) Подключаемые файлы. Подключение HTML-кода. Подключение PHP-кода.</p>
<p>Тема 8. Система управления содержимым (ПК-2, ПК-3) Главная страница. Создание проекта "Поиск шуток".</p>
<p>Тема 9. Форматирование содержимого с помощью регулярных выражений (ПК-2, ПК-3) Регулярные выражения. Замена текста с помощью регулярных выражений. Выделение в тексте.Абзацы. Гиперссылки.</p>
<p>Тема 11. Администрирование MySQL (ПК-2, ПК-3) Резервное копирование баз данных в MySQL. Резервное копирование базы данных с помощьюphpMyAdmin.</p>
<p>Тема 12. Расширенные SQL-запросы (ПК-2, ПК-3) Оператор левого объединения. Дополнительные источники информации.</p>
<p>Тема 13. Бинарные данные (ПК-2, ПК-3) Полудинамические страницы. Обеспечение загрузки файлов.</p>
<p>Тема 14. JavaScript интерактивность приложений (ПК-2, ПК-3) Что может и не может JavaScript в браузере? «Привет, мир!» на JS.</p>
<p>Тема 15. Разработка распределенных приложений на платформе .NET (ПК-2, ПК-3) Понятие распределённой системы. Определение распределённой системы.</p>
<p>Тема 16. Система управления содержимым CMS (ПК-2, ПК-3) Сайт без кода. Как работает CMS?</p>
<p>Тема 17. Мобильный сайт и мобильное приложение (ПК-2, ПК-3) Интерфейс. Быстродействие. Связь и адаптация.</p>
<p>Тема 18. Нативные приложения (ПК-2, ПК-3) Выбор языка программирования. выбор платформы. Анализ эмуляторов</p>
<p>Тема 20. Создание мобильных приложений (ПК-2, ПК-3) Выбор платформы. Эмуляторы.</p>

7.2 Содержание практических занятий и лабораторных работ

Тема 3. Разработка веб-приложений на языке PHP (ПК-2, ПК-3)

Первый PHP скрипт. Комментарии в PHP. Переменные и их типы. Константы. Арифметические операции. Строковые операции. Логические операции. Оператор эквивалентности. Массивы.

Тема 4. Организация сетевого взаимодействия с базой данных (MySQL) (ПК-2, ПК-3)

Создание базы данных. Создание таблицы. Добавление данных в таблицу. Вывод сохраненных данных.

Тема 5. Публикация данных из MySQL в Интернете (ПК-2, ПК-3)

Добавление информации в базу данных. Удаление информации из базы данных. Отправка SQL-запросов с помощью PHP. Обработка результатов выполнения команды SELECT.

Тема 6. Проектирование реляционных баз данных (ПК-2, ПК-3)

Отношение «многие ко многим». Один за многих, и многие за одного

Тема 7. Структурное программирование (ПК-2, ПК-3)

Виды включений. Разделение подключаемых файлов. Вспомогательные функции для шаблонов. Нестандартные функции и библиотеки функций.

Тема 8. Система управления содержимым (ПК-2, ПК-3)

Управление списком авторов. Удаление имен авторов. Добавление и редактирование имен авторов. Управление списком категорий. Управление списком шуток.

Тема 9. Форматирование содержимого с помощью регулярных выражений (ПК-2, ПК-3)

Передача данных в реальных условиях.

Тема 10. Куки, сессии и контроль доступа (ПК-2, ПК-3)

Управление паролями и ролями. Модерирование шуток.

Тема 11. Администрирование MySQL (ПК-2, ПК-3)

Инкрементальное резервное копирование с помощью бинарного журнала изменений. Управление доступом к MySQL. Проблемы, связанные с именем сервера. Индексы.

Тема 12. Расширенные SQL-запросы (ПК-2, ПК-3)

Сортировка результатов выполнения запроса SELECT. Установка лимитов. Транзакции в базе данных. Псевдонимы для столбцов и таблиц. Группирование результатов.

<p>Тема 13. Бинарные данные (ПК-2, ПК-3)</p> <p>Запись загруженных файлов в базу данных. Типы бинарных столбцов. Сохранение файлов. Отображение сохраненных файлов. Размер пакета в MySQL.</p>
<p>Тема 14. JavaScript интерактивность приложений (ПК-2, ПК-3)</p> <p>Современная разметка, Внешние скрипты. Таблицы совместимости. Структура кода. Инструкции. Переменные. Имена переменных. Константы. Типы данных. Взаимодействие: alert, prompt, confirm. Объекты. Литералы и свойства.</p>
<p>Тема 15. Разработка распределенных приложений на платформе .NET (ПК-2, ПК-3)</p> <p>Программные компоненты. Требования к распределенным системам. Понятие промежуточной среды. Взаимодействие компонент распределенной системы.</p>
<p>Тема 16. Система управления содержимым CMS (ПК-2, ПК-3)</p> <p>Плагины, Темы и Расширения. Популярные Системы Управления Контентом. Плюсы и минусы использования CMS. Как правильно выбрать Систему Управления Контентом. Как начать пользоваться CMS.</p>
<p>Тема 17. Мобильный сайт и мобильное приложение (ПК-2, ПК-3)</p> <p>Интеграция с платформой. Фрагментация. Ресурсы. Публикация.</p>
<p>Тема 18. Нативные приложения (ПК-2, ПК-3)</p> <p>Натив преимущества и недостатки. Инструменты для разработки нативных приложения</p>
<p>Тема 19. Гибридные приложения (ПК-2, ПК-3)</p> <p>Гибрид преимущества и недостатки. Инструменты для разработки гибридных приложения</p>
<p>Тема 20. Создание мобильных приложений (ПК-2, ПК-3)</p> <p>Мобильное приложение. Основные этапы создания, компиляция, публикация в маркетплейс</p>

7.3. Содержание самостоятельной работы

<p>Тема 2. Разработка веб-интерфейсов. Языки разметки и стили (ПК-2, ПК-3)</p> <p>Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 3. Разработка веб-приложений на языке PHP (ПК-2, ПК-3)</p> <p>Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>

<p>Тема 5. Публикация данных из MySQL в Интернете (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 6. Проектирование реляционных баз данных (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 7. Структурное программирование (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 8. Система управления содержимым (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 9. Форматирование содержимого с помощью регулярных выражений (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 10. Куки, сессии и контроль доступа (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 11. Администрирование MySQL (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 12. Расширенные SQL-запросы (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 13. Бинарные данные (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 14. JavaScript интерактивность приложений (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 15. Разработка распределенных приложений на платформе .NET (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 16. Система управления содержимым CMS (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 17. Мобильный сайт и мобильное приложение (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>

<p>Тема 18. Нативные приложения (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 19. Гибридные приложения (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 20. Создание мобильных приложений (ПК-2, ПК-3) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену
Приложение 1.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену
Приложение 2.

7.3.3. Перечень курсовых работ
Не предусмотрено

7.4. Электронное портфолио обучающегося
Материалы не размещаются.

7.5. Методические рекомендации по выполнению контрольной работы
Не предусмотрено.

7.6 Методические рекомендации по выполнению курсовой работы
Не предусмотрено

8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

По заявлению студента

В целях доступности освоения программы для лиц с ограниченными возможностями здоровья при необходимости кафедра обеспечивает следующие условия:

- особый порядок освоения дисциплины, с учетом состояния их здоровья;
- электронные образовательные ресурсы по дисциплине в формах, адаптированных к ограничениям их здоровья;
- изучение дисциплины по индивидуальному учебному плану (вне зависимости от формы обучения);
- электронное обучение и дистанционные образовательные технологии, которые предусматривают возможности приема-передачи информации в доступных для них формах.
- доступ (удаленный доступ), к современным профессиональным базам данных и информационным справочным системам, состав которых определен РПД.

9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Сайт библиотеки УрГЭУ

<http://lib.usue.ru/>

Основная литература:

2. Панов Разработка многоуровневых приложений. Тесты. Тест 3. Презентационный уровень уровня бизнес-логики [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - – Режим доступа:<http://lib.wbstatic.usue.ru/202406a/90.docx>

3. Мартишин С.А., Симонов В.Л., Храпченко М.В. Базы данных: Работа с распределенными базами данных и файловыми системами на примере MongoDB и HDFS с использованием Node.js, Express.js, Apache Spark и Scala [Электронный ресурс]: Учебное пособие. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2024. - 235 – Режим доступа:<https://znanium.com/catalog/product/2139860>

4. Тузовский А. Ф. Проектирование и разработка web-приложений [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2023. - 219 – Режим доступа:<https://urait.ru/bcode/530767>

5. Панов Разработка многоуровневых приложений. Курс лекций. Тема 1. Обзор общих принципов построения многоуровневых приложений [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 1 – Режим доступа: <https://libw.usue.ru/202408/133.mp4>

6. Панов Разработка многоуровневых приложений. Курс лекций. Тема 3. Презентационный уровень и уровень бизнес-логики [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 1 – Режим доступа: <https://libw.usue.ru/202408/135.mp4>

7. Панов Разработка многоуровневых приложений. Курс лекций. Тема 2. Ключевая роль SQL в многоуровневых приложениях [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 1 – Режим доступа: <https://libw.usue.ru/202408/134.mp4>

8. Ткаченко О.Н. Взаимодействие пользователя с интерфейсами информационных систем для мобильных устройств: исследование опыта [Электронный ресурс]: Учебное пособие. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2026. - 152 – Режим доступа:<https://znanium.com/catalog/product/2178768>

9. Тузовский А. Ф. Проектирование и разработка web-приложений [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2023. - 218 с – Режим доступа:<https://urait.ru/bcode/512113>

10. Тузовский А. Ф. Проектирование и разработка web-приложений [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2024. - 219 – Режим доступа:<https://urait.ru/bcode/537106>

11. Панов Разработка многоуровневых приложений. Тесты. Тест 2. Ключевая роль SQL в многоуровневых приложениях [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - – Режим доступа: <http://lib.wbstatic.usue.ru/202406a/89.docx>

12. Панов Разработка многоуровневых приложений. Тесты. Тест 1. Обзор общих принципов построения многоуровневых приложений [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - – Режим доступа: <http://lib.wbstatic.usue.ru/202406a/88.docx>

Дополнительная литература:

1. Никсон Р., Вильчинский Н. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5: производственно-практическое издание. - Санкт-Петербург [и др.]: Питер, 2017. - 766

2. Зубкова Е. В., Лескова Ю. В., Федоров А. Н. Принципы построения и архитектура web-приложений [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2017. - 160 – Режим доступа: <http://lib.usue.ru/resource/limit/ump/18/p490704.pdf>

3. Мартишин С.А., Симонов В.Л. Базы данных: Работа с распределенными базами данных и файловыми системами на примере MongoDB и HDFS с использованием Node.js, Express.js, Apache Spark и Scala [Электронный ресурс]: Учебное пособие. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2021. - 235 – Режим доступа: <https://znanium.com/catalog/product/1214862>

4. Кузин А. В., Чумакова Е. В. Программирование на языке Си [Электронный ресурс]: справочник: учебное пособие : учебное пособие. - Москва: Издательство "ФОРУМ", 2023. - 143 – Режим доступа: <https://znanium.ru/catalog/product/1878382>

5. Кислицын Е. В., Шишков Е. И. Разработка приложений на языке Java [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2017. - 86 – Режим доступа: <http://lib.usue.ru/resource/limit/ump/17/p488938.pdf>

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Перечень лицензионного программного обеспечения:

Microsoft Visual Studio Community. Лицензия для образовательных учреждений. Срок действия лицензии - без ограничения срока.

PostgreSQL Server. Лицензия PostgreSQL. Срок действия лицензии - без ограничения срока.

Putty. Лицензия MIT license. Срок действия лицензии - без ограничения срока.

WinSCP. Лицензия GNU General Public License. Срок действия лицензии - без ограничения срока.

Notepad++. Лицензия GNU General Public License. Срок действия лицензии - без ограничения срока.

Перечень информационных справочных систем, ресурсов информационно-телекоммуникационной сети «Интернет»:

Справочно-правовая система Консультант+. Договор № 143/223-У/2025 от 02.12.2025 Срок действия лицензии до 31.12.2026

Справочно-правовая система Гарант. Договор № 58419 от 22 декабря 2015. Срок действия лицензии - без ограничения срока

Веб-программирование

<https://openedu.ru/course/ITMOUniversity/WEBDEV/>

Разработка Android-приложений для мобильных устройств

<https://openedu.ru/course/ITMOUniversity/ANDROID/>

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Реализация учебной дисциплины осуществляется с использованием материально-технической базы УрГЭУ, обеспечивающей проведение всех видов учебных занятий и научно-исследовательской и самостоятельной работы обучающихся:

Специальные помещения представляют собой учебные аудитории для проведения всех видов занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

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

Все помещения укомплектованы специализированной мебелью и оснащены мультимедийным оборудованием спецоборудованием (информационно-телекоммуникационным, иным компьютерным), доступом к информационно-поисковым, справочно-правовым системам, электронным библиотечным системам, базам данных действующего законодательства, иным информационным ресурсам служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа презентации и другие учебно-наглядные пособия, обеспечивающие тематические иллюстрации.

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену

К зачету

1. Что такое PHP?
2. Как изолировать PHP код от HTML?
3. Как использовать комментарии в коде PHP?
4. В чем разница между одинарными и двойными кавычками в PHP?
5. Сколько в PHP типов данных? Какие это типы данных?
6. Что такое переменная и как ее использовать?
7. Что такое переменные извне?
8. В чем разница между GET и POST?
9. Что такое константы? Как их использовать?
10. Что такое оператор?
11. Какие существуют операторы?
12. Что такое арифметические операторы?
13. Что такое логические операторы?
14. Что такое оператор присваивания?
15. Что такое побитовые операторы?
16. Что такое операторы сравнения?
17. Что такое оператор управления ошибками?
18. Что такое операторы исполнения?
19. Что такое строковые операторы?
20. В чем различия между echo и print?

К экзамену

1. В чем отличие цикла while от do while?
2. Что такое goto и как он работает?
3. Как прервать выполнение for, foreach, while, do-while или switch?
4. Как пропустить часть интерации цикла внутри for, foreach, while, do-while или switch?
5. Как удалить переменную или элемент из массива?
6. Что такое функция?
7. Что такое аргументы функции?
8. Какие существуют способы передачи аргументов в функцию?
9. Как происходит передача аргументов в функцию по значению?
10. Как происходит передача аргументов в функцию по ссылке?
11. Что такое исключения?
12. Что такое ссылка?
13. Что делают ссылки?
14. Что не делают ссылки?
15. Что такое Cookie и зачем они используются?
16. Что нельзя хранить в Cookie и почему?
17. Какие есть типы cookie?
18. Что такое сессии и зачем они используются?

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену

Примерные вопросы закрытого типа (ПК-2,3)

1. Что такое многоуровневая архитектура приложений?
 1. Архитектура, которая состоит из нескольких уровней, каждый из которых отвечает за определенные функции.
 2. Архитектура, которая состоит из одного уровня, объединяющего все функции приложения.
 3. Архитектура, которая не имеет четкого разделения на уровни.
2. Какие преимущества предоставляет многоуровневая архитектура приложений?
 1. Улучшенная масштабируемость и поддерживаемость.
 2. Уменьшенная сложность и увеличенная производительность.
 3. Отсутствие необходимости в разделении функций на уровни.
3. Какие уровни обычно присутствуют в многоуровневой архитектуре приложений?
 1. Представление (интерфейс), бизнес-логика, доступ к данным.
 2. Только один уровень, объединяющий все функции.
 3. Только два уровня: клиентский и серверный.
4. Что такое уровень представления (интерфейса) в многоуровневой архитектуре?
 1. Уровень, отвечающий за взаимодействие с пользователем.
 2. Уровень, где располагается бизнес-логика приложения.
 3. Уровень, отвечающий за доступ к данным.
5. Какой уровень в многоуровневой архитектуре отвечает за обработку бизнес-логики?
 1. Уровень бизнес-логики.
 2. Уровень доступа к данным.
 3. Уровень представления (интерфейса).

Примерные вопросы открытого типа (ПК-2,3)

1. Как вы понимаете концепцию многоуровневой архитектуры приложений и почему она важна для разработки программного обеспечения?
2. Расскажите о принципах разделения функциональности на уровни в многоуровневой архитектуре приложений и какие преимущества это может принести для проекта.
3. Какие вызовы и проблемы могут возникнуть при проектировании и реализации многоуровневой архитектуры приложений, и как их можно решить?
4. Какие критерии и принципы следует учитывать при выборе технологий и инструментов для каждого уровня многоуровневой архитектуры приложений?

5. Какие сценарии использования и примеры проектов могут быть наилучшими для применения многоуровневой архитектуры приложений, и какие уроки могут быть извлечены из таких проектов?

Примерные практические задания к зачету

1. Создайте переменную \$a и присвойте ей значение 3. Выведите значение этой переменной на экран.
2. Создайте переменные \$a=10 и \$b=2. Выведите на экран их сумму, разность, произведение и частное (результат деления).
3. Создайте переменные \$c=15 и \$d=2. Просуммируйте их, а результат присвойте переменной \$result. Выведите на экран значение переменной \$result.
4. Создайте переменные \$a=10, \$b=2 и \$c=5. Выведите на экран их сумму.
5. Создайте переменные \$a=17 и \$b=10. Отнимите от \$a переменную \$b и результат присвойте переменной \$c. Затем создайте переменную \$d, присвойте ей значение 7. Сложите переменные \$c и \$d, а результат запишите в переменную \$result. Выведите на экран значение переменной \$result.
6. Создайте переменную \$text и присвойте ей значение 'Привет, Мир!'. Выведите значение этой переменной на экран.
7. Создайте переменные \$text1='Привет, ' и \$text2='Мир!'. С помощью этих переменных и операции сложения строк выведите на экран фразу 'Привет, Мир!'.
8. Создайте переменную \$name и присвойте ей ваше имя. Выведите на экран фразу 'Привет, %Имя%!'. Вместо %Имя% должно стоять ваше имя.
9. Создайте переменную \$age и присвойте ей ваш возраст. Выведите на экран 'Мне %Возраст% лет!'.
10. Создайте переменную \$text и присвойте ей значение 'abcde'. Обращаясь к отдельным символам этой строки выведите на экран символ 'a', символ 'c', символ 'e'.
11. Дана произвольная строка, например, 'abcde'. Поменяйте первую букву (то есть букву 'a') этой строки на '!'.
12. Создайте переменную \$num и присвойте ей значение '12345'. Найдите сумму цифр этого числа.
13. Создайте массив \$arr=['a', 'b', 'c']. Выведите значение массива на экран с помощью функции var_dump().
14. С помощью массива \$arr из предыдущего номера выведите на экран содержимое первого, второго и третьего элементов.
15. Создайте массив \$arr=['a', 'b', 'c', 'd'] и с его помощью выведите на экран строку 'a+b, c+d'.
16. Создайте массив \$arr с элементами 2, 5, 3, 9. Умножьте первый элемент массива на второй, а третий элемент на четвертый. Результаты сложите, присвойте переменной \$result. Выведите на экран значение этой переменной.
17. Если переменная \$a равна нулю, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
18. Если переменная \$a больше нуля, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
19. Если переменная \$a меньше нуля, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
20. Если переменная \$a больше или равна нулю, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
21. Если переменная \$a меньше или равна нулю, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
22. Если переменная \$a не равна нулю, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
23. Если переменная \$a равна 'test', то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 'test', 'тест', 3.

24. Если переменная \$a равна '1' и по значению и по типу, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном '1', 1, 3.
25. Если переменная \$a пустая, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 3, -3, 0, null, true, "", '0'.
26. Если переменная \$a НЕ пустая, то выведите 'Верно', иначе выведите 'Неверно'.
27. Если переменная \$a существует, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 3 и null.
28. Если переменная \$a НЕ существует, то выведите 'Верно', иначе выведите 'Неверно'.
29. Если переменная \$var равна true, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$var, равном true, false. Напишите два варианта скрипта - с короткой записью и с длинной.
30. Если переменная \$var НЕ равна true, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$var, равном true, false. Напишите два варианта скрипта - с короткой записью и с длинной.
31. Если переменная \$a больше нуля и меньше 5-ти, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 5, 0, -3, 2.
32. Если переменная \$a равна нулю или равна двум, то прибавьте к ней 7, иначе поделите ее на 10. Выведите новое значение переменной на экран. Проверьте работу скрипта при \$a, равном 5, 0, -3, 2.
33. Если переменная \$a равна или меньше 1, а переменная \$b больше или равна 3, то выведите сумму этих переменных, иначе выведите их разность (результат вычитания). Проверьте работу скрипта при \$a и \$b, равном 1 и 3, 0 и 6, 3 и 5.
34. Если переменная \$a больше 2-х и меньше 11-ти, или переменная \$b больше или равна 6-ти и меньше 14-ти, то выведите 'Верно', в противном случае выведите 'Неверно'.
35. Переменная \$num может принимать одно из значений: 1, 2, 3 или 4. Если она имеет значение '1', то в переменную \$result запишем 'зима', если имеет значение '2' – 'лето' и так далее. Решите задачу через switch-case.
36. В переменной \$day лежит какое-то число из интервала от 1 до 31. Определите в какую декаду месяца попадает это число (в первую, вторую или третью).
37. В переменной \$month лежит какое-то число из интервала от 1 до 12. Определите в какую пору года попадает этот месяц (зима, лето, весна, осень).
38. В переменной \$year хранится год. Определите, является ли он високосным (в таком году есть 29 февраля). Год будет високосным в двух случаях: либо он делится на 4, но при этом не делится на 100, либо делится на 400. Так, годы 1700, 1800 и 1900 не являются високосными, так как они делятся на 100 и не делятся на 400. Годы 1600 и 2000 - високосные, так как они делятся на 400.
39. Дана строка с символами, например, 'abcde'. Проверьте, что первым символом этой строки является буква 'a'. Если это так - выведите 'да', в противном случае выведите 'нет'.
40. Дана строка с цифрами, например, '12345'. Проверьте, что первым символом этой строки является цифра 1, 2 или 3. Если это так - выведите 'да', в противном случае выведите 'нет'.
41. Дана строка из 3-х цифр. Найдите сумму этих цифр. То есть сложите как числа первый символ строки, второй и третий.
42. Дана строка из 6-ти цифр. Проверьте, что сумма первых трех цифр равняется сумме вторых трех цифр. Если это так - выведите 'да', в противном случае выведите 'нет'.
43. Дан массив с элементами 2, 5, 9, 15, 0, 4. С помощью цикла foreach и оператора if выведите на экран столбец тех элементов массива, которые больше 3-х, но меньше 10.
44. Дан массив с числами. Числа могут быть положительными и отрицательными. Найдите сумму положительных элементов этого массива.

45. Дан массив с элементами 1, 2, 5, 9, 4, 13, 4, 10. С помощью цикла `foreach` и оператора `if` проверьте есть ли в массиве элемент со значением, равным 4. Если есть - выведите на экран 'Есть!' и выйдите из цикла. Если нет - ничего делать не надо.
46. Дан массив числами, например: ['10', '20', '30', '50', '235', '3000']. Выведите на экран только те числа из массива, которые начинаются на цифру 1, 2 или 5.
47. Дан массив с элементами 1, 2, 3, 4, 5, 6, 7, 8, 9. С помощью цикла `foreach` создайте строку '-1-2-3-4-5-6-7-8-9-'.
48. Составьте массив дней недели. С помощью цикла `foreach` выведите все дни недели, а выходные дни выведите жирным.
49. Составьте массив дней недели. С помощью цикла `foreach` выведите все дни недели, а текущий день выведите курсивом. Текущий день должен храниться в переменной `$day`.
50. Возведите 2 в 10 степень. Результат запишите в переменную `$st`.
51. Найдите квадратный корень из 245.
52. Дан массив с элементами 4, 2, 5, 19, 13, 0, 10. Найдите корень из суммы квадратов его элементов. Для решения воспользуйтесь циклом `foreach`.
53. Дана строка 'php'. Сделайте из нее строку 'PHP'.
54. Дана строка 'PHP'. Сделайте из нее строку 'php'.
55. Дана строка 'london'. Сделайте из нее строку 'London'.
56. Дана строка 'London'. Сделайте из нее строку 'london'.
57. Дана строка 'london is the capital of great britain'. Сделайте из нее строку 'London Is The Capital Of Great Britain'.
58. Дана строка 'LONDON'. Сделайте из нее строку 'London'.
59. Дана строка 'html css php'. Найдите количество символов в этой строке.
60. Дана переменная `$password`, в которой хранится пароль пользователя. Если количество символов пароля больше 5-ти и меньше 10-ти, то выведите пользователю сообщение о том, что пароль подходит, иначе сообщение о том, что нужно придумать другой пароль.
61. Дана строка 'html css php'. Найдите количество символов в этой строке.
62. Дана переменная `$password`, в которой хранится пароль пользователя. Если количество символов пароля больше 5-ти и меньше 10-ти, то выведите пользователю сообщение о том, что пароль подходит, иначе сообщение о том, что нужно придумать другой пароль.

Примерные практические задания к экзамену

1. Дан массив `$arr`. Подсчитайте количество элементов этого массива.
2. Дан массив `$arr`. С помощью функции `count` выведите последний элемент данного массива.
3. Дан массив с числами. Проверьте, что в нем есть элемент со значением 3.
4. Дан массив [1, 2, 3, 4, 5]. Найдите сумму элементов данного массива.
5. Дан массив [1, 2, 3, 4, 5]. Найдите произведение (умножение) элементов данного массива.
6. Дан массив `$arr`. С помощью функций `array_sum` и `count` найдите среднее арифметическое элементов (сумма элементов делить на их количество) данного массива.
7. Создайте массив, заполненный числами от 1 до 100.
8. Создайте массив, заполненный буквами от 'a' до 'z'.
9. Создайте строку '1-2-3-4-5-6-7-8-9' не используя цикл.
10. Найдите сумму чисел от 1 до 100 не используя цикл.
11. Найдите произведение чисел от 1 до 10 не используя цикл.
12. Даны два массива: первый с элементами 1, 2, 3, второй с элементами 'a', 'b', 'c'. Сделайте из них массив с элементами 1, 2, 3, 'a', 'b', 'c'.
13. Дан массив с элементами 1, 2, 3, 4, 5. С помощью функции `array_slice` создайте из него массив `$result` с элементами 2, 3, 4.

14. Дан массив [1, 2, 3, 4, 5]. С помощью функции `array_splice` преобразуйте массив в [1, 4, 5].
15. Дан массив [1, 2, 3, 4, 5]. С помощью функции `array_splice` запишите в новый массив элементы [2, 3, 4].
16. Дан массив [1, 2, 3, 4, 5]. С помощью функции `array_splice` сделайте из него массив [1, 2, 3, 'a', 'b', 'c', 4, 5].
17. Дан массив [1, 2, 3, 4, 5]. С помощью функции `array_splice` сделайте из него массив [1, 'a', 'b', 2, 3, 4, 'c', 5, 'e'].
18. Дан массив 'a'=>1, 'b'=>2, 'c'=>3'. Запишите в массив `$keys` ключи из этого массива, а в `$values` – значения.
19. Даны два массива: ['a', 'b', 'c'] и [1, 2, 3]. Создайте с их помощью массив 'a'=>1, 'b'=>2, 'c'=>3'.
20. Дан массив 'a'=>1, 'b'=>2, 'c'=>3. Поменяйте в нем местами ключи и значения.
21. Дан массив с элементами 1, 2, 3, 4, 5. Сделайте из него массив с элементами 5, 4, 3, 2, 1.
22. Дан массив ['a', '-', 'b', '-', 'c', '-', 'd']. Найдите позицию первого элемента '-'.
23. Дан массив ['a', '-', 'b', '-', 'c', '-', 'd']. Найдите позицию первого элемента '-' и удалите его с помощью функции `array_splice`.
24. Дан массив ['a', 'b', 'c', 'd', 'e']. Поменяйте элемент с ключом 0 на '!', а элемент с ключом 3 - на '!!'.
25. Дан массив '3'=>'a', '1'=>'c', '2'=>'e', '4'=>'b'. Попробуйте на нем различные типы сортировок.
26. Дан массив с элементами 'a'=>1, 'b'=>2, 'c'=>3. Выведите на экран случайный ключ из данного массива.
27. Дан массив с элементами 'a'=>1, 'b'=>2, 'c'=>3. Выведите на экран случайный элемент данного массива.
28. Дан массив `$arr`. Перемешайте его элементы в случайном порядке.
29. Заполните массив числами от 1 до 25 с помощью `range`, а затем перемешайте его элементы в случайном порядке.
30. Создайте массив, заполненный буквами от 'a' до 'z' так, чтобы буквы шли в случайном порядке и не повторялись.
31. Сделайте строку длиной 6 символов, состоящую из маленьких английских букв, расположенных в случайном порядке. Буквы не должны повторяться.
32. Дана строка '1234567890'. Найдите сумму цифр из этой строки не используя цикл.
33. Создайте массив ['a'=>1, 'b'=>2... 'z'=>26] не используя цикл.
34. Создайте массив вида [[1, 2, 3], [4, 5, 6], [7, 8, 9]] не используя цикл.
35. Дан массив с элементами 1, 2, 4, 5, 5. Найдите второй по величине элемент. В нашем случае это будет 4.
36. Дан массив с числами. Найдите среднее арифметическое его элементов (сумма элементов делить на количество) не используя цикл.
37. Найдите сумму чисел от 1 до 100 не используя цикл.
38. Выведите столбец чисел от 1 до 100 не используя цикл.
39. Заполните массив 10-ю иксами не используя цикл.
40. Заполните массив 10-ю случайными числами от 1 до 10 так, чтобы они не повторялись. Цикл использовать нельзя.
41. Найдите факториал заданного числа не используя цикл. Факториал - это произведение чисел от 1 до заданного числа включительно.
42. Дано число. Найдите сумму цифр этого числа не используя цикл.
43. Дана строка. Сделайте заглавным последний символ этой строки не используя цикл.
44. Дан массив с числами. Получите из него массив с квадратными корнями этих чисел не используя цикл.

45. Заполните массив числами от 1 до 26 так, чтобы ключами этих чисел были буквы английского алфавита: ['a'=>1, 'b'=>2...]. Сделайте это не используя цикл.
46. Дана строка с числами '1234567890'. Найдите сумму пар чисел: 12+34+56+78+90. Решите задачу, не используя цикл.
47. Выведите на экран текущий год, месяц, день, час, минуту, секунду.
48. Выведите текущую дату-время в форматах '2025-12-31', '31.12.2025', '31.12.13', '12:59:59'.
49. С помощью функций mktime и date выведите 12 февраля 2025 года в формате '12.02.2025'.
50. Создайте массив дней недели \$week. Выведите на экран название текущего дня недели с помощью массива \$week и функции date. Узнайте какой день недели был 06.06.2006, в ваш день рождения.
51. Создайте массив месяцев \$month. Выведите на экран название текущего месяца с помощью массива \$month и функции date.
52. Найдите количество дней в текущем месяце. Скрипт должен работать независимо от месяца, в котором он запущен.
53. Сделайте поле ввода, в которое пользователь вводит год (4 цифры), а скрипт определяет високосный ли год.
54. Сделайте форму, которая спрашивает дату в формате '31.12.2025'. С помощью функций mktime и explode переведите эту дату в формат timestamp. Узнайте день недели (словом) за введенную дату.
55. Сделайте форму, которая спрашивает дату в формате '2025-12-31'. С помощью функций mktime и explode переведите эту дату в формат timestamp. Узнайте месяц (словом) за введенную дату.
56. Узнайте сколько дней осталось до Нового Года. Скрипт должен работать в любом году.
57. Сделайте форму с одним полем ввода, в которое пользователь вводит год. Найдите все пятницы 13-е в этом году. Результат выведите в виде массива дат.
58. Узнайте какой день недели был 100 дней назад.
59. Сделайте функцию, которая параметрами принимает 2 числа. Если эти числа равны - пусть функция вернет true, а если не равны - false.
60. Сделайте функцию, которая параметрами принимает 2 числа. Если их сумма больше 10 - пусть функция вернет true, а если нет - false.
61. Сделайте функцию, которая параметром принимает число и проверяет - отрицательное оно или нет. Если отрицательное - пусть функция вернет true, а если нет - false.
62. С помощью цикла for сформируйте строку '123456789' и запишите ее в переменную \$str.
63. С помощью цикла for сформируйте строку '987654321' и запишите ее в переменную \$str.
64. С помощью цикла for сформируйте строку '-1-2-3-4-5-6-7-8-9-' и запишите ее в переменную \$str. Спросите у пользователя имя с помощью формы. Сделайте чекбокс: если он отмечен, то поприветствуйте пользователя, если не отмечен - попрощайтесь с пользователем.
65. Спросите у пользователя, какие из языков он знает: html, css, php, javascript. Выведите на экран те языки, которые знает пользователь.
66. Спросите у пользователя знает ли он PHP с помощью двух radio-кнопок. Выведите результат на экран. Сделайте так, чтобы по умолчанию один из вариантов был уже отмечен.
67. Спросите у пользователя его возраст с помощью нескольких radio-кнопок. Варианты ответа сделайте такими: менее 20 лет, 20-25, 26-30, более 30.
68. Спросите у пользователя его возраст с помощью select. Варианты ответа сделайте такими: менее 20 лет, 20-25, 26-30, более 30.

69. Спросите у пользователя с помощью мультиселекта, какие из языков он знает: html, css, php, javascript. Выведите на экран те языки, которые знает пользователь.