

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Современные методы проектирования распределенных
приложений»

Направление подготовки 09.04.01 Информатика и вычислительная техника

Профиль Управление программным инжинирингом

Квалификация выпускника магистр

Нормативный период обучения 2 года

Форма обучения очная

Год начала подготовки 2026

Воронеж 2026

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Цель дисциплины заключается в формировании теоретических знаний и практических навыков проектирования, разработки, развертывания и сопровождения распределенных приложений с использованием современных инструментов контейнеризации, оркестрации, брокеров сообщений, NoSQL-хранилищ, кэширования и средств контроля версий.

1.2. Задачи освоения дисциплины

Задачи освоения дисциплины:

- изучить принципы построения распределенных приложений и микросервисной архитектуры;
- освоить применение Git, Docker, Docker Compose и базовых средств Kubernetes при разработке и развертывании приложений;
- сформировать навыки использования RabbitMQ для асинхронного обмена сообщениями между компонентами приложения;
- освоить применение Redis и MongoDB для хранения данных, кэширования и организации конкурентного доступа;
- научиться проектировать, реализовывать, проверять и документировать учебное распределенное приложение в рамках лабораторных работ и курсового проекта.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Современные методы проектирования распределенных приложений» относится к дисциплинам обязательной части блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Современные методы проектирования распределенных приложений» направлен на формирование следующих компетенций:

УК-4 - Способен применять современные коммуникативные технологии, в том числе на иностранном(ых) языке(ах), для академического и профессионального взаимодействия

ОПК-3 - Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями;

ОПК-8 - Способен осуществлять эффективное управление разработкой программных средств и проектов.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
УК-4	Знать специализированную терминологию дисциплины, включая русско- и англоязычные обозначения компонентов распределенных приложений: контейнер, образ, сервис, очередь сообщений, брокер, кэш, NoSQL-хранилище, оркестрация, репликация.

	Уметь подготавливать и представлять результаты лабораторных работ и курсового проекта, аргументированно описывать архитектурные решения, ход развертывания и результаты проверки распределенного приложения. Владеть навыками деловой и технической коммуникации при обсуждении архитектуры распределенного приложения, оформлении отчета, пояснительной записки и защите проектных решений.
ОПК-3	Знать основные принципы построения распределенных приложений, микросервисного взаимодействия, контейнеризации, асинхронного обмена сообщениями, применения Redis, MongoDB и RabbitMQ. Уметь выбирать и обосновывать компоненты распределенного приложения, проектировать простую архитектуру сервисов, хранилищ данных, очередей сообщений и механизмов кэширования. Владеть навыками разработки фрагментов распределенных приложений на Python с использованием Docker, Docker Compose, Redis, MongoDB, RabbitMQ и базовых средств Kubernetes.
ОПК-8	Знать современные инструментальные средства и технологии жизненного цикла распределенного приложения: Git, Docker, Docker Compose, Kubernetes, redis-cli, MongoExpress, библиотеки доступа к БД и брокерам сообщений. Уметь применять инструменты разработки, сборки, развертывания, тестирования и отладки распределенных приложений, анализировать результаты выполнения контейнеров, очередей, кэша и баз данных. Владеть навыками практического использования систем контроля версий, контейнеров, оркестрации, брокеров сообщений и NoSQL-хранилищ при создании и проверке учебного распределенного приложения.

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Современные методы проектирования распределенных приложений» составляет 5 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры	
		1	2
Аудиторные занятия (всего)	108	54	54
В том числе:			
Лекции	36	18	18
Лабораторные работы (ЛР)	72	36	36
Самостоятельная работа	45	18	27
Курсовой проект	+	+	
Часы на контроль	27	-	27

Виды промежуточной аттестации - экзамен, зачет	+	+	+
Общая трудоемкость: академические часы	180	72	108
зач.ед.	5	2	3

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Управление исходным кодом и контейнеризация приложений средствами Git, Docker и Docker Compose	Назначение системы контроля версий Git в командной разработке. Образы и контейнеры Docker. Создание Dockerfile. Сборка и запуск контейнеров. Многоконтейнерные приложения и файл docker-compose.yml. Развертывание простого web-приложения с несколькими сервисами.	6	12	6	24
2	Оркестрация контейнеризованных приложений и основы Kubernetes	Назначение оркестрации в распределенных приложениях. Кластер Kubernetes, pod, deployment, service. Развертывание Python-приложения в локальном кластере. Масштабирование экземпляров приложения и проверка доступности сервиса.	6	12	6	24
3	Асинхронное взаимодействие компонентов распределенного приложения с использованием RabbitMQ	Брокеры сообщений в распределенных приложениях. Основные элементы RabbitMQ: exchange, queue, binding, producer, consumer. Типы обменников и маршрутизация сообщений. Реализация простого обмена сообщениями между сервисами.	6	12	6	24
		Итого по первому семестру	18	36	18	72
4	Redis как in-memory NoSQL-хранилище в распределенных приложениях	Назначение NoSQL-хранилищ класса ключ-значение. Основные структуры данных Redis: string, hash, list, set, sorted set. Подключение к Redis из контейнера и из приложения. Использование Redis для хранения состояния учебного сервиса.	6	12	9	27
5	Блокировки и кэширование средствами Redis	Проблемы конкурентного доступа в распределенных приложениях. Использование Redis для реализации простых блокировок. Кэширование результатов запросов. Анализ работы приложения без кэша, с кэшем и при одновременных запросах. Предотвращение cache stampede.	6	12	9	27
6	Документно-ориентированное хранилище MongoDB и работа с ним из Python	Основные возможности MongoDB. Базы данных, коллекции и документы. Операции добавления, поиска, изменения и удаления документов. Развертывание MongoDB и MongoExpress в контейнерах. Подключение к MongoDB из Python с использованием PyMongo.	6	12	9	27
		Итого по второму семестру	18	36	27	81
Итого			36	72	45	153

5.2 Перечень лабораторных работ

1 семестр

Лабораторная работа 1. Управление исходным кодом распределенного приложения с использованием Git.

Лабораторная работа 2. Создание и развертывание Docker-образа приложения.

Лабораторная работа 3. Разработка многоконтейнерного приложения с использованием Docker Compose.

Лабораторная работа 4. Развертывание Python-приложения в кластере Kubernetes.

Лабораторная работа 5. Настройка очередей и обменников в RabbitMQ с использованием интерфейса управления.

Лабораторная работа 6. Организация обмена сообщениями между приложениями с использованием RabbitMQ.

2 семестр

Лабораторная работа 1. Моделирование структур данных распределенного приложения в Redis.

Лабораторная работа 2. Управление конкурентным доступом в Redis с использованием распределенных блокировок.

Лабораторная работа 3. Кэширование данных в Redis и предотвращение штормов кэша при массовых запросах.

Лабораторная работа 4. Развертывание и администрирование документной базы данных MongoDB в контейнерной среде.

Лабораторная работа 5. Интеграция Python-приложения с MongoDB: создание, поиск и изменение документов.

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 1 семестре для очной формы обучения.

Примерная тематика курсового проекта: «Разработка распределенного приложения на базе микросервисной архитектуры с использованием контейнеризации и брокера сообщений»

Задачи, решаемые при выполнении курсового проекта:

- анализ предметной области, постановка задачи и формирование требований к разрабатываемому распределенному приложению;
- проектирование архитектуры приложения, состава сервисов, схемы взаимодействия компонентов и структуры используемых данных;
- реализация и развертывание приложения с использованием Docker, Docker Compose или Kubernetes, а также организация обмена сообщениями между сервисами с применением RabbitMQ или Kafka.

Курсовой проект включают в себя графическую часть и расчетно-пояснительную записку.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

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

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
УК-4	Знать специализированную терминологию дисциплины, включая русско- и англоязычные обозначения компонентов распределенных приложений: контейнер, образ, сервис, очередь сообщений, брокер, кэш, NoSQL-хранилище, оркестрация, репликация.	Активная работа на занятиях, ответы на теоретические вопросы при защите лабораторных работ и курсового проекта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь подготавливать и представлять результаты лабораторных работ и курсового проекта, аргументированно описывать архитектурные решения, ход развертывания и результаты проверки распределенного приложения.	владение материалом, в том числе, на иностранном языке при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть навыками деловой и технической коммуникации при обсуждении архитектуры распределенного приложения, оформлении отчета, пояснительной записки и защите проектных решений.	владение материалом, в том числе, на иностранном языке при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-3	Знать основные принципы построения распределенных приложений, микросервисного взаимодействия, контейнеризации, асинхронного обмена сообщениями, применения Redis, MongoDB и RabbitMQ.	Активная работа на занятиях, ответы на теоретические вопросы при защите лабораторных работ и курсового проекта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь выбирать и обосновывать компоненты распределенного приложения, проектировать простую архитектуру сервисов, хранилищ	владение материалом при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	данных, очередей сообщений и механизмов кэширования.	работ и курсового проекта	в рабочих программах	рабочих программах
	Владеть навыками разработки фрагментов распределенных приложений на Python с использованием Docker, Docker Compose, Redis, MongoDB, RabbitMQ и базовых средств Kubernetes.	владение материалом при выполнении лабораторных работ и курсового проекта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-8	Знать современные инструментальные средства и технологии жизненного цикла распределенного приложения: Git, Docker, Docker Compose, Kubernetes, redis-cli, MongoExpress, библиотеки доступа к БД и брокерам сообщений.	Активная работа на занятиях, ответы на теоретические вопросы при защите лабораторных работ и курсового проекта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь применять инструменты разработки, сборки, развертывания, тестирования и отладки распределенных приложений, анализировать результаты выполнения контейнеров, очередей, кэша и баз данных.	владение материалом при выполнении лабораторных работ и курсового проекта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть навыками практического использования систем контроля версий, контейнеров, оркестрации, брокеров сообщений и NoSQL-хранилищ при создании и проверке учебного распределенного приложения.	владение материалом при выполнении лабораторных работ и курсового проекта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 1, 2 семестре для очной формы обучения по двух/четырёхбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
УК-4	Знать специализированную терминологию дисциплины, включая русско- и англоязычные обозначения компонентов распределенных приложений: контейнер, образ, сервис, очередь сообщений, брокер, кэш, NoSQL-хранилище, оркестрация, репликация.	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь подготавливать и представлять результаты лабораторных работ и курсового проекта, аргументированно описывать архитектурные решения, ход развертывания и результаты проверки распределенного приложения.	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть навыками деловой и технической коммуникации при	Решение прикладных	Продемонстрирован верный ход	Задачи не решены

	обсуждении архитектуры распределенного приложения, оформлении отчета, пояснительной записки и защите проектных решений.	задач в конкретной предметной области	решения в большинстве задач	
ОПК-3	Знать основные принципы построения распределенных приложений, микросервисного взаимодействия, контейнеризации, асинхронного обмена сообщениями, применения Redis, MongoDB и RabbitMQ.	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь выбирать и обосновывать компоненты распределенного приложения, проектировать простую архитектуру сервисов, хранилищ данных, очередей сообщений и механизмов кэширования.	Решение стандартных практических задач	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены
	Владеть навыками разработки фрагментов распределенных приложений на Python с использованием Docker, Docker Compose, Redis, MongoDB, RabbitMQ и базовых средств Kubernetes.	Решение прикладных задач в конкретной предметной области	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены
ОПК-8	Знать современные инструментальные средства и технологии жизненного цикла распределенного приложения: Git, Docker, Docker Compose, Kubernetes, redis-cli, MongoExpress, библиотеки доступа к БД и брокерам сообщений.	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь применять инструменты разработки, сборки, развертывания, тестирования и отладки распределенных приложений, анализировать результаты выполнения контейнеров, очередей, кэша и баз данных.	Решение стандартных практических задач	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены
	Владеть навыками практического использования систем контроля версий, контейнеров, оркестрации, брокеров сообщений и NoSQL-хранилищ при создании и проверке учебного распределенного приложения.	Решение прикладных задач в конкретной предметной области	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены

или

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
УК-4	Знать специализированную терминологию дисциплины, включая русско- и англоязычные обозначения компонентов	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов

	распределенных приложений: контейнер, образ, сервис, очередь сообщений, брокер, кэш, NoSQL-хранилище, оркестрация, репликация.					
	Уметь подготавливать и представлять результаты лабораторных работ и курсового проекта, аргументированно описывать архитектурные решения, ход развертывания и результаты проверки распределенного приложения.	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть навыками деловой и технической коммуникации при обсуждении архитектуры распределенного приложения, оформлении отчета, пояснительной записки и защите проектных решений.	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-3	Знать основные принципы построения распределенных приложений, микросервисного взаимодействия, контейнеризации, асинхронного обмена сообщениями, применения Redis, MongoDB и RabbitMQ.	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	Уметь выбирать и обосновывать компоненты распределенного приложения, проектировать простую архитектуру сервисов, хранилищ данных, очередей сообщений и механизмов кэширования.	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть навыками разработки фрагментов распределенных приложений на Python с использованием Docker, Docker Compose, Redis, MongoDB, RabbitMQ и	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

	базовых средств Kubernetes.			во всех задачах		
ОПК-8	Знать современные инструментальные средства и технологии жизненного цикла распределенного приложения: Git, Docker, Docker Compose, Kubernetes, redis-cli, MongoExpress, библиотеки доступа к БД и брокерам сообщений.	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	Уметь применять инструменты разработки, сборки, развертывания, тестирования и отладки распределенных приложений, анализировать результаты выполнения контейнеров, очередей, кэша и баз данных.	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть навыками практического использования систем контроля версий, контейнеров, оркестрации, брокеров сообщений и NoSQL-хранилищ при создании и проверке учебного распределенного приложения.	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию (минимум 10 вопросов для тестирования с вариантами ответов)

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Какой командой запускаются сервисы, описанные в docker-compose.yml?

- А) docker compose up
- Б) git commit
- В) kubectl get pods
- Г) redis-cli get

2. Какой командой можно посмотреть запущенные контейнеры Docker?

- А) docker ps
- Б) docker build
- В) git status
- Г) mongo show

3. Что означает запись ports: "8080:80" в Docker Compose?

- А) порт 80 контейнера доступен на порту 8080 хоста

- Б) контейнер удаляется через 80 секунд
- В) создается 8080 контейнеров
- Г) запускается Redis

4. Какую команду Redis следует использовать для записи строкового значения?

- А) GET key
- Б) SET key value
- В) HGETALL key
- Г) LPUSH key

5. Какая команда Redis читает значение по ключу?

- А) GET key
- Б) COPY key
- В) RUN key
- Г) PUBLISH key

6. Какой компонент RabbitMQ отправляет сообщения?

- А) producer
- Б) consumer
- В) volume
- Г) pod

7. Какой компонент RabbitMQ получает и обрабатывает сообщения?

- А) producer
- Б) consumer
- В) Dockerfile
- Г) branch

8. Какая библиотека Python используется для работы с MongoDB?

- А) PyMongo
- Б) pika
- В) docker-compose
- Г) kubectl

9. Какая команда Kubernetes позволяет посмотреть список pod?

- А) kubectl get pods
- Б) docker compose up
- В) git log
- Г) redis-cli

10. Что необходимо указать Python-приложению для подключения к MongoDB?

- А) адрес сервера, порт, имя базы данных и параметры авторизации
- Б) только имя Git-ветки
- В) только номер Docker-образа
- Г) только название exchange

7.2.3 Примерный перечень заданий для решения прикладных задач

1. В приложении часто повторяются запросы к MongoDB. Какой компонент целесообразно добавить для ускорения доступа?

- А) Redis-кэш

- Б) Git-ветку
- В) новый Dockerfile без приложения
- Г) пустую очередь

2. Два экземпляра сервиса одновременно изменяют один ресурс. Какой механизм поможет избежать некорректного изменения?

- А) распределенная блокировка Redis
- Б) удаление Docker Compose
- В) создание новой ветки Git
- Г) запуск MongoExpress

3. Сервис формирует заявки, а другой сервис должен обрабатывать их асинхронно. Какой компонент следует использовать?

- А) RabbitMQ
- Б) только локальный файл
- В) GitHub README
- Г) Docker volume без приложения

4. Для приложения нужны web-сервис, база MongoDB и Redis. Где удобнее описать их совместный запуск?

- А) в docker-compose.yml
- Б) в .gitignore
- В) в одном SQL-запросе
- Г) в файле отчета

5. В системе обработки документов сообщения с ошибками должны попадать в отдельную очередь. Какой подход подходит лучше всего?

- А) маршрутизация через exchange и отдельную error-queue
- Б) удаление всех сообщений
- В) хранение ошибок только в имени контейнера
- Г) отключение consumer

6. Нужно развернуть несколько экземпляров web-приложения и обеспечить доступ к ним в Kubernetes. Какие объекты наиболее уместны?

- А) Deployment и Service
- Б) Commit и Branch
- В) Collection и Document
- Г) Exchange и Binding без pod

7. В интернет-магазине оформление заказа не должно задерживать работу корзины. Как лучше организовать передачу заказа?

- А) отправить сообщение о заказе в брокер сообщений
- Б) хранить заказ только в переменной браузера
- В) пересобрать Docker-образ вручную
- Г) удалить сервис заказов

8. При промахе кэша Redis приложение должно:

- А) получить данные из основного хранилища и записать их в кэш
- Б) удалить Docker Compose
- В) остановить MongoDB
- Г) создать новую Git-ветку

9. Для проверки работоспособности распределенного приложения следует в первую очередь:

- А) убедиться, что контейнеры запущены, сервисы доступны, сообщения обрабатываются, данные сохраняются
- Б) удалить все логи
- В) изменить название дисциплины
- Г) отключить сеть контейнеров

10. В курсовом проекте студент разрабатывает приложение из нескольких сервисов. Что должно быть отражено в архитектуре?

- А) назначение сервисов, связи между ними, хранилища данных и способ развертывания
- Б) только цвет интерфейса
- В) только фамилия студента
- Г) только список литературы

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Назначение Git при разработке распределенных приложений.
2. Основные команды Git для фиксации, просмотра и анализа изменений.
3. Понятие контейнеризации. Образ и контейнер Docker.
4. Назначение Dockerfile и основные инструкции Dockerfile.
5. Сборка Docker-образа и запуск контейнера.
6. Проброс портов, переменные окружения и тома Docker.
7. Назначение Docker Compose в разработке многоконтейнерных приложений.
8. Структура файла docker-compose.yml.
9. Понятие сервиса в Docker Compose.
10. Сети и зависимости сервисов в Docker Compose.
11. Назначение Kubernetes и задачи оркестрации контейнеров.
12. Основные объекты Kubernetes: pod, deployment, service.
13. Развертывание Python-приложения в Kubernetes.
14. Масштабирование контейнеризованного приложения.
15. Брокеры сообщений в распределенных приложениях.
16. Основные элементы RabbitMQ: exchange, queue, binding.
17. Producer и consumer в RabbitMQ.
18. Типы exchange в RabbitMQ и маршрутизация сообщений.
19. Подтверждение доставки сообщений и обработка ошибок.
20. Использование RabbitMQ при проектировании распределенного приложения.

7.2.5 Примерный перечень заданий для подготовки к экзамену

1. Принципы построения распределенных приложений.
2. Микросервисная архитектура: преимущества, ограничения и области применения.
3. Контейнеризация как способ развертывания распределенных приложений.
4. Docker и Docker Compose в жизненном цикле приложения.

5. Основы оркестрации контейнеров средствами Kubernetes.
6. Асинхронное взаимодействие сервисов с использованием RabbitMQ.
7. Очереди сообщений, exchange, binding и routing key в RabbitMQ.
8. Сценарии применения брокеров сообщений в распределенных системах.
9. NoSQL-хранилища и их роль в распределенных приложениях.
10. Redis как in-memory хранилище класса ключ-значение.
11. Основные структуры данных Redis: String, Hash, List, Set, Sorted Set.
12. Использование Redis для хранения состояния приложения.
13. Проблема конкурентного доступа в распределенных приложениях.
14. Реализация распределенных блокировок средствами Redis.
15. Кэширование данных в распределенных приложениях.
16. Проблема cache stampede и способы ее предотвращения.
17. MongoDB как документно-ориентированная СУБД.
18. Базы данных, коллекции и документы MongoDB.
19. Основные операции CRUD в MongoDB.
20. Использование MongoExpress для просмотра и администрирования данных.
21. Подключение Python-приложения к MongoDB с использованием PyMongo.
22. Проектирование архитектуры учебного распределенного приложения.
23. Выбор компонентов распределенного приложения: web-сервис, брокер сообщений, кэш, база данных.
24. Контейнерное развертывание сервисов курсового проекта.
25. Проверка работоспособности распределенного приложения и анализ результатов его выполнения.

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 6 до 10 баллов

3. Оценка «Хорошо» ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» ставится, если студент набрал от 16 до 20 баллов.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Управление исходным кодом и контейнеризация приложений средствами Git, Docker и Docker Compose	УК-4, ОПК-3, ОПК -8	Тест, контрольная работа, защита лабораторных работ, защита реферата,

			требования к курсовому проекту....
2	Оркестрация контейнеризованных приложений и основы Kubernetes	УК-4, ОПК-3, ОПК -8	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....
3	Асинхронное взаимодействие компонентов распределенного приложения с использованием RabbitMQ	УК-4, ОПК-3, ОПК -8	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....
4	Redis как in-memory NoSQL-хранилище в распределенных приложениях	УК-4, ОПК-3, ОПК -8	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....
5	Блокировки и кэширование средствами Redis	УК-4, ОПК-3, ОПК -8	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....
6	Документно-ориентированное хранилище MongoDB и работа с ним из Python	УК-4, ОПК-3, ОПК -8	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....

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

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Астапчук В.А. Распределенные информационные системы и базы данных : учебное пособие / В.А. Астапчук, Д.Н. Достовалов. — Новосибирск : Новосибирский государственный технический университет, 2024. — 72 с. — ISBN 978-5-7782-5326-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/155588.html>
2. Маркелов А.А. Введение в технологии контейнеров и Kubernetes / А.А. Маркелов. — Москва : ДМК Пресс, 2019. — 194 с. — ISBN 978-5-97060-775-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/124714.html>.
3. Милл И. Docker на практике / И. Милл, Э.Х. Сейерс ; перевод Д.А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/124536.html>.
4. Лукша М. Kubernetes в действии / М. Лукша ; перевод А.В. Логунов. — Москва : ДМК Пресс, 2019. — 672 с. — ISBN 978-5-97060-657-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/125099.html>.
5. Пселтис Э.Дж. Поточковая обработка данных. Конвейер реального времени / Э.Дж. Пселтис ; перевод А.А. Слинкин. — Москва : ДМК Пресс, 2018. — 218 с. — ISBN 978-5-97060-606-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/125003.html>.
6. Маркин А.В. Постреляционные базы данных. MongoDB : учебное пособие / А.В. Маркин. — 3-е изд. — Москва : Ай Пи Ар Медиа, 2024. — 383 с. — ISBN 978-5-4497-3243-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/141284.html>.
7. Роцин П.Г. Командная разработка программного обеспечения с помощью системы контроля версий GIT: конспект лекций : учебное пособие / П.Г. Роцин. — Москва : Национальный исследовательский ядерный университет «МИФИ», 2022. — 106 с. — ISBN 978-5-7262-2846-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/132682.html>.

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

8.2.1 Перечень лицензионного и свободно распространяемого программного обеспечения:

- Windows Professional 8.1 Single Upgrade MVL A Each Academic
- Office Professional Plus 2013 Single MVL A Each Academic

Свободно распространяемое ПО:

- Visual Studio Community
- Visual Studio Code
- Oracle VM VirtualBox

Отечественное ПО:

- Архиватор 7z
- Adobe Acrobat Reader
- ОС «Альт Образование»

8.2.2 Ресурсы информационно-телекоммуникационной сети «Интернет», современные профессиональные базы данных и информационные справочные системы:

- Образовательный портал ВГТУ: <https://old.education.cchgeu.ru/>
- Электронная библиотека ВГТУ: <https://bibl.cchgeu.ru/catalog/>
- Электронно-библиотечная система IPR SMART: <https://www.iprbookshop.ru/>
- Электронно-библиотечная система «Лань»: <https://e.lanbook.com/>
- Научная электронная библиотека: www.elibrary.ru
- Информационно -справочная система ВГТУ: <https://wiki.cchgeu.ru/>
- <https://metanit.com/>
- <http://window.edu.ru>
- <https://wiki.cchgeu.ru/>
- <https://proglib.io>
- <https://msdn.microsoft.com/ru-ru/>
- <https://docs.microsoft.com/>

8.2.3 Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине:

- лекции с применением мультимедийных средств;
- обучение прикладным информационным технологиям, ориентированным на специальность, в рамках лабораторных работ с применением лицензионного программного обеспечения.

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Для проведения лекционных занятий используется аудитория или лаборатория, оснащенная проектором для проведения лекций.

Для проведения лабораторных работ используется лаборатория, укомплектованная специализированной мебелью для обучающихся и преподавателя, оборудованная техническими средствами обучения: компьютерами с лицензионным программным для проведения лабораторного практикума, с возможностью подключения к сети «Интернет» и доступом в

электронную информационно-образовательную среду университета, мультимедиа-проектором, из следующего перечня:

- 408/2 (Лаборатория разработки программных систем)
- 412/2 (Лаборатория микропроцессорной техники);
- 415/2 (Лаборатория распределённых вычислений);
- 417/2 (Лаборатория автоматизации проектирования вычислительных комплексов и сетей);
- 419/2 (Лаборатория телекоммуникационных систем).

Лаборатории расположены по адресу: 394018, г. Воронеж, Плехановская, 11 (учебный корпус №2).

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Современные методы проектирования распределенных приложений» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, защитой курсового проекта.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.

Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	----------------------------	--

Лист согласования

Автор	Олейникова Светлана Александровна	Подписано	09.07.2026 12:30:14
Заведующий кафедрой	Барабанов Владимир Федорович	Подписано	09.07.2026 12:31:54
Руководитель образовательной программы	Олейникова Светлана Александровна	Подписано	09.07.2026 12:33:27
Декан факультета	Бредихин Алексей Вячеславович	Утверждено	09.07.2026 12:35:16