

Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет»

Кафедра Автоматизированные системы управления  
(наименование)

**Фонд оценочных средств**

по дисциплине Web-технологии  
(наименование)

Специальность 01.03.04 Прикладная математика  
(код и наименование специальности)

Направление подготовки \_\_\_\_\_  
(код и наименование направления специальности)

Направленность (профиль) \_\_\_\_\_  
(код и наименование специализации)

Квалификация  
бакалавр

Форма обучения  
очная

Фонд оценочных средств предназначен для контроля знаний обучающихся по направлению подготовки 01.03.04 Прикладная математика

(код и наименование специальности)

по дисциплине Web- технологии

(наименование)

рабочая программа рег. № 010304-2 от 26.03.2021г.

Фонд оценочных средств рассмотрен и утвержден на заседании кафедры «Автоматизированные системы управления» протокол № 1 от «30» 08 20 22 г.

Заведующий кафедрой

  
подпись

А.И. ЯКИМОВ  
И.О. Фамилия

Исполнители

ст. преподаватель  
должность

  
подпись

Н.В.Выговская  
И.О. Фамилия

СОГЛАСОВАНО:

Декан  
электротехнического факультета

  
подпись

С.В. Болотов  
И.О. Фамилия

Заведующий кафедрой ПОИТ

  
подпись

В.В.Кутузов  
И.О. Фамилия

## 1 Перечень оценочных средств

Для определения качества освоения обучающимися учебного материала по дисциплине используются следующие оценочные средства:

№ п/п	Оценочное средство	Краткая характеристика оценочного средства
1	Вопросы к экзамену	Перечень вопросов к экзамену
2	Практические задания к экзамену	Вопросы и задания для защиты практических работ
3	Экзаменационные билеты	Экзаменационный билет
4	Задания к лабораторным работам	Вопросы к защите лабораторных работ

## 2 Перечень компетенций, с указанием этапов их формирования в процессе освоения дисциплины

№ п/п	Уровни сформированности компетенции	Содержательное описание уровня	Результаты обучения
ПК-3 - Способен осуществлять непосредственное руководство процессами разработки программного обеспечения.			
3.6 Способен применять знание Web-программирования при руководстве разработкой программного кода.			
1	Пороговый уровень	Понимает назначение и базовые элементы HTML, CSS и JavaScript и принципы руководства процессами разработки программного обеспечения..	Разработка кода HTML с ошибками CSS и JavaScript
2	Продвинутый уровень	Понимает назначение и базовые элементы HTML, CSS и JavaScript и библиотеки jQuery, умеет их использовать и осуществлять непосредственное руководство процессами разработки программного обеспечения .	Разработка кода HTML с CSS и JavaScript для несложного сайта
3	Высокий уровень	<i>Понимает назначение и базовые элементы HTML, CSS и JavaScript и умеет их использовать. Умеет разрабатывать интерактивные сайты и функциональные веб-приложения с возможностями jQuery и осуществлять непосредственное руководство процессами разработки программного обеспечения на профессиональном уровне.</i>	<i>Разработка сайтов и веб-приложений различного уровня сложности по требованию.</i>

## 3. Задания для диагностической работы, необходимые для оценки планируемых результатов обучения по дисциплине (оценочные средства). Описание показателей и критериев оценивания компетенций, описание шкал оценивания.

ПК-3 - Способен осуществлять непосредственное руководство процессами разработки программного обеспечения.

### Закрытые тесты

<p><b>1. Язык программирования php нашел наиболее широкое применение в:</b></p>	<p>- Автоматизированном проектировании  <b>+ Разработке web-приложений</b>  - Создании и управлении базами данных  Нет ответа</p>
<p><b>2. Какими разделителями окружены скрипты php скрипта?</b></p>	<p>+ &lt;? php ... ?&gt;  - &lt;php&gt; ... &lt;/php&gt;  - &lt;/&gt; php &lt;/&gt;  Нет ответа</p>
<p><b>3. Используя синтаксис языка php, напишите "Hello, World!":</b></p>	<p>- printf "Hello, World!"  - &lt;p&gt; Hello, World! &lt;/p&gt;  <b>+ echo "Hello, World!"</b></p>
<p><b>4. С какого символа начинаются все переменные в php?</b></p>	<p>- &lt;&gt;  + \$  - !  Нет ответа</p>
<p><b>5. К какому языку программирования синтаксически наиболее близок php?</b></p>	<p><b>+ С и Perl</b>  - .Net  - VBScript  Нет ответа</p>
<p><b>6. Перечислите основные типы данных в php:</b></p>	<p>- Boolean, float, logic, text, subject, NULL  - String, boolean, text, massive, logic, NULL  <b>+ Boolean, integer, float, string, array, object, resource, NULL</b>  Нет ответа</p>
<p><b>7. В php используется ... типизация данных.</b></p>	<p><b>+ Динамическая</b>  - Синтаксическая  - Коммутационная  Нет ответа</p>
<p><b>8. Объясните смысл кода, изображенного на картинке:</b></p>	<p>- Оператор исполнения выполнит строку, заключенную в кавычки, с сохранением результата в буфере оперативной памяти  - Переменной с именем output присваивается значение, заключенное в кавычки  <b>+ Оператор исполнения выполнит строку, заключенную в кавычки, а затем возвратит полученный результат</b>  Нет ответа</p>
<p><b>9. Как правильно включить файл "time.inc"?</b></p>	<p><b>+ &lt;? php include "time.inc"; ?&gt;</b>  - &lt;/&gt; php include 'time.inc' &lt;&gt;  - &lt;!-- include file="time.inc" --&gt;  Нет ответа</p>
<p><b>10. Укажите правильный способ создания функции в php:</b></p>	<p>- create newFunction()  - new_function newFunction()  <b>+ function newFunction()</b>  Нет ответа</p>
<p><b>11. Укажите неверно заданное имя:</b></p>	<p><b>+ \$my-Var</b>  - \$myVar  - \$my_Var  Нет ответа</p>
<p><b>12. Укажите верно заданный массив с использованием синтаксиса языка php:</b></p>	<p>- \$months = array["September", "October", "November"]</p>

	<ul style="list-style-type: none"> <li>- \$months = "September", "October", "November"</li> <li>+ \$months = array("September", "October", "November")</li> </ul> <p>Нет ответа</p>
13. В php файлы cookie задаются следующим образом:	<ul style="list-style-type: none"> <li>+ setcookie()</li> <li>- makecookie()</li> <li>- createcookie()</li> </ul> <p>Нет ответа</p>
14. Для написания комментариев в пределах работающего кода на php используется запись вида:	<ul style="list-style-type: none"> <li>+ /*...*/</li> <li>- &lt;!--...--&gt;</li> <li>- &lt;comment&gt;...&lt;/comment&gt;</li> </ul> <p>Нет ответа</p>
15. Как правильно записать добавление 1 (единицы) к переменной \$count?	<ul style="list-style-type: none"> <li>- \$count+1</li> <li>+ \$count++</li> <li>- \$count+=+1</li> </ul> <p>Нет ответа</p>
16. В php суперглобальная переменная, содержащая информацию о местоположении скриптов, путях, заголовках – это:	<ul style="list-style-type: none"> <li>+ \$_SERVER</li> <li>- \$_GLOBALS</li> <li>- \$_GET</li> </ul> <p>Нет ответа</p>
17. Укажите оператор, написанный на языке php, который делал бы файл "filename.txt" доступным для чтения.	<ul style="list-style-type: none"> <li>+ fopen ("filename.txt","r")</li> <li>- open ("filename.txt","read")</li> <li>- fopen ("filename.txt","r+")</li> </ul> <p>Нет ответа</p>
18. Каждый php-оператор должен заканчиваться:	<ul style="list-style-type: none"> <li>- Двоеточием (:)</li> <li>+ Точкой с запятой (;)</li> <li>- Знаки препинания в php не используются</li> </ul> <p>Нет ответа</p>
19. Какой способ использования тегов рекомендован для языка программирования php?	<ul style="list-style-type: none"> <li>- &lt;?</li> <li>- &lt;php</li> <li>+ &lt;? Php</li> </ul> <p>Нет ответа</p>
20. Выберите верное утверждение, характеризующее множественное наследование в php.	<ul style="list-style-type: none"> <li>- Множественное наследование задается через подключение родственных классов. Для этого используется функция _multi()</li> <li>+ В php не существует понятия «множественное наследование», т.е. у класса может быть лишь один родитель. Эмуляция множественного наследования задается функцией _call () или трейтами</li> <li>- Организовать наследование позволяет метод _constructor(), создающий родственные экземпляры базового класса</li> </ul> <p>Нет ответа</p>

### Открытый тест

1 Пять базовых типов данных в PHP	bool, int, float, string, array, object, callable, mixed, resource, null
-----------------------------------	--

<b>2 Что такое рекурсия?</b>	Рекурсия (recursion) — это поведение функции, при котором она вызывает сама себя
<b>3 Какая система типов используется в PHP</b>	Целые числа, числа с плавающей точкой двойной точности, логические значения, строковые значения, массивы, объекты, значения NULL и ресурсы
<b>4 Какие есть основные модификаторы доступа в PHP</b>	Public, protected, private
<b>5 Что делает оператор yield.</b>	Оператор "yield" можно рассматривать как оператор "return", за исключением того, что вместо прекращения работы функции, "yield" только приостанавливает её выполнение и возвращает текущее значение, и при следующем вызове функции она возобновит выполнение с места, на котором прервалась.
<b>6 Области применения PHP</b>	Создание скриптов для выполнения на стороне сервера. Создание скриптов для выполнения в командной строке. ... Создание оконных приложений, выполняющихся на стороне клиента
<b>7 Встроенные функции PHP</b>	echo(), print(), date().
<b>8 Что такое ссылка в PHP</b>	Ссылки в PHP — это способ получить доступ к одной и той же переменной (области памяти) под разными именами.
<b>9 Что такое динамическая переменная?</b>	Динамической переменной считается та, имя которой хранится в самой переменной.
<b>10 Какими способами можно перенаправить страницу в PHP?</b>	Используя функцию PHP header() Используя JavaScript
<b>11 Что такое тип данных?</b>	Тип данных – это описание, определяющее свойства и порядок обработки данных. Например, фраза “переменная \$str имеет тип данных “строка” означает, что в этой переменной может содержаться любое число символов, а операция сложения строк представляет собой последовательное соединение слагаемых строк в одну.
<b>12 Сколько типов данных в PHP?</b>	PHP поддерживает 8 базовых типов данных
<b>13 Что такое static функция?</b>	Static принадлежит классу, а не экземпляру класса. И вызывается у класса, а не у объекта, т.е. напрямую.
<b>14 Есть ли разница между self и this в php?</b>	self используется для статических функций и членов класса, а this наоборот для нестатических.
<b>15 Что такое конструктор?</b>	В PHP (начиная с версии 5) конструктор – это метод __construct(), который автоматически вызывается ключевым словом new после создания объекта. Обычно он используется для выполнения различных автоматических инициализаций, как

	например, инициализация свойств. Конструкторы также могут принимать аргументы, в этом случае, когда указано выражение new, необходимо передать конструктору формальные параметры в круглых скобках.
<b>16 Обязательно ли писать ?&gt; в конце скрипта?</b>	Нет
<b>17 Поддерживает ли PHP множественное наследование?</b>	Нет
<b>18 Что делает функция eval() ?</b>	eval() вычисляет строку как PHP-код
<b>19 Чем отличается цикл while от do while?</b>	do-while всегда выполняет тело цикла хотя бы один раз, поскольку его условное выражение проверяется в конце цикла.
<b>20 Как перевернуть массив? Есть массив array('h', 'e', 'l', 'l', 'o'), как из него получить array('o', 'l', 'l', 'e', 'h')?</b>	Для этого в PHP есть функция array_reverse().

**4 Задания для текущего контроля и промежуточной аттестации, необходимые для оценки планируемых результатов обучения по дисциплине (оценочные средства). Описание показателей и критериев оценивания компетенций, описание шкал оценивания**

#### **Перечень вопросов к экзамену**

1. Описание возможностей PHP. История PHP.
2. Области применения PHP.
3. Способы использования (как PHP встраивается в HTML-код и простейшие примеры.)
4. Создание скрипта на PHP. Установка и настройка программного обеспечения, необходимого для работы с PHP.
5. Комментарии, переменные, константы на PHP.
6. Типы данных PHP.
7. Операции языка PHP.
8. Условные операторы IF и SWITCH
9. Операторы циклов while, do while, for, foreach
10. Операторы передачи управления break и continue, включения include.
11. Протокол HTTP и способы передачи данных на сервер.
12. Использование HTML-форм для передачи данных на сервер. Методы передачи данных GET, HEAD, POST
13. Обработка запросов с помощью PHP (S\_request)
14. Функции PHP, определяемые пользователем. Аргументы функций (передача аргументов по значению и по ссылке, значение аргументов по умолчанию).
15. Возвращение значений из функций.
16. Встроенные функции PHP: echo(), print(), date().
17. Понятие о классах и объектах PHP.
18. Наследование в PHP.
19. Конструкторы объектов в PHP.
20. Создание массивов на PHP.
21. Сортировка массивов на PHP
22. Применение функции ко всем элементам массива на PHP
23. Выделение подмассива на PHP
24. Сумма элементов массива на PHP

25. Строки на PHP
26. Поиск элемента в строке
27. Выделение подстроки
28. Замена вхождения подстроки
29. Разделение и соединение строки
30. Создание файла на PHP
31. Закрытие соединения с файлом
32. Запись данных в файл
33. Чтение данных из файла
34. Проверка существования файла
35. Удаление файла
36. Загрузка файла на сервер
37. Использование СУБД MySQL
38. Взаимодействие PHP и MySQL
39. Построение интерфейса для добавления информации в БД
40. Отображение данных, хранящихся в MySQL
41. Авторизация доступа на PHP с помощью сессий
42. Механизм сессий Cookies
43. Настройка сессий
44. Работа с сессиями
45. Безопасность на PHP
46. Понятие регулярного выражения
47. Регулярные выражения в PHP
48. Синтаксис регулярных выражений
49. Работа с сервером MySQL
50. Команды языка SQL
51. Работа PHP с MySQL

**Пример экзаменационного билета**

Межгосударственное образовательное учреждение высшего образования

Белорусско-Российский университет

(название УВО)

Кафедра «Автоматизированные системы управления»

Экзаменационный билет № □1□

Специальность 01.03.04 Прикладная математика

Дисциплина «Основы web-программирования»

- 1 Строки на PHP.
- 2 Наследование в PHP.

Заведующий кафедрой \_\_\_\_\_ А.И. Якимов Преподаватель \_\_\_\_\_

Дата утверждения \_\_\_\_ 20\_\_ г., протокол №\_\_



## Вопросы к защите лабораторных работ

1. Назовите простые типы данных, поддерживаемые в PHP.
2. Что такое инкремент и декремент, в чем разница между префиксным и постфиксным инкрементом и декрементом?
3. Что такое рекурсия?
4. В чем разница между =, == и ===?
5. Какие знаете принципы ООП?
6. Какая система типов используется в PHP? Опишите плюсы и минусы.
7. Чем отличаются ключевые слова: include и require, mysql\_connect и mysql\_pconnect?
8. Что такое интерфейсы? Используете ли вы их? Если да — расскажите об этом.
9. Что такое абстрактный класс и чем он отличается от интерфейса?
10. Может ли абстрактный класс содержать частный метод?
11. Какие модификаторы видимости есть в PHP?
12. Какие магические методы вы знаете и как их применяют?
13. Что такое генераторы и как их использовать?
14. Что делает оператор yield?
15. Что такое traits? Альтернативное решение? Приведите пример.
16. Опишите поведение при использовании traits с одинаковыми именами полей и / или методов.
17. Будут ли доступны частные методы trait в классе?
18. Можно ли компоновать traits в trait?
19. Расскажите об обработке ошибок и исключения (try catch, finally и throw).
20. Что такое type hinting, как работает, зачем нужен?
21. Что такое namespaces и зачем они нужны?
22. Сравнение значений переменных в PHP и подводные камни? Приведение типов. Что изменилось в PHP 8 в этом контексте?
23. Как работает session в PHP, где хранится, как инициализируется?
24. Суперглобальные массивы. Какие знаете? Как использовали?
25. Сравните include vs required, include\_once vs required\_once.
26. Что означает сложность алгоритма?
27. Что такое замыкание в PHP? Приведите пример.
28. В чем разница между замыканием в PHP и JavaScript?
29. Что такое позднее связывание? Расскажите о поведении и применении static.
30. Как переопределить хранение сессий?
31. Расскажите о SPL-библиотеке (Reflection, autoload, структуры данных).
32. Расскажите о принципах SOLID.
33. Расскажите о шаблонах GRASP.
34. Расскажите о Dependency Injection: что такое DI-контейнеры? Какие есть варианты реализаций?
35. Что вам известно о MVC?
36. Что вам известно о шаблонах GoF?
37. Что вам известно о шаблонах, которые применяются в ORM?
38. Напишите на PHP пример реализации паттерна Singleton.
39. Что такое Docker? Каков принцип его работы?
40. Что такое LAMP / NAMP?
41. Расскажите о regexp.
42. Расскажите о SSH-протоколе.
43. Что такое PDO?
44. Что нового появилось в PHP 8?

45. Что такое PHP PEAR?
46. Какие версии PHP до сих пор поддерживаются?
47. В чем разница между GET и POST?
48. Чем отличаются операторы BREAK и CONTINUE?
49. Есть ли разница между одинарными и двойными кавычками?

### Вопросы и задания для защиты практических работ

1. Как передаются переменные (по значению или по ссылке)?
2. Какие процессы происходят, когда пользователь вводит в браузере URL?
3. Что такое вариативная функция или splat-оператор?
4. Что такое OWASP?
5. Какие типы уязвимостей знаете? Как от них защищаться?
6. Что такое идемпотентные методы? Какие HTTP-методы являются идемпотентными для REST?
7. Что такое stateless?
8. SOAP vs REST. В чем разница?
9. Какие методы авторизации используют для построения API?
10. Что может содержать интерфейс?
11. Класс содержит свойство, которое, в свою очередь, является объектом. Что будет содержать это свойство в клонированном объекте: ссылку на тот же дочерний объект или копию дочернего объекта? Что нужно сделать, чтобы это изменить?
12. Что такое Mock? Где используют и зачем?
13. Что такое PSR?
14. Опишите реализацию одного из шаблонов проектирования.
15. Что такое Redis?
16. Как хранятся данные в Redis / Memcached?
17. Расскажите о целесообразности применения Redis / Memcached для кэширования. Каковы плюсы и минусы?
18. Назовите различия между nginx и Apache.
19. Что такое Opcache? Как он работает?
20. Что такое JIT? Как он работает?

### Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Зачет

Оценка	Зачтено	Не зачтено
Баллы	51-100	0-50

Экзамен, дифференцированный зачет

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Баллы	87-100	65-86	51-64	0-50

### Оценивание выполнения практических заданий (пример)

Баллы	Показатели	Критерии
5	1. Полнота выполнения практического задания; 2. Своевременность выполнения задания; 3. Последовательность и	Задание решено самостоятельно. При этом составлен правильный алгоритм решения задания, в логических рассуждениях, в выборе формул и решении нет ошибок, получен верный ответ, задание решено рациональным способом.

<b>4</b>	рациональность выполнения задания; 4. Самостоятельность решения; 5. и т.д.	Задание решено с помощью преподавателя. При этом составлен правильный алгоритм решения задания, в логическом рассуждении и решении нет существенных ошибок; правильно сделан выбор формул для решения; есть объяснение решения, но задание решено нерациональным способом или допущено не более двух несущественных ошибок, получен верный ответ.
<b>3</b>		Задание решено с подсказками преподавателя. При этом задание понято правильно, в логическом рассуждении нет существенных ошибок, но допущены существенные ошибки в выборе формул или в математических расчетах; задание решено не полностью или в общем виде.
<b>0-2</b>		Задание не решено.

### Оценивание выполнения тестов (пример)

Баллы	Показатели	Критерии
<b>5</b>	1. Полнота выполнения тестовых заданий; 2. Своевременность выполнения;	Выполнено ___ % заданий предложенного теста, в заданиях открытого типа дан полный, развернутый ответ на поставленный вопрос
<b>4</b>	3. Правильность ответов на вопросы; 4. Самостоятельность тестирования;	4 Выполнено ___ % заданий предложенного теста, в заданиях открытого типа дан полный, развернутый ответ на поставленный вопрос; однако были допущены неточности в определении понятий, терминов и др.
<b>3</b>	5. и т.д.	Выполнено ___ % заданий предложенного теста, в заданиях открытого типа дан неполный ответ на поставленный вопрос, в ответе не присутствуют доказательные примеры, текст со стилистическими и орфографическими ошибками.
<b>0-2</b>		Выполнено ___ % заданий предложенного теста, на поставленные вопросы ответ отсутствует или неполный, допущены существенные ошибки в теоретическом материале (терминах, понятиях).

### Оценивание ответа на экзамене (примерное в зависимости от структуры билета)

Баллы	Показатели	Критерии
<b>10</b>	1. Полнота изложения теоретического материала; 2. Полнота и правильность решения практического задания; 3. Правильность и/или аргументированность изложения (последовательность действий); 4.	Дан полный, в логической последовательности развернутый ответ на поставленный вопрос, где он продемонстрировал знания предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно, и исчерпывающе отвечает на дополнительные вопросы, приводит собственные примеры по проблематике поставленного вопроса, решил предложенные практические задания без ошибок.
<b>9</b>		
<b>8</b>	Самостоятельность	
<b>7</b>		

<b>6</b>	ответа; 5. Культура речи; 6. и т.д.	
<b>5</b>		
<b>4</b>		
<b>0-3</b>		Дан ответ, который содержит ряд серьезных неточностей, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы, незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов, неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Выводы поверхностны. Решение практических заданий не выполнено, т.е студент не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя.

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

### **Описание показателей и критериев оценивания компетенций, описание шкал оценивания**

#### **Критерии оценки лабораторных работ**

Каждая лабораторная работа оценивается максимально на 5 баллов в зависимости от уровня знаний студента по тематике работы.

При защите лабораторной работы студент предоставляет отчет. Защита лабораторной работы проводится устно, путем ответов на вопросы по теме работы и демонстрации навыков, полученных при выполнении работы, с показом работоспособности кода. Правильные ответы оцениваются согласно оценочным уровням сформированности компетенций по изучаемой теме. Удовлетворительными считаются оценки от 3 баллов включительно.

#### **Критерии оценки контрольной работы**

Контрольная работа содержит 30 вопросов и проводится в обучающей системе Moodle в виде теста. Максимальная оценка устанавливается в 5 баллов. Оценки рассчитываются тестирующим модулем системы автоматически в зависимости от грамотности, полноты, скорости ответа (количества правильно выполненных пунктов теста). Удовлетворительными считаются оценки от 3 баллов включительно.

#### **Критерии оценки зачета**

Зачет

Оценка	Зачтено	Не зачтено
Баллы	51-100	0-50

Зачет по дисциплине «Типы и структуры данных» оценивает в комплексе работу студента: его теоретические знания по всем разделам учебной программы, умения применять полученные знания при решении практических задач.

Количество баллов за зачет рассчитывается автоматически системой контроля платформы Moodle как сумма баллов за все задания, приведенная к 40-балльной шкале.

С учетом всех видов рейтинг-контроля за семестр в качестве итогового значения рейтинга после сдачи зачета студент набирает **51-100 баллов** в случае удовлетворительного ответа:

<b>Вид рейтинг-контроля</b>	<b>Диапазон значений (в случае удовлетворительной успеваемости)</b>	<b>Суммарное значение (в случае удовлетворительной успеваемости)</b>
Промежуточный рейтинг-контроль	36-60	<b>51-100</b>

Итоговый рейтинг-контроль (текущая аттестация, зачет)	15-40	
--	-------	--

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

### 5.1 Критерии оценки лабораторных работ

Каждая выполненная и защищенная лабораторная работа оцениваются в диапазоне от 2 до 4 баллов. При этом 2 балла начисляется за выполнение работы и 1 или 2 балла за оформление отчета и защиту работы в зависимости от качества оформления и уровня знаний студента по тематике работы. Если по окончании модуля лабораторная работа выполнена, но не защищена, то баллы по ней не начисляются и она попадает в разряд задолженности.

### 5.2 Критерии оценки контрольных работ

Каждая выполненная контрольная работа оцениваются в 2 балла. При этом 2 балла начисляются за выполнение 76-100% контрольных заданий. 1 балл начисляется за выполнение 51-75% контрольных заданий. За выполнение менее 50% контрольных заданий баллы не начисляются и она попадает в разряд задолженности.

### 5.3 Критерии оценки тестовых заданий

Каждая выполненное тестовое задание оцениваются от 3 до 4 баллов. Критерием определения количества баллов является количество правильных ответов на тестовые вопросы, определяемое в процентах. За выполнение менее 50% тестовых вопросов баллы не начисляются и она попадает в разряд задолженности.

Баллы определяются по следующей формуле:

Балл (>50%) = (Макс. Балл) (%отв/100%) [Балл], где %отв – правильные ответы в процентах, 50% - допустимое значение правильных ответов, при котором итоговый рейтинг-контроль полагают успешным, Макс. Балл равен от 3 до 4 в зависимости от тестового задания.

### 5.4 Критерии оценки экзамена

Допустимые погрешности и ошибки при определении учебных достижений студентов на экзаменах с устной и / или письменной формой.

Шкала соответствия	Уровень соответствия	Баллы	Количество ошибок, погрешности / несущественные / существенные
Соответствие	Высокий	40	0/0/0
		39	1/1/0
		38	2/1/1
		37	3/2/1
	Средний	36	5/2/1
		35	6/3/1
		34	6/4/1
		33	7/1/1
		32	7/2/1
		31	7/3/1
		30	7/4/1
		29	7/1/2
	Достаточный	28	7/2/1
		27	7/2/1
		26	7/3/1

		25	7/4/1
		24	4/1/2
		23	5/2/2
		22	6/3/2
		21	6/4/2
		20	6/5/2
		19	7/1/2
		18	7/2/2
		17	7/3/2
		16	7/4/2
	Минимально необходимый	15	7/4/3
Несоответствие	Низкий	<14	8/5/4

Итоговый контроль в форме компьютерного тестирования проводится в сетевом классе. Критерием определения экзаменационной оценки является количество правильных ответов на тестовые вопросы, определяемое в процентах.

Баллы определяются по следующей методике.

1) Количество правильных ответов не более 50%:

Балл ( $<50\%$ ) =  $(\%отв - 20\%)/2$  [Балл], где %отв – правильные ответы в процентах, 20% - эмпирическое количество правильных ответов при случайном выборе.

Пример 1. Получено %отв = 46%.

Балл ( $<50\%$ ) =  $(46\% - 20\%)/2 = 13$  Баллов.

2) Количество правильных ответов более 50%:

Балл ( $>50\%$ ) =  $(\%отв - 50\%)/2 + 15$  [Балл], где %отв – правильные ответы в процентах, 50% - допустимое значение правильных ответов, при котором итоговый рейтинг-контроль полагают успешным.

Пример 2. Получено %отв = 86%.

Балл ( $>50\%$ ) =  $(86\% - 50\%)/2 + 15 = 33$  Балла.

Итоговая оценка определяется как сумма текущего контроля и промежуточной аттестации и соответствует баллам:

Экзамен

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Баллы	87-100	65-86	51-64	0-50