



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ИНСТИТУТ ТЕХНОЛОГИЙ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
В Г. ВОЛГОДОНСКЕ РОСТОВСКОЙ ОБЛАСТИ**

(Институт технологий (филиал) ДГТУ в г. Волгодонске)



УТВЕРЖДАЮ

И.о. директора

Н.М. Сидоркина

«22» апреля 2024 г

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
(ОЦЕНОЧНЫЕ СРЕДСТВА)**

**для проведения текущего контроля и промежуточной аттестации
по дисциплине**

«Технологии Web программирования»

для обучающихся по направлению подготовки

09.03.02 Информационные системы и технологии

программа бакалавриата Информационные системы

2024 года набора

Волгодонск
2024

Лист согласования

Оценочные материалы (оценочные средства) по дисциплине Технологии Web
программирования

(наименование)

составлены в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности)

09.03.02 Информационные системы и технологии

(код направления (специальности), наименование)

Рассмотрены и одобрены на заседании кафедры «ТСИИТ» протокол № 9
от «22» 04 2024 г

Разработчики оценочных материалов (оценочных средств)

доцент  Н.В. Кочковая

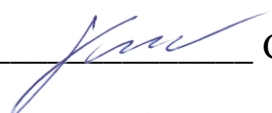
подпись

Заведующий кафедрой


 Н.В. Кочковая

подпись

Согласовано:

директор НПЦ «Микроэлектроника»  С.Л. Бондаренко

подпись

Начальник отдела ПО ООО «Топаз-сервис»  Д.В. Чубукин

подпись

**Лист визирования оценочных материалов (оценочных средств)
на очередной учебный год**

Оценочные материалы (оценочные средства) по дисциплине «Технологии Web программирования» проанализированы и признаны актуальными для использования на 20__ - 20__ учебный год.

Протокол заседания кафедры «ТСиИТ» от «__» _____ 20__ г. № _____
Заведующий кафедрой «ТСиИТ» _____ Н.В. Кочковая
«__» _____ 20__ г.

Оценочные материалы (оценочные средства) по дисциплине «Технологии Web программирования» проанализированы и признаны актуальными для использования на 20__ - 20__ учебный год.

Протокол заседания кафедры «ТСиИТ» от «__» _____ 20__ г. № _____
Заведующий кафедрой «ТСиИТ» _____ Н.В. Кочковая
«__» _____ 20__ г.

Оценочные материалы (оценочные средства) по дисциплине «Технологии Web программирования» проанализированы и признаны актуальными для использования на 20__ - 20__ учебный год.

Протокол заседания кафедры «ТСиИТ» от «__» _____ 20__ г. № _____
Заведующий кафедрой «ТСиИТ» _____ Н.В. Кочковая
«__» _____ 20__ г.

Оценочные материалы (оценочные средства) по дисциплине «Технологии Web программирования» проанализированы и признаны актуальными для использования на 20__ - 20__ учебный год.

Протокол заседания кафедры «ТСиИТ» от «__» _____ 20__ г. № _____
Заведующий кафедрой «ТСиИТ» _____ Н.В. Кочковая
«__» _____ 20__ г.

Содержание

С.

1 Паспорт оценочных материалов (оценочных средств)	
1.1 Перечень компетенций, формируемых дисциплиной (модулем), с указанием этапов их формирования в процессе освоения ОПОП	5
1.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования	9
1.3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, описание шкал оценивания	13
2 Контрольные задания (демоверсии) для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы	14

1 Паспорт оценочных материалов (оценочных средств)

Оценочные материалы (оценочные средства) прилагаются к рабочей программе дисциплины и представляет собой совокупность контрольно-измерительных материалов (типовые задачи (задания), контрольные работы, тесты и др.) и методов их использования, предназначенных для измерения уровня достижения обучающимся установленных результатов обучения.

Оценочные материалы (оценочные средства) используются при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся.

1.1 Перечень компетенций, формируемых дисциплиной, с указанием этапов их формирования в процессе освоения ОПОП

Перечень компетенций, формируемых в процессе изучения дисциплины:

ПК-1: Способность принимать участие во внедрении информационных систем, настраивать, эксплуатировать и сопровождать информационные системы и сервисы.

ПК-5: Способность программировать и разрабатывать прикладное программное обеспечение.

Конечными результатами освоения дисциплины являются сформированные когнитивные дескрипторы «знать», «уметь», «владеть», расписанные по отдельным компетенциям. Формирование дескрипторов происходит в течение всего семестра по этапам в рамках контактной работы, включающей различные виды занятий и самостоятельной работы, с применением различных форм и методов обучения (табл. 1).

Таблица 1 Формирование компетенций в процессе изучения дисциплины

Код и наименование общепрофессиональной компетенции	Код и наименование индикатора достижения общепрофессиональной компетенции	Планируемые результаты обучения (показатели достижения заданного уровня компетенции)	Вид учебных занятий, работы ¹ , формы и методы обучения, способствующие формированию и развитию компетенции ²	Контролируемые разделы и темы дисциплины ³	Оценочные материалы (оценочные средства), используемые для оценки уровня сформированности и компетенции	Критерии оценивания компетенций ⁴
ПК-5: Способность программировать и разрабатывать прикладное программное обеспечение	ПК-5.1: Знает технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ в различных режимах; основы объектно-ориентированного подхода к программированию	особенности моделей	Лекции, практические занятия (устный опрос, выполнение заданий и решение задач), СРС (анализ ситуаций)	1.1 – 1.6	УО, ТЗ, РЗ, ДЗ	посещаемость занятий; подготовка докладов; познавательная активность на занятиях, качество подготовки докладов и презентаций по разделам дисциплины, выполнение практических заданий,
	ПК-5.2: Умеет работать с современными системами программирования, включая объектно-ориентированные; ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования	определять оптимальные методы моделирования в обработке информации в глобальных сетях	Практ. занятия (устный опрос, выполнение заданий и решение задач), СРС (анализ ситуации)		УО, ТЗ, РЗ, ДЗ	

¹ Лекционные занятия, практические занятия, лабораторные занятия, самостоятельная работа

² Необходимо указать активные и интерактивные методы обучения (например, интерактивная лекция, работа в малых группах, методы мозгового штурма, решение творческих задач, работа в группах, проектные методы обучения, ролевые игры, тренинги, анализ ситуаций и имитационных моделей и др.), способствующие развитию у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств

³ Указать номера тем в соответствии с рабочей программой дисциплины

⁴ Необходимо выбрать критерий оценивания компетенции: посещаемость занятий; подготовка к практическим занятиям; подготовка к лабораторным занятиям; ответы на вопросы преподавателя в рамках занятия; подготовка докладов, эссе, рефератов; умение отвечать на вопросы по теме лабораторных работ, познавательная активность на занятиях, качество подготовки рефератов и презентацией по разделам дисциплины, контрольные работы, экзамены, умение делать выводы и др.

	ПК-5.3: Владеет языками процедурного и объектноориентированного программирования, навыками разработки и отладки и тестирования программ не менее, чем на одном из алгоритмических процедурных языков программирования высокого уровня	навыками применения моделей	Практ. занятия (устный опрос, выполнение заданий и решение задач), СРС (анализ ситуации)		УО, ТЗ, РЗ, ДЗ	контрольных работ, умение делать выводы
ПК-1: Способность принимать участие во внедрении информационных систем, настраивать, эксплуатировать и сопровождать информационные системы и сервисы	ПК-1.1: Знает назначение и виды прикладных ИС, принципы работы технических устройств ИКТ, состав функциональных и обеспечивающих подсистем ИС, модели и процессы жизненного цикла ИС, стадии создания прикладных ИС, методы информационного обслуживания, методы анализа прикладной области, информационных потребностей, формирования требований к ИС, методологию и технологию проектирования прикладных ИС	методы организации процесса сбора и обработки информации	Лекц. Практ. занятия (устный опрос, выполнение заданий и решение задач), СРС (анализ ситуации)	1.1 – 1.6 2.1 – 2.6	УО, ТЗ, РЗ, ДЗ	посещаемость занятий; подготовка докладов; познавательная активность на занятиях, качество подготовки докладов и презентаций по разделам дисциплины, выполнение практических заданий, контрольных работ, умение делать выводы
	ПК-1.2: Умеет проводить работы по инсталляции программного обеспечения ИС и загрузке баз данных ; настраивать параметры ИС и тестировать результаты настройки; использовать различные операционные системы, выбирать и оценивать архитектуру ИС и их подсистем	применять методы поиска источников информации	Практ. занятия (устный опрос, выполнение заданий и решение задач), СРС (анализ ситуации)		УО, ТЗ, РЗ, ДЗ	
	ПК-1.3: Владеет навыками работы с технологиями сбора, накопления, обработки, передачи и распространения информации; навыками работы с инструментальными средствами моделирования предметной области, прикладных и информационных процессов; навыками работы в современной программно-технической среде	средствами поиска информации	Практ. занятия (устный опрос, выполнение заданий и решение задач), СРС (анализ ситуации)		УО, ТЗ, РЗ, ДЗ	

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

Оценивание результатов обучения по дисциплине осуществляется в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся.

По дисциплине «*Технологии Web программирования*» предусмотрены следующие виды контроля: текущий контроль (осуществление контроля всех видов аудиторной и внеаудиторной деятельности обучающегося с целью получения первичной информации о ходе усвоения отдельных элементов содержания дисциплины); промежуточная аттестация (оценивается уровень и качество подготовки по дисциплине в целом).

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы обучающихся. Текущий контроль служит для оценки объёма и уровня усвоения обучающимся учебного материала одного или нескольких разделов дисциплины (модуля) в соответствии с её рабочей программой и определяется результатами текущего контроля знаний обучающихся.

Текущий контроль осуществляется два раза в семестр по календарному графику учебного процесса.

Текущий контроль предполагает начисление баллов за выполнение различных видов работ. Результаты текущего контроля подводятся по шкале балльно-рейтинговой системы. Регламент балльно-рейтинговой системы определен Положением о системе «Контроль успеваемости и рейтинг обучающихся».

Текущий контроль является результатом оценки знаний, умений, навыков и приобретенных компетенций обучающихся по всему объёму учебной дисциплины, изученному в семестре, в котором стоит форма контроля в соответствии с учебным планом.

Текущий контроль успеваемости предусматривает оценивание хода освоения дисциплины: теоретических основ и практической части.

При обучении по заочной форме обучения текущий контроль не предусмотрен.

Промежуточная аттестация по дисциплине «*Технологии Web программирования*» проводится в форме экзамена.

В табл. 2 приведено весовое распределение баллов и шкала оценивания по видам контрольных мероприятий.

Таблица 2 Весовое распределение баллов и шкала оценивания по видам контрольных мероприятий

Текущий контроль (50 баллов ⁵)				Промежуточная аттестация (50 баллов)	Итоговое количество баллов по результатам текущего контроля и промежуточной аттестации
Блок 1		Блок 2			
Лекционные занятия (X_1)	Практические занятия (Y_1)	Лекционные занятия (X_2)	Практические занятия (Y_2)	от 0 до 50 баллов	Менее 41 балла – неудовлетворительно; 41-60 баллов – удовлетворительно; 61-80 баллов – хорошо; 81-100 баллов – отлично
5	15	5	25		
Сумма баллов за 1 блок = $X_1 + Y_1=20$		Сумма баллов за 2 блок = $X_2 + Y_2=30$			

⁵ Вид занятий по дисциплине (лекционные, практические, лабораторные) определяется учебным планом. Количество столбцов таблицы корректируется в зависимости от видов занятий, предусмотренных учебным планом.

Распределение баллов по блокам, по каждому виду занятий в рамках дисциплины определяет преподаватель. Распределение баллов по дисциплине утверждается протоколом заседания кафедры.

По заочной форме обучения мероприятия текущего контроля не предусмотрены.

Для определения фактических оценок каждого показателя выставляются следующие баллы (табл.3):

Таблица 3– Распределение баллов по дисциплине

Вид учебных работ по дисциплине	Количество баллов	
	1 блок	2 блок
<i>Текущий контроль (50 баллов)</i>		
Посещение занятий	4	4
Выполнение заданий по дисциплине (УО, ТЗ, РЗ, ДЗ), в том числе:	16	25
- устный опрос (УО)	3	3
- выполнение тестовых заданий (ТЗ)	4	4
- решение задач (РЗ)	9	9
- выполнение дополнительных заданий- (ДЗ -подготовка доклад к конференции, статьи)	0	9
	20	30
<i>Промежуточная аттестация (50 баллов)</i>		
Экзамен в письменной форме		
Сумма баллов по дисциплине 100 баллов		

Экзамен является формой итоговой оценки качества освоения обучающимся образовательной программы по дисциплине в целом или по разделу дисциплины. По результатам экзамена обучающемуся выставляется оценка «отлично», «хорошо», «удовлетворительно», или «неудовлетворительно».

Оценка «отлично» (81-100 баллов) выставляется обучающемуся, если:

- обучающийся набрал по текущему контролю необходимые и достаточные баллы для выставления оценки автоматом⁶;
 - обучающийся знает, понимает основные положения дисциплины, демонстрирует умение применять их для выполнения задания, в котором нет явно указанных способов решения;
 - обучающийся анализирует элементы, устанавливает связи между ними, сводит их в единую систему, способен выдвинуть идею, спроектировать и презентовать свой проект (решение);
 - ответ обучающегося по теоретическому и практическому материалу, содержащемуся в вопросах экзаменационного билета, является полным, и удовлетворяет требованиям программы дисциплины;
 - обучающийся продемонстрировал свободное владение концептуально-понятийным аппаратом, научным языком и терминологией соответствующей дисциплины;
 - на дополнительные вопросы преподавателя обучающийся дал правильные ответы.
- Компетенция (и) или ее часть (и) сформированы на высоком уровне (уровень 3) (см. табл. 1).

Оценка «хорошо» (61-80 баллов) выставляется обучающемуся, если:

- обучающийся знает, понимает основные положения дисциплины, демонстрирует умение применять их для выполнения задания, в котором нет явно указанных способов решения; анализирует элементы, устанавливает связи между ними;

⁶ Количество и условия получения необходимых и достаточных для получения автомата баллов определены Положением о системе «Контроль успеваемости и рейтинг обучающихся»

- ответ по теоретическому материалу, содержащемуся в вопросах экзаменационного билета, является полным, или частично полным и удовлетворяет требованиям программы, но не всегда дается точное, уверенное и аргументированное изложение материала;

- на дополнительные вопросы преподавателя обучающийся дал правильные ответы;
- обучающийся продемонстрировал владение терминологией соответствующей дисциплины.

Компетенция (и) или ее часть (и) сформированы на среднем уровне (уровень 2) (см. табл. 1).

Оценка «удовлетворительно» (41-60 баллов) выставляется обучающемуся, если:

- обучающийся знает и воспроизводит основные положения дисциплины в соответствии с заданием, применяет их для выполнения типового задания, в котором очевиден способ решения;

- обучающийся продемонстрировал базовые знания важнейших разделов дисциплины и содержания лекционного курса;

- у обучающегося имеются затруднения в использовании научно-понятийного аппарата в терминологии курса;

- несмотря на недостаточность знаний, обучающийся имеет стремление логически четко построить ответ, что свидетельствует о возможности последующего обучения.

Компетенция (и) или ее часть (и) сформированы на базовом уровне (уровень 1) (см. табл. 1).

Оценка «неудовлетворительно» (менее 41 балла) выставляется обучающемуся, если:

- обучающийся имеет представление о содержании дисциплины, но не знает основные положения (темы, раздела, закона и т.д.), к которому относится задание, не способен выполнить задание с очевидным решением;

- у обучающегося имеются существенные пробелы в знании основного материала по дисциплине;

- в процессе ответа по теоретическому материалу, содержащемуся в вопросах экзаменационного билета, допущены принципиальные ошибки при изложении материала.

Компетенция(и) или ее часть (и) не сформированы.

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

Оценивание результатов обучения студентов по дисциплине «Технологии Web программирования» осуществляется по регламенту текущего контроля и промежуточной аттестации.

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы студентов. Результаты текущего контроля подводятся по шкале балльно - рейтинговой системы, реализуемой в ДГТУ.

Текущий контроль осуществляется два раза в семестр по календарному графику учебного процесса в рамках проведения контрольных точек.

Формы текущего контроля знаний:

- устный опрос;

- выполнение тестовых заданий (ТЗ);

- решение практических заданий и задач (РЗ);

- дополнительные задания (ДЗ).

Проработка конспекта лекций и учебной литературы осуществляется студентами в течение всего семестра, после изучения новой темы. Перечень вопросов для устного опроса определен содержанием темы в РПД и методическими рекомендациями по изучению дисциплины.

Защита практических заданий производится студентом в день их выполнения в соответствии с расписанием занятий. Преподаватель проверяет правильность выполнения практического задания студентом, контролирует знание студентом пройденного материала с помощью контрольных вопросов или тестирования.

Оценка компетентности осуществляется следующим образом: в процессе защиты выявляется информационная компетентность в соответствии с практическим заданием, затем преподавателем дается комплексная оценка деятельности студента.

Высокую оценку получают студенты, которые при подготовке материала для самостоятельной работы сумели самостоятельно составить логический план к теме и реализовать его, собрать достаточный фактический материал, показать связь рассматриваемой темы с современными проблемами науки и общества, сонаправлением обучения студента и каков авторский вклад в систематизацию, структурирование материала.

Оценка качества подготовки на основании выполненных заданий ведется преподавателям (с обсуждением результатов), баллы начисляются в зависимости от сложности задания.

Итоговый контроль освоения умения и усвоенных знаний дисциплины «Технологии Web программирования» осуществляется в процессе промежуточной аттестации на экзамене. Условием допуска к экзамену является положительная текущая аттестация по всем практическим работам учебной дисциплины, ключевым теоретическим вопросам дисциплины.

2 Контрольные задания (демоверсии) для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

2.1 Задания для оценивания результатов обучения в виде знаний и умений:

2.1.1 Вопросы устного опроса (УО) для оценивания результатов обучения в виде знаний и умений:

1. Структура HTML-документа и элементы разметки заголовка документа.
2. Типовая структура HTML-документа и содержание его заголовка.
3. Содержание элементов разметки.
4. Контейнеры тела документа.
5. Элементы разметки тела HTML - документа.
6. Типизация, назначение и применение.
7. Списки.
8. Гиперссылки и якоря.
9. Спецификация якорей и гиперссылок.
10. Взаимосвязи документов: элемент LINK.
11. Гиперссылки вперёд и назад.
12. Гиперссылки и машины поиска.
13. Информация пути: элемент BASE.
14. Графика. Принципы применения графических образов при HTML-разметке.
15. Карты изображений.
16. Таблицы в HTML.
17. Принципы применения таблиц в HTML - разметке.
18. Табличная организация текста.
19. Табличная координатная сетка.
20. Назначение и применение CSS.
21. Основные понятия CSS, их назначение, определение и использование при форматировании HTML- документа.

22. Блочные и строковые элементы.
23. Описание, форматирование и свойства.
24. Цвет и шрифт.
25. Управление отображением цветом и текста и фоном, на котором отображается текст.
26. Использование гарнитур шрифтов.
27. Текст и списки.
28. Свойства текстовых фрагментов: межбуквенные расстояния, высота строк, выравнивание, отступ в первой строке параграфа, преобразования начертания.
29. Управление формой и отображением списков.
30. Введение в PHP. История языка PHP.
31. Возможности PHP (краткий перечень платформ, протоколов, баз данных, приложений электронной коммерции и функций, которые поддерживаются PHP).
32. Способы использования.
33. Установка и настройка программного обеспечения, необходимого для работы с PHP. Основы синтаксиса. Основной синтаксис PHP.
34. Управляющие конструкции.
35. Условный оператор (if, switch). Циклы (while, for, do while).
36. Операторы включения (include, require).
37. Механизм получения данных из HTML-форм, и их обработка с помощью PHP.
38. Функции в PHP. Понятие функции. Функции, определяемые пользователем.

Критерии оценки устного опроса:

- качество ответов (ответы должны быть полными, четко выстроены, логичными (аргументированными));

- владение научным и профессиональной терминологией.

Шкала оценивания устного опроса.

Каждый вопрос оценивается по следующей шкале:

- 0 баллов - обучающийся дал неправильный ответ на вопрос или не ответил;

- 1 балл - ответ обучающегося является не полным, не точным, не уверенным и не аргументированным;

- 2 балла – ответ обучающегося является полным, но не точным, не уверенным и не аргументированным;

- 3 - ответ обучающегося является полным, точным, уверенным и аргументированным.

По результатам опросов выводится средняя оценка, которая округляется до целой величины и выставляется при первой рейтинговой оценке.

2.1.2 Тестовые задания (ТЗ) для оценивания результатов обучения в виде знаний:

- 1) Расшифруйте аббревиатуру HTML
 - a) HyperText Markup Language
 - b) HeadText Modulation Language
 - c) HeadingText Mode Language
- 2) Где следует располагать tag title?
 - a) В теге: head
 - b) В теге: body

- c) В теге: footer
- 3) Какую роль выполняет !DOCTYPE ?
 - a) Он предназначен для указания типа текущего документа HTML или XHTML, узнав которую, браузер начинает разбирать код в соответствии с версией
 - b) Это открывающий тег, используемый для объявления заголовков
 - c) Это элемент, который говорит браузеру, о том что эта страница главнее остальных
- 4) Существует ли у тега input закрывающий тег в синтаксисе HTML?
 - a) Да
 - b) Нет
 - c) Такого тега в стандартах HTML не существует
- 5) Если Вам потребуются создать ссылку в документе, какой тег Вы будете использовать?
 - a) Тег: p
 - b) Тег: link
 - c) Тег: a
- 6) Какой атрибут следует указывать для определения URL-адреса в ссылке?
 - a) Атрибут : alt
 - b) Атрибут : href
 - c) Атрибут : name
- 7) Какая из предложенных ссылок является абсолютной?
 - a) <http://www.page.html>
 - b) <catalog/page.html>
 - c) ../catalog/page.html
- 8) Если Вам нужно вывести изображение, какой тег следует использовать?
 - a) Тег: images
 - b) Тег: img
 - c) Тег: image
- 9) Элемент div является строчным или блочным элементом?
 - a) Блочным
 - b) Строчным
 - c) Является как строчным, так и блочным элементом
- 10) Можно ли в строчные элементы вкладывать блочные?
 - a) Нельзя. Это не семантическая верстка
 - b) Можно. На семантику это не влияет
 - c) Нет необходимости
- 11) К каким тегам в документе поисковые системы обращаются для получения описания сайта, ключевых слов и других данных?
 - a) Тег: output
 - b) Тег: head
 - c) Тег: meta
- 12) Если Вам нужно вывести описание содержимого тега в виде всплывающей подсказки, какой атрибут Вы используете?
 - a) Атрибут: type
 - b) Атрибут: title
 - c) Атрибут: value
- 13) Если Вам нужно связать элемент на странице со стилевым оформлением, то какой атрибут Вы используете?

- a) Атрибут: type
 - b) Атрибут: data
 - c) Атрибут: class
- 14) Сколько раз ID элемента может повторяться в коде документа?
- a) Один раз и более
 - b) Сколько угодно
 - c) Только один раз
- 15) Вам нужен нумерованный список. Какой элемент Вы используете?
- a) ul
 - b) ol
 - c) dl
- 16) Начнем с простого: как в PHP создать новую переменную с именем animal и строковым значением 'cat'?
- a) animal = 'cat';
 - b) \$animal = cat
 - c) string animal = "cat";
 - d) var animal = 'cat';
 - e) \$animal = 'cat';
- 17) С помощью какой функции можно удалить Cookie?
- a) setcookie
 - b) deletcookie
 - c) readcookie
 - d) getcookie
- 18) Как получить данные POST-запроса?
- a) через константу POST
 - b) через переменную \$_POST
 - c) через функцию getPostData()
- 19) Как нужно создавать новый массив?
- a) \$a = [];
 - b) \$a = array();
 - c) \$a = new Array();
 - d) \$a = null;
- 20) Что будет в переменной \$result после выполнения кода \$result = require 1.php, если в 1.php написан код <?php return 7 % 4;?
- a) 7
 - b) 4
 - c) 3
 - d) 2
- 21) С помощью какой функции можно прочитать файл?
- a) array_merge
 - b) print_r
 - c) file_get_contents
- 22) Что делает оператор break при вызове внутри цикла?
- a) останавливает работу цикла
 - b) переводит цикл на следующую итерацию, обрывая выполнение текущей
- 23) Что будет в переменной \$result после выполнения кода \$result = 2 ** 3;?

- a) 8
 - b) 6
 - c) 2
 - d) 4
- 24) Как называется возможность указывать типы аргументов у функции `function(int $x)...`?
- a) в PHP нет такой возможности
 - b) тайп-хинтинг
 - c) приведение типов
 - d) строгая типизация
- 25) С помощью какого символа в PHP можно склеить 2 строки в одну?
- a) символ доллара
 - b) символ точки
 - c) символ запятой
- 26) Что будет в переменной `$x` после выполнения кода `$x = 2; $x -= 2;?`
- a) -2
 - b) 4
 - c) 2
 - d) 0
- 27) Возможно ли на PHP писать рекурсивные функции?
- a) нет
 - b) да
- 28) Какая из этих конструкций не относится к циклам?
- a) `isset`
 - b) `foreach`
 - c) `while`
 - d) `for`
- 29) Что будет в переменной `$result` после выполнения кода `$result = 2 + 2 * 2;?`
- a) 2
 - b) 4
 - c) 8
 - d) 6
- 30) Продолжит ли код выполнение после выражения `include 1.php`; если файл `1.php` не будет найден?
- a) да, без каких либо ошибок
 - b) нет, выведет ошибку и завершится
 - c) да, но выведет ошибку
- 31) Как присвоить переменной значение?
- a) `$x => 5`
 - b) `$x = 5`
 - c) `$x ==== 5`
 - d) `$x == 5`
- 32) Что будет в переменной `$result` после выполнения кода `$i = 5; $result = $i++;?`
- a) 5
 - b) 7
 - c) 6
 - d) 4

- 33) Каким является язык PHP?
- a) интерпретируемым
 - b) компилируемым
- 34) Что будет в переменной \$x после выполнения кода \$x = 5; \$x % 2;?
- a) 5
 - b) 0
 - c) 2
 - d) 1
- 35) Какой тип значения будет в переменной \$a после выполнения кода \$a = '123';?
- a) float
 - b) int
 - c) boolean
 - d) string

Тестовые задания (ТЗ) выполняются студентами перед контрольной точкой текущей аттестации соответственно по разделам.

Максимальное количество баллов по разделу – 4.

Оценка 4 балла выставляется обучающемуся, если он правильно ответил на 80% и более тестовых заданий;

Оценка 3 балла выставляется обучающемуся, если он правильно ответил на 61-79% тестовых заданий;

Оценка 2 балла выставляется обучающемуся, если он правильно ответил на 41-60% тестовых заданий;

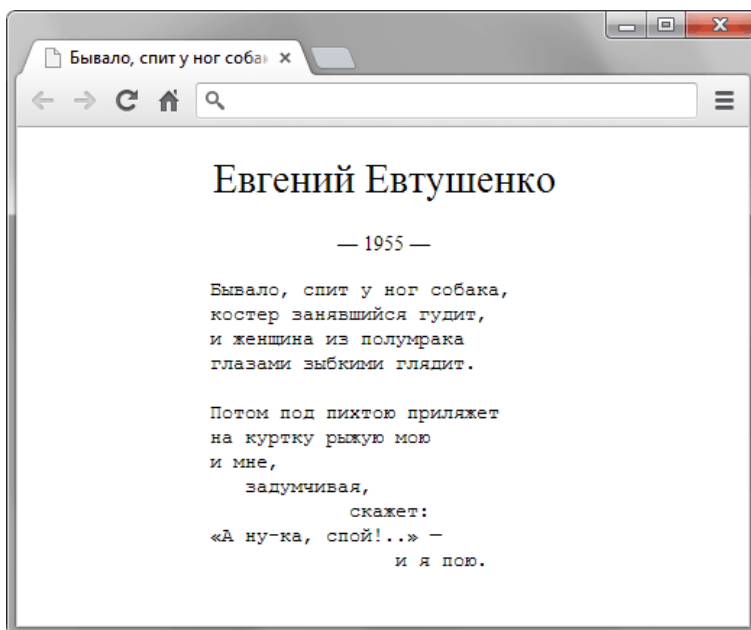
Оценка 1 балл выставляется обучающемуся, если он правильно ответил на 21-40% тестовых заданий;

Оценка 0 баллов выставляется обучающемуся, если он правильно ответил на 20% и менее тестовых заданий.

2.2 Задания для оценивания результатов обучения в виде владений и умений

2.2.1 Комплекс практических заданий и задач (РЗ)

1. Оформите стихотворение, как показано на рисунке:



Евгений Евтушенко

— 1955 —

Бывало, спит у ног собака,
костер занявшийся гудит,
и женщина из полумрака
глазами зыбкими глядит.

Потом под пихтою приляжет
на куртку рыжую мою
и мне, задумчивая, скажет:
«А ну-ка, спой!..» —
и я пою.

2. Сделайте ссылку, которая при наведении на неё курсора мыши меняла свой вид, как показано на рис. 1. Вверху показана исходная ссылка, внизу ссылка после наведения курсора.

Комментировать

Комментировать

3. Исправьте ошибки в приведенном коде:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html>
<title>Glossary</title> <meta http-equiv="content-type" content="text/html; charset=utf-8">
<body> <a href="glosstop.html"><h1>Glossary of Terms</h1></a> <span>
<h2><p>Algorithmic Oriented Language.</h2></p> </span> <span> <h2><p>Creates new
project</p></h2> </span> </body> </html>
```

4. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"> <html> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-1"> <body> <h1 align="justify">Галион</h1> <p align="justify">Галион - большое

трехмачтовое судно особо прочной постройки, снабженное тяжелой артиллерией.

Эти суда служили для перевозки товаров и драгоценных металлов из испанских и португальских колоний в Европу.

Флагманский корабль был мощным *галионом*, вооруженным сорока Восьмью большими пушками и восьмью малыми.

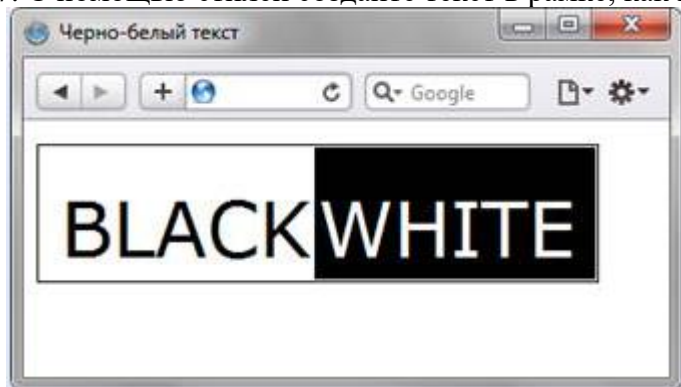
```
5. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"> <html> <head> <meta
http-equiv="Content-Type" content="text/html; charset=windows-1251"> <head> <body>
<h2>Опрос общественного мнения показал</h2> <p>Диего Веласкес писал свои полотна в
стиле: империализма, кубизма и импрессионизма;</p> конфуцианство возникло в:
Италии, Корее и Франции;</p> <p>богами торговли и воровства в Древней Греции были:
Марс, Меркурий и Дионис;</p> <i>богами загробного мира в Египте были: Аид, Анубис
и Ассирис;</i></p> <p>столица Золотой Орды - Монголия;</p> <p>главные противники
Александра Македонского: татаро-монголы, Дедал и Цезарь;</p> <p>в Древней Греции
морями повелевали боги: Плутон, Нептун и Тритон.</p> </ul> </body> </html>
```

6. Сделайте страницу с изображением флага Японии, как показано на рисунке. Размер 300x200 пикселей, диаметр круга 120 пикселей. Любые картинки применять запрещено, всё надо сделать с помощью CSS. Страница должна корректно отображаться во всех современных браузерах.

*Для круга следует использовать свойство `background: radial-gradient(circle, #bc002d 58px, #fff 60px);`



7. С помощью стилей создайте текст в рамке, как показано на рисунке:



8. С помощью тегов `` и `` постройте списки, показанные на рис. 1. При этом у вас должен быть валидный XHTML и CSS. Списки должны корректно отображаться в браузерах IE8, Firefox 6, Safari 5, Opera 11, Chrome 8 и в их старших версиях.

Сделать 2 варианта: 1 – с помощью html, 2 – с помощью CSS.

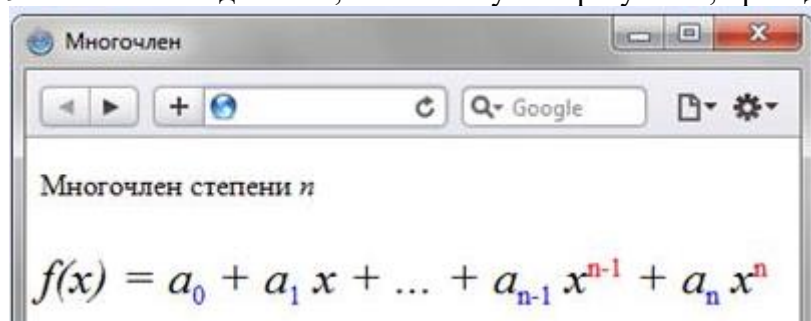
Первый список, начинается с 1

01. Баал
02. Агарес
03. Марбас
04. Пруфлас
05. Аамон

Второй список, начинается с 6

06. Барбатос
07. Буер
08. Гасион
09. Ботис

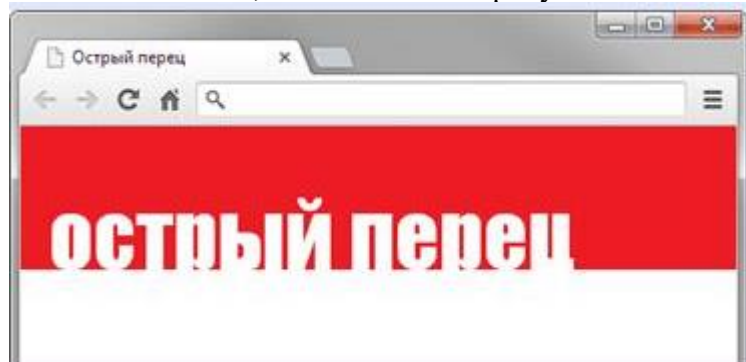
9. Напишите код HTML, чтобы получить результат, приведенный на рисунке



10. Измените стиль для таблицы, чтобы она получилась, как показано на рисунке. Вносить изменения в код таблицы нельзя, всё оформление должно делаться только через стили.

	2004	2005	2006
Рубины	43	51	79
Изумруды	28	34	48
Сапфиры	29	57	36
Аметисты	23	64	97

11. Сделайте текст, как показано на рисунке. В качестве шрифта укажите Impact.



12. Сверстайте форму регистрации, показанную на рисунке. Ширина формы и её полей фиксирована.

Регистрация

E-mail

Пароль

Ещё раз

Отправить

13. Установите **WAMP** сборку, если вы работаете под Windows (м.б. вам понадобится LAMP или MAMP, если вы используете другую операционную систему)
14. Узнайте каков объем памяти выделяется PHP-скрипту. Примечание: задание нужно сделать двумя способами — посмотреть в конфигурационном файле или получить параметр из **php-функции**.
15. Узнайте какова максимальная длительность выполнения PHP-скрипта. Примечание: см. предыдущую задачу.
16. Установите параметры из первых двух задач в значения, отличные от начальных. Убедитесь, что изменения применены и используются.
17. Вызовите в вашем скрипте функцию **phpinfo()** и поверхностно познакомьтесь с показанными параметрами
18. Вызовите оператор **echo** с параметром «привет»; убедитесь в использовании нужной кодировки на сервере.
19. Проверьте вашу текущую версию PHP.
20. * Пропишите путь к PHP-интерпретатору в системной переменной PATH.
21. * Вызовите PHP-скрипт из консоли.
22. Распечатайте результат вызова функции **date('d-m-Y H:i:s');**
23. Испытайте работу всех PHP-тегов, которые доступны в вашей версии PHP
24. Возьмите один из своих HTML-файлов. Измените его расширение на .php, убедитесь в доступности файла на сервере.
25. **Создайте PHP переменную \$htmlab** и поместите в нее значение «gu». Распечатайте результат.
26. **Создайте константу NPP** (news per page — количество новостей на страницу) и задайте ей значение 10.
27. Создайте две **целочисленных переменных** и распечатайте результат выполнения над ними математических операторов. Примечание: например, \$num1=23 и \$num2=67; результат оператора сложения **echo \$num1+\$num2**
28. Создайте переменную \$test= «345» и узнайте ее тип, при помощи функции **gettype()**; посмотрите в документации, какая функция устанавливает тип переменной.
29. При помощи разных способов приведения типов данных, приведите переменную \$test из предыдущего задания в числовой тип.
30. **Проверьте**, что выведет выражение **echo «103»+2.**
31. Тут идет ряд задач связанных с js
32. Напишите скрипт, который будет показывать фон на сайте в зависимости от того, четный или нечетный час. Примечание: попробуйте применить тернарный оператор.
33. Напишите скрипт, который будет выводить предупреждение, если на скрипт выделяется объем памяти меньший 128Mb
34. **Напишите скрипт**, который будет, в зависимости от дня недели, выводить надпись. Например: сегодня среда. Примечание: используйте оператор **switch**

35. Напишите **RНР цикл**, который выводит числа от 1 до 100.
36. Напишите **RНР цикл**, который выводит числа от 23 до 78.
37. Напишите **RНР цикл**, который выводит нумерованный список из 10 пунктов.
38. Создайте массив из 100 случайных чисел.
39. Вывести массив из предыдущего задания, при помощи цикла **while**, а потом при помощи **foreach**.
40. Создайте массив из 10 строк и выведите их любым циклом внутри HTML-элемента **div**.
41. * **Создайте массив**, каждый элемент которого тоже массив с ключами title, description, price. Выведите все элементы этого массива, так, чтобы заголовки были в HTML-элементе h2, описания в p, а цена в гиперссылке.
42. * При выводе элементов из предыдущего задания покрасьте фон элементов ниже определенной цены в отличный от других цвет.
43. Создайте массив из 50 случайных чисел от 0 до 100. Найти все числа меньше 72 и поместить их в отдельный массив
44. **Создайте цикл**, который выводит числа то 0 до 100 в HTML-элементах **div**; окраска HTML-элементов должна чередоваться («зебра»).
45. Создать **функцию в RНР** — getPlus10(), которая будет принимать число и распечатывать сумму этого числа и 10.
46. Изменить функцию из предыдущего задания: она должна возвращать сумму, а не сразу выводить.
47. Напишите функцию pythagoras(), которая принимает значения двух катетов прямоугольного треугольника и возвращает размер гипотенузы этого прямоугольного треугольника. Примечание: может пригодиться **встроенная RНР-функция** pow().
48. * Создать функцию, которая находит ипотечный платеж
49. Создайте функцию col(), которая распечатает количество **переданных аргументов**. Например: col(12,6,123) должна распечатать число 3.

2.3 Типовые экзаменационные материалы

Перечень вопросов для проведения экзамена(теоретические вопросы)

1. Понятие и функции поисковой системы
2. История развития поисковых систем
3. Состав и принципы работы поисковой системы
4. Тематический индекс цитирования
5. Индексы поисковых систем
6. Поисковый сервер
7. Результаты поиска
8. Поисковая форма
9. Структура HTML-документов
10. Заголовки в HTML
11. Обычный текст или абзац в HTML
12. Ссылки в HTML
13. Списки в HTML
14. Рисунки в HTML
15. Таблицы в HTML
16. Формы HTML
17. Поиск в HTML-документах
18. Фреймы

19. Стили CSS
20. Переменные. Типы переменных в PHP
21. Объекты и свойства в PHP
22. Операторы в PHP
23. Функции в PHP
24. Методы в PHP
25. События в PHP. Комментарии
26. Случайные числа в PHP
27. Ветвление в PHP
28. Массивы в PHP
29. Цикл WHILE в PHP
30. Цикл FOR в PHP
31. Переменные и типы данных в PHP
32. Работа с файлами в PHP
33. Взаимосвязь PHP и MySQL
34. Безопасность в PHP
35. Регулярные выражения

Структура экзаменационного билета:

1. Теоретический вопрос.
2. Практическое задание (задача).

Пример экзаменационного билета



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ИНСТИТУТ ТЕХНОЛОГИЙ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
 ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
 «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
 В Г. ВОЛГОДОНСКЕ РОСТОВСКОЙ ОБЛАСТИ**

(Институт технологий (филиал) ДГТУ в г. Волгодонске)

Факультет Технологии и менеджмент

Кафедра Технический сервис и информационные технологии

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

на 2022/2023 учебный год

Дисциплина Технологии Web программирования

1. Понятие и функции поисковой системы
2. Создайте список с подпунктами состоящий минимум из 8 пунктов.

Экзаменатор	_____	К.А. Чернышов	30.08.2022
подпись	Ф.И.О.	дата	
Зав.кафедрой	_____	Н.В.Кочковая	30.08.2018
Подпись	Ф.И.О.	Дата	

АКТУАЛЬНО НА

20__/20__уч.год	_____	20__/20__уч.год	_____
Подпись	Ф.И.О. зав.каф.	Подпись	Ф.И.О. зав.каф.

20__/20__уч.год	_____	20__/20__уч.год	_____
Подпись	Ф.И.О. зав.каф.	Подпись	Ф.И.О. зав.каф.

Карта тестовых заданий

Направление подготовки: Информационные системы и технологии

Профиль: Информационные системы

Компетенция: ПК-5: Способность программировать и разрабатывать прикладное программное обеспечение

Дисциплина: Технологии web-программирования

Описание теста:

Описание теста:

1. Тест состоит из 70 заданий, которые проверяют уровень освоения компетенций обучающегося. При тестировании каждому обучающемуся предлагается 30 тестовых заданий по 15 открытого и закрытого типов разных уровней сложности.

2. За правильный ответ тестового задания обучающийся получает 1 условный балл, за неправильный ответ – 0 баллов. По окончании тестирования, система автоматически определяет «заработанный итоговый балл» по тесту, согласно критериям оценки

3 Максимальная общая сумма баллов за все правильные ответы составляет – 100 баллов.

4. Тест успешно пройден, если обучающийся правильно ответил на 70% тестовых заданий (61 балл).

5. На прохождение тестирования, включая организационный момент, обучающимся отводится не более 45 минут. На каждое тестовое задание в среднем по 1,5 минуты.

6. Обучающемуся предоставляется одна попытка для прохождения компьютерного тестирования.

Кодификатором теста по дисциплине является раздел рабочей программы «4. Структура и содержание дисциплины (модуля)»

Комплект тестовых заданий

Задания закрытого типа

Задания альтернативного выбора

*Выберите **один** или несколько правильных ответов*

Простые (1 уровень)

1 Под протоколом HTTP принято понимать _____

А) Базовый протокол сети гипертекстовых ресурсов

Б) Протокол режима командной строки

В) Первый обозреватель с графическим интерфейсом

Г) Протокол коммутации пакетов, предполагающий разбиение запросов данных на небольшие фрагменты

2 Механизм, который позволяет серверу хранить информацию на компьютере клиента и извлекать её оттуда - это _____

А) Механизм Cookie

Б) Механизм World Wide Web

В) Механизм HTTP

Г) Механизм Frame

3 В многоуровневной архитектуре клиент-сервер большая часть бизнес-логики находится в _____

А) терминале

Б) сервере приложений

В) сервере баз данных

Г) веб-сервисе

4 Этим утверждением **НЕЛЬЗЯ** охарактеризовать архитектуру Веб-приложения

- А) постоянная необходимость использовать дополнительное ПО на стороне клиента
- Б) возможность подключения практически неограниченного количества клиентов
- В) недоступность при отсутствии работоспособности сервера или каналов связи
- Г) доступность при работоспособности сервера и каналов связи

5 Это утверждение относительно метода POST **НЕВЕРНО**:

- А) отправляемая информация не показывается в адресной строке
- Б) метод POST позволяет передавать запросу файлы
- В) сгенерированные страницы можно добавить в закладки
- Г) нарушается логика работы кнопки "Назад" в браузере

Средне –сложные (2 уровень)

6 На этом этапе жизненного цикла страницы устанавливаются свойства страницы, например Request и Response

- А) запрос страницы
- Б) запуск
- В) инициализация страницы
- Г) загрузка

7 Основой паттерна является

- А) отделение модели данных приложения, его логики и представления данных друг от друга
- Б) отделение клиента от сервера
- В) отделение модели данных приложения от логики
- Г) отделение модели данных приложения от его представления данных

8 Паттерн MVC **НЕ** включает в себя

- А) модель
- Б) представление
- В) контроллер
- Г) правила роутинга

9 К статическим функциям, определенным в некотором классе, следует обращаться как

- А) \$имя_экземпляра_класса -> имя_функции
- Б) \$имя_экземпляра_класса :: имя_функции
- В) имя_класса => имя_функции
- Г) имя_класса -> имя_функции

10 Если в классе не объявлен абстрактный метод, а сам класс объявлен абстрактным, то _____

- А) будет сгенерирована ошибка
- Б) будет сгенерировано предупреждение
- В) **ни ошибки, ни предупреждения не будет сгенерировано - такое использование механизма абстракции является корректным**
- Г) ни ошибки, ни предупреждения не будет сгенерировано, однако такой класс нельзя будет использовать в качестве родителя при наследовании

11 Механизм интерфейсов в PHP _____

- А) **поддерживает множественное наследование**
- Б) множественного наследования не поддерживает, в отличие от абстрактных классов
- В) поддерживает множественное наследование только в случае, если все предки являются производными одного и того же интерфейса
- Г) отсутствует

12 В классе определено одно свойство, имеющее модификатор доступа public, и одно, имеющее модификатор доступа protected. Для класса используется механизм итерации по умолчанию. Цикл, построенный на использовании foreach, даст доступ

- А) и к public, и к protected свойствам
- Б) **только к public свойству**
- В) только к protected свойству

Г) ни к одному свойству доступ не будет предоставлен

13 Входные скрипты Yii2 _____

А) включают в себя полностью все средства для MVC

Б) ответственны за запуск и обработку входящего запроса

В) осуществляют корректную работу различных компонентов приложения и их координацию для обработки запроса

Г) принимают входные данные от пользователя и преобразовывают их в понятный для моделей формат

14 В состав маршрута **НЕ** входит

А) Id действия

Б) Id контроллера

В) Id модуля

Г) Id скрипта

15 Контроллеры Yii2 **НЕ** могут _____

А) иметь доступ к данным запроса

Б) вызывать методы моделей

В) использовать HTML или другую разметку

Г) отвечать за фильтрацию и авторизацию

16 Модели Yii2 **НЕ** поддерживают

А) пре-фильтрацию

Б) атрибуты

В) массовое присвоение

Г) правила валидации

17 Для создания в форме элемента с меткой и любой применимой валидацией с помощью JavaScript, вызывается метод _____

А) ActiveForm::begin()

Б) ActiveForm::field()

В) ActiveForm::input()

Г) ActiveForm::label()

18 В Yii2 для создания выпадающего списка используется метод _____

А) ActiveForm::list()

Б) ActiveForm::dropdownlist()

В) ActiveForm::radio()

Г) ActiveForm::select()

19 Роутинг - это процесс, который _____

А) отвечает за создание URL из заданного маршрута и соответствующих параметров запроса

Б) позволяет приложению распознавать различные форматы URL без внесения изменений в существующий код приложения

В) разбирает URL в маршрут, который используется при создании соответствующего экземпляра действия контроллера для обработки запроса

Г) позволяет временно перевести приложение в режим обслуживания и отображать одно информационное сообщение для всех запросов

20 В Yii2 для создания ссылки на действие контроллера используется метод _____

А) createUrl()

Б) generateUrl()

В) actionUrl()

Г) linkUrl()

21 : \$count = Customer::find()->where(['status' => Customer::STATUS_ACTIVE])->count();

Как результат работы представленный код _____

А) возвращает количество активных покупателей

Б) возвращает всех покупателей массивом, индексированным их идентификаторами

- В) возвращает всех активных покупателей, сортируя их по идентификаторам
 Г) возвращает количество покупателей

22 В Yii2 для задания параметров пагинации используется метод _____

- А) **paginate()**
 Б) **limit()**
 В) **offset()**
 Г) **page()**

23 Встроенные валидаторы Yii2 – это

- А) **boolean**
 Б) **date**
 В) **compare**
 Г) **null**
 Д) **datetime**
 Е) **html**
 Ж) **file**
 З) **email**

24 Виджеты Yii2 для работы с данными - это

- А) **DetailView**
 Б) **DataProvider**
 В) **ActiveRecordView**
 Г) **DetailView**
 Д) **ContentView**
 Е) **ListView**
 Ж) **TableView**
 З) **GridView**

25 При подключении к базе данных в Yii2 в конфигурационном файле необходимо указать параметры

- А) **class**
 Б) **dsn**
 В) **dbname**
 Г) **username**
 Д) **password**
 Е) **host**
 Ж) **server**
 З) **db**

26 Методы класса ActiveForm - это

- А) **field()**
 Б) **textarea()**
 В) **password**
 Г) **passwordInput()**
 Д) **hint()**
 Е) **checkboxList()**
 Ж) **prompt**
 З) **multiply**

27 Класс AppAsset содержит свойства

- А) **\$basePath**
 Б) **\$baseUrl**
 В) **\$asset**
 Г) **\$view**
 Д) **\$css**
 Е) **\$js**
 Ж) **\$resource**
 З) **\$depends**

28 Для настройки ЧПУ следует установить параметры компонента urlManager

- А) **enablePrettyUrl**
 Б) **enableFriendlyUrl**
 В) **showIndexName**

- Г) SEO-URL
- Д) validateUrl
- Е) showScriptName**
- Ж) enableStrictParsing**
- З) rules**

- 29 Компонент request позволяет получить
- А) url**
 - Б) absoluteUrl**
 - В) server
 - Г) param
 - Д) hostInfo**
 - Е) scriptName
 - Ж) queryString**
 - З) port

- 30 К методам построителя запросов относят
- А) join()**
 - Б) select()**
 - В) from()**
 - Г) get()
 - Д) having()**
 - Е) groupBy()**
 - Ж) group()
 - З) index()

- 31 К базовым хелперам Yii2 относят
- А) ArrayHelper**
 - Б) Html**
 - В) Url helper**
 - Г) FormHelper
 - Д) CaptchaHelper
 - Е) SessionHelper
 - Ж) CookieHelper
 - З) DbHelper

- 32 Наиболее известные PHP-фреймворки - это
- А) Yii2**
 - Б) Laravel**
 - В) Spring
 - Г) Symfony**
 - Д) Django
 - Е) Angular
 - Ж) React
 - З) Vue

Сложные (3 уровень)

- 33 При объявлении связи **НЕ** указывается
- А) кратность связи
 - Б) название таблицы, с которой идет связь**
 - В) название связанного Active Record класса
 - Г) связь между двумя типами данных

- 34 Этот метод может вызвать проблемы производительности, когда вам понадобится получить доступ к тем же самым свойствам связей для нескольких объектов Active Record
- А) Жадная загрузка
 - Б) Отложенная загрузка**
 - В) Загрузка методом join
 - Г) DAO-построитель запросов

35 Гость имеет доступ к действиям _____

```
class SiteController extends Controller
{
  public function behaviors()
  {
    return [
      'access' => [
        'class' => AccessControl::class,
        'only' => ['login', 'logout', 'signup'],
        'rules' => [
          [
            'allow' => true,
            'actions' => ['login', 'signup'],
            'roles' => ['?'],
          ],
          [
            'allow' => true,
            'actions' => ['logout'],
            'roles' => ['@'],
          ],
        ],
      ],
    ];
  }
  // ...
}
```

- A) login и signup
- Б) login, logout и signup
- В) logout
- Г) для гостя не прописана роль

Задания на установление соответствия

Установите соответствие между левым и правым столбцами.

Простые (1 уровень)

36 Установите соответствие (1-А, 2-Б, 3-Г):

- | | |
|----------------|---|
| 1 Инкапсуляция | А) размещение одного объекта или класса внутри другого для разграничения доступа к ним |
| 2 Наследование | Б) способность объекта или класса базироваться на другом объекте или классе |
| 3 Полиморфизм | В) гарантия реализации функционала
Г) реализация задач одной и той же идеи разными способами |

37 Установите соответствие (1-В, 2-Б, 3-А):

- | | |
|-------------|---|
| 1 public | А) свойство или метод могут быть доступны ТОЛЬКО внутри класса |
| 2 protected | Б) к свойству или методу можно получить доступ внутри класса и классов, производных от этого класса |
| 3 private | В) свойство или метод доступны отовсюду
Г) программные сущности должны быть открыты для расширения, но закрыты для изменений |

Средне-сложные (2 уровень)**38 Установите соответствие (1-Г, 2-Д, 3-Б, 4-А):**

- | | |
|-----|---------------------|
| 1 С | А) удаление данных |
| 2 R | Б) изменение данных |
| 3 U | В) миграция данных |
| 4 D | Г) создание данных |
| | Д) чтение данных |

39 Установите соответствие (1-А, 2-Б, 3-Г):

- | | |
|--------------|---|
| 1 Model | А) отвечает за данные, определяет структуру программного приложения |
| 2 View | Б) обеспечивает взаимодействие с пользователем |
| 3 Controller | В) обеспечивает ведение логов данных |
| | Г) определяет, как веб-сайт реагирует на действия пользователя |

40 Установите соответствие: (1-Г, 2-В, 3-А):

- | | |
|-------------|--|
| 1 web.php | А) точка входа в приложение Yii |
| 2 assets/ | Б) сторонние библиотеки |
| 3 index.php | В) скрипты, используемые приложением (js, css) |
| | Г) конфигурация Web приложения |

41 Установите соответствие (1-Б, 2-А, 3-В):

- | | |
|------------|--|
| 1 exists() | А) возвращает количество записей |
| 2 count() | Б) возвращает значение указывающее, что выборка содержит результат |
| 3 column() | В) возвращает первый столбец результата. |
| | Г) возвращает дескриптор соединения |

42 Установите соответствие (1-А, 2-Б, 3-В):

- | | |
|---------------------|--|
| 1 findIdentity() | А) находит экземпляр identity class, используя ID пользователя |
| 2 getAuthKey() | Б) возвращает ключ, используемый для основанной на cookie аутентификации |
| 3 validateAuthKey() | В) реализует логику проверки ключа |
| | Г) обращается к базе данных для поиска пользователя по ID |

43 Установите соответствие (1-Б, 2-Г, 3-А):

- | | |
|------------|--|
| 1 required | А) обрезает пробелы вокруг входящих данных |
| 2 safe | Б) проверяет, являются ли входящие значения не пустыми |
| 3 trim | В) проверяет соответствие регистра |
| | Г) указывает, что атрибут является безопасным |

44 Укажите верный порядок параметров при установлении связи в классе Active Record (**1-А, 2-Б, 3-В**):

- А) кратность связи
- Б) название связанного Active Record класса
- В) связь между двумя типами данны: указываются столбцы с помощью которых два типа данных связаны
- Г) имя базы данных

Сложные (3 уровень)

45 Установите верную последовательность жизненного цикла запроса пользователя (**1-Б, 2-В, 3-Г, 4-Д, 5-Е, 6-Ж, 7-А**):

- А) генерация вида
- Б) создание экземпляра приложения
- В) определение маршрута запроса
- Д) создание экземпляра контроллера
- Е) создание действия и наложение фильтров
- Ж) загрузка модели

Задания открытого типа

Задания на дополнение

Напишите пропущенное слово.

Простые (1 уровень)

46 Специальная программа, позволяющая просматривать содержимое сети Интернет - это _____ (**браузер, веб-браузер**)

47 Чтобы запретить доступ к методу в производных классах, следует объявить метод как _____ (**private**)

48 Чтобы сохранить доступность метода в производных классах, но запретить его переопределение, следует объявить метод как _____ (**final**)

49 Пакетный менеджер уровня приложений для языка программирования PHP, который предоставляет средства по управлению зависимостями в PHP-приложении - это _____ (**Composer**)

50 Если класс модели расширяется напрямую от `yii\base\Model` то все не статичные публичные переменные - это _____ (**атрибутами, атрибутом, атрибуты**)

Средне-сложные (2 уровень)

51 Для подключения представления Yii2 используется метод _____ (**render, render()**)

52 Метод Yii2, который может вставить или обновить строку данных в таблице базы данных - это _____ (**save(), save**)

53 Объекты Yii2, которые могут включать в себя представления, содержать различную логику и быть использованы в различных представлениях, - это _____ (**виджет, виджеты, widget**)

54 Метод, в котором задаются метки атрибутов - это _____ (**attributelabels(), attributelabels**)

55 Валидатор Yii2, который указывает, что атрибут является безопасным - это _____ (**safe**)

56 Валидатор Yii2, который проверяет, что входящее значение совпадает с указанным регулярным выражением - это _____ (**match**)

57 Свойство компонента URL manager, позволяющее выполнить переключение на ЧПУ - это _____ (**\$enablePrettyUrl, enablePrettyUrl**)

58 Метод, который используется при миграции баз данных для отката изменений - это _____ (**down(), down**)

59 Виджет Yii2, который использует данные, как провайдер данных и отображает каждую строку используя `columns` для предоставления данных в таблице - это _____ (**GridView**)

60 Виджет, который отображает детали по данным для одной model - это _____
(**DetailView**)

61 Класс Yii2, который обеспечивает объектно-ориентированный интерфейс для доступа и манипулирования данными, хранящимися в базах данных - это _____ (**Active Record, ActiveRecord**)

62 Файлы ресурсов, расположенные на другом Web сервере, отличного от веб-хостинга вашего приложения - это _____ ресурсы (**внешние**)

63 Приложению распознает различные форматы URL без внесения изменений в существующий код приложения за счет настройки в конфигурации приложения компонента _____ (**urlManager**)

64 Законченные программные блоки, состоящие из моделей, представлений, контроллеров и других вспомогательных компонентов - это _____ (**Модули, Модуль**)

65 Переменная, доступная внутри шаблона, которая содержит результат рендеринга вида, который рендерится при вызове метода render() в контроллерах - это _____ (**\$content, content**)

66 При создании видов, которые генерируют HTML страницы, важно кодировать и/или фильтровать данные, которые приходят от пользователей перед тем как их показывать. Метод хелпера Html, который позволяет показать обычный текст после его кодирования - это _____ (**encode(), encode**)

67 Метод, который вызывает событие EVENT_BEGIN_PAGE – это _____.
(**beginPage(), beginPage**)

68 Для создания поля ввода пароля необходимо использовать метод
<?= \$form->field(\$model, 'password')->_____ ?>
(**passwordInput()**)

69 Объявлять фильтры в классе контроллера можно путём перекрытия метода _____
(**behaviors(), behaviors**)

70 Код задаёт основной комплект ресурсов используемый в шаблоне базового приложения
class _____ extends AssetBundle
(**AppAsset**)

71 Запрос возвращает покупателя с идентификатором 123.
\$customer = Customer::_____ (123);
(**findOne**)

72 : class Site _____ extends Controller
(**Controller**)

73 return \$this->_____ ('entry-confirm', ['model' => \$model]);
(**render**)

74 Заполните пропуск
public function _____
{

```

return [
    [['name', 'email'], 'required'],
    ['email', 'email'],
];
}
(rules())

```

75 Заполните пропуск для выполнения запроса: получить все строки из таблицы "country" и отсортировать их по "name"

```

$countries = Country::find()->_____ ('name')->all();
(orderBy)

```

```

76 $pagination = new _____([
    'defaultPageSize' => 5,
    'totalCount' => $query->count(),
]);
(Pagination)

```

77 Свойство AssetBundle, которое задаёт Web доступную директорию, содержащую файлы ресурсов текущего комплекта - это _____(basePath)

78 Число методов, которые доступны в шаблоне приложения Yii2 - _____ (5)

Сложные (3 уровень)

79 Компонент, который управляет статусом аутентификации пользователя - это _____(user)

80 Виджет, который используется для работы с датой и временем - это _____(DatePicker)

Задания свободного изложения

Напишите развернутый ответ в свободной форме, изложив основные положения, факты, применив важнейшие понятия и сделав обобщение по теме задания

Простые (1 уровень)

81 Опишите модель для работы с таблицей users (username, email, password)

Ответ:

```

class User extends ActiveRecord
{
    public static function tableName()
    {
        return '{{%users}}';
    }
    public function rules()
    {
        return [
            [['username', 'email', 'password'], 'required'],
            ['email', 'email'],
            ['email', 'unique'],
        ];
    }
}

```

}

Средне-сложные (2 уровень)

82 Напишите код метода, позволяющий добавить нового пользователя в базу данных. Для работы используется таблица users (username, email, password)

Ответ:

```
public function saveUser($username, $email, $password)
{
    $user = User::findOne(['email' => $email]);
    if ($user !== null) {
        return false;
    }

    $user = new User();
    $user->username = $username;
    $user->email = $email;
    $user->password = Yii::$app->security->generatePasswordHash($password);
    return $user->save();
}
```

83 Напишите код метода, позволяющего удалить пользователя из базы данных. Для работы используется таблица users (username, email, password)

Ответ:

```
public function deleteUser($id)
{
    $user = User::findOne($id);
    if ($user !== null) {
        return $user->delete();
    }
    return false;
}
```

84 Опишите правила валидации, используемые при регистрации нового пользователя. Необходимо добавить следующие правила валидации. Поле "username" должно быть обязательным и содержать только буквы, цифры и символы подчеркивания. Поле "email" должно быть обязательным и содержать корректный email-адрес. Поле "email" должно быть уникальным в базе данных. Поле "password" должно быть обязательным и содержать не менее 6 символов.

Ответ:

```
public function rules()
{
    return [
        [['username', 'email', 'password'], 'required'],
        [['username', 'match', 'pattern' => '/^[a-zA-Z0-9_]+$/', 'message' => 'Username can only contain letters, numbers and underscores.'],
        ['email', 'email'],
        ['email', 'unique', 'targetClass' => '\app\models\User', 'message' => 'This email address has already been taken.'],
        ['password', 'string', 'min' => 6],
    ];
}
```

Сложные (3 уровень)

85 Напишите на Yii2 запрос на поиск пользователей, которые оставили более 5 комментариев. Вывод найденных пользователей должен быть реализован с использованием виджета работы с данными.

Ответ:

```
use app\models\User;
use yii\data\ArrayDataProvider;
use yii\grid\GridView;
use yii\db\Query;

$users = (new Query())
    ->select(['user.*', 'COUNT(comment.id) AS comment_count'])
    ->from('user')
    ->leftJoin('comment', 'comment.user_id = user.id')
    ->groupBy('user.id')
    ->having('COUNT(comment.id) > 5')
    ->all();

$dataProvider = new ArrayDataProvider([
    'allModels' => $users,
    'sort' => [
        'attributes' => ['id', 'username', 'email', 'comment_count'],
    ],
]);

echo GridView::widget([
    'dataProvider' => $dataProvider,
    'columns' => [
        'id',
        'username',
        'email',
        'comment_count',
    ],
]);
```

Карта учета тестовых заданий

Направление подготовки	Информационные системы и технологии			
Профиль	Информационные системы и технологии			
Дисциплина	Технологии web-программирования			
Компетенция	ПК-3: Способность выполнять работы и управление работами по созданию (модификации) и сопровождению информационных систем, автоматизирующих задачи организационного управления и бизнес-процессы			
Индикатор	ПК-3.5: Способен использовать знания и технологии Web-программирования в процессе выполнения работ и управления работами по созданию (модификации) и сопровождению информационных систем, автоматизирующих задачи организационного управления и бизнес-процессы			
Уровень освоения	Тестовые задания			Итого
	Закрытого типа		Открытого типа	
	Альтернативный выбор	Установление соответствия/последовательности	На дополнение Свободного изложения	

1.1.1 (20%)	5	2	5	1	13
1.1.2 (70 %)	27	7	28	3	65
1.1.3 (10 %)	3	1	2	1	7
Итого:	35	10	35	5	85

Критерии оценивания

Критерии оценивания тестовых заданий

Критерии оценивания: правильное выполнение одного тестового задания оценивается 1 условным баллом, неправильное – 0 баллов.

Максимальная общая сумма баллов за все правильные ответы составляет наивысший балл – 100 баллов.

Шкала оценивания результатов компьютерного тестирования обучающихся (рекомендуемая)

Оценка	Процент верных ответов	Баллы
«удовлетворительно»	70-79%	61-75 баллов
«хорошо»	80-90%	76-90 баллов
«отлично»	91-100%	91-100 баллов

Ключи ответов

№ тестовых заданий	Номер и вариант правильного ответа
1	А) Базовый протокол сети гипертекстовых ресурсов
2	А) Механизм Cookie
3	Б) сервере приложений
4	А) постоянная необходимость использовать дополнительное ПО на стороне клиента
5	В) сгенерированные страницы можно добавить в закладки
6	Б) запуск
7	А) отделение модели данных приложения, его логики и представления данных друг от друга
8	правила роутинга
9	Г) имя_класса -> имя_функции
10	В) ни ошибки, ни предупреждения не будет сгенерировано - такое использование механизма абстракции является корректным
11	А) поддерживает множественное наследование
12	Б) только к public свойству
13	Б) ответственны за запуск и обработку входящего запроса
14	Г) Id скрипта
15	В) использовать HTML или другую разметку
16	А) пре-фильтрацию
17	Б) ActiveForm::field()
18	Б) ActiveForm::dropdownlist()
19	В) разбирает URL в маршрут,

46	браузер, веб-браузер
47	private
48	final
49	Composer
50	атрибутами, атрибутом, атрибуты
51	render, render()
52	save(), save
53	виджет, виджеты, widget
54	attributelabels(),attributelabels
55	safe
56	match
57	(\$enablePrettyUrl, enablePrettyUrl
58	down(), down
59	GridView
60	DetailView
61	Active Record, ActiveRecord
62	внешние
63	urlManager
64	Модули, Модуль
65	\$content, content

	который используется при создании соответствующего экземпляра действия контроллера для обработки запроса
20	A) createUrl()
21	A) возвращает количество активных покупателей
22	A) paginate()
23	A) boolean Б) date В) compare Ж) file З) email
24	Г) DetailView Е) ListView З) GridView
25	A) class Б) dsn Г) username Д) password
26	A) field() Б) textarea() Г) passwordInput() Д) hint() Е) checkBoxList()
27	A) \$basePath Б) \$baseUrl Д) \$css Е) \$js З) \$depends
28	A) enablePrettyUrl Е) showScriptName Ж) enableStrictParsing З) rules
29	A) url Б) absoluteUrl Д) hostInfo Ж) queryString
30	A) join() Б) select() В) from() Д) having() Е) groupBy()
31	A) ArrayHelper Б) Html В) Url helper
32	A) Yii2 Б) Laravel Г) Symfony
33	Б) название таблицы, с которой идет связь
34	Б) Отложенная загрузка
35	A) login и signup

66	encode(), encode
67	beginPage(), beginPage
68	passwordInput()
69	behaviors(), behaviors
70	AppAsset
71	findOne
72	Controller
73	render
74	rules()
75	orderBy
76	Pagination
77	basePath
78	5
79	user
80	DatePicker
81	<pre>class User extends ActiveRecord { public static function tableName() { return '{{%users}}'; } public function rules() {</pre>

36	1-А, 2-Б, 3-Г
37	1-В, 2-Б, 3-А
38	1-Г, 2-Д, 3-Б, 4-А
39	1-А, 2-Б, 3-Г

	<pre> return [[['username', 'email', 'password'], 'required'], ['email', 'email'], ['email', 'unique'],]; } </pre>
82	<pre> public function saveUser(\$username, \$email, \$password) { \$user = User::findOne(['email' => \$email]); if (\$user !== null) { return false; } \$user = new User(); \$user->username = \$username; \$user->email = \$email; \$user->password = Yii::\$app- >security- >generatePasswordHash(\$password); return \$user->save(); } </pre>
83	<pre> public function deleteUser(\$id) { \$user = User::findOne(\$id); if (\$user !== null) { return \$user->delete(); } return false; } </pre>
84	<pre> public function rules() { return [[['username', 'email', 'password'], 'required'], ['username', 'match', 'pattern' => '/^[a-zA-Z0-9_]+\$/', 'message' => 'Username can only contain letters, numbers and underscores.'], ['email', 'email'], ['email', 'unique', 'targetClass' => '\app\models\User', 'message' => 'This email address has already been taken.'], ['password', 'string', 'min' => 6],]; } </pre>
85	<pre> use app\models\User; use yii\data\ArrayDataProvider; use yii\grid\GridView; use yii\db\Query; </pre>

40	1-Г, 2-Б, 3-А
41	1-Б, 2-А, 3-Б
42	1-А, 2-Б, 3-Б
43	1-Б, 2-Г, 3-А
44	1-А, 2-Б, 3-Б
45	1-Б, 2-Б, 3-Г, 4-Д, 5-Е, 6-Ж, 7-А

	<pre> \$users = (new Query()) ->select(['user.*', 'COUNT(comment.id) AS comment_count']) ->from('user') ->leftJoin('comment', 'comment.user_id = user.id') ->groupBy('user.id') ->having('COUNT(comment.id) > 5') ->all(); \$dataProvider = new ArrayDataProvider(['allModels' => \$users, 'sort' => ['attributes' => ['id', 'username', 'email', 'comment_count'],],]); echo GridView::widget(['dataProvider' => \$dataProvider, 'columns' => ['id', 'username', 'email', 'comment_count',],]); </pre>