

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Минцаев Магомед Шавалович

Должность: Ректор

Дата подписания: 05.06.2026 10:03:16

Уникальный программный ключ:

236bcc35c296f119d6aafdc22836b21db52dbc07971a86865a582569f54304cc

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ГРОЗНЕНСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

имени академика М.Д. Миллиошчикова

«УТВЕРЖДАЮ»

Первый проректор

И.Г. Гаирабеков



РАБОЧАЯ ПРОГРАММА

дисциплины

«Облачные сервисы и серверная разработка мобильных приложений»

Направление подготовки/специальность

09.03.03 Прикладная информатика

Направленность/специализация

«Разработка мобильных и Web-приложений»

Квалификация

Бакалавр

Год начала подготовки

2026

Грозный 2026

1. Цели и задачи освоения дисциплины

Целью освоения дисциплины является подготовка обучающихся к организационно-управленческому виду деятельности по направлению подготовки 09.03.03 Прикладная информатика (профиль подготовки: Разработка мобильных и Web-приложений) посредством обеспечения этапов формирования компетенций, предусмотренных ФГОС, в части представленных ниже знаний, умений и навыков.

Задачами дисциплины является изучение понятийного аппарата дисциплины, основных теоретических положений и методов, формирование умений и привитие навыков применения теоретических знаний для решения практических и прикладных задач.

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к «Часть, формируемая участниками образовательных отношений» блока 1.

В свою очередь, данный курс, помимо самостоятельного значения, является предшествующей дисциплиной для курсов: Продвинутой разработка мобильных приложений: код, архитектура, алгоритмы», «Информационная безопасность мобильных Web-приложений».

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Таблица 1

Код по ФГОС	Индикаторы достижения	Планируемые результаты обучения по дисциплине (ЗУВ)
Универсальные		
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.	УК.1.1. Выбирает источники информации, адекватные поставленным задачам. УК.1.2. Демонстрирует умение осуществлять поиск информации, рассматривать различные точки зрения для решения поставленных задач.	Знает источники информации, адекватные поставленным задачам. Умеет осуществлять поиск информации, рассматривать различные точки зрения для решения поставленных задач. Владеет методами и инструментами критического анализа информации, способен оценивать их эффективность и применять их в практических ситуациях.
Общепрофессиональные		
ОПК-1. Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального	ОПК-1.1. Применяет основы математики, вычислительной техники и программирования в профессиональной деятельности. ОПК-1.2. Решает стандартные профессиональные задачи с применением естественнонаучных и	Знает основы математики, вычислительной техники и программирования в профессиональной деятельности. Умеет решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний,

исследования в профессиональной деятельности.	общеинженерных знаний, методов математического анализа и моделирования.	методов математического анализа и моделирования. Владеет методами и инструментами экспериментального исследования, способен проводить теоретические и экспериментальные исследования профессиональной деятельности.
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1. Выбирает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. ОПК-2.2. Применяет современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.	Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. Умеет применять современные информационные технологии и программные средства, в том числе отечественного производства, для анализа данных, автоматизации процессов и решения задач профессиональной деятельности. Владеет методами настройки, оптимизации и адаптации информационных технологий и программных средств, способен обеспечивать их надежное и эффективное функционирование в рамках профессиональной деятельности.
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения.	ОПК-7.1. Применяет языки программирования и языки работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ОПК-7.2. Программирует, выполняет отладку и тестирование прототипов программно-технических комплексов задач.	Знает языки программирования и языки работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. Умеет создавать прототипы программно-технических комплексов с использованием средств разработки. Владеет навыками программирования на выбранном языке программирования и умениями использования средств разработки для создания прототипов программно-технических комплексов.

Профессиональные		
ПК – 1. Способен разрабатывать требования и проектировать программное обеспечение.	ПК-1.1. Анализирует возможности реализации требований к компьютерному программному обеспечению. ПК-1.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие. ПК-1.3. Проектирует компьютерное программное обеспечение.	Знает возможности реализации требований к компьютерному программному обеспечению. Умеет разрабатывать технические спецификации на программные компоненты и их взаимодействие. Владеет навыками проектирования компьютерного программного обеспечения.
ПК-5. Способен разрабатывать и адаптировать прикладное программное обеспечение.	ПК-5.1. Проектирует и разрабатывает прототипы ИС. ПК-5.3. Обеспечивает соответствие разработанного кода и процесса кодирования на языках программирования принятым в организации или проекте стандартам и технологиям.	Знает принципы проектирования информационных систем (ИС) и их прототипов. Умеет писать код на языках программирования в соответствии с принятыми стандартами и технологиями. Владеет навыками проектирования прототипов информационных систем с учетом требований заказчика.

4. Объем дисциплины и виды учебной работы

Таблица 2

Вид учебной работы	Всего часов/ зач.ед.	Всего часов/ зач.ед.
	ОФО 7 семестр	ЗФО 8 семестр
Контактная работа (всего)	68/1,9	20/0,6
В том числе:		
Лекции	34/0,9	10/0,3
Практические занятия		
Семинары		
Лабораторные работы	34/0,9	10/0,3
Самостоятельная работа (всего)	76/2,1	124/3,4
В том числе:		
Курсовая работа (проект)		
Расчетно-графические работы		
Индивидуальное задание	33/0,9	52/1,4
Рефераты		
Доклады		
Презентации		
<i>И (или) другие виды самостоятельной работы:</i>		
Подготовка к лабораторным работам	33/0,9	36/1
Подготовка к практическим занятиям		
Подготовка к зачету	10/0,3	36/1
Подготовка к экзамену		
Вид отчетности	Зачет	Зачет
Общая трудоемкость дисциплины	144	144
	4	4

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Таблица 3

№ п/п	Наименование раздела дисциплины по семестрам	Часы лекц. занятий ОФО	Часы лаборат. занятий ОФО	Часы лекц. занятий ЗФО	Часы лаборат. занятий ЗФО
1.	Тема 1. Архитектурные паттерны для серверных приложений.	4	4	2	2
2.	Тема 2. Управление базами данных в облачных приложениях.	4	4	2	2
3.	Тема 3. Аутентификация и авторизация в мобильных приложениях.	4	4	2	2
4.	Тема 4. Разработка API для мобильных клиентов.	4	4	2	2
5.	Тема 5. Обработка и хранение медиа-контента.	4	4	-	-
6.	Тема 6. Оптимизация производительности серверной части.	4	4	-	-
7.	Тема 7. Обработка ошибок и мониторинг серверных приложений.	4	4	2	-
8.	Тема 8. Безопасность серверных приложений в облачных средах.	6	6	-	2
	ИТОГО	34	34	10	10

5.2. Лекционные занятия

Таблица 4

Раздел	Наименование раздела дисциплины	Содержание раздела
1	Тема 1. Архитектурные паттерны для серверных приложений.	Рассмотрение различных архитектурных паттернов, таких как монолитная, микросервисная и serverless, и их применение в разработке серверной части мобильных приложений.
2	Тема 2. Управление базами данных в облачных приложениях.	Исследование различных типов баз данных, их выбор для мобильных приложений и стратегий масштабирования и резервного копирования.
3	Тема 3. Аутентификация и авторизация в мобильных приложениях.	Обзор методов аутентификации пользователей и управления доступом к ресурсам сервера, включая JWT, OAuth и другие механизмы.
4	Тема 4. Разработка API для мобильных клиентов.	Проектирование и реализация RESTful или GraphQL API для обеспечения взаимодействия между мобильными приложениями и сервером.
5	Тема 5. Обработка и хранение медиа-контента.	Исследование способов обработки и хранения изображений, видео и других медиа-контента на сервере для мобильных приложений.
6	Тема 6. Оптимизация производительности серверной части.	Рассмотрение методов оптимизации производительности серверных приложений, включая кэширование данных, асинхронные запросы и масштабирование.
7	Тема 7. Обработка ошибок и мониторинг серверных приложений.	Создание стратегий обработки ошибок и реализация мониторинга для обнаружения и устранения проблем в серверной части мобильных приложений.
8	Тема 8. Безопасность серверных приложений в облачных средах.	Анализ методов обеспечения безопасности серверных приложений, включая защиту от атак, шифрование данных и сетевую безопасность.

5.3. Практические (семинарские) занятия

Практические занятия учебным планом не предусмотрены.

5.4 Лабораторные занятия

Таблица 5

№ п/п	Наименование работы	Содержание лабораторной работы
1	Лабораторная работа 1.	<i>Цель лабораторной работы:</i> <u>«Локальные сервера»</u> JSON формат передачи данных, глубокое клонирование объектов.
2	Лабораторная работа 2.	<i>Цель лабораторной работы:</i> <u>«AJAX и общение с сервером»</u> Реализация скрипта отправки данных на сервер. Красивое оповещение пользователя.
3	Лабораторная работа 3.	<i>Цель лабораторной работы:</i> <u>«Работа с Promise (ES6)»</u> Fetch API. Методы перебора массивов.
4	Лабораторная работа 4.	<i>Цель лабораторной работы:</i> <u>«Получение данных с сервера. Async/Await (ES8)»</u> Дополнительно: Что такое библиотеки. Библиотека axios.
5	Лабораторная работа 5.	<i>Цель лабораторной работы:</i> <u>«Как сохранить данные без БД. Работа с localStorage»</u> Создание интерфейса мобильного приложения, который адаптируется к различным устройствам и разрешениям экранов.
6	Лабораторная работа 6.	<i>Цель лабораторной работы:</i> <u>«Настройка Express + nodemon + TypeScript»</u> Установка библиотек Express. Создание модулей и контроллеров.
7	Лабораторная работа 7.	<i>Цель лабораторной работы:</i> <u>«Развёртывание проекта»</u> Развёртывание проекта на heroku, deploy.
8	Лабораторная работа 8.	<i>Цель лабораторной работы:</i> <u>«Express middleware»</u> Express middleware, chain of responsibility.

6. Самостоятельная работа студентов по дисциплине

6.1. Разработка облачного сервиса для синхронизации данных в мобильном приложении

Целью данной работы является разработка и демонстрация облачного сервиса, который обеспечивает синхронизацию данных между мобильным приложением и облачным хранилищем.

Задачи:

1. Анализ требований: Изучение требований к облачной синхронизации данных для мобильного приложения, включая типы данных, частоту синхронизации и безопасность.
2. Проектирование архитектуры: Разработка архитектуры облачного сервиса, включая определение API, выбор базы данных и принятие решений о безопасности и масштабируемости.
3. Реализация бекэнда: Написание кода серверной части облачного сервиса с

использованием подходящих технологий и фреймворков, таких как Node.js, Django или Spring Boot.

4. Интеграция с мобильным приложением: Разработка клиентской части мобильного приложения для взаимодействия с облачным сервисом, включая синхронизацию данных и обработку ошибок.

5. Тестирование и отладка: Проведение тестирования облачного сервиса и мобильного приложения для проверки функциональности, производительности и безопасности.

6. Документация: Написание документации по использованию и развертыванию облачного сервиса, а также инструкций для разработчиков мобильного приложения.

7. Оценка производительности и масштабируемости: Проведение тестов производительности и масштабируемости для оценки работы облачного сервиса при различных нагрузках.

8. Выводы и дальнейшие направления исследований: Анализ результатов работы, выявление достижений и проблем, а также предложение направлений для дальнейших исследований и улучшений.

Для выполнения данной работы можно использовать простое мобильное приложение для заметок, которое будет сохранять созданные пользователем заметки в облачном хранилище и обеспечивать синхронизацию данных между мобильным устройством и облачным сервисом.

Шаги для получения готового мобильного приложения:

1. Выбор платформы: Определите, для какой платформы вы хотите разработать мобильное приложение - Android, iOS или оба.

2. Поиск открытых исходных кодов: Используйте ресурсы для поиска открытых исходных кодов мобильных приложений. GitHub является одним из самых популярных ресурсов, где можно найти множество проектов с открытым исходным кодом. Например, вы можете искать проекты на основе ключевых слов, таких как "заметки", "todo list" и т.д.

3. Выбор и адаптация приложения: Выберите подходящее мобильное приложение для своей работы и адаптируйте его под ваши цели. Убедитесь, что лицензия приложения позволяет вам использовать его в учебных целях.

Процесс синхронизации между приложением и облачным сервисом:

1. Авторизация пользователя: Пользователь входит в приложение с помощью своих учетных данных.

2. Отправка данных на сервер: Когда пользователь создает, редактирует или удаляет заметку в мобильном приложении, эти изменения отправляются на серверный бекэнд через API.

3. Обработка данных на сервере: Серверный бекэнд получает запросы от мобильного приложения, обрабатывает их и вносит соответствующие изменения в базу данных.

4. Синхронизация с другими устройствами: Если у пользователя есть несколько устройств, подключенных к облачному сервису, изменения, внесенные на одном устройстве, должны быть синхронизированы с другими устройствами пользователя.

5. Обновление данных на мобильном устройстве: После того как сервер обновит данные, мобильное приложение получает уведомление об изменениях и обновляет свои данные в соответствии с обновленной информацией из облачного хранилища.

6. Обработка конфликтов: В случае возникновения конфликтов (например,

одновременного редактирования заметки на двух устройствах), сервер и/или мобильное приложение должны иметь механизмы разрешения конфликтов.

Таким образом, сам процесс синхронизации включает в себя взаимодействие между мобильным приложением и серверным бекэндом через API, а также обработку и согласование изменений данных между различными устройствами пользователя.

7. Оценочные средства

7.1. Вопросы к первой рубежной аттестации

1. Чем отличается монолитная архитектура от микросервисной?
2. Какие преимущества предоставляет микросервисная архитектура по сравнению с монолитной?
3. Какие недостатки могут быть у микросервисной архитектуры и как их решить?
4. Что такое серверless архитектура и какие преимущества она предоставляет?
5. Какие задачи лучше всего подходят для реализации с помощью серверless архитектуры?
6. Какие основные типы баз данных существуют, и в чем их различия?
7. Как выбрать подходящую базу данных для мобильного приложения?
8. Какие стратегии масштабирования баз данных используются в облачных приложениях?
9. Как осуществляется резервное копирование данных в облачных приложениях?
10. Какие механизмы обеспечивают высокую доступность и надежность баз данных в облачных средах?
11. Чем отличается аутентификация от авторизации?
12. Какие протоколы аутентификации используются в мобильных приложениях?
13. Как работает механизм JWT (JSON Web Token) и в каких случаях его целесообразно применять?
14. Какие преимущества предоставляет протокол OAuth для аутентификации и авторизации в мобильных приложениях?
15. Какие механизмы обеспечивают безопасность и защиту от атак при реализации аутентификации и авторизации в мобильных приложениях?

*Образец аттестационного билета
(за 1 рубежную аттестацию)*

**Грозненский государственный нефтяной технический университет
им.акад. М.Д. Миллионщикова**

Институт "Цифровой экономики и технологического предпринимательства"

Группа "ПИ" Семестр "7"

Дисциплина " Облачные сервисы и серверная разработка мобильных приложений"

Билет № 1

1. Какие механизмы обеспечивают высокую доступность и надежность баз данных в облачных средах?
2. Какие основные типы баз данных существуют, и в чем их различия?

Подпись преподавателя _____ **Подпись заведующего кафедрой** _____

Вопросы ко второй рубежной аттестации

1. Какие методы обработки изображений могут быть использованы на сервере для мобильных приложений?
2. Какие форматы хранения видео наиболее распространены при разработке серверной части для мобильных приложений?
3. Какие механизмы сжатия изображений помогают оптимизировать хранение медиа-контента на сервере?
4. Каким образом можно обеспечить безопасность хранения медиа-контента на сервере?
5. Какие аспекты следует учитывать при реализации механизма загрузки и скачивания медиа-контента в мобильном приложении?
6. Как кэширование данных может повысить производительность серверных приложений для мобильных клиентов?
7. Какие типы запросов можно асинхронизировать для улучшения производительности сервера?
8. Каким образом масштабирование серверных приложений влияет на их производительность?
9. Какие методы существуют для балансировки нагрузки и оптимизации производительности серверов?
10. Какие инструменты можно использовать для мониторинга производительности серверных приложений?
11. 11. Какие стратегии обработки ошибок могут быть применены в серверной части мобильных приложений?
12. Каким образом можно реализовать логирование для обнаружения и отслеживания ошибок в серверных приложениях?
13. Какие инструменты мониторинга существуют для обнаружения проблем в работе серверной части мобильных приложений?
14. Какие параметры следует отслеживать для обеспечения оптимальной производительности серверов?
15. Как можно автоматизировать процесс обнаружения и устранения проблем в серверных приложениях?

*Образец аттестационного билета
(за 2 рубежную аттестацию)*

**Грозненский государственный нефтяной технический университет
им.акад. М.Д. Миллионщикова**

**Институт "Цифровой экономики и технологического предпринимательства"
Группа "ПИ" Семестр "7"**

**Дисциплина "Облачные сервисы и серверная разработка мобильных приложений"
Билет № 8**

1. Как можно автоматизировать процесс обнаружения и устранения проблем в серверных приложениях?
2. Каким образом можно обеспечить безопасность хранения медиа-контента на сервере?

Подпись преподавателя _____ **Подпись заведующего кафедрой** _____

7.2. Вопросы на зачет

1. Чем отличается монолитная архитектура от микросервисной?
2. Какие преимущества предоставляет микросервисная архитектура по сравнению с монолитной?
3. Какие недостатки могут быть у микросервисной архитектуры и как их решить?
4. Что такое серверless архитектура и какие преимущества она предоставляет?
5. Какие задачи лучше всего подходят для реализации с помощью серверless архитектуры?
6. Какие основные типы баз данных существуют, и в чем их различия?
7. Как выбрать подходящую базу данных для мобильного приложения?
8. Какие стратегии масштабирования баз данных используются в облачных приложениях?
9. Как осуществляется резервное копирование данных в облачных приложениях?
10. Какие механизмы обеспечивают высокую доступность и надежность баз данных в облачных средах?
11. Чем отличается аутентификация от авторизации?
12. Какие протоколы аутентификации используются в мобильных приложениях?
13. Как работает механизм JWT (JSON Web Token) и в каких случаях его целесообразно применять?
14. Какие преимущества предоставляет протокол OAuth для аутентификации и авторизации в мобильных приложениях?
16. Какие механизмы обеспечивают безопасность и защиту от атак при реализации аутентификации и авторизации в мобильных приложениях?
15. Какие методы обработки изображений могут быть использованы на сервере для мобильных приложений?
16. Какие форматы хранения видео наиболее распространены при разработке серверной части для мобильных приложений?
17. Какие механизмы сжатия изображений помогают оптимизировать хранение медиа-контента на сервере?
18. Каким образом можно обеспечить безопасность хранения медиа-контента на сервере?
19. Какие аспекты следует учитывать при реализации механизма загрузки и скачивания медиа-контента в мобильном приложении?
20. Как кэширование данных может повысить производительность серверных приложений для мобильных клиентов?
21. Какие типы запросов можно асинхронизировать для улучшения производительности сервера?
22. Каким образом масштабирование серверных приложений влияет на их производительность?
23. Какие методы существуют для балансировки нагрузки и оптимизации производительности серверов?
24. Какие инструменты можно использовать для мониторинга производительности серверных приложений?
25. Какие стратегии обработки ошибок могут быть применены в серверной части мобильных приложений?
26. Каким образом можно реализовать логирование для обнаружения и отслеживания ошибок в серверных приложениях?

27. Какие инструменты мониторинга существуют для обнаружения проблем в работе серверной части мобильных приложений?

28. Какие параметры следует отслеживать для обеспечения оптимальной производительности серверов?

29. Как можно автоматизировать процесс обнаружения и устранения проблем в серверных приложениях?

Образец билета на зачет

**Грозненский государственный нефтяной технический университет
им.акад. М.Д. Миллионщикова**

Институт "Цифровой экономики и технологического предпринимательства"

Группа "ПИ" Семестр "7"

Дисциплина " Облачные сервисы и серверная разработка мобильных приложений "

Билет № 4

1. Какие типы запросов можно асинхронизировать для улучшения производительности сервера?
2. Как можно автоматизировать процесс обнаружения и устранения проблем в серверных приложениях?

Подпись преподавателя_____ **Подпись заведующего кафедрой**_____

7.3. Текущий контроль

В качестве оценочных средств текущего контроля используются средства контроля выполнения и защиты лабораторных работ по дисциплине. Защита лабораторной работы – ответ на контрольные вопросы после выполнения лабораторной работы.

Лабораторная работа 1.	Цель лабораторной работы: <u>«Локальные сервера»</u>
Лабораторная работа 2.	Цель лабораторной работы: <u>«AJAX и общение с сервером»</u>
Лабораторная работа 3.	Цель лабораторной работы: <u>«Работа с Promise (ES6)»</u>
Лабораторная работа 4.	Цель лабораторной работы: <u>«Получение данных с сервера. Async/Await (ES8)»</u>
Лабораторная работа 5.	Цель лабораторной работы: <u>«Как сохранить данные без БД. Работа с localStorage»</u>
Лабораторная работа 6.	Цель лабораторной работы: <u>«Настройка Express + nodemon + TypeScript»</u>
Лабораторная работа 7.	Цель лабораторной работы: <u>«Развёртывание проекта»</u>
Лабораторная работа 8.	Цель лабораторной работы: <u>«Express middleware»</u>

Лабораторная работа 1

Цель лабораторной работы: «Локальные сервера». JSON формат передачи данных, глубокое клонирование объектов.

Что такое Node.js и как она работает

До появления Node.js приложения, которые написаны на языке программирования JavaScript, можно было запускать только в браузере. С появлением платформы стало возможно писать на JavaScript не только в браузере, но и на сервере.

Платформу разработал Райан Дал, программист из Америки, в 2009 году. Она работает на движке V8, транслирующем JavaScript в машинный код. Простыми словами, Node.js — это приложение C++, которое получает на входе код JavaScript и выполняет его. Чтобы взаимодействовать с устройствами ввода-вывода на компьютере, в платформе предусмотрен собственный интерфейс на C++. Таким образом, платформа превращает специализированный скриптовый язык JavaScript в язык общего назначения, поэтому на Node.js можно писать любые компьютерные программы. Node.js позволяет пользоваться единым языком JavaScript для написания кода и на стороне клиента (Frontend), и на сервере (Backend). Эти возможности Node.js важны для разработки приложений реального времени, которые основаны на событиях.

Для чего нужен Node.js

Платформу используют frontend-разработчики, backend-разработчики и другие. Она позволяет написать программу для разных ОС: Linux, OS X и Windows, может использоваться для создания API. Также Node.js применяется для разработки кросс-платформенных приложений: например, списка задач, который должен работать на разных платформах, синхронизировать данные в реальном времени и отправлять на мобильное устройство. Node.js используется при создании сервисов с постоянным обменом информацией с пользователем: социальных сетей, онлайн-игр, чатов, систем совместной работы над проектом, онлайн-редакторов текста и пр.

Установка Node.js

Первый шаг — установка консоли. На Windows уже предустановлен терминал `cmd.exe`, но как основное «место работы» он не подходит. В качестве аналога используется эмулятор консоли, встроенный в VS Code терминал. После этого шага с официального сайта Node.js необходимо [скачать](#) последнюю версию платформы.

Загрузки

Последняя текущая версия: **18.14.2** (Содержит npm 9.5.0)

Загрузите исходный код Node.js или готовый установщик для вашей платформы и начните разработку уже сегодня.

LTS
Рекомендованный для большинства

Текущая
Последняя функциональность


Установщик Windows
node-v18.14.2-x64.msi


Установщик macOS
node-v18.14.2.pkg


Исходный код
node-v18.14.2.tar.gz

Установщик Windows (.msi)	32-bit	64-bit
Бинарный Windows (.zip)	32-bit	64-bit
Установщик macOS (.pkg)	64-bit / ARM64	
Бинарный macOS (.tar.gz)	64-bit	ARM64
Бинарный Linux (x64)	64-bit	
Бинарный Linux (ARM)	ARMv7	ARMv8
Исходный код	node-v18.14.2.tar.gz	

Рис 1 Установка Node JS

Теперь можно начать работу в Node JS. Для этого создайте файл с расширением `.js` и откройте его в VS Code. Для проверки напишем небольшую программу в виде таймера, которая будет выводить новое значение каждую секунду, пока мы не остановим программу.

```
let counter = 0;

const timer = setInterval(() => {
  counter++;
  console.log(counter);
}, 1000)
```

Для вывода результата откройте терминал и введите следующую команду:

```
node название_вашего_файла.js
```

Остановить действие программы можно сочетанием клавиш **ctrl + c**.

Создание собственного сервера в Node JS

```
const http = require('http');
```

```
const server = http.createServer((request, response) => {  
  response.write('hello world')  
  response.end()  
})  
  
server.listen(3003)
```

Для того, чтобы читать информацию на собственном сервере, нам необходимо подключить модуль, отвечающий за отправку и принятие запросов.

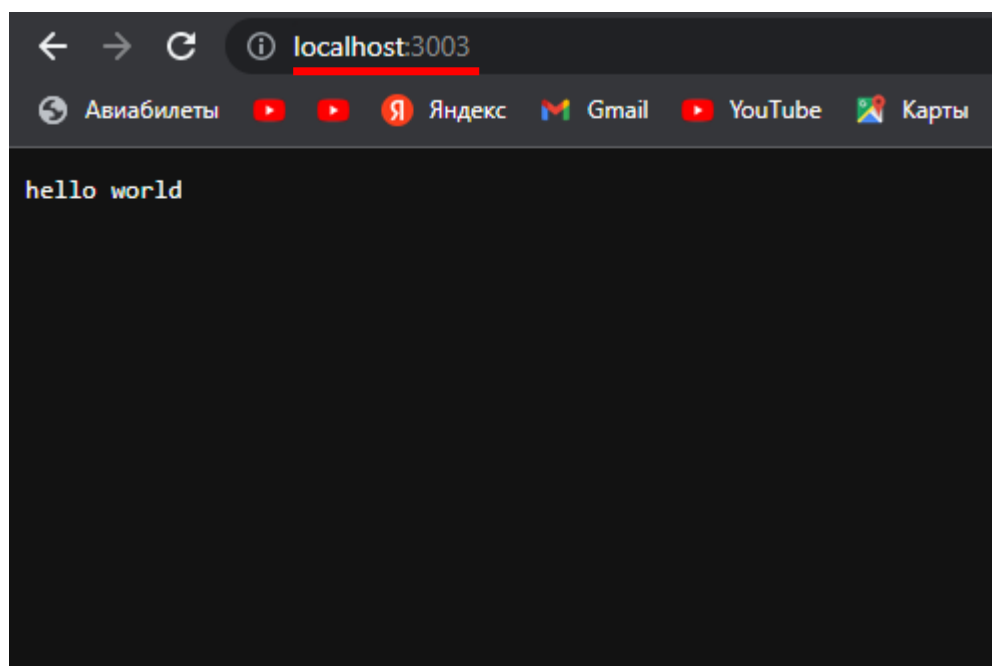
Дело в том, что Node.js состоит из блоков, называемых модулями; и каждый отдельный файл по своей сути — отдельный блок (модуль), чья область видимости изолирована от других таких же блоков (модулей) служит для подключения одного независимого модуля (файла) к другому независимому модулю (файлу). Принцип подключения через `require` заключается в создании в целевом модуле объекта и помещении в этот объект подключаемого модуля. В нашем случае этот модуль отвечает за http-запросы.

Команда `.createServer` позволяет создать новый сервер, куда мы передаем два параметра (`request` – запрос, `response` - ответ).

С помощью метода `write('hello world')` мы выводим информацию на сервер.

Для того, чтобы запустить программу необходимо прописать команду, с которым мы уже знакомы: `node` название вашего файла.js

И указать в браузере адрес сервера, который был указан в `server.listen(3003)`



Самостоятельно:

- поменяйте выводимое значение на любое другое;
- добавьте к этому значению новое, постоянно меняющееся значение с помощью таймера, который реализовали ранее;

- поменяйте выводимое значение в зависимости от URL – сервера, например:

Если URL равен <http://localhost:3003/> выводите ‘hello world’, если

<http://localhost:3003/home> выводите сообщение ‘i`ts my home address’ и т.д.

7.4. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания.

Таблица 7

Планируемые результаты освоения компетенции	Критерии оценивания результатов обучения				Наименование оценочного средства
	менее 41 баллов (неудовлетворительно)	41-60 баллов (удовлетворительно)	61-80 баллов (хорошо)	81-100 баллов (отлично)	
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.					
Знать: источники информации, адекватные поставленным задачам.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	задания для лабораторной работы, билеты рубежных аттестаций, темы докладов
Уметь: осуществлять поиск информации, рассматривать различные точки зрения для решения поставленных задач.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	
Владеть: методами и инструментами критического анализа информации, способен оценивать их эффективность и применять их в практических ситуациях.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	

ОПК-1. Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности.

Знать: основы математики, вычислительной техники и программирования в профессиональной деятельности.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	<i>задания для лабораторной работы, билеты рубежных аттестаций, темы докладов</i>
Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	
Владеть: методами и инструментами экспериментального исследования, способен проводить теоретические и экспериментальные исследования профессиональной деятельности.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	

ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности

Знать: современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	<i>задания для лабораторной работы, билеты рубежных аттестаций, темы докладов</i>
--	----------------------	-----------------	--	---------------------------------------	---

Уметь: применять современные информационные технологии и программные средства, в том числе отечественного производства, для анализа данных, автоматизации процессов и решения задач профессиональной деятельности.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	
Владеть: методами настройки, оптимизации и адаптации информационных технологий и программных средств, способен обеспечивать их надежное и эффективное функционирование в рамках профессиональной деятельности.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения.					
Знать: языки программирования и языки работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	<i>задания для лабораторной работы, билеты рубежных аттестаций, темы докладов</i>
Уметь: создавать прототипы программно-технических комплексов с использованием средств разработки.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	

Владеть: навыками программирования на выбранном языке программирования и умениями использования средств разработки для создания прототипов программно-технических комплексов.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	
ПК – 1. Способен разрабатывать требования и проектировать программное обеспечение.					
Знать: возможности реализации требований к компьютерному программному обеспечению.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	<i>задания для лабораторной работы, билеты рубежных аттестаций, темы докладов</i>
Уметь: разрабатывать технические спецификации на программные компоненты и их взаимодействие.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	
Владеть: навыками проектирования компьютерного программного обеспечения.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	
ПК-5. Способен разрабатывать и адаптировать прикладное программное обеспечение.					

Знать: принципы проектирования информационных систем (ИС) и их прототипов.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	задания для лабораторной работы, билеты рубежных аттестаций, темы докладов
Уметь: писать код на языках программирования в соответствии с принятыми стандартами и технологиями.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	
Владеть: навыками проектирования прототипов информационных систем с учетом требований заказчика	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	

8. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Для осуществления процедур текущего контроля успеваемости и промежуточной аттестации обучающихся созданы фонды оценочных средств, адаптированные для инвалидов и лиц с ограниченными возможностями здоровья и позволяющие оценить достижение ими запланированных в основной образовательной программе результатов обучения и уровень сформированности всех компетенций, заявленных в образовательной программе. Форма проведения текущей аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.). При тестировании для слабовидящих студентов используются фонды оценочных средств с укрупненным шрифтом.

На экзамен приглашается сопровождающий, который обеспечивает техническое сопровождение студенту. При необходимости студенту-инвалиду предоставляется дополнительное время для подготовки ответа на экзамене (или зачете). Обучающиеся с ограниченными возможностями здоровья и обучающиеся инвалиды обеспечиваются печатными и электронными образовательными ресурсами (программы, учебные пособия для самостоятельной работы и т.д.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

1) для инвалидов и лиц с ограниченными возможностями здоровья **по зрению:**

- **для слепых:** задания для выполнения на семинарах и практических занятиях оформляются рельефно-точечным шрифтом Брайля или в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением для слепых, либо зачитываются ассистентом; письменные задания выполняются на бумаге рельефно-точечным шрифтом Брайля или на компьютере со специализированным программным обеспечением для слепых либо надиктовываются ассистенту; обучающимся для выполнения задания при необходимости предоставляется комплект письменных принадлежностей и бумага для письма рельефно-точечным шрифтом Брайля, компьютер со специализированным программным обеспечением для слепых;

- **для слабовидящих:** обеспечивается индивидуальное равномерное освещение не менее 300 люкс; обучающимся для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; задания для выполнения заданий оформляются увеличенным шрифтом;

2) для инвалидов и лиц с ограниченными возможностями здоровья **по слуху:**

- для **глухих и слабослышащих:** обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования; предоставляются услуги сурдопереводчика;

- для **слепоглухих** допускается присутствие ассистента, оказывающего услуги тифлосурдопереводчика (помимо требований, выполняемых соответственно для слепых и глухих);

3) для лиц с тяжелыми нарушениями речи, глухих, слабослышащих лекции и семинары, проводимые в устной форме, проводятся в письменной форме;

4) для инвалидов и лиц с ограниченными возможностями здоровья, **имеющих нарушения опорно-двигательного аппарата:**

- для лиц с нарушениями опорно-двигательного аппарата, нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей: письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту; выполнение заданий (тестов, контрольных работ), проводимые в письменной форме, проводятся в устной форме путем опроса, беседы с обучающимся.

9. Учебно-методическое и информационное обеспечение дисциплины

9.1. Литература

1. Соколова В.В. Разработка мобильных приложений : учебное пособие / Соколова В.В.. — Томск : Томский политехнический университет, 2014. — 176 с. — ISBN 978-5-4387-0369-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/34706.html>
 2. Нужный А.М. Разработка мобильных приложений на языке Java с использованием Android Studio : учебное пособие / Нужный А.М., Гребенникова Н.И., Сафронов В.В.. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. — 93 с. — ISBN 978-5-7731-0906-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/111479.html>
 3. Проектирование клиент-серверных приложений : учебное пособие для проведения практических занятий / . — Ростов-на-Дону : Северо-Кавказский филиал Московского технического университета связи и информатики, 2018. — 54 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/89512.html>
- Мухамедзянов Р.Р. JAVA. Серверные приложения / Мухамедзянов Р.Р.. — Москва : СОЛОН-Р, 2016. — 336 с. — ISBN 5-93455-134-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/90352.html>

9.2. Методические указания для обучающихся по освоению дисциплины (приложение)

10. Материально-техническое обеспечение дисциплины

10.1. Материально-техническая база

Лекционная аудитория, оснащенная компьютером, видеопроекционным оборудованием, в том числе для презентаций, средствами звуковоспроизведения, экраном.

Мультимедийные средства и другая техника для презентаций учебного материала, офисный пакет программ MSWindows (MS Excel, MSWord) для оформления расчетов экономической эффективности информационных систем, OpenOfficeGoogleChrome.

10.2. Помещения для самостоятельной работы

Помещение для самостоятельной работы (Новый главный учебный корпус ФГБОУ ВО «Грозненский государственный нефтяной технический университет» 364902, Чеченская республика, г. Грозный, проспект им. Х.А. Исаева, 100. Аудитория оснащена необходимой компьютерной техникой, в наличии есть необходимое ПО: WinPro 10 RUS Upgrd OLP NL Acdmc; OfficeStd RUS OLP NL Acdmc (право на использование согласно Контракту № 267-ЭА/19 от 15.09.2019 г.).

Система ГАРАНТ (проприетарная лицензия) Visual Studio-(Freemium) 1С Предприятие договор от 02.12.2020 регистрационные номера продуктов (9334859; 9334952) Sublime Text- (открытый доступ) Notepad++ (открытый доступ).

**Методические указания по освоению дисциплины
«Облачные сервисы и серверная разработка мобильных приложений»**

1. Методические указания для обучающихся по планированию и организации времени, необходимого для освоения дисциплины.

Изучение рекомендуется начать с ознакомления с рабочей программой дисциплины, ее структурой и содержанием разделов (модулей), фондом оценочных средств, ознакомиться с учебно-методическим и информационным обеспечением дисциплины.

Дисциплина «Облачные сервисы и серверная разработка мобильных приложений» состоит из 8 связанных между собой тем, обеспечивающих последовательное изучение материала.

Обучение по дисциплине «Облачные сервисы и серверная разработка мобильных приложений» осуществляется в следующих формах:

1. Аудиторные занятия (лекции, практические занятия).
2. Самостоятельная работа студента (подготовка к лекциям, лабораторным занятиям, рефератам__и иным формам письменных работ, индивидуальная консультация с преподавателем).
3. Интерактивные формы проведения занятий (лекция-дискуссия, групповое решение кейса и др. формы).

Учебный материал структурирован и изучение дисциплины производится в тематической последовательности. Каждому практическому занятию и самостоятельному изучению материала предшествует лекция по данной теме. Обучающиеся самостоятельно проводят предварительную подготовку к занятию, принимают активное и творческое участие в обсуждении теоретических вопросов, разборе проблемных ситуаций и поисков путей их решения. Многие проблемы, изучаемые в курсе, носят дискуссионный характер, что предполагает интерактивный характер проведения занятий на конкретных примерах.

Описание последовательности действий обучающегося:

При изучении курса следует внимательно слушать и конспектировать материал, излагаемый на аудиторных занятиях. Для его понимания и качественного усвоения рекомендуется следующая последовательность действий:

1. После окончания учебных занятий для закрепления материала просмотреть и обдумать текст лекции, прослушанной сегодня, разобрать рассмотренные примеры (10 – 15 минут).
2. При подготовке к лекции следующего дня повторить текст предыдущей лекции, подумать о том, какая может быть следующая тема (10 - 15 минут).
3. В течение недели выбрать время для работы с литературой в библиотеке (по 1 часу).
4. При подготовке к практическому занятию повторить основные понятия по теме, изучить примеры. Решая конкретную ситуацию, - предварительно понять, какой теоретический материал нужно использовать. Наметить план решения, попробовать на его основе решить лабораторные задания.

2. Методические указания по работе обучающихся во время проведения лекций.

Лекции дают обучающимся систематизированные знания по дисциплине, концентрируют их внимание на наиболее сложных и важных вопросах. Лекции обычно излагаются в традиционном или в проблемном стиле. Для студентов в большинстве случаев в проблемном стиле. Проблемный стиль позволяет стимулировать активную познавательную деятельность обучающихся и их интерес к дисциплине, формировать творческое мышление, прибегать к противопоставлениям и сравнениям, делать обобщения, активизировать внимание обучающихся путем постановки проблемных вопросов, поощрять дискуссию.

Во время лекционных занятий рекомендуется вести конспектирование учебного материала, обращать внимание на формулировки и категории, раскрывающие суть того или иного явления, или процессов, выводы и практические рекомендации.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает преподаватель, отмечая наиболее важные моменты в лекционном материале замечаниями «важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек, подчеркивая термины и определения.

Целесообразно разработать собственную систему сокращений, аббревиатур и символов. Однако при дальнейшей работе с конспектом символы лучше заменить обычными словами для быстрого зрительного восприятия текста.

Работая над конспектом лекций, необходимо использовать не только основную литературу, но и ту литературу, которую дополнительно рекомендовал преподаватель. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

Тематика лекций дается в рабочей программе дисциплины.

3. Методические указания обучающимся по подготовке к лабораторным/семинарским занятиям.

На практических занятиях приветствуется активное участие в обсуждении конкретных ситуаций, способность на основе полученных знаний находить наиболее эффективные решения поставленных проблем, уметь находить полезный дополнительный материал по тематике семинарских занятий.

Студенту рекомендуется следующая схема подготовки к семинарскому занятию:

1. Ознакомление с планом практического занятия, который отражает содержание предложенной темы;

2. Проработать конспект лекций;

3. Прочитать основную и дополнительную литературу.

В процессе подготовки к практическим занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал из-за лимита аудиторных часов. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов отношение к конкретной проблеме. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса;

4. Ответить на вопросы плана практического занятия;

5. Выполнить домашнее задание;

6. Проработать тестовые задания и задачи;

7. При затруднениях сформулировать вопросы к преподавателю.

Результат такой работы должен проявиться в способности студента свободно ответить

на теоретические вопросы практикума, выступать и участвовать в коллективном обсуждении вопросов изучаемой темы, правильно выполнять практические задания и иные задания, которые даются в фонде оценочных средств дисциплины.

4. Методические указания обучающимся по организации самостоятельной работы.

Цель организации самостоятельной работы по дисциплине «Облачные сервисы и серверная разработка мобильных приложений» — это углубление и расширение знаний в области гуманитарных наук; формирование навыка и интереса к самостоятельной познавательной деятельности.

Самостоятельная работа обучающихся является важнейшим видом освоения содержания дисциплины, подготовки к практическим занятиям и к контрольной работе. Сюда же относятся и самостоятельное углубленное изучение тем дисциплины. Самостоятельная работа представляет собой постоянно действующую систему, основу образовательного процесса и носит исследовательский характер, что послужит в будущем основанием для написания выпускной квалификационной работы, практического применения полученных знаний.

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению, с учетом потребностей и возможностей личности.

Правильная организация самостоятельных учебных занятий, их систематичность, целесообразное планирование рабочего времени позволяет студентам развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивать высокий уровень успеваемости в период обучения, получить навыки повышения профессионального уровня.

Подготовка к лабораторному занятию включает, кроме проработки конспекта и презентации лекции, поиск литературы (по рекомендованным спискам и самостоятельно), решение задач из перечня лабораторных работ (текущий контроль). Лабораторное занятие – выполнение поставленных перед студентом задач с использованием ПК и специального программного обеспечения.

При подготовке к контрольной работе обучающийся должен повторять пройденный материал в строгом соответствии с учебной программой, используя конспект лекций и литературу, рекомендованную преподавателем. При необходимости можно обратиться за консультацией и методической помощью к преподавателю.

Самостоятельная работа реализуется:

- непосредственно в процессе аудиторных занятий - на лекциях, практических занятиях;
- в контакте с преподавателем вне рамок расписания - на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.
- в библиотеке, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Виды СРС и критерии оценок

(по балльно-рейтинговой системе ГГНТУ, СРС оценивается в 15 баллов)

1. Реферат/Доклад/Презентация
2. Индивидуальное задание

Темы для самостоятельной работы прописаны в рабочей программе дисциплины. Эффективным средством осуществления обучающимся самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем.

Составитель:

Доцент «ИСЭ»



/Хасухаджиев А.С-А./

СОГЛАСОВАНО:

Зав. выпускающей каф. «ИСЭ»



/Магомаева Л.Р./

Директор ДУМР



/Магомаева М.А./