

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Усынин Максим Валерьевич  
Должность: Ректор  
Дата подписания: 25.12.2020 11:39:36  
Уникальный программный ключ:  
f498e59e83f65dd7c3ce7bb8a25cbbabb33ebc58

**Частное образовательное учреждение высшего образования  
«Международный Институт Дизайна и Сервиса»  
(ЧОУВО МИДиС)**

Кафедра математики и информатики

**ФОНД  
ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО  
КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ  
АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ  
ВВЕДЕНИЕ В РАЗРАБОТКУ ПРИЛОЖЕНИЙ НА NODEJS**

Направление подготовки: 09.03.03 Прикладная информатика  
Направленность (профиль): Разработка компьютерных игр и приложений с  
виртуальной и дополненной реальностью  
Квалификация выпускника: Бакалавр  
Год набора: 2021

Автор-составитель: Чеботарев С.С.

Челябинск 2024

## СОДЕРЖАНИЕ

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы .....	3
2. Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания .....	5
3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.....	6
4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.....	12

## 1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины «Введение в разработку приложений на NodeJS»  
направлен на формирование следующих компетенций:

Код и наименование компетенций выпускника	Код и наименование индикатора достижения компетенций
ПК-3 Способен проектировать и разрабатывать информационные системы в соответствии с требованиями заказчика	ПК-3.1. Выполнять действия разработке прототипов информационных систем, мобильных и Web приложений ПК-3.2. Выполнять действия по проектированию, верификации информационных систем, мобильных и Web приложений в соответствии с требованиями заказчика. ПК 3.3. Владеть инструментами и методами разработки и тестирования баз данных информационных систем
ПК-4 Способен проектировать информационные ресурсы (web, мобильных приложений) составлять формализованные описания решений, поставленных задач, в соответствии с требованиями, принятыми в организации нормативных документов	ПК-4.1. Составлять формализованные описания решений поставленных задач в соответствии с требованиями, принятыми в организации нормативных документов, выполнять действия по проектированию структур баз данных и дизайну программных интерфейсов. ПК-4.2. Использовать существующие типовые решения и шаблоны информационных ресурсов (web, мобильных приложений), применять методы и средства проектирования и дизайна информационных ресурсов, баз данных и программных интерфейсов. ПК-4.3. Применять типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке информационных ресурсов (web, мобильных приложений)
ПК-5 Способен разрабатывать мобильные приложения	ПК-5.1. Осуществлять деятельность по разработке и отладке мобильных приложений. ПК-5.2. Работать со стандартными сервисами платформ и со встроенными устройствами для получения данных, использовать технологии для работы с различными протоколами обмена данными. ПК-5.3. Применять программные средства, технологии и платформы для разработки мобильных приложений, знать основы информационной безопасности

№ п/п	Код компетенции	Наименование компетенции	Этапы формирования компетенций
1.	ПК-3	Способен проектировать и разрабатывать информационные системы в соответствии с требованиями заказчика	<p><i>1 Этап – Знать:</i> ПК-3.1. - способы разработки прототипов информационных систем, мобильных и Web приложений</p> <p><i>2 Этап – Уметь:</i> ПК-3.2. - выполнять действия разработке прототипов информационных систем, мобильных и Web приложений;</p>

			<p>- выполнять действия по проектированию, верификации информационных систем, мобильных и Web приложений в соответствии с требованиями заказчика;</p> <p>- владеть инструментами и методами разработки и тестирования баз данных информационных систем</p>
			<p><i>3 Этап – Владеть:</i> ПК-3.3. - инструментами и методами разработки и тестирования баз данных информационных систем</p>
2.	ПК-4	Способен проектировать информационные ресурсы (web, мобильных приложений) составлять формализованные описания решений, поставленных задач, в соответствии с требованиями, принятыми в организации нормативных документов	<p><i>1 Этап – Знать:</i> ПК-4.1. - формализованные описания решений поставленных задач в соответствии с требованиями, принятыми в организации нормативных документов;</p> <p>- действия по проектированию структур баз данных и дизайну программных интерфейсов</p>
			<p><i>2 Этап – Уметь:</i> ПК-4.2. - использовать существующие типовые решения и шаблоны информационных ресурсов (web, мобильных приложений);</p> <p>- применять методы и средства проектирования и дизайна информационных ресурсов, баз данных и программных интерфейсов</p>
			<p><i>3 Этап – Владеть:</i> ПК-4.3. - способами применения типовых решений, библиотек программных модулей, шаблонов, классов объектов, используемых при разработке информационных ресурсов (web, мобильных приложений)</p>
3.	ПК-5	Способен разрабатывать мобильные приложения	<p><i>1 Этап – Знать:</i> ПК-5.1. - технологии разработки и отладки мобильных приложений</p>
			<p><i>2 Этап – Уметь:</i> ПК-5.2. - работать со стандартными сервисами платформ и со встроенными устройствами для получения данных, использовать технологии для работы с различными протоколами обмена данными</p>
			<p><i>3 Этап – Владеть:</i> ПК-5.3. - программными средствами, технологиями для разработки мобильных приложений;</p>

			- основами информационной безопасности
--	--	--	--

## 2. ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

№ п/п	Код компетенции	Наименование компетенции	Критерии оценивания компетенций на различных этапах формирования	Шкала оценивания
1.	ПК-3	Способен проектировать и разрабатывать информационные системы в соответствии с требованиями заказчика	<p><i>1 Этап – Знать:</i> ПК-3.1. - способы разработки прототипов информационных систем, мобильных и Web приложений</p> <p><i>2 Этап – Уметь:</i> ПК-3.2. - выполнять действия разработке прототипов информационных систем, мобильных и Web приложений; - выполнять действия по проектированию, верификации информационных систем, мобильных и Web приложений в соответствии с требованиями заказчика; - владеть инструментами и методами разработки и тестирования баз данных информационных систем</p> <p><i>3 Этап – Владеть:</i> ПК-3.3. - инструментами и методами разработки и тестирования баз данных информационных систем</p>	<p>«ЗАЧТЕНО»:</p> <ol style="list-style-type: none"> <li>1. Усвоение программного материала.</li> <li>2. Умение применять основные приемы и методы обработки данных.</li> <li>3. Выполнение практических заданий и самостоятельной работы за семестр.</li> <li>4. Точность и обоснованность выводов.</li> <li>5. Точные, полные и логичные ответы на дополнительные вопросы.</li> </ol> <p>«НЕ ЗАЧТЕНО»:</p> <ol style="list-style-type: none"> <li>1. Незнание значительной части программного материала</li> <li>2. Невыполнение практических заданий и самостоятельной работы за семестр.</li> <li>3. Грубые ошибки при выполнении практических заданий и самостоятельной работы.</li> <li>4. Неумение выделить главное, сделать выводы и обобщения.</li> </ol>
2.	ПК-4	Способен проектировать информационные ресурсы (web, мобильных приложений) составлять формализованные описания решений, поставленных задач, в соответствии с требованиями,	<p><i>1 Этап – Знать:</i> ПК-4.1. - формализованные описания решений поставленных задач в соответствии с требованиями, принятых в организации нормативных документов; - действия по проектированию структур баз данных и дизайну программных интерфейсов</p> <p><i>2 Этап – Уметь:</i> ПК-4.2. - использовать существующие типовые решения и шаблоны информационных ресурсов (web,</p>	<ol style="list-style-type: none"> <li>2. Невыполнение практических заданий и самостоятельной работы за семестр.</li> <li>3. Грубые ошибки при выполнении практических заданий и самостоятельной работы.</li> <li>4. Неумение выделить главное, сделать выводы и обобщения.</li> </ol>

		принятых в организации нормативных документов	<p>мобильных приложений);</p> <p>- применять методы и средства проектирования и дизайна информационных ресурсов, баз данных и программных интерфейсов</p> <p><i>3 Этап – Владеть:</i> ПК-4.3.</p> <p>- способами применения типовых решений, библиотек программных модулей, шаблонов, классов объектов, используемых при разработке информационных ресурсов (web, мобильных приложений)</p>	5. Неправильные ответы на дополнительные вопросы.
3.	ПК-5	Способен разрабатывать мобильные приложения	<p><i>1 Этап – Знать:</i> ПК-5.1. - технологии разработки и отладки мобильных приложений</p> <p><i>2 Этап – Уметь:</i> ПК-5.2. - работать со стандартными сервисами платформ и со встроенными устройствами для получения данных, использовать технологии для работы с различными протоколами обмена данными</p> <p><i>3 Этап – Владеть:</i> ПК-5.3. - программными средствами, технологиями для разработки мобильных приложений;</p> <p>- основами информационной безопасности</p>	

**3. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

**1 ЭТАП – ЗНАТЬ**

**Тестовые вопросы**

Вопрос 1 - Какое из следующих утверждений верно о Node.js?

A - Node.js является на основе JavaScript рамки / платформа, построенная на браузере Chrome JavaScript V8 Engine.

B - Node.js используется для delevor ввода / вывода интенсивные веб - приложения, такие как потоковое видео сайтов, приложений одной странице и других веб - приложений.

C - Node.js является открытым исходным кодом и совершенно свободно использовать.

D - Все вышеперечисленное.

Ответ: D

Explanation

Вопрос 2 - Что такое Node.js?

A - Node.js является веб - сервер.

B - Node.js является на основе JavaScript рамки / платформа, построенная на браузере Chrome JavaScript V8 Engine.

C - Node.js является основой на основе Java.

D - Ни один из выше.

Ответ: B

Вопрос 3 - Все API - интерфейсы из Node.js являются.

A - Асинхронный

B - Синхронный

C - Оба выше.

D - Ни один из выше.

Вопрос 4 - Почему код, написанный на Node.js довольно быстро, хотя пишут в JavaScript?

A - Node.js конвертирует JavaScript код на основе Java кода, а затем выполнить то же самое.

B - Node.js внутренне преобразует код JavaScript на основе C кода, а затем выполнить то же самое.

C - Будучи построен на Google Chrome, V8 JavaScript Engine.

D - Ни один из выше.

Ответ: C

Вопрос 5 - Как Node на основе веб - серверы отличаются от традиционных веб - серверов?

Запрос узла на основе серверный процесс гораздо быстрее, чем традиционный сервер -.

B - на основе серверного узла использует одну поточную модель и может услуги значительно большее количество запросов, чем традиционные сервера , как Apache HTTP Server.

C - Там нет большой разницы между ними.

D - Ни один из выше.

Ответ: B

Вопрос 6 В каких из следующих областей, Node.js идеально подходит для использования?

A - ввод / вывод Приложения

B - потоковые данные Приложения

C - Data Intensive в реальном времени приложений (DIRT)

D - Все вышеперечисленное.

Ответ: D

Вопрос 7 В каких из следующих областей, Node.js не рекомендуется использовать?

A - Single Приложения Page

Приложения на основе JSON API - интерфейсы - B

C - ресурсоемкие приложения

D - Data Intensive Realtime Приложения (сop)

Ответ: C

Вопрос 8 Какая из справедливо следующее утверждение использовать модуль Node HTTP в приложения на базе узла?

A - переменная = HTTP требуется ("HTTP");

B - переменная HTTP = импорта ("HTTP");

C - пакет HTTP;

D - импорт HTTP;

Ответ:

Вопрос 9 REPL стоит.

A - Исследования Eval Программа Learn

B - Read Eval Loop печати

C - прочитать Заработать Точка Узнать

D - Read Loop Eval точки

Ответ: B

Вопрос 10 Какие из следующей команды запускает сеанс REPL?

A - \$ узел

Начальный узел \$ - B

C - \$ узел REPL

Консоль узла \$ - D

Вопрос 11 - Что такое использование Underscore переменной в REPL сессии?

A - чтобы получить последнюю команду, используемую.

B - чтобы получить последний результат.

C - для сохранения результата.

D - Ни один из выше.

Ответ: B

Вопрос 12 -Что NPM означает?

A - Node Package Manager

B - Узел Менеджер проекта

C - Новый руководитель проектов

D - Новый менеджер пакетов

Вопрос 13 - Какой из следующей команды покажет версию Node?

A - \$ NPM --version

B - \$ узел --version

C - \$ NPM GetVersion

D - \$ узел GetVersion

Ответ: B

Вопрос 14 - Какой из следующей команды покажет версию NPM?

A - \$ NPM --version

B - \$ узел --version

C - \$ NPM GetVersion

D - \$ узел GetVersion

Вопрос 15 - По умолчанию, NPM устанавливает какую - либо зависимость в локальном режиме.

A - верно

B - ложь

Вопрос 16 - По умолчанию, NPM устанавливает какую - либо зависимость в глобальном режиме.

A - верно

B - ложь

Ответ: B



Вопрос 17 - Какой из следующей команды покажет все модули, установленные по всему миру?

- A - \$ НПМ Ls -g
- B - \$ НПМ Ls
- C - \$ узловые Ls -g
- D - \$ узловые Ls

Вопрос 18 - Какой из следующей команды будут показаны все модули, установленные локально.

- A - \$ НПМ Ls -g
  - B - \$ НПМ Ls
  - C - \$ узловые Ls -g
  - D - \$ узловые Ls
- Ответ: B

Вопрос 19 - Какое из следующих утверждений о package.json?

- A - package.json присутствует в корневом каталоге любого узла приложения / модуля.
- B - package.json используется для определения свойств пакета.
- C - package.json могут быть использованы для обновления зависимостей приложения Node.
- D - Все вышеперечисленное.

Ответ: D

Вопрос 20 - Что такое обратный вызов?

- A - Обратный вызов представляет собой асинхронный эквивалент для функции.
- B - Обратный вызов представляет собой метод, в котором метод обратного вызова метода вызывающего абонента.
- C - Оба выше.
- D - Ни один из выше.

Вопрос 21 - Узел JS является однопоточен приложение, но поддерживает параллелизм.

- A - верно
- B - ложь

Ответ:

Вопрос 22 - Что из перечисленного верно по отношению к узлу.

- A - Каждый API узла JS являются асинхронными.
- B - узел является одним потоком, и использует функцию асинхронной вызовов для поддержания параллельности.
- C - Узел нить сохраняет цикл событий и всякий раз, когда любая задача получить закончена, она запускает соответствующее событие, которое сигнализирует функцию слушателя событий, чтобы выполнить.
- D - Все вышеперечисленное.

Ответ: D

Вопрос 23 - Какой из следующих вариантов предусматривает встроенные события.

- A - события
- B - обратный вызов
- C - бросок
- D - обработчик

Вопрос 24 - Какое из следующих утверждений о EventEmitter.on собственности?

- А - на имущество используется для стрельбы событие.
  - В - на имущество используется для связывания функции с событием.
  - С - на имущество используется для обнаружения обработчика событий.
  - Д - Ни один из выше.
- Ответ: В

Вопрос 25 - Какое из следующих утверждений верно в отношении EventEmitter.emit собственности?

- А - испускают свойство используется для обнаружения обработчика событий.
  - В - испускают свойство используется для связывания функции с событием.
  - С - испускают свойство используется для стрельбы событие.
  - Д - Ни один из выше.
- Ответ: С

## 2 ЭТАП – УМЕТЬ

### Лабораторные работы

- Лабораторная работа № 1. Инсталляция и ознакомление с Node.js
- Лабораторная работа № 2. Основы работы с модулями в Node.js
- Лабораторная работа № 3. Создание собственного модуля
- Лабораторная работа № 4. Приёмы работы с модулями.
- Лабораторная работа № 5. Введение в NPM - менеджер пакетов для Node.js.
- Лабораторная работа № 6. Структура пакета NPM.
- Лабораторная работа № 7. Глобальные модули.
- Лабораторная работа № 8. Модуль util и наследование.
- Лабораторная работа № 9. Модуль console.
- Лабораторная работа № 10. Наследование от ошибок Error.
- Лабораторная работа № 11. Чтение параметров из командной строки и окружения.
- Лабораторная работа № 12. События, EventEmitter и утечки памяти.
- Лабораторная работа № 13. Работа с базой данных MYSQL в PHP. Авторизация пользователя
- Лабораторная работа № 14. Проверка безопасности веб-приложения
- Лабораторная работа № 15. Разработка веб-приложений с использованием AJAX

## 3 ЭТАП – ВЛАДЕТЬ

### Контрольные вопросы:

Раздел 1. Введение

Тема 1.1. Фундаментальный node.js

1. Настройка окружения, редактора, инструменты для разработки и отладки.
2. Модули: организация приложения.
3. Встроенные модули http, path, fs, events и другие.
4. Жизнь и смерть Node.JS-процесса, важные особенности работы Event Loop в примерах.
5. Потоки в Node.JS, частые ошибки при работе с ними.

Тема 1.2. Сервер на node.js

1. Создание чат-сервера на Node.JS, частые ошибки.
2. Тестирование с использованием mocha, supertest и других фреймворков.

### Тема 1.3. Асинхронный код

1. Асинхронность через callback'и, модуль async (old school).
2. Promises, Iterators, Generators, их особенности в Node.JS.
3. Объединение Promises + Generators, "плоский" асинхронный код через библиотеку "co".

## Раздел 2. Архитектура

### Тема 2.1. Современный веб-сервис

1. Фреймворк Коа.JS: основы, роутинг.
2. Конфигурация через модуль config.
3. Коа.JS: структура middleware, готовые middleware – какие внешние модули для чего использовать.
4. Правильное логирование и обработка ошибок.

### Тема 2.2. Работа с базой данных

1. База MongoDB, её особенности.
2. Объектно-ориентированная работа с MongoDB через Mongoose.
3. Полная картина работы с базой: запросы, схемы, валидация, плагины и middleware.

### Тема 2.3. Авторизация с passport.js

1. Модель пользователя, регистрация. Сессии с Node.JS.
2. Passport.JS: сериализация, стратегии, авторизация.
3. CSRF-защита от взлома для форм и AJAX

### Тема 2.4. Архитектура проекта

1. Архитектура HMVC для Коа.JS.
2. Организация шаблонов, тестов, клиентских скриптов.

### **Темы групповых и/или индивидуальных творческих проектов по дисциплине «ВВЕДЕНИЕ В РАЗРАБОТКУ ПРИЛОЖЕНИЙ НА NODEJS»**

1. Автоматизированная система контроля посещений учебного заведения.
2. Автоматизированная система управления персональными данными.
3. АРМ «Деканат».
4. Геометрия задач линейного программирования.
5. Исследование информационной проводимости социальных сетей.
6. Применение динамического программирования для решения экстремальных задач.
7. Автоматизированная система управления в организации железнодорожных перевозок.
8. Разработка и использование сетевой тестовой оболочки.
9. Автоматизированная система управления в организации Каталога фильмов
10. Разработка Гостевой книги
11. Автоматизированная система управления в организации кинотеатра
12. Поддержка электронного документооборота в организации

### **Вопросы к зачету**

1. Фундаментальный Node.JS
2. Настройка окружения, редактора, инструменты для разработки и отладки.
3. Модули: организация приложения
4. Встроенные модули http, path, fs, events и другие.
5. Жизнь и смерть Node.JS-процесса, важные особенности работы Event Loop в примерах.
6. Поток в Node.JS, частые ошибки при работе с ними.
7. Сервер на Node.JS.

8. Создание чат-сервера на Node.JS, частые ошибки.
9. Тестирование с использованием mocha, supertest и других фреймворков.
10. Асинхронный код
11. Асинхронность через callback'и, модуль async (old school).
12. Promises, Iterators, Generators, их особенности в Node.JS
13. Объединение Promises + Generators, "плоский" асинхронный код через библиотеку "co"..
14. Раздел 2. Архитектура
15. Современный веб-сервис.
16. Конфигурация через модуль config.
17. Кoa.JS: структура middleware, готовые middleware – какие внешние модули для чего использовать.
18. Правильное логирование и обработка ошибок
19. Работа с базой данных
20. База MongoDB, её особенности.
21. Объектно-ориентированная работа с MongoDB через Mongoose.
22. Полная картина работы с базой: запросы, схемы, валидация, плагины и middleware.
23. Авторизация с Passport.JS
24. Модель пользователя, регистрация.
25. Сессии с Node.JS.
26. Passport.JS: сериализация, стратегии, авторизация.
27. CSRF-защита от взлома для форм и AJAX..
28. Архитектура проекта.
29. Архитектура HMVC для Koa.JS.
30. Организация шаблонов, тестов, клиентских скриптов.

#### 4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

### 1 ЭТАП – ЗНАТЬ

#### Методические рекомендации по решению тестовых заданий

1. Отвечая на вопрос с одним правильным ответом, щелкните на радиокнопке рядом с правильным ответом.
2. Отвечая на вопрос с несколькими правильными вариантами ответа, щелкните на чекбоксах рядом со всеми правильными ответами.
3. Отвечая на вопросы, требующие определить правильную последовательность, впишите по порядку номера в поле ввода рядом с ответом.
4. Вопросов в тесте – 30.
5. Время на выполнение теста – 40 минут.

#### Критерии оценивания решения тестовых заданий

Оценка	Правильность (ошибочность) выполнения задания
«отлично»	91-100% правильных ответов
«хорошо»	74-90% правильных ответов
«удовлетворительно»	59-73% правильных ответов
«неудовлетворительно»	58% и менее правильных ответов

## 2 ЭТАП – УМЕТЬ

### *Критерии оценивания лабораторных работ*

Оценка	Правильность (ошибочность) выполнения задания
«работа зачтено»	Использованы все графические ресурсы. Выполнены все этапы технического задания
«решение не зачтено»	Не выполнены этапы технического задания

## 3 ЭТАП – ВЛАДЕТЬ

### *Критерии оценки ответов студентов на контрольные вопросы*

Оценка	Правильность (ошибочность) выполнения задания
«отлично»	полно и аргументировано отвечает по содержанию задания; понимает материал, обосновывает свои суждения, применяет знания на практике, иллюстрирует ответ примерами не только по предложенной литературе; излагает материал последовательно и правильно.
«хорошо»	полно и аргументировано отвечает по содержанию задания; понимает материал, обосновывает свои суждения, применяет знания на практике, иллюстрирует ответ примерами не только по предложенной литературе; излагает материал последовательно и правильно, но допускает 1-2 ошибки, которые исправляет сам.
«удовлетворительно»	знает и понимает основные положения по содержанию задания; излагает материал неполно, но допускает неточности в определении понятий или формулировке правил; не достаточно глубоко и доказательно обосновать свои суждения иллюстрирует ответ примерами только по предложенной литературе; излагает материал непоследовательно и допускает 3-4 ошибки.
«неудовлетворительно»	допускает существенные ошибки в формулировке определений и правил, искажающие их смысл; излагает материал непоследовательно, неуверенно и не по существу задания; допускает существенные ошибки, не позволяющие раскрыть смысл задания, являющиеся серьезным препятствием к успешному овладению следующим материалом.

### **Критерии оценивания выполнения проектов**

Оценка «отлично» выставляется студенту, если он выполнил работу на 90%, при защите работы студент показывает глубокое знание вопросов темы.

Оценка «хорошо» выставляется студенту, если он выполнил работу на 75%, при защите работы студент без затруднений отвечает на вопросы.

Оценка «удовлетворительно» выставляется студенту, если он выполнил работу на 60% с незначительными ошибками; при защите показывает слабое знание вопросов темы, не всегда дает исчерпывающие и обоснованные вопросы на заданные вопросы, допускает существенные ошибки.

Оценка «неудовлетворительно» работа не выполнена.

### **Критерии оценивания знаний на зачете**

Оценка «ЗАЧТЕНО»:

1. Усвоение программного материала.
2. Умение применять основные приемы и методы обработки данных.
3. Выполнение практических заданий и самостоятельной работы за семестр.
4. Точность и обоснованность выводов.
5. Точные, полные и логичные ответы на дополнительные вопросы.

Оценка «НЕ ЗАЧТЕНО»:

1. Незнание значительной части программного материала
2. Невыполнение практических заданий и самостоятельной работы за семестр.
3. Грубые ошибки при выполнении практических заданий и самостоятельной работы.
4. Неумение выделить главное, сделать выводы и обобщения.
5. Неправильные ответы на дополнительные вопросы.