

Документы
Информация о владельце: **Федеральное государственное автономное образовательное учреждение высшего образования "Самарский государственный экономический университет"**
ФИО: Кандрашина Елена Александровна
Должность: И.о. ректора ФГАОУ ВО «Самарский государственный экономический университет»
Дата подписания: 13.07.2026 14:11:33
Уникальный программный ключ:
2db64eb9605ce27edd3b8e8fdd32c70e0674ddd2

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ) «РАЗРАБОТКА РАСПРЕДЕЛЕННЫХ ПРИЛОЖЕНИЙ»

Уровень высшего образования: бакалавриат

Направление подготовки: 09.03.03 Прикладная информатика

Направленность (профиль) подготовки: Интеллектуальные цифровые системы и сервисы в управлении

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2026

Срок получения образования: 4 года

Объем:
в зачетных единицах: 3 з.е.
в академических часах: 108 ак.ч.

г. Самара, 2026

Разработчики:

Не имеет Колотилина М. А.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика, утвержденного приказом Минобрнауки от 19.09.2017 № 922, с учетом трудовых функций профессиональных стандартов: "Менеджер по информационным технологиям", утвержден приказом Минтруда России от 30.08.2021 № 588н; "Специалист по информационным системам", утвержден приказом Минтруда России от 13.07.2023 № 586н; "Руководитель проектов в области информационных технологий", утвержден приказом Минтруда России от 27.04.2023 № 369н; "Системный аналитик", утвержден приказом Минтруда России от 27.04.2023 № 367н.

Согласование и утверждение

№	Подразделение или коллегиальный орган	Ответственное лицо	ФИО	Виза	Дата, протокол (при наличии)
1	Кафедра прикладной информатики	Заведующий кафедрой, руководитель подразделения, реализующего ОП	Ермолина Л. В.	Рассмотрено	21.05.2026, № 9

1. Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины - является формирование у студентов системного понимания принципов построения, проектирования и реализации распределенных программных систем, а также приобретение практических навыков разработки масштабируемых, отказоустойчивых и безопасных приложений, функционирующих в гетерогенных сетевых средах.

Дисциплина направлена на подготовку специалистов, способных грамотно выбирать архитектурные стили, протоколы взаимодействия и методы управления данными при создании современных распределенных систем (микросервисных, событийно-ориентированных, облачных), а также оценивать компромиссы между согласованностью, доступностью и устойчивостью к сбоям.

Задачи изучения дисциплины:

- Теоретическая задача – изучить фундаментальные концепции распределенных систем: модели взаимодействия (RPC, REST, gRPC), логические и физические часы, CAP-теорему, типы согласованности данных, механизмы репликации и шардирования, а также основные архитектурные паттерны (микросервисы, событийно-ориентированная архитектура, CQRS, Saga).;

- Практико-ориентированная задача – освоить инструментарий и технологии разработки распределенных приложений: протоколы передачи данных (HTTP/2, HTTP/3, WebSocket), форматы сериализации (JSON, Protobuf, Avro), библиотеки для реализации RPC/gRPC, системы кэширования и очереди сообщений, а также научиться применять шаблоны повышения отказоустойчивости (Circuit Breaker, Retry, Bulkhead) и механизмы обнаружения сбоев.

;

- Инженерно-проектная задача – выработать навыки проектирования распределенной системы под заданные нефункциональные требования (масштабируемость, доступность, производительность, безопасность), включая выбор архитектуры, протоколов синхронизации, стратегий обработки транзакций, методов аутентификации/авторизации (JWT, OAuth2, mTLS), а также организацию логирования, трассировки и мониторинга для диагностики распределенных сценариев..

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

Компетенции, индикаторы и результаты обучения

ПК-1 Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы

ПК-1.1 Разрабатывает архитектуру и прототип ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы

Знать:

ПК-1.1/Зн1 Фундаментальные принципы построения распределенных систем (масштабируемость, отказоустойчивость, CAP-теорема); модели взаимодействия (RPC, REST, gRPC), протоколы (HTTP/2, HTTP/3, WebSocket) и форматы сериализации (JSON, Protobuf, Avro); архитектурные стили (микросервисы, событийно-ориентированная архитектура, CQRS, Saga, API Gateway); методы управления данными (репликация, шардирование, распределенные транзакции, 2PC, конечная согласованность); механизмы синхронизации времени (логические и физические часы); подходы к обеспечению отказоустойчивости (Circuit Breaker, Retry, Bulkhead, детекторы сбоев) и безопасности (JWT, OAuth2, mTLS, управление секретами).

Уметь:

ПК-1.1/Ум1 Анализировать бизнес-требования и формулировать нефункциональные требования (доступность, производительность, масштабируемость) для обоснованного выбора архитектуры распределенной ИС; выбирать протоколы взаимодействия, форматы данных и схемы интеграции компонентов; применять архитектурные паттерны для декомпозиции бизнес-процессов на сервисы; проектировать стратегии управления данными (репликация, шардирование, согласованность) с учетом требований транзакций; внедрять механизмы отказоустойчивости (таймауты, ретраи, восстановление) и безопасности (аутентификация, авторизация, защита каналов) на этапе архитектурного проектирования.

Владеть:

ПК-1.1/Нв1 Навыками разработки архитектурных схем и прототипов распределенных ИС с выбором топологии, протоколов и форматов обмена (в т.ч. с использованием gRPC и REST API); навыками декомпозиции предметной области на микросервисы и событийно-ориентированные компоненты с проектированием API-шлюзов; навыками проектирования моделей данных в распределенных системах с выбором стратегий репликации, шардирования и кэширования (Redis, Memcached); навыками применения шаблонов отказоустойчивости и обнаружения сбоев для обеспечения высокой доступности ИС; навыками организации логирования, распределенной трассировки и мониторинга для сопровождения распределенных систем; навыками внедрения базовых механизмов безопасности (настройка JWT/OAuth2, mTLS, управление секретами) в прототипах ИС.

ПК-1.2 Проектирует, создает дизайн и разрабатывает базы данных ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС

Знать:

ПК-1.2/Зн1 Принципы проектирования реляционных и нереляционных баз данных в контексте распределенных систем (нормализация, денормализация, ER-моделирование); модели данных NoSQL (ключ-значение, документная, колоночная, графовая) и их применимость для распределенных ИС; механизмы обеспечения согласованности данных (ACID/BASE, CAP-теорема, конечная согласованность); стратегии масштабирования данных (горизонтальное и вертикальное шардирование, репликация master-slave/multi-master, consistent hashing); подходы к управлению распределенными транзакциями (2PC, Saga, компенсирующие транзакции); методы оптимизации производительности (индексирование, кэширование, материализованные представления, партиционирование); основы безопасности данных на уровне БД (шифрование, маскирование, управление доступом, аудит).

Уметь:

ПК-1.2/Ум1 Анализировать предметную область и бизнес-процессы для построения концептуальных, логических и физических моделей данных распределенной ИС; выбирать тип СУБД (реляционная/NoSQL) и схему хранения в зависимости от требований к нагрузке, согласованности и доступности; проектировать распределенные схемы данных с применением шардирования и репликации, обосновывая выбор ключа шардирования и топологии реплик; разрабатывать структуры для хранения событий в событийно-ориентированных архитектурах (Event Sourcing) и очередей сообщений; проектировать модели для CQRS-систем с разделением на команды и запросы; оптимизировать запросы и структуры хранения с учетом сетевых задержек и распределенности данных; разрабатывать стратегии резервного копирования, восстановления и миграции данных в распределенной среде.

Владеть:

ПК-1.2/Нв1 Навыками создания ER-диаграмм и схем баз данных для распределенных приложений с использованием CASE-средств; навыками работы с СУБД различных типов (PostgreSQL, MySQL — для реляционных; MongoDB, Redis, Cassandra — для NoSQL) в распределенных конфигурациях; навыками настройки репликации и шардирования на примерах конкретных СУБД; навыками написания оптимизированных запросов (SQL/NoSQL) с учетом распределенного характера данных и планов выполнения; навыками применения шаблонов Saga и компенсирующих транзакций для обеспечения согласованности в распределенных бизнес-процессах; навыками организации кэширующих слоев на основе Redis/Memcached для снижения нагрузки на БД; навыками проектирования систем логирования изменений данных (CDC — Change Data Capture) и аудита для сопровождения распределенных ИС.

ПК-1.3 Реализует процесс обеспечения и контроля качества в соответствии с регламентами организации в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС

Знать:

ПК-1.3/Зн1 Современные стандарты, методологии и регламенты обеспечения качества программного обеспечения (ISO 25010, ГОСТ Р ИСО/МЭК 12207, Agile, DevOps); подходы к тестированию распределенных систем на различных уровнях (модульное, интеграционное, системное, приемочное, нагрузочное, стресс-тестирование); методы контроля качества в распределенных средах, включая тестирование отказоустойчивости (Chaos Engineering), сетевых задержек и сбоев; инструменты автоматизации тестирования для распределенных приложений (Postman, JMeter, Gatling, Selenium, Pact для контрактного тестирования); принципы организации мониторинга, логирования и распределенной трассировки (Prometheus, Grafana, ELK Stack, Jaeger, Zipkin) как инструментов контроля качества в эксплуатации; подходы к оценке нефункциональных характеристик (производительность, масштабируемость, доступность, надежность, безопасность) и методы их измерения (SLA, SLO, SLI, бюджеты ошибок); регламенты управления инцидентами, дефектами и изменениями качества в процессе сопровождения

Уметь:

ПК-1.3/Ум1 Разрабатывать стратегию и план обеспечения качества для распределенной ИС с учетом бизнес-требований и регламентов организации; определять метрики качества (SLA/SLO/SLI) для компонентов распределенной системы и настраивать их мониторинг; проектировать сценарии нагрузочного и стресс-тестирования для проверки поведения системы при пиковых нагрузках и сбоях; применять инструменты Chaos Engineering для проверки отказоустойчивости распределенного приложения; организовывать систему сбора и анализа логов и трейсов для диагностики проблем и контроля качества в реальном времени; проводить анализ результатов тестирования и мониторинга, выявлять узкие места и формировать рекомендации по улучшению качества ИС; внедрять процессы code review, статического анализа кода и автоматизированного тестирования в CI/CD-пайплайны для непрерывного контроля качества.

Владеть:

ПК-1.3/Нв1 Навыками разработки и внедрения регламентов обеспечения качества в процессы создания и сопровождения распределенных ИС; навыками использования инструментов автоматизированного тестирования (JUnit/TestNG, PyTest, Postman/Newman, JMeter, Gatling) для распределенных систем; навыками настройки систем мониторинга (Prometheus + Grafana) и логирования (ELK Stack) для контроля эксплуатационных показателей качества; навыками проведения нагрузочных испытаний и анализа их результатов с выработкой рекомендаций по оптимизации; навыками применения Chaos Engineering (инструменты Chaos Monkey, Gremlin) для проверки устойчивости распределенной системы к сбоям; навыками организации распределенной трассировки (Jaeger/Zipkin) для контроля качества взаимодействия между сервисами; навыками формирования отчетности по качеству и управления дефектами в рамках сопровождения ИС (JIRA/YouTrack).

3. Место дисциплины в структуре ОП

Дисциплина (модуль) «Разработка распределенных приложений» относится к формируемой участниками образовательных отношений части образовательной программы и изучается в семестре(ах): 3.

В процессе изучения дисциплины студент готовится к решению типов задач профессиональной деятельности, предусмотренных ФГОС ВО и образовательной программой.

Компетенция	Предшествующие дисциплины	Последующие дисциплины
ПК-1 - Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы		

ПК-1.1 Разрабатывает архитектуру и прототип ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	Архитектура интеллектуальных информационных систем, Веб-программирование, Облачные ресурсы для разработки интеллектуальных сервисов, Облачные технологии и сервисы	Автоматизированное тестирование программного обеспечения, Архитектура интеллектуальных информационных систем, Блокчейн и его приложения, Веб-программирование, Выполнение и защита выпускной квалификационной работы, Интеллектуальные информационные системы в менеджменте, Информационно-коммуникационные технологии в профессиональной деятельности, Корпоративные интеллектуальные системы, Методы и средства проектирования цифровых сервисов в управлении, Нейросетевые технологии в социальных медиа, Проектирование и разработка экспертных интеллектуальных систем, Производственная практика: преддипломная практика, Производственная практика: технологическая (проектно-технологическая) практика, Разработка цифровых сервисов на основе искусственного интеллекта, Системы машинного обучения в управлении, Современные технологии разработки приложений, Технологии блокчейн, Технологии больших данных, Управление качеством разработки приложений, Учебная практика: ознакомительная практика, Цифровой дизайн, инфографика и визуализация данных в управлении
---	---	---

ПК-1.2 Проектирует, создает дизайн и разрабатывает базы данных ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	Архитектура интеллектуальных информационных систем, Веб-программирование, Облачные ресурсы для разработки интеллектуальных сервисов, Облачные технологии и сервисы	Автоматизированное тестирование программного обеспечения, Архитектура интеллектуальных информационных систем, Блокчейн и его приложения, Веб-программирование, Выполнение и защита выпускной квалификационной работы, Интеллектуальные информационные системы в менеджменте, Информационно-коммуникационные технологии в профессиональной деятельности, Корпоративные интеллектуальные системы, Нейросетевые технологии в социальных медиа, Проектирование и разработка экспертных интеллектуальных систем, Производственная практика: преддипломная практика, Производственная практика: технологическая (проектно-технологическая) практика, Разработка цифровых сервисов на основе искусственного интеллекта, Системы машинного обучения в управлении, Современные технологии разработки приложений, Технологии блокчейн, Технологии больших данных, Управление качеством разработки приложений, Учебная практика: ознакомительная практика, Цифровой дизайн, инфографика и визуализация данных в управлении
--	--	---

ПК-1.3 Реализует процесс обеспечения и контроля качества в соответствии с регламентами организации в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	Архитектура интеллектуальных информационных систем, Веб-программирование, Облачные ресурсы для разработки интеллектуальных сервисов, Облачные технологии и сервисы	Автоматизированное тестирование программного обеспечения, Архитектура интеллектуальных информационных систем, Блокчейн и его приложения, Веб-программирование, Выполнение и защита выпускной квалификационной работы, Интеллектуальные информационные системы в менеджменте, Информационно-коммуникационные технологии в профессиональной деятельности, Корпоративные интеллектуальные системы, Методы и средства проектирования цифровых сервисов в управлении, Нейросетевые технологии в социальных медиа, Проектирование и разработка экспертных интеллектуальных систем, Производственная практика: преддипломная практика, Производственная практика: технологическая (проектно-технологическая) практика, Разработка цифровых сервисов на основе искусственного интеллекта, Системы машинного обучения в управлении, Современные технологии разработки приложений, Технологии блокчейн, Технологии больших данных, Управление качеством разработки приложений, Учебная практика: ознакомительная практика, Цифровой дизайн, инфографика и визуализация данных в управлении
---	--	--

4. Объем дисциплины (модуля) и виды учебной работы

Период обучения	Общая трудоемкость (часы)	Общая трудоемкость (ЗЕТ)	Контактная работа (часы, всего)	Лабораторные занятия (часы)	Лекционные занятия (часы)	Индивидуальная контактная работа (часы)	Самостоятельная работа (часы)	Промежуточная аттестация
Третий семестр	108	3	36	18	18	0,15	53,85	Зачет
Всего	108	3	36	18	18	0,15	53,85	18

5. Содержание дисциплины (модуля)

5.1. Разделы, темы дисциплины и виды занятий
(часы промежуточной аттестации не указываются)

Наименование раздела, темы	Всего	Лабораторные занятия	Лекционные занятия	Самостоятельная работа
Раздел 1. Распределенные приложения	90	18	18	53,85
Тема 1.1. Введение в распределенные системы Определение, свойства (масштабируемость, отказоустойчивость, прозрачность). CAP-теорема. Классификация: клиент-сервер, peer-to-peer, микросервисы. Примеры современных систем.	5	2	2	1
Тема 1.2. Модели взаимодействия и удаленный вызов процедур (RPC) Синхронное/асинхронное взаимодействие. RPC vs REST vs gRPC. Сериализация (JSON, Protobuf, Avro). Обзор фреймворков (gRPC, Thrift).	13	2	2	9
Тема 1.3. Сетевые основы и протоколы прикладного уровня OSI/TCP/IP. HTTP/2, HTTP/3, WebSocket. Особенности работы в WAN и дата-центрах. DNS, балансировка нагрузки (L4/L7).	14	2	2	10
Тема 1.4. Архитектурные стили: Микросервисы и Event-Driven Монолит vs микросервисы. Декомпозиция по бизнес-capabilities. Event Sourcing, CQRS. Шаблоны интеграции (API Gateway, Backend-for-Frontend).	14,15	2	2	10
Тема 1.5. Синхронизация времени и логические часы Проблема времени в распределенных системах. Физические (NTP) и логические часы (Лампорт, векторные метки). Частичный порядок событий. Применение в трассировке.	9	2	2	5

Тема 1.6. Распределенные транзакции и согласованность ACID vs BASE. Двухфазный коммит (2PC) и его проблемы. Компенсирующие транзакции (Saga). Конечная согласованность. Выбор уровня изоляции.	9	2	2	5
Тема 1.7. Управление данными в распределенных системах Репликация (master-slave, multi-master) и шардирование (consistent hashing). Распределенные кэши (Redis, Memcached). Eventual consistency, quorum-подходы.	9	2	2	5
Тема 1.8. Отказоустойчивость и обнаружение сбоев Виды отказов (fail-stop, византийские). Heartbeat-механизмы, детекторы недоступности. Таймауты, ретрай, экспоненциальная задержка. Pattern: Circuit Breaker, Retry, Bulkhead.	9	2	2	5
Тема 1.9. Безопасность распределенных приложений Аутентификация (JWT, OAuth2/OIDC), авторизация (RBAC/ABAC). TLS/mTLS. Защита от атак (Man-in-the-middle, Replay). Управление секретами (Vault). Аудит и логирование.	7,85	2	2	3,85

5.2. Контрольные мероприятия по дисциплине

Вид контроля	Форма контроля/Оценочное средство
Текущий контроль	Тестовое задание
Промежуточная аттестация	Зачет

№ п/п	Наименование раздела	Вид контроля/ используемые оценочные материалы	
		Текущий	Промежут. аттестация
1	Распределенные приложения	Тестовое задание	Зачет

6. Оценочные материалы текущего контроля

1. Распределенные приложения Тестовое задание

№ п/п	Содержание вопроса		Компетенция
		Правильный ответ (ключ ответа)	

1	При разработке архитектуры распределенной ИС для интернет-магазина перед вами стоит выбор между согласованностью и доступностью данных. Какое утверждение корректно описывает CAP-теорему, которую необходимо учитывать при проектировании архитектуры распределенной системы? 1. В распределенной системе одновременно гарантируются согласованность, доступность и устойчивость к разделению сети. 2. В распределенной системе невозможно одновременно обеспечить все три свойства CAP — необходимо выбирать два из трех. 3. CAP-теорема применима только к реляционным базам данных и не распространяется на NoSQL-хранилища. 4. CAP-теорема требует всегда жертвовать доступностью ради согласованности в распределенных системах. 5. CAP-теорема описывает только проблемы сетевых задержек и не влияет на архитектурные решения ИС.	ПК-1
	Ответ: 2	
2	При проектировании архитектуры ИС на основе микросервисов необходимо выбрать протокол взаимодействия между компонентами распределенной системы. Какой стек технологий (протокол + формат сериализации) используется в gRPC для обеспечения высокопроизводительного обмена данными между сервисами? 1. REST + JSON 2. SOAP + XML 3. HTTP/2 + Protobuf 4. WebSocket + Avro 5. HTTP/1.1 + Thrift	ПК-1
	Ответ: 3	
3	При разработке распределенной базы данных для высоконагруженной ИС с интенсивной записью данных и необходимостью равномерного распределения нагрузки, какую стратегию шардирования следует выбрать для предотвращения «горячих точек» (hot spots) в архитектуре хранения данных? 1. Шардирование по диапазону значений (range-based sharding) 2. Шардирование на основе списка значений (list-based sharding) 3. Шардирование на основе хеширования (consistent hashing) 4. Шардирование по географическому признаку 5. Шардирование по времени создания записи	ПК-1
	Ответ: 3	
4	При проектировании базы данных ИС для управления каталогом товаров с иерархической и изменяемой структурой (вложенные категории, вариативные атрибуты) в распределенной системе, какая модель данных NoSQL наиболее подходит для эффективного хранения и извлечения таких данных? 1. Ключ-значение (Key-Value) 2. Документная (Document-oriented) 3. Колоночная (Column-family) 4. Графовая (Graph) 5. Реляционная (Relational)	ПК-1
	Ответ: 2	

5	В рамках процесса обеспечения качества ИС и контроля эксплуатационных характеристик распределенной системы, какой показатель используется для оценки доступности сервиса и определяется как процент успешных запросов за определенный период времени в соответствии с регламентами организации? 1. Среднее время ответа (Average Response Time) 2. Пропускная способность (Throughput) 3. Уровень обслуживания (SLO — Service Level Objective) 4. Индекс удовлетворенности пользователей (CSAT) 5. Количество инцидентов за месяц	ПК-1										
	Ответ: 3											
6	Установите соответствие между архитектурным паттерном распределенной ИС и его функциональным назначением. В ответе запишите последовательность цифр (например, А-1, Б-2, В-3). <table><tr><th>Архитектурный паттерн распределенной ИС</th><th>Назначение паттерна в архитектуре системы</th></tr><tr><td>А. API Gateway</td><td>1. Обеспечение согласованности данных в распределенных транзакциях через компенсирующие операции</td></tr><tr><td>Б. CQRS</td><td>2. Единая точка входа для маршрутизации запросов к <u>микросервисам</u>, агрегации ответов и аутентификации</td></tr><tr><td>В. Saga</td><td>3. Автоматическое восстановление системы после сбоя компонента</td></tr><tr><td>Г. Circuit Breaker</td><td>4. Разделение команд на запись и запросов на чтение для оптимизации производительности</td></tr></table>	Архитектурный паттерн распределенной ИС	Назначение паттерна в архитектуре системы	А. API Gateway	1. Обеспечение согласованности данных в распределенных транзакциях через компенсирующие операции	Б. CQRS	2. Единая точка входа для маршрутизации запросов к <u>микросервисам</u> , агрегации ответов и аутентификации	В. Saga	3. Автоматическое восстановление системы после сбоя компонента	Г. Circuit Breaker	4. Разделение команд на запись и запросов на чтение для оптимизации производительности	ПК-1
Архитектурный паттерн распределенной ИС	Назначение паттерна в архитектуре системы											
А. API Gateway	1. Обеспечение согласованности данных в распределенных транзакциях через компенсирующие операции											
Б. CQRS	2. Единая точка входа для маршрутизации запросов к <u>микросервисам</u> , агрегации ответов и аутентификации											
В. Saga	3. Автоматическое восстановление системы после сбоя компонента											
Г. Circuit Breaker	4. Разделение команд на запись и запросов на чтение для оптимизации производительности											
	Ответ: А-2, Б-4, В-1, Г-3											
7	Установите соответствие между типом СУБД и сценарием использования в распределенной ИС. В ответе запишите последовательность цифр (например, А-1, Б-2, В-3). <table><tr><th>Тип СУБД</th><th>Сценарий использования в распределенной системе</th></tr><tr><td>А. Реляционная (<u>PostgreSQL</u>)</td><td>1. Хранение сессий пользователей и кэширование результатов частых запросов</td></tr><tr><td>Б. Документная (<u>MongoDB</u>)</td><td>2. Управление транзакционными данными с жесткими требованиями ACID (финансовые операции)</td></tr><tr><td>В. Ключ-значение (<u>Redis</u>)</td><td>3. Хранение данных с множеством связей и отношений (социальные сети)</td></tr><tr><td>Г. <u>Графовая</u> (<u>Neo4j</u>)</td><td>4. Хранение каталога товаров с изменяемой вложенной структурой</td></tr></table>	Тип СУБД	Сценарий использования в распределенной системе	А. Реляционная (<u>PostgreSQL</u>)	1. Хранение сессий пользователей и кэширование результатов частых запросов	Б. Документная (<u>MongoDB</u>)	2. Управление транзакционными данными с жесткими требованиями ACID (финансовые операции)	В. Ключ-значение (<u>Redis</u>)	3. Хранение данных с множеством связей и отношений (социальные сети)	Г. <u>Графовая</u> (<u>Neo4j</u>)	4. Хранение каталога товаров с изменяемой вложенной структурой	ПК-1
Тип СУБД	Сценарий использования в распределенной системе											
А. Реляционная (<u>PostgreSQL</u>)	1. Хранение сессий пользователей и кэширование результатов частых запросов											
Б. Документная (<u>MongoDB</u>)	2. Управление транзакционными данными с жесткими требованиями ACID (финансовые операции)											
В. Ключ-значение (<u>Redis</u>)	3. Хранение данных с множеством связей и отношений (социальные сети)											
Г. <u>Графовая</u> (<u>Neo4j</u>)	4. Хранение каталога товаров с изменяемой вложенной структурой											
	Ответ: А-2, Б-4, В-1, Г-3											
8	Установите соответствие между инструментом контроля качества и его функциональным назначением в процессе обеспечения качества распределенной ИС. В ответе запишите последовательность цифр (например, А-1, Б-2, В-3). <table><tr><th>Инструмент контроля качества</th><th>Назначение в процессе обеспечения качества</th></tr><tr><td>А. <u>Prometheus</u> + <u>Grafana</u></td><td>1. Сбор, агрегация и анализ логов событий компонентов распределенной системы</td></tr><tr><td>Б. ELK Stack (Elasticsearch, Logstash, Kibana)</td><td>2. Распределенная трассировка запросов между <u>микросервисами</u> для диагностики узких мест</td></tr><tr><td>В. <u>Jaeger</u> / <u>Zipkin</u></td><td>3. Мониторинг метрик производительности и визуализация <u>дашбордов</u> состояния системы</td></tr><tr><td>Г. <u>JMeter</u> / <u>Gatling</u></td><td>4. Нагрузочное и стресс-тестирование распределенных приложений</td></tr></table>	Инструмент контроля качества	Назначение в процессе обеспечения качества	А. <u>Prometheus</u> + <u>Grafana</u>	1. Сбор, агрегация и анализ логов событий компонентов распределенной системы	Б. ELK Stack (Elasticsearch, Logstash, Kibana)	2. Распределенная трассировка запросов между <u>микросервисами</u> для диагностики узких мест	В. <u>Jaeger</u> / <u>Zipkin</u>	3. Мониторинг метрик производительности и визуализация <u>дашбордов</u> состояния системы	Г. <u>JMeter</u> / <u>Gatling</u>	4. Нагрузочное и стресс-тестирование распределенных приложений	ПК-1
Инструмент контроля качества	Назначение в процессе обеспечения качества											
А. <u>Prometheus</u> + <u>Grafana</u>	1. Сбор, агрегация и анализ логов событий компонентов распределенной системы											
Б. ELK Stack (Elasticsearch, Logstash, Kibana)	2. Распределенная трассировка запросов между <u>микросервисами</u> для диагностики узких мест											
В. <u>Jaeger</u> / <u>Zipkin</u>	3. Мониторинг метрик производительности и визуализация <u>дашбордов</u> состояния системы											
Г. <u>JMeter</u> / <u>Gatling</u>	4. Нагрузочное и стресс-тестирование распределенных приложений											
	Ответ: А-3, Б-1, В-2, Г-4											
9	Установите правильную последовательность этапов разработки архитектуры ИС при создании распределенной системы. В ответе запишите последовательность цифр (например, 1-2-3-4). 1. Выбор архитектурного стиля (микросервисы/монолит/событийно-ориентированная) 2. Декомпозиция бизнес-процессов и выделение сервисов/компонентов 3. Анализ нефункциональных требований (масштабируемость, доступность, производительность) 4. Определение протоколов взаимодействия (REST/gRPC) и моделей данных	ПК-1										
	Ответ: 3-2-1-4											

10	Установите правильную последовательность шагов проведения нагрузочного тестирования в процессе обеспечения качества распределенной ИС в соответствии с регламентами организации. В ответе запишите последовательность цифр (например, 1-2-3-4). 1. Анализ результатов тестирования и формирование рекомендаций по оптимизации 2. Определение целевых показателей производительности (SLO) и профиля нагрузки 3. Настройка инфраструктуры тестирования и запуск тестовых сценариев 4. Разработка сценариев нагрузочных тестов, имитирующих реальные бизнес-операции	ПК-1
	Ответ: 2-4-3-1	
11	Дайте развернутый письменный ответ (3-5 предложений). В рамках разработки архитектуры распределенной ИС для интернет-платформы используется микросервисный подход. При оформлении заказа необходимо атомарно обновить данные в трех сервисах: заказов, инвентаризации (списание товара) и оплаты. Опишите, какие архитектурные паттерны распределенных систем вы примените для обеспечения согласованности данных между сервисами. Обоснуйте выбор с учетом требований к доступности и отказоустойчивости ИС.	ПК-1
	Ответ: Применение паттерна Saga (хореография/оркестрация) для распределенных транзакций, так как 2PC снижает доступность. Использование компенсирующих транзакций при сбоях. Внедрение событийно-ориентированной архитектуры с брокером сообщений (Kafka) для асинхронного обмена.	
12	Дайте развернутый письменный ответ (3-5 предложений). При эксплуатации распределенной ИС в часы пиковой нагрузки один из сервисов начинает отвечать с таймаутами, вызывая каскадные сбои. Какие шаблоны повышения отказоустойчивости архитектуры ИС вы примените для защиты системы? Опишите их назначение и порядок применения.	ПК-1
	Ответ: Circuit Breaker — отключение вызовов при превышении порога ошибок. Retry с экспоненциальной задержкой — ограниченные повторные попытки. Bulkhead — изоляция пулов потоков. Настройка таймаутов и детекторов недоступности.	
13	Дайте развернутый письменный ответ (3-5 предложений). При проектировании распределенной базы данных для социальной сети с 100 млн пользователей предложите стратегию шардирования и репликации. Обоснуйте выбор ключа шардирования, типа шардирования и количества шардов в архитектуре хранения данных ИС. Опишите возможные проблемы и способы их решения.	ПК-1
	Ответ: Consistent hashing по user_id для равномерного распределения. 16-32 шарда. Репликация (master-slave) для доступности. Проблема «горячих» пользователей — решение через кэширование (Redis) или дополнительное шардирование по вторичному ключу.	
14	Дайте развернутый письменный ответ (3-5 предложений). Для разработки базы данных ИС управления заказами требуется обеспечить высокую производительность чтения отчетов, гибкость записи и полный аудит изменений. Опишите применение паттернов CQRS и Event Sourcing в архитектуре распределенной системы. Как организовать хранение данных и синхронизацию между компонентами?	ПК-1
	Ответ: CQRS — разделение команд и запросов. Event Sourcing — хранение изменений как событий в event store. Материализованные представления для чтения, обновляемые асинхронно через обработку событий. Независимое масштабирование чтения и записи.	
15	Дайте развернутый письменный ответ (3-5 предложений). В рамках процесса обеспечения качества ИС разработайте план контроля эксплуатационных характеристик для распределенной системы в соответствии с регламентами организации. Определите ключевые метрики качества (SLA/SLO/SLI), инструменты мониторинга, логирования и трассировки, а также регламенты реагирования на инциденты.	ПК-1
	Ответ: Метрики: SLA (доступность 99.9%), SLO (время ответа < 200 мс), SLI (частота ошибок < 0.1%). Инструменты: Prometheus + Grafana (мониторинг), ELK Stack (логирование), Jaeger (трассировка). Регламенты: алертинг, ежедневный мониторинг, Chaos Engineering, анализ инцидентов.	

7. Оценочные материалы промежуточной аттестации

Зачет третий семестр

№ п/п	Содержание вопроса		Компетенция
		Правильный ответ (ключ ответа)	
1	Дайте развернутый письменный ответ (4-6 предложений) В рамках выполнения работ по созданию ИС для интернет-магазина с высокими требованиями к доступности (99.99%) архитектор выбирает стратегию обработки данных. Опишите, как CAP-теорема влияет на разработку архитектуры распределенной ИС для данного бизнес-процесса. Какой выбор должен сделать архитектор и почему? Приведите пример компромиссного решения.		ПК-1

	Ответ: CAP-теорема утверждает, что в распределенной системе невозможно одновременно обеспечить согласованность (Consistency), доступность (Availability) и устойчивость к разделению сети (Partition Tolerance) — необходимо выбирать два из трех. Для интернет-магазина с требованием высокой доступности архитектор должен выбрать AP-систему (доступность + устойчивость к разделению), жертвуя строгой согласованностью в пользу конечной согласованности (eventual consistency). Это означает, что при сетевых сбоях система продолжает принимать запросы, а данные синхронизируются позже (например, при обновлении остатков товара). Компромиссное решение — использование асинхронной репликации и механизмов разрешения конфликтов (например, техника «последняя запись побеждает» или векторные метки времени).	
2	Дайте развернутый письменный ответ (4-6 предложений) При выполнении работ по модификации ИС управления логистикой принято решение перейти от монолитной архитектуры к микросервисной. Опишите, как осуществляется декомпозиция бизнес-процессов при разработке архитектуры распределенной ИС на основе микросервисов. Какие критерии используются для выделения сервисов? Приведите 2-3 примера сервисов для логистической ИС. Ответ: Декомпозиция бизнес-процессов при переходе к микросервисной архитектуре выполняется на основе предметных областей (Domain-Driven Design, DDD) и бизнес-способностей (business capabilities). Критерии выделения сервисов: независимость изменений, отдельный цикл разработки/развертывания, слабая связанность, высокая связанность внутри сервиса, масштабируемость по нагрузке, право собственности у одной команды. Для логистической ИС можно выделить сервисы: Сервис управления заказами, Сервис инвентаризации (складских остатков), Сервис доставки (маршрутизация), Сервис уведомлений, Сервис оплаты. Каждый сервис владеет своими данными и взаимодействует через синхронные (REST/gRPC) или асинхронные (брокеры сообщений) интерфейсы.	ПК-1
3	Дайте развернутый письменный ответ При разработке архитектуры ИС с интенсивным обменом данными между 20 микросервисами архитектор выбирает протокол взаимодействия. Сравните gRPC и REST в контексте проектирования распределенных ИС. Укажите преимущества и ограничения каждого подхода. В каких сценариях бизнес-процессов предпочтительно использовать gRPC? Ответ: REST — архитектурный стиль на основе HTTP/1.1 с форматом JSON, прост в реализации, широко поддерживается, легко отлаживается, но имеет низкую производительность из-за текстового формата и отсутствия строгой типизации. gRPC — фреймворк на базе HTTP/2 с бинарной сериализацией Protobuf, обеспечивает высокую производительность, строгую контрактность (protobuf-схемы), поддержку стриминга (однаправленный и двунаправленный), встроенные механизмы таймаутов и retry. Ограничения gRPC: сложность отладки (бинарный формат), ограниченная поддержка браузерами, необходимость генерации кода. gRPC предпочтителен для внутреннего взаимодействия между микросервисами при высоких нагрузках, в системах реального времени, стриминговых сценариях и при необходимости строгой типизации контрактов. REST предпочтителен для публичных API и интеграции с внешними системами.	ПК-1
4	Дайте развернутый письменный ответ При разработке базы данных ИС для социальной сети, где основной нагрузкой являются чтение ленты новостей, хранение профилей пользователей с вложенными структурами и обработка связей между пользователями, необходимо выбрать типы СУБД. Какие модели данных NoSQL вы предложите для разных компонентов системы? Обоснуйте выбор с учетом требований к проектированию распределенной базы данных ИС. Ответ: Для социальной сети целесообразно использовать полиглотное хранение — комбинацию нескольких СУБД в зависимости от типа данных. Для хранения профилей пользователей с вложенной структурой (интересы, настройки, фото) подходит документная БД (MongoDB) — гибкая схема и быстрые запросы по документу. Для ленты новостей и кэширования — ключ-значение (Redis) — высокая производительность чтения/записи в память. Для хранения связей между пользователями (друзья, подписки, рекомендации) — графовая БД (Neo4j) — эффективные обходы связей любой глубины. Для аналитических отчетов и агрегаций — колоночная БД (Cassandra) для высокой пропускной способности записи. Такой подход позволяет оптимально использовать сильные стороны каждой модели данных в распределенной среде.	ПК-1
5	Дайте развернутый письменный ответ При проектировании распределенной базы данных ИС для системы управления заказами с ожидаемой нагрузкой 50 тысяч запросов в секунду необходимо реализовать горизонтальное масштабирование. Опишите механизм шардирования на основе consistent hashing. В чем преимущества данного подхода по сравнению с диапазонным шардированием? Как выбрать ключ шардирования для системы заказов? Ответ: Consistent hashing — стратегия шардирования, при которой ключи и узлы отображаются на кольцо хеш-пространства; каждый ключ принадлежит ближайшему узлу по часовой стрелке. Преимущества по сравнению с диапазонным шардированием: минимальное перераспределение данных при добавлении/удалении узлов (только соседние диапазоны, а не все данные), устойчивость к «горячим точкам» за счет виртуальных нод, естественная поддержка репликации. Для системы заказов ключом шардирования рекомендуется order_id (или его хеш) для равномерного распределения по всем узлам. Для запросов, фильтрующих по пользователю, можно использовать вторичные индексы или композитный ключ (user_id + order_id) с дополнительным кэшированием.	ПК-1

6	Дайте развернутый письменный ответ При разработке базы данных ИС для банковской системы с требованиями к полному аудиту всех изменений, высокой производительности чтения отчетов и гибкости записи, предложите применение паттернов CQRS и Event Sourcing. Как организовать хранение данных и синхронизацию между командами и запросами в архитектуре распределенной системы? Ответ: CQRS (Command Query Responsibility Segregation) разделяет операции записи (команды) и чтения (запросы) на разные модели данных, что позволяет масштабировать их независимо. Event Sourcing хранит все изменения как неизменяемую последовательность событий в event store (например, EventStoreDB или Apache Kafka), обеспечивая полный аудит и возможность восстановления состояния на любой момент времени. При записи команды создаются события, которые сохраняются в event store. Для чтения строятся материализованные представления (read models) в отдельной БД (например, PostgreSQL или Elasticsearch), которые обновляются асинхронно через обработчики событий (event handlers). Это позволяет обеспечить строгую согласованность в записи и высокую производительность в чтении, а также полную воспроизводимость всех операций для аудита.	ПК-1
7	Дайте развернутый письменный ответ В рамках сопровождения ИС интернет-платформы выявлена проблема: при оформлении заказа списание товара и оплата выполняются в разных сервисах, и при сбое одного из них данные становятся несогласованными. Опишите, как паттерн Saga решает проблему распределенных транзакций в архитектуре ИС. Сравните хореографию и оркестрацию. Какой подход вы предпочтете для сложного бизнес-процесса и почему? Ответ: Saga — паттерн управления распределенными транзакциями, где каждая бизнес-операция разбивается на последовательность локальных транзакций в разных сервисах; при сбое выполняются компенсирующие транзакции для отката. В хореографии каждый сервис публикует события и реагирует на них без центрального координатора — децентрализованно, но сложно отслеживать процесс. В оркестрации центральный координатор (оркестратор) управляет последовательностью шагов и вызывает компенсации при ошибках — централизованно, проще контролировать и логировать. Для сложного бизнес-процесса оформления заказа (3+ сервисов с зависимостями) я предпочту оркестрацию — она обеспечивает лучшую наблюдаемость, управляемость и упрощает внесение изменений в последовательность шагов, что критично для сопровождения ИС.	ПК-1
8	Дайте развернутый письменный ответ При выполнении работ по созданию ИС критически важной для бизнеса (система бронирования авиабилетов) необходимо обеспечить высокую отказоустойчивость. Какие шаблоны повышения отказоустойчивости распределенной ИС вы примените? Опишите каждый шаблон и последовательность их применения при возникновении сбоя в сервисе оплаты. Ответ: Применяются следующие шаблоны: Circuit Breaker — при превышении порога ошибок вызовы к проблемному сервису блокируются (состояние «open»), возвращается fallback-ответ (например, «оплата временно недоступна, повторите позже»), периодически проверяется восстановление (состояние «half-open»). Retry с экспоненциальной задержкой — ограниченное количество повторных попыток с увеличивающимся интервалом (100ms, 200ms, 400ms) при временных сбоях. Bulkhead — изоляция пулов потоков или подключений к сервису оплаты, чтобы его сбой не исчерпал ресурсы других сервисов. Timeout — ограничение времени ожидания ответа (например, 3 секунды). Health Check / Heartbeat — периодическая проверка доступности сервиса оплаты. Последовательность: при сбое сервиса оплаты Circuit Breaker переходит в состояние open, возвращается fallback, Retry не применяется (так как сбой устойчив). При восстановлении сервиса Circuit Breaker переходит в half-open, пробный запрос проходит, возвращается в closed — система восстанавливается.	ПК-1
9	Дайте развернутый письменный ответ В процессе обеспечения и контроля качества ИС распределенной системы с 50 микросервисами необходимо организовать наблюдаемость (observability) для диагностики проблем и контроля производительности. Опишите три основных столпа наблюдаемости — мониторинг, логирование и распределенную трассировку. Какие инструменты вы предложите для каждого столпа в соответствии с регламентами организации? Ответ: Три столпа наблюдаемости: Мониторинг — сбор и визуализация метрик (CPU, память, количество запросов, ошибок, время ответа). Инструменты: Prometheus для сбора метрик и Grafana для дашбордов и алертинга. Логирование — запись структурированных событий (JSON-логи) с уровнями (info, warn, error). Инструменты: ELK Stack (Elasticsearch, Logstash, Kibana) или Loki для централизованного сбора, хранения и поиска логов. Распределенная трассировка — отслеживание пути запроса через все микросервисы с идентификатором trace_id, измерение времени в каждом сервисе. Инструменты: Jaeger или Zipkin с интеграцией OpenTelemetry. В соответствии с регламентами организации настраиваются алерты на превышение порогов метрик (SLO-нарушения), логи агрегируются с корреляцией по trace_id для полной диагностики инцидентов.	ПК-1

10	Дайте развернутый письменный ответ В рамках процесса обеспечения качества ИС перед релизом новой версии распределенной системы необходимо провести нагрузочное тестирование. Опишите план проведения нагрузочного тестирования в соответствии с регламентами организации: какие этапы включает процесс, какие метрики качества (SLA/SLO/SLI) вы определите, и какие инструменты используете. Как вы интерпретируете результаты и принимаете решение о готовности системы к релизу?	ПК-1
	Ответ: План нагрузочного тестирования включает этапы: Определение SLO (Service Level Objectives) и профиля нагрузки на основе бизнес-требований (например, RPS — 10 000 запросов/сек, среднее время ответа < 200 мс, ошибки < 0.1%). Разработка сценариев тестов, имитирующих реальные бизнес-операции (пользовательские сценарии). Настройка инфраструктуры (распределенные нагрузочные агенты) и запуск тестов с постепенным увеличением нагрузки. Сбор метрик SLI (Service Level Indicators) — процент успешных запросов, время ответа, пропускная способность, количество ошибок. Анализ результатов: сравнение фактических SLI с целевыми SLO, выявление узких мест (бутылочных), формирование отчета с рекомендациями. Инструменты: JMeter, Gatling, k6 или Yandex.Tank. Решение о готовности: если все SLI соответствуют SLO при 120% ожидаемой нагрузки в течение часа — система готова к релизу. При отклонениях — требуется оптимизация и повторное тестирование.	

7.1. Уровни овладения

Компетенция: ПК-1 Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы.

Индикатор достижения компетенции: ПК-1.1 Разрабатывает архитектуру и прототип ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы.

Уровень	Характеристика	Оценка в баллах
Повышенный	Достигнуто полное овладение знаниями, умениями и навыками. Студент свободно владеет терминологией, умеет применять теоретические знания в различных ситуациях для решения поставленных задач.	81-100
Базовый	Достигнуто достаточное овладение знаниями, умениями и навыками. Студент уверенно владеет терминологией, умеет применять теоретические знания в различных ситуациях для решения поставленных задач.	61-80
Пороговый	Достигнуто овладение минимально необходимыми знаниями, умениями и навыками. Студент владеет основной терминологией, умеет применять теоретические знания для решения поставленных задач в стандартных ситуациях.	41-60
Ниже порогового	Компетенция не освоена	0-40

Индикатор достижения компетенции: ПК-1.2 Проектирует, создает дизайн и разрабатывает базы данных ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС.

Уровень	Характеристика	Оценка в баллах
---------	----------------	-----------------

Повышенный	Достигнуто полное овладение знаниями, умениями и навыками. Студент свободно владеет терминологией, умеет применять теоретические знания в различных ситуациях для решения поставленных задач.	81-100
Базовый	Достигнуто достаточное овладение знаниями, умениями и навыками. Студент уверенно владеет терминологией, умеет применять теоретические знания в различных ситуациях для решения поставленных задач.	61-80
Пороговый	Достигнуто овладение минимально необходимыми знаниями, умениями и навыками. Студент владеет основной терминологией, умеет применять теоретические знания для решения поставленных задач в стандартных ситуациях.	41-60
Ниже порогового	Компетенция не освоена	0-40

Индикатор достижения компетенции: ПК-1.3 Реализует процесс обеспечения и контроля качества в соответствии с регламентами организации в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС.

Уровень	Характеристика	Оценка в баллах
Повышенный	Достигнуто полное овладение знаниями, умениями и навыками. Студент свободно владеет терминологией, умеет применять теоретические знания в различных ситуациях для решения поставленных задач.	81-100
Базовый	Достигнуто достаточное овладение знаниями, умениями и навыками. Студент уверенно владеет терминологией, умеет применять теоретические знания в различных ситуациях для решения поставленных задач.	61-80
Пороговый	Достигнуто овладение минимально необходимыми знаниями, умениями и навыками. Студент владеет основной терминологией, умеет применять теоретические знания для решения поставленных задач в стандартных ситуациях.	41-60
Ниже порогового	Компетенция не освоена	0-40

8. Материально-техническое и учебно-методическое обеспечение дисциплины

8.1. Перечень основной и дополнительной учебной литературы

Основная литература

1. Бабичев, С. Л. Распределенные системы: учебник для вузов / С. Л. Бабичев, К. А. Коньков. - Москва: Юрайт, 2026. - 507 с - 978-5-534-11380-8. - Текст: электронный // ИКО Юрайт: [сайт]. - URL: <https://urait.ru/bcode/587607> (дата обращения: 21.05.2026). - Режим доступа: по подписке

2. Петрухнова,, Г. В. Введение в распределенные системы: учебное пособие / Г. В. Петрухнова,. - Введение в распределенные системы - Воронеж: Воронежский государственный технический университет, ЭБС АСВ, 2021. - 81 с. - 978-5-7731-0925-9. - Текст: электронный // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/111462.html> (дата обращения: 21.05.2026). - Режим доступа: по подписке

Дополнительная литература

1. Распределенные базы и хранилища данных / А.М. Марасанов, Н.П. Аносова, О.О. Бородин, Е.С. Гаврилов. - Москва: Национальный Открытый Университет ИНТУИТ, 2024. - 254 с. - intuit483. - Текст: электронный // ibooks: [сайт]. - URL: <https://ibooks.ru/reading.php?short=1&productid=394510> (дата обращения: 21.05.2026). - Режим доступа: по подписке

8.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся

Профессиональные базы данных

1. <https://lks.dap.gov.ru/> - Цифровая аналитическая платформа предоставления статистических данных» (ГИС ЦАП)
2. <https://minfin.gov.ru/> - Министерство финансов Российской Федерации (Минфин России)
3. <https://fgistp.economy.gov.ru/design/main> - Федеральная государственная информационная система территориального планирования
4. <http://www.gov.ru/> - Профессиональная база данных «Информационные системы Министерства экономического развития Российской Федерации в сети Интернет» (Портал «Официальная Россия»)
5. <https://programs.economy.gov.ru/> - Аналитическая информационная система мониторинга и анализа исполнения государственных программ

Ресурсы «Интернет»

1. <https://www.rusprofile.ru/> - Система комплексного анализа рынков и компаний (СКАР) / Rusprofile
2. https://info216.wiki.uib.no/index.php?title=Lab:_OWL_2&oldid=2704 - Открытые лабораторные работы «INFO216: OWL 2 Lab» дают отличную пошаговую инструкцию по созданию онтологии в Protégé, включая определение классов, свойств, использование механизма логического вывода (Reasoner) и запросов SPARQL
3. <https://stepik.org> - Платформа с онлайн-курсами от авторов-практиков

8.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

1. MySQL Server 8.0.45;
2. MySQL Workbench 8.0.45;
3. Python версии 3.14.4 ;
4. Консультант Плюс;
5. Мой офис;

Перечень информационно-справочных систем

(обновление выполняется еженедельно)

Не используется.

8.4. Специальные помещения, лаборатории и лабораторное оборудование

Учебные аудитории для проведения занятий лекционного типа	Комплекты ученической мебели Мультимедийный проектор Доска Экран
---	---

Учебные аудитории для проведения практических занятий (занятий семинарского типа)	Комплекты ученической мебели Мультимедийный проектор Доска Экран Компьютеры с выходом в сеть «Интернет» и ЭИОС СПб
Учебные аудитории для групповых и индивидуальных консультаций	Комплекты ученической мебели Мультимедийный проектор Доска Экран Компьютеры с выходом в сеть «Интернет» и ЭИОС СПб
Учебные аудитории для текущего контроля и промежуточной аттестации	Комплекты ученической мебели Мультимедийный проектор Доска Экран Компьютеры с выходом в сеть «Интернет» и ЭИОС СПб
Помещения для самостоятельной работы	Комплекты ученической мебели Мультимедийный проектор Доска Экран Компьютеры с выходом в сеть «Интернет» и ЭИОС СПб
Помещения для хранения и профилактического обслуживания оборудования	Комплекты специализированной мебели для хранения