

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Якушин Владимир Андреевич
Должность: ректор, д.ю.н., профессор
Дата подписания: 06.10.2024
Уникальный программный ключ:
a5427c2559e1ff4b007ed9b1994671e27053e0dc

Министерство науки и высшего образования РФ
Образовательная автономная некоммерческая организация
высшего образования
«Волжский университет имени В.Н. Татищева» (институт)

УТВЕРЖДАЮ

Ректор Якушин В.А.
от 27.06.2024г.

Рабочая программа

Электронный бизнес

Направление подготовки 09.03.02 Информационные системы и технологии

Квалификация (степень) выпускника – бакалавр

Форма обучения – очная, заочная, очно-заочная

Тольятти, 2024 г.

Рабочая программа **Электронный бизнес** составлена с требованиями ФГОС, ВО, ОПОП по направлению подготовки 09.03.02 Информационные системы и технологии (уровень высшего образования: бакалавриат) и учебного плана.

Одобрено Учебно-методическим советом вуза
протокол № 63 от 07.05.2024г

Председатель УМС
к.п.н. И.И. Муртаева

1. ПЕРЕЧЕНЬ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

В результате освоения дисциплины у обучающихся должны быть сформированы следующие общепрофессиональные компетенции и профессиональные компетенции:

Наименование компетенции	Код компетенции
Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-1

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Данная учебная дисциплина к части, формируемой участниками образовательных отношений 09.03.02 «Информационные системы и технологии».

В таблице 1 представлен перечень компетенций с указанием перечня дисциплин, формирующих эти компетенции согласно учебному плану ОПОП

Таблица 1

Код компетенции	Наименование компетенции, формируемой в рамках освоения дисциплины	Предшествующие дисциплины, формирующие указанную компетенцию	Последующие дисциплины, формирующие указанную компетенцию
ПК-1	Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	Моделирование Системное программное обеспечение Базовые технологии и процессы Сети и телекоммуникации Надежность систем	Методы и средства проектирование информационных систем и технологий Защита информации Научно исследовательская работа Инструментальные средства информационных систем Архитектура информационных систем Производственная практика. Технологическая (проектно-технологическая) практика Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты

* в качестве этапа формирования компетенций используются номера семестров согласно учебного плана ОПОП

Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы направления подготовки, представлен в таблице:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ПК-1 Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-1.1. Планирует процедуры создания, сопровождения и интеграции программных модулей и компонент ИС, автоматизирующих задачи организационного управления и бизнес-процессы ПК 1.2. Разрабатывает, сопровождает и интегрирует программные модули и компоненты ИС, автоматизирующих задачи организационного управления и бизнес-процессы ПК 1.3. Организует разработку и сопровождение ИС, автоматизирующих задачи организационного управления и бизнес-процессы ПК 1.4. Организует интеграцию программных модулей и компонент и верификацию программного продукта

3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Всего	Семестр
		6
Общая трудоёмкость дисциплины	108 час 3 з.е.	108 час 3 з.е.
Контактная работа с преподавателем (всего)	64	64
В том числе:		
Лекции	32	32
Практические / семинарские занятия	-	-
Лабораторные занятия	32	32
Консультации	-	-
Самостоятельная работа (всего)	44	44
В том числе (если есть):		
Курсовой проект / работа	-	-
Расчетно-графическая работа	-	-
Контрольная работа	-	-
Реферат / эссе / доклад	-	-
Иное	44	44
Вид промежуточной аттестации (зачет, экзамен)	Зачет	Зачет

ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Всего	Семестр
		8
Общая трудоёмкость дисциплины	108 час 3 з.е.	108 час 3 з.е.
Контактная работа с преподавателем (всего)	16	16
В том числе:		
Лекции	8	8
Практические / семинарские занятия	-	-
Лабораторные занятия	8	8
Консультации	-	-
Самостоятельная работа (всего)	92	92
В том числе (если есть):		
Курсовой проект / работа	-	-
Расчетно-графическая работа	-	-
Контрольная работа	-	-
Реферат / эссе / доклад	-	-
Иное	92	92
Вид промежуточной аттестации (зачет, экзамен)	Зачет	Зачет

ОЧНО-ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Всего	Семестр
		8
Общая трудоёмкость дисциплины	108 час 3 з.е.	108 час 3 з.е.
Контактная работа с преподавателем (всего)	32	32
В том числе:		
Лекции	16	16
Практические / семинарские занятия	-	-
Лабораторные занятия	16	16
Консультации	-	-
Самостоятельная работа (всего)	76	76
В том числе (если есть):		
Курсовой проект / работа	-	-
Расчетно-графическая работа	-	-
Контрольная работа	-	-
Реферат / эссе / доклад	-	-
Иное	76	76
Вид промежуточной аттестации (зачет, экзамен)	Зачет	Зачет

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

ОЧНАЯ ФОРМА ОБУЧЕНИЯ

№ п/п	Тема	Количество часов на			
		лекции	практически е /семинарски е занятия	лабораторн ые занятия	самостояте льную работу
1	ИНТЕРНЕТ – ОСНОВА ИНТЕРАКТИВНОГО БИЗНЕСА	10		10	15
2	ПРИНЦИПЫ РАБОТЫ ИНТЕРНЕТ-ПРИЛОЖЕНИЙ	10		10	15
3	РАЗРАБОТКА ИНТЕРНЕТ- ПРИЛОЖЕНИЙ НА ЯЗЫКЕ PHP	12		12	14
Итого		32		32	44

ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

№ п/п	Тема	Количество часов на			
		лекции	практически е /семинарски е занятия	лабораторн ые занятия	самостояте льную работу

1	ИНТЕРНЕТ – ОСНОВА ИНТЕРАКТИВНОГО БИЗНЕСА	2		2	30
2	ПРИНЦИПЫ РАБОТЫ ИНТЕРНЕТ-ПРИЛОЖЕНИЙ	2		2	30
3	РАЗРАБОТКА ИНТЕРНЕТ-ПРИЛОЖЕНИЙ НА ЯЗЫКЕ PHP	4		4	32
Итого		8		8	92

ОЧНО-ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

№ п/п	Тема	Количество часов на			
		лекции	практически е /семинарски е занятия	лабораторн ые занятия	самостояте льную работу
1	ИНТЕРНЕТ – ОСНОВА ИНТЕРАКТИВНОГО БИЗНЕСА	4		4	25
2	ПРИНЦИПЫ РАБОТЫ ИНТЕРНЕТ-ПРИЛОЖЕНИЙ	4		4	25
3	РАЗРАБОТКА ИНТЕРНЕТ-ПРИЛОЖЕНИЙ НА ЯЗЫКЕ PHP	8		8	26
Итого		16		16	76

4.2. КРАТКОЕ СОДЕРЖАНИЕ ЛЕКЦИОННОГО КУРСА

Тема 1. ИНТЕРНЕТ – ОСНОВА ИНТЕРАКТИВНОГО БИЗНЕСА

Электронный бизнес – закономерный этап инновационной спирали

Особенности Интернет как среды электронной коммерции

Основные определения

Этапы разработки и продвижения Интернет-проекта: Планирование Интернет-проекта. Определение контента сайта. Размещение сайта в Интернет. Регистрация в поисковых системах. Регистрация в рейтингах. Баннерная реклама. Обмен текстовыми ссылками. Массовая рассылка рекламных сообщений. Размещение объявлений о сайте на электронных досках. Рассылка в телеконференции (группы новостей).

Тема 2. ПРИНЦИПЫ РАБОТЫ ИНТЕРНЕТ-ПРИЛОЖЕНИЙ

Web-дизайн и web-программирование.

Архитектура «клиент-сервер».

Отладка web-приложений на сервере.

Сравнительный анализ возможностей web-технологий: HTML, Java и JavaScript, CGI, SSI, PHP, Macromedia Flash, ActiveX, ASP.

Использование баз данных в Интернет.

Интерактивное взаимодействие с помощью форм: Элементы управления формы. Способы передачи параметров (Строка передаваемых параметров, Метод GET, Метод POST).

Тема 3. РАЗРАБОТКА ИНТЕРНЕТ-ПРИЛОЖЕНИЙ НА ЯЗЫКЕ PHP

Область применения языка PHP.

Теги сценария PHP.

Константы.

Переменные.

Типы данных: Скалярные типы. Составные типы (Массивы. Приведение типов).
 Управляющие конструкции.
 Операторы.
 Функции: Определение и вызов функций. Вложенные функции. Возврат значений из функций. Рекурсивные функции (Объекты (классы)).
 Работа с файлами: Стандартные функции PHP для работы с файлами. Обработка данных в нескольких файлах (Использование атрибута ACTION. Операторы включения файлов)
 Поддержка баз данных в PHP.
 Использование СУБД (История создания MySQL. Взаимодействие с СУБД MySQL. Способ хранения таблиц и баз данных (Типы таблиц. Типы данных в СУБД MySQL).
 Стандартные функции PHP для работы с MySQL (Соединение с сервером. Выбор базы данных. Выполнение указанного запроса. Извлечение информации. Ввод данных).
 Обработка ошибок.

4.3. ТЕМАТИКА ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Лабораторная работа №1. PHP. Переменные, массивы
 Лабораторная работа №2 PHP. Классы
 Лабораторная работа № 3. Основы работы в СУБД MySQL
 Лабораторная работа №4 PHP. Доступ к MySQL
 Лабораторная работа №5 PHP. Использование операторов включения файлов, глобальных массивов, сессий
 Лабораторная работа №6 PHP. Динамическая загрузка содержимого страницы с помощью оператора INCLUDE

5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

5.1 Основная литература

Тузовский, А. Ф. Проектирование и разработка web-приложений : учебник для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2024. — 219 с. — (Высшее образование). — ISBN 978-5-534-16300-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561176>

Полужктова, Н. Р. Разработка веб-приложений : учебник для вузов / Н. Р. Полужктова. — 2-е изд. — Москва : Издательство Юрайт, 2024. — 204 с. — (Высшее образование). — ISBN 978-5-534-18645-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567610>

5.2 Дополнительная литература

Гаврилов, Л. П. Электронная коммерция : учебник и практикум для вузов / Л. П. Гаврилов. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 579 с. — (Высшее образование). — ISBN 978-5-534-17867-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560846>

Гаврилов, Л. П. Цифровой бизнес : учебник и практикум для вузов / Л. П. Гаврилов. — 6-е изд. — Москва : Издательство Юрайт, 2024. — 311 с. — (Высшее образование). — ISBN 978-5-534-17869-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/568755>

5.3. Ресурсы информационно-коммуникационной сети «Интернет»

Адрес Интернет ресурса	Название Интернет ресурса	Режим доступа
------------------------	---------------------------	---------------

http://intuit.ru/	Интернет-университет информационных технологий	Свободный
http://vkit.ru/	Сайт журнала «Вестник компьютерных и информационных технологий»	Свободный
http://ru.wikipedia.org/	Свободная общедоступная мультязычная универсальная интернет-энциклопедия	Свободный

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ

Дисциплина «**Электронный бизнес**» изучается в течение одного семестра. При планировании и организации времени, необходимого на изучение обучающимся дисциплины, необходимо придерживаться следующих рекомендаций.

В период между сессиями студенты должны вести конспект лекций, изучать теоретический материал в соответствии с программой курса, выполнять предложенные преподавателем задания для самостоятельной работы, готовиться к сдаче зачета и экзамена, прорабатывая необходимый материал согласно перечню вопросов для подготовки к зачету и экзамену и списку рекомендованной литературы.

Выполнение лабораторных работ относится к числу обязательных видов работ. Перед выполнением работы необходимо внимательно ознакомиться с теоретическим материалом, представленным в методических указаниях к соответствующей лабораторной работе. При необходимости можно воспользоваться рекомендуемой литературой. В ходе выполнения работы необходимо руководствоваться порядком выполнения лабораторной работы и указаниями преподавателя, при этом должны соблюдаться правила техники безопасности. Результатом выполнения работы является отчет, который должен быть аккуратно оформлен и выполнен в соответствии с требованиями, приведенными в методических указаниях.

В указанное преподавателем время обучающиеся защищают отчеты. Защита проводится в виде собеседования по контрольным вопросам, приведенным в методических указаниях. Кроме того, преподаватель может задавать дополнительные вопросы, касающиеся результатов эксперимента, выводов по результатам опытов и т.п. К промежуточной аттестации допускаются обучающиеся, выполнившие все лабораторные работы и защитившие отчеты по ним. При наличии задолженности по лабораторным работам, по согласованию с преподавателем, возможна замена работы по выполнению отчета на реферат по теме соответствующего лабораторного занятия с последующей его защитой.

В течение семестра и во время сессии основным видом подготовки являются самостоятельные занятия. Они включают в себя изучение вопросов, вынесенных на самостоятельное изучение, оформление отчетов по лабораторным работам, курсовое проектирование, а так же подготовку к промежуточной аттестации.

Систематическая работа в соответствии с программой дисциплины – условие успешного освоения материала.

Методические рекомендации по обучению лиц с ограниченными возможностями здоровья и инвалидов

Освоение дисциплины обучающимися с ограниченными возможностями здоровья может быть организовано как совместно с другими обучающимися, так и в отдельных группах. Предполагаются специальные условия для получения образования обучающимися с ограниченными возможностями здоровья.

Профессорско-педагогический состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья, индивидуальными программами реабилитации инвалидов (при наличии). При

необходимости осуществляется дополнительная поддержка преподавания тьюторами, психологами, социальными работниками, прошедшими подготовку ассистентами.

В соответствии с методическими рекомендациями Минобрнауки РФ (утв. 8 апреля 2014 г. N АК-44/05вн) в курсе предполагается использовать социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Подбор и разработка учебных материалов производятся с учетом предоставления материала в различных формах: аудиальной, визуальной, с использованием специальных технических средств и информационных систем.

Медиа материалы также следует использовать и адаптировать с учетом индивидуальных особенностей обучения лиц с ОВЗ.

Освоение дисциплины лицами с ОВЗ осуществляется с использованием средств обучения общего и специального назначения (персонального и коллективного использования). Материально-техническое обеспечение предусматривает приспособление аудиторий к нуждам лиц с ОВЗ.

Форма проведения аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей. Для студентов с ОВЗ предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной или электронной форме (для лиц с нарушениями опорно-двигательного аппарата);
- в печатной форме или электронной форме с увеличенным шрифтом и контрастностью (для лиц с нарушениями слуха, речи, зрения);
- методом чтения ассистентом задания вслух (для лиц с нарушениями зрения).

Студентам с инвалидностью увеличивается время на подготовку ответов на контрольные вопросы. Для таких студентов предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге или набором ответов на компьютере (для лиц с нарушениями слуха, речи);
- выбором ответа из возможных вариантов с использованием услуг ассистента (для лиц с нарушениями опорно-двигательного аппарата);
- устно (для лиц с нарушениями зрения, опорно-двигательного аппарата).

При необходимости для обучающихся с инвалидностью процедура оценивания результатов обучения может проводиться в несколько этапов.

7. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ

Windows (для академических организациях, лицензия Microsoft Imagine (ранее MSDN AA, Dream Spark);

Open Office (свободное ПО)

Denwer3_Base_2013-06-02_a2.2.22_p5.3.13_m5.5.25_pma3.5.1_xdebug

8. НЕОБХОДИМАЯ МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА

Оборудование лекционных аудиторий Б-609: офисная мебель на 20 мест, 9 ПК с доступом в Интернет и ЭИОС, демонстрационное оборудование: проектор – 1 шт.; экран, доска ученическая, рабочее место преподавателя.

Оборудование аудиторий для лабораторных занятий ауд. Б-609: офисная мебель на 20 мест, 9 ПК с доступом в Интернет и ЭИОС, демонстрационное оборудование: проектор – 1 шт.; экран, доска ученическая, рабочее место преподавателя.

Оборудование аудиторий для самостоятельной работы: читальный зал НТБ: 5 ПК с доступом в Интернет; ауд. Б-609: офисная мебель на 20 мест, 9 ПК с доступом в Интернет и ЭИОС, демонстрационное оборудование: проектор – 1 шт.; экран, доска ученическая, рабочее место преподавателя.

Разработчик:

Кафедра ИиСУ

(место работы)

**доцент кафедры
ИиСУ**

(занимаемая должность)

Е.Л. Румянцева

(инициалы, фамилия)

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ОБРАЗОВАТЕЛЬНАЯ АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛЖСКИЙ УНИВЕРСИТЕТ имени В.Н. ТАТИЩЕВА» (институт)**

Фонд оценочных средств

«Электронный бизнес»

для направления подготовки

09.03.02 «Информационные системы и технологии»

Квалификация (степень) выпускника – бакалавриат

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

Оценочные средства разработаны для оценки профессиональных компетенций: ПК-1.

1. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты освоения основной профессиональной образовательной программы (ОПОП) – компетенции обучающихся определяются требованиями стандарта по направлению подготовки (специальности) и формируются в соответствии с матрицей компетенций ОПОП (Таблица 2)

Планируемые результаты обучения по дисциплине – знания, умения, навыки и (или) опыт деятельности, характеризующие этапы формирования компетенций и обеспечивающие достижение планируемых результатов освоения образовательной программы, формируются в соответствии с картами компетенций ОПОП.

Таблица 1

Планируемые результаты обучения по дисциплине

Код и наименование общепрофессиональной компетенции	Код и наименование индикатора достижения общепрофессиональной компетенции
2	3
ПК-1 Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-1.1. Планирует процедуры создания, сопровождения и интеграции программных модулей и компонент ИС, автоматизирующих задачи организационного управления и бизнес-процессы ПК 1.2. Разрабатывает, сопровождает и интегрирует программные модули и компоненты ИС, автоматизирующих задачи организационного управления и бизнес-процессы ПК 1.3. Организует разработку и сопровождение ИС, автоматизирующих задачи организационного управления и бизнес-процессы ПК 1.4. Организует интеграцию программных модулей и компонент и верификацию программного продукта

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

Результаты обучения по дисциплине «Электронный бизнес» направления подготовки 09.03.02 «Информационные системы и технологии» определяются показателями и критериями оценивания сформированности компетенций на этапах их формирования представлены в табл. 2.

Таблица 2

Матрица соответствия оценочных средств запланированным результатам обучения

Компетенции	Оценочные средства		
	Текущий контроль		Промежуточный контроль
	Оценочное	Оценочное	Зачет (вопросы к зачету)

	средство 1 (лабораторное задание)	средство 2	
ПК-1	ПК-1.1. ПК -1.2. ПК -1.3. ПК-1.4.		ПК-1.1. ПК -1.2. ПК -1.3. ПК-1.4.

Показатели и критерии оценивания сформированности компетенций (промежуточного контроля)

На этапе промежуточной аттестации используется система оценки успеваемости обучающихся, которая позволяет преподавателю оценить уровень освоения материала обучающимися. Критерии оценивания сформированности планируемых результатов обучения (дескрипторов) представлены в карте компетенции ОПОП.

Форма оценки знаний: оценка - 5 «отлично»; 4 «хорошо»; 3 «удовлетворительно»; 2 «неудовлетворительно». Лабораторные работы, практические занятия, практика оцениваются: «зачет», «незачет». Возможно использование балльно-рейтинговой оценки.

Шкала оценивания:

«Зачет» – выставляется, если сформированность заявленных дескрипторов компетенций на 51 % и более оценивается не ниже «удовлетворительно» при условии отсутствия критерия «неудовлетворительно». Выставляется, когда обучающийся показывает хорошие знания изученного учебного материала; самостоятельно, логично и последовательно излагает и интерпретирует материалы учебного курса; полностью раскрывает смысл предлагаемого вопроса; владеет основными терминами и понятиями изученного курса; показывает умение переложить теоретические знания на предполагаемый практический опыт.

«Отлично» – выставляется, если сформированность заявленных дескрипторов компетенций 85 % более (в соответствии с картами компетенций ОПОП) оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно»: студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций;

«Хорошо» – выставляется, если сформированность заявленных дескрипторов компетенций на 61 % и более (в соответствии с картами компетенций ОПОП) оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно», допускается оценка «удовлетворительно»: обучающийся показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций;

«Удовлетворительно» – выставляется, если сформированность заявленных дескрипторов компетенций 51 % и более (в соответствии с картами компетенций ОПОП) оценивается критериями «удовлетворительно», «хорошо» и «отлично»: обучающийся показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение конкретной практической задачи из числа предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой;

«Неудовлетворительно» «Незачет» – выставляется, если сформированность заявленных дескрипторов компетенций менее чем 51 % (в соответствии с картами компетенций ОПОП): при ответе обучающегося выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины.

Ответы и решения обучающихся оцениваются по следующим общим критериям:

распознавание проблем; определение значимой информации; анализ проблем; аргументированность; использование стратегий; творческий подход; выводы; общая грамотность.

Соответствие критериев оценивания сформированности планируемых результатов обучения (дескрипторов) системам оценок представлено в табл.

Таблица 4

Интегральная оценка

Критерии	Традиционная оценка	Балльно-рейтинговая оценка
5	5	86 - 100
4	4	61-85
3	3	51-60
2 и 1	2, Незачет	0-50
5, 4, 3	Зачет	51-100

Обучающиеся обязаны сдавать все задания в сроки, установленные преподавателем. Оценка «Удовлетворительно» по дисциплине, может выставляться и при неполной сформированности компетенций в ходе освоения отдельной учебной дисциплины, если их формирование предполагается продолжить на более поздних этапах обучения, в ходе изучения других учебных дисциплин.

Показатели и критерии оценки достижений студентом запланированных результатов освоения дисциплины в ходе текущего контроля и промежуточной аттестации

Оценка, уровень	Критерии
«отлично», повышенный уровень	Студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций
«хорошо», пороговый уровень	Студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций
«удовлетворительно», пороговый уровень	Студент показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение конкретной практической задачи из числа предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой
«неудовлетворительно», уровень не сформирован	При ответе студента выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины

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

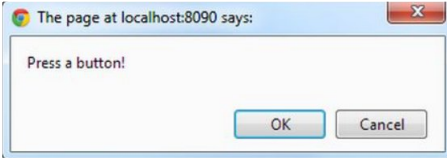
Перечень компетенций и индикаторов достижения компетенций, для оценки

сформированности которых используется данный ФОС

Код и наименование компетенции	Код и наименование индикатора достижения компетенции, реализуемые дисциплиной
ПК-1 Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-1.1. Планирует процедуры создания, сопровождения и интеграции программных модулей и компонент ИС, автоматизирующих задачи организационного управления и бизнес-процессы ПК 1.2. Разрабатывает, сопровождает и интегрирует программные модули и компоненты ИС, автоматизирующих задачи организационного управления и бизнес-процессы ПК 1.3. Организует разработку и сопровождение ИС, автоматизирующих задачи организационного управления и бизнес-процессы ПК 1.4. Организует интеграцию программных модулей и компонент и верификацию программного продукта

Номер задания	Содержание вопроса	Правильный ответ на задание
1.	Выберите правильный вариант ответа. Поставлена задача выбрать язык программирования для создания интерактивных веб-страниц, сценарии которого выполняются в основном на стороне клиента. A) PHP <u>B) JavaScript</u> C) Python D) C++	В
2.	Выберите правильный вариант ответа. Поставлена задача выбрать язык программирования для обработки сценариев, которые выполняются в основном на стороне сервера. <u>A) PHP</u> B) JavaScript C) Python D) C++	А
3.	Выберите правильный вариант ответа. Java-программа, встроенная в веб-страницу. Запускается внутри веб-браузера и работает на стороне клиента. Размещается на веб-сервере используется для динамичности сайта. С помощью каких тегов внедряется в HTML-страницу? A) alert B) let age <u>C) OBJECT или APPLET</u> D) OBJECT	С
4.	Выберите правильный вариант ответа. Представлен код JavaScript. Выберите результат его работы.	А

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre><!DOCTYPE html> <script> "use strict"; alert('Привет'); alert('Мир'); </script></pre> A) Подтвердите действие на странице run.plnkr.co Привет, Мир!  B) Подтвердите действие на странице run.plnkr.co Привет  C) Подтвердите действие на странице run.plnkr.co Мир  D) Подтвердите действие на странице run.plnkr.co 	
5.	Выберите правильный вариант ответа. Представлен код JavaScript. Выберите результат его работы. <pre><!DOCTYPE html> <script> "use strict"; alert("не число" / 2); </script></pre> A) Infinity числовое значение B) object ошибка в языке, на самом деле это не объект C) NaN, такое деление является ошибкой D) "не число" / 2	C
6.	Выберите правильный вариант ответа. Представлен код JavaScript. Выберите окончательное значение переменной \$. <pre><!DOCTYPE html> <script> "use strict"; let name = "Иван"; alert(`Привет, \${name}!`); alert(`результат: \${1 + 2}`); </script></pre>	A

Номер задания	Содержание вопроса	Правильный ответ на задание
	A) 3 B) Привет C) Иван D) Привет Иван	
7.	Выберите правильный вариант ответа. Пример модального окна JavaScript. A)  B)  C)  D) 	В
8.	Выберите правильный вариант ответа. Представлен код JavaScript. Какое значение необходимо внести в модальное окно для положительного срабатывания конструкции if? <pre> <!DOCTYPE html> <script> "use strict"; let year = prompt('В каком году была опубликована спецификация ECMAScript-2015?', ''); if (year == 2015) { alert("Правильно!"); alert("Вы такой умный!"); } else { alert('А вот и неправильно!'); } </script> </pre> A) Правильно B) 2015 C) Неправильно D) А вот и неправильно!	В
9.	Выберите правильный вариант ответа.	А

Номер задания	Содержание вопроса	Правильный ответ на задание
	Представлен код JavaScript. Условие работы цикла. <pre><!DOCTYPE html> <script> "use strict"; let i = 3; while (i) { alert(i); i--; } </script></pre> <u>A) когда i будет равно 0, условие станет ложным, и цикл остановится</u> B) когда i будет равно 1, условие станет ложным, и цикл остановится C) когда i будет равно 3, условие станет ложным, и цикл остановится D) когда i будет равно 5, условие станет ложным, и цикл остановится	
10.	Выберите правильный вариант ответа. Представлен код JavaScript. Какие значения будет выдавать цикл? <pre><!DOCTYPE html> <script> "use strict"; for (let i = 0; i < 3; i++) { alert(i); } </script></pre> <u>A) 0 1 2</u> B) 1 2 3 C) 1 2 D) 0 3	A
11.	Выберите правильный вариант ответа. Представлен код JavaScript. Какие значения будет выдавать цикл в соответствии с условием if? <pre><!DOCTYPE html> <script> "use strict"; for (let i = 0; i < 10; i++) { if (i % 2 == 0) continue; alert(i); } </script></pre> A) 0 1 3 5 7 9 <u>B) 1 3 5 7 9</u> C) 1 2 3 4 5 6 7 8 9	B

Номер задания	Содержание вопроса	Правильный ответ на задание
	D) 0 1 2 3 4 5 6 7 8 9	
12.	Выберите правильный вариант ответа. Представлен код JavaScript. Количество вызовов функции. <pre><!DOCTYPE html> <script> "use strict"; function showMessage() { alert('Всем привет!'); } showMessage(); showMessage(); </script></pre> A) 2 B) 3 C) 5 D) 0	A
13.	Выберите правильный вариант ответа. Представлен пример конструкции <pre>let age = 16; let state = 'CA'; if (state == 'CA') { if (age >= 16) { console.log("You can drive."); } }</pre> A) вложенного оператора If...else B) оператора If C) вложенного оператора If D) оператора else	C
14.	Выберите правильный вариант ответа. Представлен код PHP. Определите отображение данных работы кода. <pre>\$i = 0; for (; \$i < 10;) { echo \$i; \$i += 2; }</pre> A) 02468 B) 0123456789 C) 0246810 D) 13579	A
15.	Выберите правильный вариант ответа. Представлен код PHP. Определите отображение данных работы кода. <pre><?php \$array = array("foo", "bar", "hello", "world"</pre>	C

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>); var_dump(\$array); ?></pre> A) <pre>array(4) { ["foo"]=> string(3) "bar" ["bar"]=> string(3) "foo" [100]=> int(-100) [-100]=> int(100) }</pre> B) <pre>array(4) { [0]=> string(1) "a" [1]=> string(1) "b" [6]=> string(1) "c" [7]=> string(1) "d" }</pre> C) <pre>array(4) { [0]=> string(3) "foo" [1]=> string(3) "bar" [2]=> string(5) "hello" [3]=> string(5) "world" }</pre> D) <pre>array(4) { [1]=> string(3) "bar" [2]=> string(5) "hello" [3]=> string(5) "world" }</pre>	
16.	При выборе языка программирования необходимо учесть достоинства и недостатки языка. Перечислите, с вашей точки зрения, недостатки языка JavaScript.	Недостатки языка JavaScript: 1. Отсутствие чтения и загрузки файлов, из за этого низкая безопасность от вредоносных программ 2. Динамическая типизация, из за этого нет возможности выявить ошибки заранее, только на этапе работы.
17.	При выборе языка программирования необходимо учесть достоинства и недостатки языка. Перечислите, с вашей точки зрения, достоинства языка JavaScript.	Достоинства языка JavaScript: 1. Доступность. 2. Схожесть с другими языками (типичность). 3. Повсеместная поддержка. 4. Полная интеграция с браузером.

Номер задания	Содержание вопроса	Правильный ответ на задание
18.	Для каких целей указывают высоту и ширину в теге <code>applet</code> ? <code><applet code=Applet1 width=100 height=50></code> <code></applet></code>	В данном примере в теге <code>applet</code> указаны высота и ширина окна для отображения апплета.
19.	Определите по представленному коду JavaScript имена используемых переменных <code><!DOCTYPE html></code> <code><script></code> <code>"use strict";</code> <code>let \$ = 1;</code> <code>let _ = 2;</code> <code>alert(\$ + _);</code> <code></script></code>	В данном коде объявлены две переменные с именем "\$" и с именем "_".
20.	Представлен код JavaScript. Слово Test будет записью по умолчанию поля ввода модального окна? <code><!DOCTYPE html></code> <code><script></code> <code>"use strict";</code> <code>let test = prompt("Test", "");</code> <code></script></code>	Нет. Слово Test будет надписью над полем ввода модального окна.
21.	Представлен код JavaScript. Будет ли содержать поле ввода первое диалоговое окно? <code><!DOCTYPE html></code> <code><script></code> <code>"use strict";</code> <code>let isBoss = confirm("Ты здесь главный?");</code> <code>alert(isBoss);</code> <code></script></code>	Нет. Первое диалоговое окно будет содержать только вопрос "Ты здесь главный?".
22.	Представлен код JavaScript. Какое значение будет содержать переменная после выполнения программы? <code><!DOCTYPE html></code> <code><script></code> <code>"use strict";</code> <code>alert(Number(" 123 "));</code> <code>alert(Number("123z"));</code> <code>alert(Number(false));</code> <code></script></code>	После выполнения программы переменная Number будет содержать 0.
23.	Представлен код JavaScript. Какое значение будет содержать переменная	После выполнения программы переменная Number будет содержать 1.

Номер задания	Содержание вопроса	Правильный ответ на задание
	после выполнения программы? <pre><!DOCTYPE html> <script> "use strict"; alert(Number(" 123 ")); alert(Number("123z")); alert(Number(true)); </script></pre>	
24.	Представлен код JavaScript. Перечислите значения отображаемые в модальных окнах? <pre><!DOCTYPE html> <script> "use strict"; alert('2' > 1); alert('01' == 1); </script></pre>	В модальных окнах будут отображаться результаты работы операторов сравнения: true true
25.	Представлен код JavaScript. При каком значении переменной age значение переменной isAgeEnough true. <pre>const isAgeEnough = age >= 18; /* переменная isAgeEnough принимает значение true или false */ /* если true, то */ if (isAgeEnough) { console.log('Sarah can start driving license'); }</pre>	Переменная isAgeEnough принимает значение true при age равное 18, 19, 20 и т.д.
26.	Представлен код JavaScript. Для каких целей поставлен в теле цикла break? <pre>for (var i = 1; ; i++) { if (i >= 8) { // условие прерывания цикла break; } console.log(i); }</pre>	Без условия цикл будет выполняться бесконечное количество раз. В этом случае чтобы его прервать (выйти из цикла) необходимо использовать инструкцию break.
27.	Представлен код JavaScript. Перечислите параметры объявленной функции. <pre>function sayWelcome (userFirstName, userLastName) { console.log(Добро пожаловать, \${userLastName} \${userFirstName} `); } sayWelcome('Иван', 'Иванов'); sayWelcome('Петр', 'Петров');</pre>	Функция sayWelcome имеет два параметра userFirstName и userLastName.
28.	Представлен код JavaScript. Кратко опишите причину проставления скобок у второго условия.	Реализована проверка, что x должно принадлежать [2; 7], а у не принадлежать [0; 5]. И обратите

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>let x = 4, y = -2; if(x >= 2 && x <= 7 && (y < 0 y > 5)) console.log("x попадает в [2; 7], y не попадает в [0; 5]");</pre>	внимание вот на эти круглые скобки. Приоритет у операции && больше, чем у , поэтому поставлены скобки.
29.	Представлен код JavaScript. Какую встроенную (стандартную) функцию использовали? <pre><!DOCTYPE html> <script> "use strict"; function showMessage() { alert('Всем привет!'); } showMessage(); showMessage(); </script></pre>	В данном коде использовали встроенную функцию alert.
30.	Представлен код страницы HTML с скриптом на JavaScript. Кратко опишите действия скрипта. Файл HTML <pre><div class="password"> <input type="password" id="password- input" placeholder="Введите пароль" name="password"> Показать пароль </div></pre> Файл JavaScript. <pre>\$('body').on('click', '.password-control', function() { if (\$('#password-input').attr('type') == 'password') { \$(this).html('Скрыть пароль'); \$('#password-input').attr('type', 'text'); } else { \$(this).html('Показать пароль'); \$('#password-input').attr('type', 'password'); } return false; });</pre>	Показывает пароль по ссылке.
31.	Представлен код на JavaScript. Кратко опишите для чего использовался цикл for. <pre>var arr = ["a", "b", "c"]; // массив for (var i = 0, length = arr.length; i < length; i++) { console.log(arr[i]); }</pre>	В представленном коде цикл «for» использовался для перебора элементов массива.
32.	Представлен код на JavaScript. Какая из	cb – её будем использовать в роли

Номер задания	Содержание вопроса	Правильный ответ на задание
	функций используется как колбэк . <pre>function cb() { console.log('callback'); } function fnWithCb(cbFn) { console.log('before calling the callback function'); cbFn(); } fnWithCb(cb);</pre>	колбэк функции, т.е. вызывать в fnWithCb.
33.	Представлен код WEB страницы. Определите действия PHP. <pre><!DOCTYPE html> <html> <head> <title>METANIT.COM</title> <meta charset="utf-8" /> </head> <body> <?php \$num = 10; echo \$num; ?> </body> </html></pre>	В части PHP кода на предложенной WEB странице выполнено: - определение переменной \$num - вывод значения переменной \$num на веб-страницу
34.	Представлен код PHP. Результат выполнения кода при a=4. <pre><?php \$a = 4; if(\$a > 0){ echo "Переменная a больше нуля"; } else{ echo "Переменная a меньше нуля"; } echo "
конец выполнения программы"; ?></pre>	В браузере будет отображен текст: Переменная a больше нуля
35.	Перечислите 4 типа переменных в PHP.	В зависимости от области видимости переменные PHP делятся на четыре типа: 1) локальные переменные 2) параметры функций 3) глобальные переменные 4) статические переменные
36.	Представлен код PHP. Информация полученная в браузере? <pre><?php phpinfo(); ?></pre>	Вызов функции phpinfo() показывает множество полезной информации о вашей системе и настройке, такой как доступные предопределенные

Номер задания	Содержание вопроса	Правильный ответ на задание
		переменные, загруженные РНР-модули и параметры настройки.
37.	Какие зарезервированные (предопределенные) объекты вы знаете?	В РНР используют: 1) зарезервированные идентификаторы в РНР 2) предопределенные классы 3) предопределенные константы
38.	Представлен код РНР. Результат выполнения кода. <pre><?php \$colors = array('red', 'blue', 'green', 'yellow'); foreach (\$colors as \$color) { echo "Вам нравится \$color?\n"; } ?></pre>	Вам нравится red? Вам нравится blue? Вам нравится green? Вам нравится yellow?
39.	Представлена часть кода РНР. Кратко опишите действия кода. <pre>\$age = array("Petya"=>"24", "Boris"=>"26", "Janna"=>"32");</pre>	Один из способов создания ассоциативного массива.
40.	Представлена часть кода РНР. Кратко опишите действия кода. <pre>\$age['Petya'] = "24"; \$age['Boris'] = "26"; \$age['Janna'] = "32";</pre>	Один из способов создания ассоциативного массива.
41.	Представлена код РНР. Кратко опишите действия кода. <pre><?php \$age = array("Petya"=>"24", "Boris"=>"26", "Janna"=>"32"); echo "Petya is " . \$age['Petya'] . " years old."; ?></pre>	Именованные ключи "Petya"=>"24", "Boris"=>"26", "Janna"=>"32"
42.	Представлена часть кода РНР. Кратко опишите действия кода. <pre><?php \$age = array("Petya"=>"24", "Boris"=>"26", "Janna"=>"32"); foreach(\$age as \$x => \$x_value) { echo "Key=" . \$x . ", Value=" . \$x_value; echo ""; } ?></pre>	Просмотр и отображение всех значений ассоциативного массива, с использованием цикла foreach
43.	Представлена часть кода РНР. Кратко опишите действия кода. <pre><?php</pre>	Сортировка массива в порядке возрастания осуществляется с помощью функции sort

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>\$fruits = ['lemon', 'orange', 'banana', 'apple']; sort(\$fruits);</pre>	
44.	Представлена часть кода PHP. Кратко опишите действия функции shuffle. <pre><?php \$languages = ['en' => 'English', 'fr' => 'French', 'es' => 'Spanish', 'pt' => 'Portuguese',]; shuffle(\$languages);</pre>	Для перемешивания элементов в массиве случайным образом, можно воспользоваться функцией shuffle.
45.	Представлена часть кода PHP. Кратко опишите действия функции array_reverse. <pre><?php \$languages = ['en' => 'English', 'fr' => 'French', 'es' => 'Spanish', 'pt' => 'Portuguese',]; \$languagesReverse = array_reverse(\$languages); echo '<pre>'; print_r(\$languagesReverse); echo '</pre>';</pre>	Для создания нового массива, в котором элементы нужно выстроить в обратном порядке, можно воспользоваться функцией array_reverse.
46.	В коде PHP представлены два массива \$base как основной массив и \$arrays как дополнительный. Кратко опишите действие представленного ниже кода PHP. <pre>array_diff(array \$base, array ...\$arrays): array</pre>	Функция array_diff вычисляет разницу между массивами. В качестве результата она возвращает массив со значениями основного массива, которых нет ни в одном из других массивов.
47.	Представлена часть кода PHP. Кратко опишите действия функции in_array. <pre>if (in_array('apple', \$fruits)) { // значение 'apple' имеется в массиве \$fruits }</pre>	Функция in_array проверяет есть ли значение 'apple' в массиве \$fruits.
48.	Опишите варианты доступа к элементам массива в PHP.	Способы доступа к элементам массива: - по их ключу или индексу - использовать цикл foreach для итерации по элементам массива (или цикл for)
49.	Перечислите типы массивов в PHP.	Типы массивов в PHP 1) Численно-индексированные

Номер задания	Содержание вопроса	Правильный ответ на задание
		массивы 2) Ассоциативные массивы 3) Многомерные массивы
50.	Представлена часть кода PHP. Как изменить код для организации сортировки в порядке убывания <pre><?php \$fruits = ['lemon', 'orange', 'banana', 'apple']; sort(\$fruits);</pre>	Функцию sort заменить на rsort или krsort.
51.	Представлен код JavaScript. Какой блок перехвата исключений использовался? <pre>function divide(v1, v2, dp) { try { return (v1 / v2).toFixed(dp); } catch(e) { console.log(`       error name   : \${ e.name }       error message: \${ e.message }     `); return 'ERROR'; } }</pre> A) try...catch B) catch...return C) try...return D) error...return	A
52.	Выберите правильный вариант ответа. Команда в коде программы позволяющая приостановить выполнение скрипта. A) debugger B) alert C) return D) catch	A
53.	Выберите правильный вариант ответа. Представлен код HTML со скриптом JS организации части форму . Определите тип организованной в коде структуры настройки формы. <pre><html> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <script> <!-- function btnClick() { var Txt1 = ""; var Txt2 = ""; Txt1 =</pre>	C

Номер задания	Содержание вопроса	Правильный ответ на задание								
	<pre>document.Test.bt.value; Txt2 = document.Test.bt.name; document.getElementById('ex1').innerHTML="<HR>" + "Вы нажали кнопку: " + Txt1.bold() + " с именем: " + Txt2.bold() + "<HR>"; } //--> </script> </head> <body> <H1>Нажатие кнопки</H1> <div id="ex1"></div> <FORM NAME="Test"> <INPUT TYPE="button" NAME="bt" VALUE="Щелкни здесь!" onClick="btnClick();"> </FORM> </body> </html></pre> A) <table><tr><td>Фамилия:</td><td> </td></tr><tr><td>Имя:</td><td> </td></tr><tr><td>Возраст:</td><td> 18 </td></tr><tr><td>Телефон:</td><td> </td></tr></table> Готово Сброс B). Флажок ☐ Состояние флажка можно изменить и этой кнопкой Смена состояния C) Нажатие кнопки Щелкни здесь! D) Message Author Name Gender male Magic number 42 Submit	Фамилия:		Имя:		Возраст:	18	Телефон:		
Фамилия:										
Имя:										
Возраст:	18									
Телефон:										
54.	Выберите правильный вариант ответа. Представлен код HTML со скриптом JS организации части форму . Определите тип	В								

Флажок ☐

Состояние флажка можно изменить и этой кнопкой

Смена состояния

Щелкни здесь!

Message

Author

Gender

Magic number

Submit

Номер задания	Содержание вопроса	Правильный ответ на задание								
	организованной в коде структуры настройки формы. <pre><html> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> </head> <body> <H1>Метод click флажка</H1> <FORM NAME="Test"> Флажок <INPUT TYPE="checkbox" NAME="ch">
Состояние флажка можно изменить и этой кнопкой <INPUT TYPE="button" VALUE="Смена состояния" onClick="document.Test.ch.click();" > </FORM> </body> </html></pre> A) <table><tr><td>Фамилия:</td><td> </td></tr><tr><td>Имя:</td><td> </td></tr><tr><td>Возраст:</td><td> 18 </td></tr><tr><td>Телефон:</td><td> </td></tr></table> Готово Сброс B) Флажок ☐ Состояние флажка можно изменить и этой кнопкой Смена состояния C) Нажатие кнопки Щелкни здесь! D) Message Author Name Gender male Magic number 42 Submit	Фамилия:		Имя:		Возраст:	18	Телефон:		
Фамилия:										
Имя:										
Возраст:	18									
Телефон:										
55.	Выберите правильный вариант ответа. Представлен код HTML со скриптом JS организации части форму . Определите тип организованной в коде структуры	A								

Номер задания	Содержание вопроса	Правильный ответ на задание
	настройки формы. <pre> <html> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <script> <!-- function Complete() { var Elem="Фамилия: " + document.Sell.Family.value + "\nИмя: " + document.Sell.Name.value + "\nВозраст: " + document.Sell.Age.value + "\nТелефон: " + document.Sell.Phone.value; alert (Elem); } function CheckAge (age) { if (age<18) return "18"; else return age; } //--> </script> </head> <body> <H1>Заполните анкету</H1> <FORM NAME="Sell"> <!-- Анкета --> <TABLE> <TR><TD>Фамилия:</TD> <TD><INPUT NAME="Family" SIZE=20 onBlur="this.value=this.value.toUpp erCase()" "></TD></TR> <TR><TD>Имя:</TD> <TD><INPUT NAME="Name" SIZE=20 onBlur="this.value=this.value.toUpp erCase()" "></TD></TR> <TR><TD>Возраст:</TD> <TD><INPUT NAME="Age" SIZE=3 VALUE="18" onBlur="this.value=CheckAge (this.va lue)" onFocus="this.select()" "></TD></TR> <TR><TD>Телефон:</TD> <TD><INPUT NAME="Phone" SIZE=10></TD></TR> </TABLE> <!-- Кнопки готовности и сброса --> </pre>	

Номер задания	Содержание вопроса	Правильный ответ на задание								
	<pre><INPUT TYPE="button" VALUE="Готово" onClick="Complete();"> <INPUT TYPE="reset" VALUE="Сброс"> </FORM> </body> </html></pre> A) <table><tr><td>Фамилия:</td><td> </td></tr><tr><td>Имя:</td><td> </td></tr><tr><td>Возраст:</td><td> 18 </td></tr><tr><td>Телефон:</td><td> </td></tr></table> Готово Сброс B) Флажок ☐ Состояние флажка можно изменить и этой кнопкой Смена состояния C) Нажатие кнопки Щелкни здесь! D) Message Author Name Gender male Magic number 42 Submit	Фамилия:		Имя:		Возраст:	18	Телефон:		
Фамилия:										
Имя:										
Возраст:	18									
Телефон:										
56.	Выберите правильный вариант ответа. Представлен код HTML со скриптом JS организации части формы. Определите вид передачи данных для заполнения списка при настройке формы. <pre><html> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> </head> <body> <H1>Динамическое заполнение списка</H1> <FORM NAME="Sel"> <!-- Пустой список ссылок--> <SELECT NAME="ListOfLinks"> </SELECT> <!-- Кнопка активизации выбранной ссылки--></pre>	В								

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> <INPUT TYPE="button" VALUE="Переход" onClick="window.location.href=document.links[Sel.ListOfLinks.selectedIndex];"> </FORM> <script><!-- // Цикл по всем ссылкам for(i=0; i<document.links.length; i++) { // Создание i-элемента списка и запись в него ссылки document.Sel.ListOfLinks.options[i] = new Option(document.links[i], i, false, false); } // Выделение первого элемента в списке document.Sel.ListOfLinks.selectedIndex = 0; //--> </script> </body> </html> </pre> A) статический <u>B) динамический</u> C) смешанный D) иерархический	
57.	Выберите правильный вариант ответа. Представлен код HTML со скриптом JS организации части формы. Определите элементов для внесения данных и количество кнопок сформированных данным кодом. <pre> <html> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <script><!-- function Complete1() { if(document.Sel3.Pwd.value==document.Sel3.Pwd1.value) alert("Вас зарегистрировали\nID="+document.Sel3.Id.value+"\nPassword="+document.Sel3.Pwd.value); else </pre>	A

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> alert("Ошибка при вводе пароля\nПопробуйте еще раз"); } //--> </script> </head> <body> <H1>Регистрация</H1> <FORM NAME="Sel3"> <TABLE> <TR><TD>Идентификатор:</TD> <TD><INPUT NAME="Id" SIZE=20 onBlur="this.value=this.value.toUpp erCase()" "></TD></TR> <TR><TD>Пароль:</TD> <TD><INPUT TYPE="password" NAME="Pwd" SIZE=20 onFocus="this.select();" "></TD></TR> <TR><TD>Проверка пароля:</TD> <TD><INPUT TYPE="password" NAME="Pwd1" SIZE=20 onFocus="this.select();" "></TD></TR> </TABLE> <INPUT TYPE="button" VALUE="Готово" onClick="Complete1();" "> <INPUT TYPE="reset" VALUE="Сброс"> </FORM> </body> </html> </pre> <u>А) Количество окон – 3, количество кнопок – 2.</u> В) Количество окон – 3, количество кнопок – 3. С) Количество окон – 2, количество кнопок – 2. В) Количество окон – 2, количество кнопок – 3	
58.	Выберите правильный вариант ответа. Определите действия представленного кода. <pre> <SCRIPT LANGUAGE="JavaScript"> <!-- document.frm.name.focus(); //--> </SCRIPT> </pre> А) и вызывает метод focus() для поля	В

Номер задания	Содержание вопроса	Правильный ответ на задание
	document формы frm. <u>B) вызывается метод focus() для поля name формы frm</u> C) и вызывает метод focus() для поля document D) вызывается метод focus() для поля frm	
59.	Выберите правильный вариант ответа. Рассмотрите предложенный код и определите вызов функции. function someName() { console.log("Вы вызвали функцию someName!"); } someName(); <u>A) someName();</u> B) console.log C) function D) someName!	A
60.	Выберите правильный вариант ответа. Представлен код WEB страницы. Определите значение переменной отображенной на WEB странице в браузере. <!DOCTYPE html> <html> <head> <title>METANIT.COM</title> <meta charset="utf-8" /> </head> <body> <?php \$num = 10; echo \$num; \$num = 22; echo \$num; ?> </body> </html> <u>A) 10 22</u> B) 10 C) 22 D) 22...10	A
61.	Выберите правильный вариант ответа. Представлен код PHP. Определите значение переменной отображенной на WEB странице в браузере. <?php	B

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> \$firstquarter = array(1 => 'Январь', 'Февраль', 'Март'); print_r(\$firstquarter); ?> A) Array (=> 'Январь' => 'Февраль' => 'Март') B) Array ([1] => 'Январь' [2] => 'Февраль' [3] => 'Март') C) Array ('Январь' 'Февраль' 'Март') </pre>	
62.	Выберите правильный вариант ответа. Представлен код WEB страницы. Определите значение отображенное на WEB странице в браузере. <pre> <!DOCTYPE html> <html> <body> <?php \$fruits = array("яблоки", "груши", "сливы"); echo "Я люблю " . \$fruits[0] . ", " . \$fruits[1] . " и " . \$fruits[2] . " ."; ?> </body> </html> A) Я люблю яблоки, груши и сливы. B) Я люблю яблоки, груши C) Я люблю яблоки и сливы. D) Яблоки, груши и сливы. </pre>	А
63.	Выберите правильный вариант ответа. Место расположения PHP кода в структуре WEB страницы. <pre> < html > < head > </pre>	А

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>< title >< /title > < /head > < body > < /body > < /html > A) < body > ...< /body > B) < title >...< /title > C) < head >< /head > D) < html >...< /html ></pre>	
64.	Выберите правильный вариант ответа. Представлен код PHP. Определите значение отображенное на WEB странице в браузере. <pre>\$array = array(1 => 'Значение 1', 2 => 'Значение 2', 3 => 'Значение 3', 4 => 'Значение 4',); foreach (\$array as \$key => \$val) { echo \$key . '
'; } foreach (\$array as \$key => \$val) { echo \$val . '
'; }</pre> A) 1 1 2 2 3 3 4 4 B) 1 Значение 1 2 Значение 2 3 Значение 3 4 Значение 4 5 C)	С

Номер задания	Содержание вопроса	Правильный ответ на задание
	1 1 2 2 3 3 4 4 5 6 Значение 1 7 Значение 2 8 Значение 3 9 Значение 4	
65.	Выберите правильный вариант ответа. Представлен код PHP. Определите отображенное на WEB странице в браузере. <pre>\$array = array('Значение 1', 'Значение 2', 'Значение 3', 'Значение 4',); for (\$n = 0; \$n < count(\$array); \$n++) { echo \$array[\$n] . '
'; }</pre> A) 1 1 2 2 3 3 4 4 <u>B)</u> 1 Значение 1 2 Значение 2 3 Значение 3 4 Значение 4 5 C) 1 1 2 2 3 3 4 4 5 6 Значение 1 7 Значение 2 8 Значение 3 9 Значение 4	В
66.	Дайте краткое описание базового синтаксиса конструкции if...else JavaScript.	Краткое описание базового синтаксиса конструкции if...else

Номер задания	Содержание вопроса	Правильный ответ на задание
		JavaScript: 1. Ключевое слово <code>if</code> расположено перед круглыми скобками. 2. Условие для проверки (<code>condition</code>), расположено внутри круглых скобок (например "это значение больше другого значения?", или "это значение существует?"). Это условие использует операторы сравнения (comparison operators), которые мы изучим позже, и возвратит нам <code>true</code> или <code>false</code>. 3. Внутри скобок <code>{ }</code> расположен код, который будет выполняться только в том случае, если условие (<code>condition</code>) верно (<code>true</code>). 4. Ключевое слово <code>else</code> (иначе). 5. Ещё скобки <code>{ }</code>, код внутри которых выполнится, только если условие не верно (не <code>true</code>).
67.	Место в данном коде программы, где следует определить переменную <code>century</code>. <pre>const birthYear = 1991; if (birthYear <= 2000) { century = 20; } else { century = 21; } console.log(century);</pre>	Переменную <code>century</code> следует определить заранее - до конструкции <code>'if', 'else'</code> .
68.	Причины использования оператора <code>switch</code> в программах JavaScript.	Оператор <code>switch</code> используется, когда из множества возможных вариантов нужно выбрать какой-то один.
69.	Представлен код JavaScript. Перечислите параметры функции <pre>function sayWelcome (userFirstName, userLastName) { console.log(`Добро пожаловать, \${userLastName} \${userFirstName}`); } // 'Иван' и 'Иванов' – аргументы sayWelcome('Иван', 'Иванов'); // Добро пожаловать, Иванов Иван // 'Петр' и 'Петров' – аргументы sayWelcome('Петр', 'Петров'); // Добро пожаловать, Петров Петр</pre>	Параметры – это по сути переменные, которые описываются в круглых скобках на этапе объявления функции. В данном примере параметры <code>userFirstName</code>, <code>userLastName</code>
70.	Приведите примеры стандартных ошибок JavaScript.	Стандартные типы ошибок 1. Ошибка синтаксиса 2. Ошибка при доступе к несуществующей переменной

Номер задания	Содержание вопроса	Правильный ответ на задание
		3. Ошибка типа 4. Ошибка, когда значение не входит в набор или диапазон допустимых значений 5. Ошибка, выдаваемая функциями обработки URI
71.	Какие решения вы можете предложить для повышения безопасности кода JavaScript?	Для повышения безопасности написания кода JavaScript необходимо установить и использовать: 1) Утилиты для тестирования кода на JS 2) Утилиты для статического анализа 3) Утилиты для динамического анализа
72.	Кратко опишите реакцию системы при обнаружении ошибки оператора JavaScript.	Когда оператор JavaScript приводит к ошибке, говорят, что он генерирует (выбрасывает) исключение. JavaScript создаёт и выбрасывает объект Error, описывающий ошибку.
73.	Представлен код JavaScript. Опишите действия системы при отсутствии строки <code>i++</code>; <pre> <!DOCTYPE html> <script> "use strict"; let i = 0; while (i < 3) { // выводит 0, затем 1, затем 2 alert(i); i++; } </script> </pre>	Если бы строка <code>i++</code> отсутствовала в примере выше, то цикл бы повторялся (в теории) вечно. На практике, конечно, браузер не позволит такому случиться, он предоставит пользователю возможность остановить «подвисший» скрипт, а JavaScript на стороне сервера придётся «убить» процесс.
74.	Приведите структуру исключения JavaScript.	<pre> try { // код ... } catch (err) { // обработка ошибки } </pre>
75.	Представлен код JavaScript. Какому параметру функции присвоено дефолтное значение? <pre> function setBgColor(selector, color = 'green') { const el = document.querySelector(selector); el.style.backgroundColor = color; } </pre>	Функции <code>setBgColor</code> присвоено дефолтное значение параметру <code>color</code> равно <code>'green'</code>.

Номер задания	Содержание вопроса	Правильный ответ на задание
	<code>setBgColor('body');</code>	
76.	Представлен код JavaScript. Кратко опишите работу кода. <pre>// division calculation function divide(v1, v2, dp) { try { return (v1 / v2).toFixed(dp); } catch(e) { console.log(error name : \${ e.name } error message: \${ e.message }); return 'ERROR'; } }</pre>	Функция выполняет код в блоке try {}, но при возникновении исключения выполняется блок catch {} и получает выброшенный объект ошибки. result показывает ERROR. Консоль показывает имя ошибки и сообщение, но это выводится оператором console.log и не завершает работу программы.
77.	Дан код файла jscode.js, проверяющий пароль на доступ к сайту. Почему при сравнении с паролем использовался оператор ===? <pre>var userName=prompt('Введите Ваш логин'); if (userName===null) alert('Вход отменен'); else if (userName==='Админ') { var userPas=prompt('Введите Ваш пароль'); if (userPas===null) alert('Вход отменен'); else if (userPas==='Чёрный Властелин') alert('Добро пожаловать!'); else alert('Пароль неверен'); } else alert('Я вас не знаю');</pre>	Оператор === сравнивает на идентичность. Плюс оператора === состоит в том, что он не приводит два значения к одному типу. Именно из-за этого он обычно и используют в данного вида программах.
78.	При настройке сайта были сформированы несколько файлов. Кратко опишите их работу. Файл «home.html»: <pre><html> <head> <script language="JavaScript"> function nextPage() { self.location = "next.html?name=" + escape(document.theFo rm.userName.value); // Use escape() any time there might be spaces or } // non-alpa characters </script></pre>	"домашняя" страница запрашивает имя пользователя, затем использует это имя, чтобы обратиться к пользователю на следующих страницах. Можно взять имя, используя форму, затем использовать JavaScript для передачи имени пользователя в следующую страницу, используя URL. Последующая страница могла бы затем анализировать имя пользователя из URL, используя информацию в document.search.

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> </head> <body> <form onSubmit = "nextPage();return false;"> Enter your name: <input type=text name=userName> <input type=submit> </form> </body> </html> </pre> Файл «next.html»: <pre> userName = document.search; userName = userName.substring(userName.indexOf(" ")+1); document.write("Greetings, " + userName + "<P>"); </pre>	
79.	Представлен код страницы с формой. Дайте краткое описание кода. <pre> <form name="myform"> <table cellpadding="10"> <tr> <td> Пользователь: <input type="text" name="username" size="15"> </td> <td> Пароль: <input type="password" name="password" size="15"> </td> </tr> <tr> <td> <input type="reset" name="btnClear" value=" Очистить "> </td> <td> <input type="button" name="btnPsw" value=" Показать пароль " onClick="ShowPassword()"> </td> </tr> </table> </form> <script language="JavaScript"> <!-- document.write(document.forms[0].eleme </pre>	Форма с четырьмя элементами: поле ввода текста (text), поле ввода пароля (password), кнопка очистки полей ввода (reset) и просто кнопка (button). Для обычной кнопки имеется обработчик ShowPassword() события onClick, который выводит окно с паролем (знаки, введенные в поле password, невидимы пользователю). Для кнопки reset обработчик события создавать необязательно, потому что нажатие на эту кнопку автоматически очищает все поля формы. В начале сценария использовался способ доступа к элементам формы через массивы форм и элементов: document.write(document.forms[0].elements[2].value); Этот код выведет слово “Очистить”, поскольку третьим элементом формы является кнопка с надписью “Очистить”.

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>nts[2].value); function ShowPassword() { alert("Пароль: " + myform.password.value); } // --> </script></pre>	
80.	Представлен код страницы HTML с скриптом на JavaScript. Кратко опишите действия скрипта. <pre><HTML> <HEAD> <TITLE>Анкета</TITLE> <SCRIPT LANGUAGE="JavaScript"> <!-- function Complete() { var szStr=""; szStr="Имя: " + frm.name.value + "\nФамилия: " + frm.family.value + "\nE-mail: " + frm.email.value; alert(szStr); } // --> </SCRIPT> </HEAD> <BODY> <H1>Анкета</H1> <FORM NAME="frm"> <TABLE> <TR> <TD>Имя:</TD> <TD><INPUT NAME="name" TYPE="text" SIZE="15"></TD> <TD rowspan="4" valign="top">Пожалуйста, заполните анкету</TD> </TR> <TR> <TD>Фамилия:</TD> <TD><INPUT NAME="family" TYPE="text" SIZE="15"></TD> </TR> <TR> <TD>E-mail:</TD> <TD><INPUT NAME="email" TYPE="text" SIZE="20"></TD></pre>	Скрипт JavaScript формирует окно с данными из формы (имя, фамилия, E-mail)

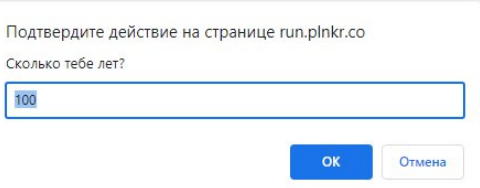
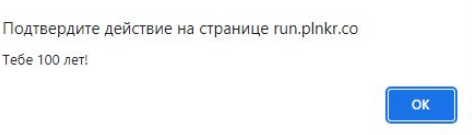
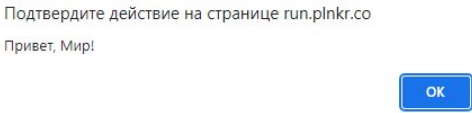
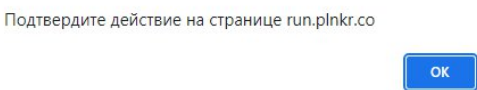
Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> </TR> <TR> <TD>&nbsp;</TD> <TD><INPUT VALUE="Показать" TYPE="button" onClick="Complete();"></TD> </TR> </TABLE> </FORM> </BODY> </HTML> </pre>	
81.	Опишите действия методов focus() и blur() объекта elem в данном коде. <pre> <!doctype html> <body> <style> .error { background: red; } </style> Ваш email: <input type="email" id="input"> <input type="text" style="width:280px" placeholder="введите неверный email и кликните сюда"> <script> input.onblur = function() { if (!this.value.includes('@')) { this.classList.add("error"); input.focus(); } else { this.classList.remove("error"); } }; </script> </body> </pre>	Методы elem.focus() и elem.blur() устанавливают/ снимают фокус.
82.	Представлен код страницы HTML с скриптом на JavaScript. Кратко опишите действия скрипта. Файл HTML <pre> <div class="password"> <input type="password" id="password-input" placeholder="Введите пароль" name="password"> </div> </pre>	Показывает пароль по иконке.

Номер задания	Содержание вопроса	Правильный ответ на задание
	Файл JavaScript. <pre>function show_hide_password(target){ var input = document.getElementById('password-input'); if (input.getAttribute('type') == 'password') { target.classList.add('view'); input.setAttribute('type', 'text'); } else { target.classList.remove('view'); input.setAttribute('type', 'password'); } return false; }</pre>	
83.	Что такое логгер?	Логгер — это специальный модуль, библиотека или отдельная программа, которая реагирует на события в программе и записывает всё, что там происходит. Эти записи называются логами, и чаще всего это обычный текстовый файл.
84.	Выберите правильный вариант ответа. Представлен код WEB страницы. Определите значение переменной отображенной на WEB странице в браузере если она не была инициализирована. <pre><?php \$num; echo \$num; ?></pre>	При попытке вывести значение не инициализированной переменной мы получим диагностическое сообщение о том, что переменная не определена
85.	Представлен код WEB страницы. Определите действия команды <code>echo</code> при отображении WEB страницы в браузере. <pre><?php \$num_1 = 11; \$num_2 = 35; echo "num_1 = \$num_1 num_2=\$num_2"; ?></pre>	Команда <code>echo</code> применяется для вывода значений: <code>num_1 = 11</code> <code>num_2=35</code>
86.	Приведите не менее трех типов переменных PHP.	Три примера можно выбрать из данного списка типов переменных PHP: bool (логический тип) int (целые числа) float (дробные числа) string (строки) array (массивы) object (объекты) callable (функции) mixed (любой тип) resource (ресурсы)

Номер задания	Содержание вопроса	Правильный ответ на задание
		null (отсутствие значения)
87.	Представлен код PHP. Результат выполнения кода при a=-4. <pre><?php \$a = -4; if(\$a > 0){ echo "Переменная a больше нуля"; } else{ echo "Переменная a меньше нуля"; } echo "
конец выполнения программы"; ?></pre>	В браузере будет отображен текст: Переменная a меньше нуля
88.	Представлен код PHP. Какие изменения в отображения об ошибках вносит данный код. <pre>< ?php Error_Reporting(E_ALL); ? ></pre>	Данный PHP код включает вывод всех сообщений об ошибках и предупреждениях.
89.	Представлен код PHP. Какие изменения в отображения об ошибках вносит данный код. <pre>< ?php Error_Reporting(E_ALL & ~E_NOTICE); ? ></pre>	Данный PHP код отключает предупреждающие сообщения.
90.	Представлен код PHP. Какие изменения в отображения об ошибках вносит данный код. <pre>< ?php Error_Reporting(E_ALL - E_NOTICE); ? ></pre>	Данный PHP код отключает предупреждающие сообщения.
91.	Представлен код PHP. Какие изменения в отображения об ошибках вносит данный код. <pre>< ?php Error_Reporting(E_ALL - (E_NOTICE+E_WARNING)); ? ></pre>	Данный PHP код отключает предупреждающие сообщения.
92.	Представлен код PHP. Кратко опишите действия кода. <pre><?php \$x = 0; do { \$x = readline("Введите x>0:"); if (\$x <= 0) { echo("Неверный ввод\n"); } }</pre>	Данный PHP код защита от неверного ввода данных пользователем.

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>} while (\$x <= 0); echo("x = " . \$x);</pre>	
93.	Представлен код WEB страницы. Определите стиль использования дескриптора. <pre><html><head> <title>Школьник – результат заказа</title> </head><body> <h1>ОАО «Школьник»</h1> <h2>Результат заказа</h2> <SCRIPT LANGUAGE='php'> echo "<p>Заказ обработан."; </SCRIPT> </body> </html></pre>	Данный дескриптор выполнен в SCRIPT-стиле JavaScript.
94.	Представлен код WEB страницы. Определите стиль использования дескриптора. <pre><html><head> <title>Школьник – результат заказа</title> </head><body> <h1>ОАО «Школьник»</h1> <h2>Результат заказа</h2> <?php echo "<p>Заказ обработан.";?> </body></html></pre>	Данный PHP-дескриптор выполнен в XML-стиле.
95.	Представлен код WEB страницы. Определите стиль использования PHP-дескриптора. <pre><html><head> <title>Школьник – результат заказа</title> </head><body> <h1>ОАО «Школьник»</h1> <h2>Результат заказа</h2> <? echo "<p>Заказ обработан..."; </body></html></pre>	Данный PHP-дескриптор выполнен в сокращенном стиле.
96.	Представлен код WEB страницы. Определите стиль использования PHP-дескриптора. <pre><html><head> <title>Школьник – результат заказа</title> </head><body> <h1>ОАО «Школьник»</h1> <h2>Результат заказа</h2> <% echo "<p>Заказ обработан."; %></pre>	Данный PHP-дескриптор выполнен в ASP-стиле, который используется в случае если на Active Server Pages (ASP) включен параметр настройки конфигурации asp_tags.

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> </body> </html> </pre>	
97.	Опишите аргументы функции PHP <code>date()</code> .	Аргумент, передаваемый в функцию <code>date()</code> , должен быть строкой формата, задающей требуемый стиль вывода. Каждая буква в строке представляет часть строки даты и времени. Н — представляет часы в 12-часовом формате, i — минуты с ведущим нулем, когда это требуется, j — день месяца без ведущего нуля, S — представляет обычный суффикс (в данном случае "th"), а F - - год, представленный четырьмя цифрами.
98.	Представлена часть кода PHP. Опишите для чего использовалась функция <code>file_exists()</code>. <pre> if (file_exists("\$DOCUMENT_ROOT/./orders/orders.txt")) echo "There are orders waiting to be processed."; else echo "There are currently no orders."; </pre>	<code>file_exists()</code> - проверка существования файла.
99.	Представлена часть кода PHP. Опишите для чего использовалась функция <code>filesize()</code>. <pre> echo filesize("\$DOCUMENT_ROOT/./orders/orders.txt"); </pre>	<code>filesize()</code> – используется для выяснения размера файла, возвращает размер файла, выраженный в байтах.
100.	Представлена часть кода PHP. Кратко опишите действия кода. <pre> \$myArray = ['name' => 'Sergei', 'age' => 48, 'hometown' => ['name' => 'Moscow', 'state' => 'Russia']]; </pre>	Инициализация ассоциативного массива при его объявлении.
101.	Выберите правильный вариант ответа. По предложенному коду JavaScript определите первое окно отображения. <pre> <!DOCTYPE html> <script> "use strict"; let age = prompt('Сколько тебе лет?', 100); alert(`Тебе \${age} лет!`); // Тебе 100 лет! </script> </pre>	А

Номер задания	Содержание вопроса	Правильный ответ на задание
	A)  B)  C)  D) 	
102.	Выберите правильный вариант ответа. Представлен код JavaScript. Результат работы программы при внесении возраста 18? <pre> <!DOCTYPE html> <script> "use strict"; let age = prompt('Возраст?', 18); let message = (age < 3) ? 'Здравствуй, малыш!' : (age < 18) ? 'Привет!' : (age < 100) ? 'Здравствуйте!' : 'Какой необычный возраст!'; alert(message); </script> </pre> A) Здравствуй, малыш! B) Привет! C) Здравствуйте! D) Какой необычный возраст!	С
103.	Выберите правильный вариант ответа. Представлен код JavaScript. По данному коду определите шаг цикла. <pre> <!DOCTYPE html> <script> "use strict"; let i = 0; for (i = 0; i < 3; i++) { // используем существующую переменную </pre>	А

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> alert(i); // 0, 1, 2 } alert(i); // 3, переменная доступна, т.к. была о бъявлена снаружи цикла </script> </pre> <u>А) значение i при каждой итерации увеличивается на 1</u> В) значение i при каждой итерации увеличивается на 2 С) значение i при каждой итерации увеличивается на 3 D) значение i при каждой итерации увеличивается на 4	
104.	Выберите правильный вариант ответа. Представлен код JavaScript. Проанализируйте алгоритм кода и определите тип используемой переменной b. <pre> let a = 7; function sum(a) { let b = 8; console.log(a + b); } sum(a); console.log(b); </pre> <u>А) локальная переменная</u> В) внешняя переменная С) аргументируемая переменная D) переменная модуля	А
105.	Выберите правильный вариант ответа. Представлен код JavaScript. Проанализируйте алгоритм кода и определите количество возможных вариантов ответа. <pre> <!DOCTYPE html> <script> "use strict"; function checkAge(age) { if (age >= 18) { return true; } else { return confirm('А родители разрешили?'); } } let age = prompt('Сколько вам лет?', 18); if (checkAge(age)) { alert('Доступ получен'); } else { alert('Доступ закрыт'); } </pre>	В

Номер задания	Содержание вопроса	Правильный ответ на задание
	<code></script></code> A) 2 <u>B) 3</u> C) 4 D) 6	
106.	Выберите правильный вариант ответа. Поставлена задача JavaScript. Нужно сохранить в переменную одно, либо другое значение, в зависимости от условия, то можно это сделать двумя путями. <u>A)</u> <code>let value = 1 // значение по умолчанию</code> <code>if (day === 'Tuesday') {</code> <code> value = 50</code> <code>}</code> B) <code>let value = 0 // значение по умолчанию</code> <code>if (externalValue) {</code> <code> value = externalValue // установить значение, если в externalValue что-либо хранится</code> <code>}</code> C) <code>const value = externalValue ?? 42</code> D) <code>let value = 0</code> <code>if (day === 'Tuesday') {</code> <code> value = 50</code> <code>}</code>	A
107.	Выберите правильный вариант ответа. Поставлена задача JavaScript. Установить значение по умолчанию, если в переменной нет значения A) <code>let value = 1 // значение по умолчанию</code> <code>if (day === 'Tuesday') {</code> <code> value = 50</code> <code>}</code> <u>B)</u> <code>let value = 0 // значение по умолчанию</code> <code>if (externalValue) {</code> <code> value = externalValue // установить значение, если в externalValue что-либо хранится</code> <code>}</code> C) <code>const value = externalValue ?? 42</code> D) <code>let value = 0</code>	B

Номер задания	Содержание вопроса	Правильный ответ на задание																																																																																																																																																	
	<pre>if (day === 'Tuesday') { value = 50 }</pre>																																																																																																																																																		
108.	Выберите правильный вариант ответа. В данной конструкции программного кода JavaScript значение переменной shoppingDone. <pre>var shoppingDone = false; if (shoppingDone === true) { var childsAllowance = 10; } else { var childsAllowance = 5; }</pre><u>A) shoppingDone всегда равный false</u> B) shoppingDone меняется на false C) shoppingDone не меняется на false D) shoppingDone не меняется на true	A																																																																																																																																																	
109.	Выберите правильный вариант ответа. Проанализируйте алгоритм кода JavaScript и выберите правильный вариант отображения данных программы. <pre>var output = " for (var i = 1; i <= 9; i++) { for (var j = 1; j <= 9; j++) { output += ' ' + i * j; if (i * j < 10) { output += ' '; } } } console.log(output); output = " }</pre>A) 1 2 3 4 5 6 7 8 9 2 4 6 8 10 12 14 16 18 3 <u>B)</u> <table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>2</td><td>4</td><td>6</td><td>8</td><td>10</td><td>12</td><td>14</td><td>16</td><td>18</td></tr><tr><td>3</td><td>6</td><td>9</td><td>12</td><td>15</td><td>18</td><td>21</td><td>24</td><td>27</td></tr><tr><td>4</td><td>8</td><td>12</td><td>16</td><td>20</td><td>24</td><td>28</td><td>32</td><td>36</td></tr><tr><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td><td>30</td><td>35</td><td>40</td><td>45</td></tr><tr><td>6</td><td>12</td><td>18</td><td>24</td><td>30</td><td>36</td><td>42</td><td>48</td><td>54</td></tr><tr><td>7</td><td>14</td><td>21</td><td>28</td><td>35</td><td>42</td><td>49</td><td>56</td><td>63</td></tr><tr><td>8</td><td>16</td><td>24</td><td>32</td><td>40</td><td>48</td><td>56</td><td>64</td><td>72</td></tr><tr><td>9</td><td>18</td><td>27</td><td>36</td><td>45</td><td>54</td><td>63</td><td>72</td><td>81</td></tr></table>C) <table><tr><td>4</td><td>6</td><td>8</td><td>10</td><td>12</td><td>14</td><td>16</td><td>18</td></tr><tr><td>6</td><td>9</td><td>12</td><td>15</td><td>18</td><td>21</td><td>24</td><td>27</td></tr><tr><td>8</td><td>12</td><td>16</td><td>20</td><td>24</td><td>28</td><td>32</td><td>36</td></tr><tr><td>10</td><td>15</td><td>20</td><td>25</td><td>30</td><td>35</td><td>40</td><td>45</td></tr><tr><td>12</td><td>18</td><td>24</td><td>30</td><td>36</td><td>42</td><td>48</td><td>54</td></tr><tr><td>14</td><td>21</td><td>28</td><td>35</td><td>42</td><td>49</td><td>56</td><td>63</td></tr><tr><td>16</td><td>24</td><td>32</td><td>40</td><td>48</td><td>56</td><td>64</td><td>72</td></tr><tr><td>18</td><td>27</td><td>36</td><td>45</td><td>54</td><td>63</td><td>72</td><td>81</td></tr></table>D)	1	2	3	4	5	6	7	8	9	2	4	6	8	10	12	14	16	18	3	6	9	12	15	18	21	24	27	4	8	12	16	20	24	28	32	36	5	10	15	20	25	30	35	40	45	6	12	18	24	30	36	42	48	54	7	14	21	28	35	42	49	56	63	8	16	24	32	40	48	56	64	72	9	18	27	36	45	54	63	72	81	4	6	8	10	12	14	16	18	6	9	12	15	18	21	24	27	8	12	16	20	24	28	32	36	10	15	20	25	30	35	40	45	12	18	24	30	36	42	48	54	14	21	28	35	42	49	56	63	16	24	32	40	48	56	64	72	18	27	36	45	54	63	72	81	B
1	2	3	4	5	6	7	8	9																																																																																																																																											
2	4	6	8	10	12	14	16	18																																																																																																																																											
3	6	9	12	15	18	21	24	27																																																																																																																																											
4	8	12	16	20	24	28	32	36																																																																																																																																											
5	10	15	20	25	30	35	40	45																																																																																																																																											
6	12	18	24	30	36	42	48	54																																																																																																																																											
7	14	21	28	35	42	49	56	63																																																																																																																																											
8	16	24	32	40	48	56	64	72																																																																																																																																											
9	18	27	36	45	54	63	72	81																																																																																																																																											
4	6	8	10	12	14	16	18																																																																																																																																												
6	9	12	15	18	21	24	27																																																																																																																																												
8	12	16	20	24	28	32	36																																																																																																																																												
10	15	20	25	30	35	40	45																																																																																																																																												
12	18	24	30	36	42	48	54																																																																																																																																												
14	21	28	35	42	49	56	63																																																																																																																																												
16	24	32	40	48	56	64	72																																																																																																																																												
18	27	36	45	54	63	72	81																																																																																																																																												

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> 4 6 8 10 6 9 12 15 8 12 16 20 10 15 20 25 12 18 24 30 14 21 28 35 16 24 32 40 18 27 36 45 </pre>	
110.	Выберите правильный вариант ответа. Рассмотрите алгоритм предложенного кода и определите количество функций в коде. <pre> const factorial = (n) => { if (n === 0) { return 1; } const iter = (counter, acc) => { if (counter === 1) { return acc; } return iter(counter - 1, counter * acc); }; return iter(n, 1); }; </pre> A) 0 B) 1 C) 2 D) 4	С
111.	Выберите правильный вариант ответа. Рассмотрите предложенный код и определите характеристику вычисления. <pre> const factorial = (n) => { if (n === 0) { return 1; } const iter = (counter, acc) => { if (counter === 1) { return acc; } return iter(counter - 1, counter * acc); }; return iter(n, 1); }; </pre> A) Итеративный алгоритм вычисления B) Рекурсивный алгоритм вычисления C) Пассивный алгоритм вычисления D) Иерархический алгоритм вычисления	В
112.	Выберите правильный вариант ответа. Представлен код РНР. Определите отображение данных работы кода.	А

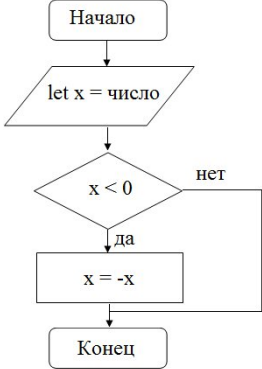
Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>\$i = 0; for (; \$i < 10;) { echo \$i; \$i += 1; }</pre> <u>A) 0123456789</u> B) 02468 C) 0246810 D) 02468	
113.	Выберите правильный вариант ответа. Представлен код PHP. Определите количество выполненных итераций <pre>for (\$i=1, \$sum=-1; false, true, false; \$i = \$i + 3, \$sum = \$sum+3) { print "\$i "; print "\$sum< br >"; }</pre> <u>A) 0</u> B) 1 C) 2 D) 4	A
114.	Выберите правильный вариант ответа. Представлен код PHP. Определите индекс последнего массива <pre>\$parts[2] = "K155ЛАЗ"; \$parts[7] = "K155ЛА8"; \$parts[20] = "K155ЛН1"; \$parts[] = "КРЕН5В";</pre> A) 20 <u>B) 21</u> C) 22 D) 23	B
115.	Выберите правильный вариант ответа. Для чего используется цикл foreach. A) цикл foreach используется для работы со списками B) цикл foreach используется для работы с переменными <u>C) цикл foreach используется для работы с массивом</u> D) цикл foreach используется для работы с объектами	C
116.	По предложенному коду JavaScript опишите алгоритм изменения переменной \$. <pre><!DOCTYPE html> <script> "use strict";</pre>	Переменная \$ в ходе программы изменялась: Иван 3

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>let name = "Иван"; alert(`Привет, \${name}!`); alert(`результат: \${1 + 2}`); </script></pre>	
117.	По предложенному коду JavaScript опишите алгоритм изменения переменной \$. <pre><!DOCTYPE html> <script> "use strict"; let name = "Иван"; alert(`Привет, \${name}!`); alert(`результат: \${1 + 2}`); </script></pre>	Переменная \$ в ходе программы изменялась: 1 Name Ilya
118.	По предложенному коду JavaScript опишите алгоритм изменения переменной Number. <pre><!DOCTYPE html> <script> "use strict"; alert(Number(" 123 ")); alert(Number(false)); alert(Number(true)); </script></pre>	Переменная Number в ходе программы изменялась: 123 0 1
119.	По предложенному коду JavaScript опишите алгоритм отображения данных в модальных окнах. <pre><!DOCTYPE html> <script> "use strict"; alert(Boolean(1)); alert(Boolean(0)); alert(Boolean("Привет!")); alert(Boolean("")); // </script></pre>	В модальных окнах отображаются следующие данные: true false true false
120.	По предложенному коду JavaScript опишите какие действия применены к переменной x. <pre><!DOCTYPE html> <script> "use strict"; let x = 1; x = -x; alert(x); </script></pre>	К значению переменной x применен унарный минус. Значение x стало -1.
121.	Представлен код JavaScript. Перечислите значения отображаемые в модальных	В модальных окнах будут отображаться значения остатков от

Номер задания	Содержание вопроса	Правильный ответ на задание
	окнах? <pre><!DOCTYPE html> <script> "use strict"; alert(5 % 2); alert(8 % 3); alert(8 % 4); </script></pre> Ответ:	деления: 1 2 0
122.	Представлен код JavaScript. Перечислите значения отображаемые в модальных окнах? <pre><!DOCTYPE html> <script> "use strict"; alert(2 ** 2); alert(2 ** 3); alert(2 ** 4); </script></pre>	В модальных окнах будут отображаться значения возведения в степень 2: 4 8 16
123.	Представлен код JavaScript. Перечислите значения отображаемые в модальных окнах? <pre><!DOCTYPE html> <script> "use strict"; alert(4 ** (1/2)); alert(8 ** (1/3)); </script></pre>	В модальных окнах будут отображаться значения взятию корня и куба 2: 2 2
124.	Представлен код JavaScript. Перечислите значения отображаемые в модальных окнах? <pre><!DOCTYPE html> <script> "use strict"; alert(2 > 1); alert(2 == 1); alert(2 != 1); </script></pre>	В модальных окнах будут отображаться результаты работы операторов сравнения: true false true
125.	Представлен код JavaScript. По алгоритму данного кода определите действия после знака «?». <pre><!DOCTYPE html> <script> "use strict"; let company = prompt('Какая компания создала JavaScript?', ''); (company == 'Netscape') ? alert('Верно!') : alert('Неправильно.');</pre>	В зависимости от условия company == 'Netscape', будет выполнена либо первая, либо вторая часть после ?.
126.	Представлен код JavaScript. Кратко	Директива break полностью

Номер задания	Содержание вопроса	Правильный ответ на задание
	опишите действия директивы <code>break</code> в данной программе. <pre><!DOCTYPE html> <script> "use strict"; let sum = 0; while (true) { let value = +prompt("Введите число", ""); if (!value) break; sum += value; } alert('Сумма: ' + sum); </script></pre>	прекращает выполнение цикла и передаёт управление на строку за его телом, то есть на <code>alert</code>.
127.	Представлен код JavaScript. Кратко опишите действия директивы <code>continue</code> в данной программе. <pre><!DOCTYPE html> <script> "use strict"; for (let i = 0; i < 10; i++) { if (i % 2 == 0) continue; alert(i); // 1, затем 3, 5, 7, 9 } </script></pre>	Для чётных значений <code>i</code>, директива <code>continue</code> прекращает выполнение тела цикла и передаёт управление на следующую итерацию <code>for</code> (со следующим числом). <code>alert</code> вызывается только для нечётных значений.
128.	Представлен пример конструкции <pre>const number = prompt("Введите число: "); if (number >= 0) { if (number == 0) { console.log("Это ноль."); } else { console.log("Это положительное число."); } } else { console.log("Это отрицательное число."); }</pre>	Вложенные операторы <code>if...else</code>.
129.	Представлен код JavaScript. Перечислите аргументы объявленной функции при ее запуске. <pre>function sayWelcome (userFirstName, userLastName) { console.log(`Добро пожаловать, \${userLastName} \${userFirstName}`); } sayWelcome('Иван', 'Иванов'); sayWelcome('Петр', 'Петров'); sayWelcome('Дмитрий');</pre>	Для функции <code>sayWelcome</code> в данном коде использовались аргументы: Иван, Иванов Петр, Петров Дмитрий
130.	Представлен код JavaScript. Проанализируйте алгоритм кода и опишите кратко действие данного кода.	Представлен код проверки: принимает ли некая переменная одно из значений 1, 2, 3 или 4.

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>let item = 3; switch(item) { case 1: console.log("item = 1"); case 2: console.log("item = 2"); case 3: console.log("item = 3"); case 4: console.log("item = 4"); default: console.log("item другое значение"); }</pre>	
131.	Представлен код JavaScript. Проанализируйте алгоритм кода и опишите работу строки <code>user.firstname = 'Александр'</code>; <pre>const someUser = { firstname: 'Петр', lastname: 'Петров' } function changeUserName(user) { user.firstname = 'Александр'; } changeUserName(someUser); console.log(someUser.firstname);</pre>	Объявлена переменная <code>someUser</code> и присвоен ей объект, состоящий из двух свойств. Строкой <code>user.firstname = 'Александр'</code> ; меняем значение свойства <code>firstname</code> на новое. В итоге при вызове функции свойство <code>firstname</code> будет <code>'Александр'</code> .
132.	Опишите понятие Колбэк функция (от английского <code>callback function</code>) в JavaScript.	Колбэк функция- обычная функция, которая просто вызывается внутри другой функции. Такие функции ещё называются функциями обратного вызова. Они очень часто применяются в асинхронном коде.
133.	Опишите понятие Рекурсия в JavaScript.	Рекурсия возможность вызова функцией внутри самой себя.
134.	Представлен код страницы HTML с скриптом на JavaScript. Кратко опишите действия скрипта. Файл HTML <pre><div class="password"> <input type="password" id="password-input" placeholder="Введите пароль" name="password"> <label><input type="checkbox" class="password-checkbox"> Показать пароль</label> </div></pre> Файл JavaScript. <pre>\$('body').on('click', '.password-checkbox', function() { if (\$(this).is(':checked')) { \$('#password-input').attr('type', 'text'); } else { \$('#password-input').attr('type',</pre>	Показывает пароль по чекбоксу.

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>'password'); } });</pre>	
135.	Опишите кратко алгоритм, представленный на рисунке.  <pre> graph TD Start([Начало]) --> Assign[/let x = число/] Assign --> Cond{x < 0} Cond -- да --> Negate[x = -x] Cond -- нет --> End([Конец]) Negate --> End </pre>	Вычисление модуля числа, хранящегося в переменной x.
136.	Проанализируйте алгоритм программного кода JavaScript. Кратко опишите действия кода. <pre>let x = 4; if(x >= 2 && x <= 7) console.log("x попадает в [2; 7]"); else console.log("x не попадает в [2; 7]");</pre>	Две проверки: первая –проверяем, что $x \geq 2$ и вторая – проверяем, что $x \leq 7$. Для проверки используется логический оператор И (&&) (логическая конъюнкция). Если оба этих условия выполняются одновременно, то x попадает в диапазон [2; 7].
137.	Проанализируйте алгоритм программного кода JavaScript. Кратко опишите действия кода. <pre>let x = 40; if(x < 2 x > 7) console.log("x не попадает в [2; 7]"); else console.log("x попадает в [2; 7]");</pre>	В составном условии используется связка по ИЛИ (две вертикальные черты – это ИЛИ) и оно будет истинно, если истинно или первое, или второе условие. То есть, в нашем случае, если $x < 2$ или $x > 7$, то делается вывод о невхождении переменной x в указанный диапазон.
138.	Представлен алгоритм конструкции цикла for в JavaScript. По предложенному коду определите инициацию.  <pre> graph TD Init[Инициализация] --> Cond{Условие} Cond -- ложно --> Instr[Инструкции] Instr --> Cond Cond -- истинно --> Update[Обновление счетчика] Update --> Instr </pre> <pre>for (var i = 1; i <= 8; i++) { console.log(i); }</pre>	Инициализация - это выражение, которое выполняется один раз перед выполнением цикла; обычно используется для инициализации счётчика. Инициализация в предложенном коде: <code>var i = 1</code> (объявление переменной i и присвоение ей значения 1).
139.	Представлен алгоритм конструкции цикла for в JavaScript. По предложенному коду определите условие.	Условие - это выражение, истинность которого проверяется перед каждой итерацией; если выражение вычисляется как истина, то выполняется итерация; в

Номер задания	Содержание вопроса	Правильный ответ на задание
	 <pre>for (var i = 1; i <= 8; i++) { console.log(i); }</pre>	противном случае цикл «for» завершает работу. Условие выполнения цикла в данном коде: $i \leq 8$ (пока значение переменной i меньше или равно 8).
140.	Представлен алгоритм конструкции цикла for в JavaScript. По предложенному коду определите инструкцию.  <pre>for (var i = 1; i <= 8; i++) { console.log(i); }</pre>	Инструкция в предложенном коде, которую нужно выполнять: console.log(i) (выведение значения счётчика в консоль).
141.	Представлен алгоритм конструкции цикла while в JavaScript. По предложенному коду определите условие.  <pre>let a = 0; while (a <= 8) { a++; if (a % 2 !== 0) { continue; } console.log(a); }</pre>	Цикл while с условием $a \leq 8$
142.	Представлен алгоритм конструкции цикла while в JavaScript. По предложенному коду опишите кратко инструкции кода.	Инструкции прописаны в фигурных скобках: 1) увеличим значение переменной a на 1 2) если число нечётное (остаток от

Номер задания	Содержание вопроса	Правильный ответ на задание
	 <pre> let a = 0; while (a <= 8) { a++; if (a % 2 !== 0) { continue; } console.log(a); } </pre>	деления на 2 не равен 0), то... 3) пропустим дальнейшее выполнение текущей итерации и перейдем к следующей 4) выведем значение переменной a в консоль
143.	Представлен алгоритм конструкции цикла do...while в JavaScript. По предложенному коду определите инструкции (тело цикла).  <pre> let num, sum = 0; do { num = +prompt ('Введите число', ''); if (num) { sum += num; } } while (num); console.log(sum); </pre>	Инструкции прописаны в фигурных скобках между операторами do и while.
144.	Проанализируйте алгоритм программного кода JavaScript. Что из себя представляет someLabel. <pre> someLabel: while (условие) { if (условие) { break someLabel; } } </pre>	someLabel является меткой. Вызов break someLabel приведёт к переходу в конец цикла, перед которым данная метка указана.
145.	Представлен код JavaScript. Кратко опишите действия данного кода с точки зрения использования метки. <pre> passLoop: for (n = 0; n < 4; n++) { </pre>	При достижении условия (n === 2 && i === 2) команда continue passLoop завершает текущую итерацию и переносит поток

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> for (i = 0; i < 4; i++) { if (n === 2 && i === 2) { continue passLoop; } console.log('n = ' + n + ', i = ' + i); } </pre>	выполнения кода в начало верхнего цикла.
146.	Представлен код на JavaScript. Какая из функций используется как колбэк . <pre> function printToLog(message) { console.log(message); } function sum(num1, num2, callback) { const result = num1 + num2; callback(result); } sum(5, 11, printToLog); </pre>	printToLog – её будем использовать в роли колбэк функции, т.е. вызывать в sum
147.	Поясните понятие «динамическая типизация» применяемая в PHP <pre> <?php \$id = 123; echo "<p>id = \$id</p>"; \$id = "jhveruuyeru"; echo "<p>id = \$id</p>"; ?> </pre>	PHP - язык с динамической типизацией, мы можем присваивать одной и той же переменной значения разных типов в процессе выполнения кода.
148.	Представлен код PHP. Результат выполнения кода при a=0. <pre> \$a = 5; if(\$a>0){ echo "Переменная a больше нуля"; } elseif(\$a < 0){ echo "Переменная a меньше нуля"; } else{ echo "Переменная a равна нулю"; } </pre>	В браузере будет отображен текст: Переменная a равна нулю
149.	Дайте характеристику применения второго кода PHP на приведенной странице WEB. <pre> <!DOCTYPE html> <html> <head> <title>METANIT.COM</title> <meta charset="utf-8" /> </head> <body> <?php \$a = 5; ?> </pre>	Можем переключаться внутри конструкции на код HTML. В данном случае само условие указывается в отдельном блоке php: <?php if (\$a > 0) { ?>. Важно, что при этом этот блок содержит только открывающую фигурную скобку "{". Завершается конструкция if другим блоком php, который содержит закрывающую фигурную скобку: <?php } ?>

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre><?php if (\$a > 0) { ?> <h2>Переменная а больше нуля</h2> <?php } ?> </body> </html></pre>	
150.	Представлен код PHP. Изучите алгоритм кода. Кратко опишите действие кода. <pre>\$user = array("name" => "Sergei Loftoff", "email" => "loftoff@gmail.com", "age" => 23); //Получение доступа к значениям в array echo "Name: " . \$user["name"] . "\n"; echo "Email: " . \$user["email"] . "\n"; echo "Age: " . \$user["age"] . "\n";</pre>	В этом примере мы создали ассоциативный массив \$user, в котором хранится информация о пользователе. Мы можем получить доступ к значениям, хранящимся в нем, используя ключи (например, "name", "email", "age"), указанные в квадратных скобках после имени массива.