



Федеральное государственное образовательное бюджетное учреждение высшего образования
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)

Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных

Документ подписан усиленной неквалифицированной электронной подписью.

Организация: «ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»

Сертификат: mKXfUY/6IMvgvpqCsZhSpqsn/tSjwI

Утверждено: Проректор по учебной и методической работе, Е.А. Каменева

Дата: 26.06.2026 г.

А.В. Поляков

Микросервисная архитектура

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки

09.03.03 - Прикладная информатика,

Образовательная программа «Прикладные информационные системы в экономике и
финансах»

Профиль:

«Прикладные информационные системы в экономике и финансах»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 09 от 16.06.2026 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 04 от 25.05.2026 г.)*



Содержание

1. Наименование дисциплины	3
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3. Место дисциплины в структуре образовательной программы	7
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	7
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	8
5.1. Содержание дисциплины	8
5.2. Учебно – тематический план	10
5.3. Содержание семинаров, практических занятий	11
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	13
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	13
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю	16
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	17
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	4
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	7
10. Методические указания для обучающихся по освоению дисциплины	34
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)	36
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	36



1. Наименование дисциплины

Микросервисная архитектура

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКП-3	Способен выполнять интеграцию программных модулей и компонентов и проверять работоспособность выпусков программного продукта	Разрабатывает процедуры интеграции программных модулей	Знать: принципы микросервисной архитектуры и способы взаимодействия сервисов; подходы к интеграции программных модулей через REST API, gRPC, брокеры сообщений; методы проектирования контрактов взаимодействия между микросервисами; способы версионирования API и управления совместимостью сервисов; принципы организации CI/CD-процессов для интеграции микросервисных компонентов; методы обеспечения отказоустойчивости интеграционных сценариев (retry, circuit breaker, saga). Уметь: проектировать процедуры интеграции микросервисов в распределенной системе; разрабатывать спецификации API и правила обмена сообщениями между сервисами; настраивать маршруты взаимодействия между программными модулями; разрабатывать сценарии автоматизированной интеграции в CI/CD-конвейере; выбирать инструменты оркестрации и мониторинга интеграционных



Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
			процессов; документировать процедуры интеграции и требования к взаимодействию компонентов.
		Выполняет интеграцию программных модулей и компонентов и проверку работоспособности выпусков программного продукта	Знать: технологии контейнеризации и оркестрации микросервисов; методы интеграционного, контрактного и end-to-end тестирования; средства автоматической сборки, развертывания и выпуска микросервисных приложений; методы мониторинга, логирования и трассировки распределенных систем; критерии оценки работоспособности релизов программного продукта; подходы к rollback и blue-green/canary deployment. Уметь: выполнять интеграцию микросервисов в единую программную систему; использовать CI/CD-инструменты для автоматической сборки и развертывания релизов; проводить интеграционное и регрессионное тестирование взаимодействующих сервисов; проверять корректность работы API, очередей сообщений и сервисных зависимостей; анализировать логи, метрики и трассировки для выявления ошибок интеграции; выполнять выпуск программного продукта и подтверждать его работоспособность в тестовой и продуктивной среде.
ПКП-8	Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению информационных систем (ИС), автоматизи-	Разрабатывает модели бизнес-процессов заказчика в рамках проекта создания (модификации) ИС	Знать: нотации моделирования бизнес-процессов: BPMN 2.0, UML (Use Case, Activity), EPC; методологии описания процессов «as-is» / «to-be» и принципы их декомпозиции; подходы к выявлению требований: интервью, наблюдение, анализ документов, прототипирование; способы до-



Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
	зирующих задачи организационного управления и бизнес-процессы		кументирования требований: user stories, use cases, глоссарий, матрица трассировки. Уметь: строить модели процессов «as-is» на основе собранных требований и выявлять узкие места; проектировать модели «to-be» с учётом автоматизации и ограничений микросервисной архитектуры; транслировать бизнес-процессы в функциональные требования к сервисам и их границам (Bounded Context); согласовывать модели с заказчиком: презентовать, фиксировать правки, версионировать артефакты.
		Разрабатывает архитектуру ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	Знать: основные понятия архитектуры информационных систем (монолитная, микросервисная, клиент-серверная, сервис-ориентированная, событийно-ориентированная); типовые архитектурные стили и шаблоны (layered, pipe-and-filter, MVC, CQRS, Event Sourcing, Hexagonal и др.); нотации и методы описания архитектуры (UML, C4 model, ArchiMate, BPMN); нефункциональные требования (производительность, масштабируемость, надежность, безопасность, сопровождаемость) и способы их учёта в архитектуре; принципы проектирования (SOLID, DRY, KISS, YAGNI, separation of concerns); подходы к интеграции систем (REST, gRPC, message brokers, ESB, GraphQL); жизненный цикл создания и сопровождения ИС (ГОСТ 34, DevOps, Agile); стандарты оформления архитектурной документации (ADR – Architecture Decision



Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
			Record, техническое задание на архитектуру); инструменты моделирования и анализа архитектуры (Draw.io, Enterprise Architect, PlantUML, Miro). Уметь: анализировать функциональные и нефункциональные требования к ИС; выбирать и обосновывать архитектурный стиль и технологический стек под заданный контекст; проектировать высокоуровневую и детальную архитектуру ИС с выделением компонентов, модулей и их взаимодействий; создавать архитектурные диаграммы (контекста, контейнеров, компонентов, последовательностей); документировать архитектурные решения с помощью ADR и других форматов; оценивать влияние изменений требований на архитектуру и предлагать варианты модификации; представлять и защищать архитектурные решения перед стейкхолдерами (заказчик, разработчики, тестировщики); использовать инструменты для совместной работы над архитектурой (например, Git + Markdown + диаграммы as code).



3. Место дисциплины в структуре образовательной программы

Дисциплина «Микросервисная архитектура» относится к «Модулю "разработке распределенных приложений"».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 8 (в часах)
Общая трудоемкость дисциплины	3/108	108
Контактная работа – Аудиторные занятия	10	10
<i>Лекции</i>	<i>2</i>	<i>2</i>
<i>Семинары, практические занятия</i>	<i>8</i>	<i>8</i>
<i>Самостоятельная работа</i>	<i>98</i>	<i>98</i>
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет



5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Введение в микросервисную архитектуру

Понятие микросервиса и микросервисной архитектуры. Свойства, характерные для микросервисной архитектуры. Компонентное представление через сервисы. Гетерогенность. Куб масштабирования и микросервисы. Микросервисы как разновидность модульности. Микросервисная архитектура для FTGO. Сравнение микросервисной и сервис-ориентированной архитектур. Достоинства и недостатки микросервисной архитектуры. Язык шаблонов микросервисной архитектуры. Организация и процессы. Переход на микросервисы.

Тема 2. Стратегии декомпозиции

Определение микросервисной архитектуры приложения. Определение системных операций. Разбиение на сервисы по бизнес-возможностям. Разбиение на сервисы по проблемным областям. Декомпозиция программного продукта на микросервисы. Рекомендации по декомпозиции. Трудности при разбиении приложения на сервисы. Определение API сервисов.

Тема 3. Межпроцессное взаимодействие в микросервисной архитектуре

Обзор межпроцессного взаимодействия в микросервисной архитектуре. Стили взаимодействия. Описание API в микросервисной архитектуре. Взаимодействие на основе удаленного вызова процедур. Использование REST. Использование gRPC. Обнаружение сервисов. Взаимодействие с помощью асинхронного обмена сообщениями. Использование асинхронного обмена сообщениями для улучшения доступности.

Тема 4. Управление транзакциями в микросервисной архитектуре

Управление транзакциями с помощью повествований. Микросервисная архитектура и необходимость в распределенных транзакциях. Проблемы с распределенными транзакциями. Координация повествований.

Тема 5. Проектирование микросервисов и систем на основе микросервисной архитектуры



Шаблоны организации бизнес-логики. Изучение основ проектирования микросервисов, декомпозиции проекта на микросервисы, организации взаимодействия между компонентами сложных систем.

Тема 6. Разработка микросервисов

Изучение инструментов и подходов к разработке микросервисов, настройка взаимодействия между микросервисами. Реализация хранилища событий. Совместное использование повествований и порождения событий. Шаблоны внешних API. Проблемы с проектированием внешних API. Реализация API-шлюза. Разработка безопасных сервисов. Проектирование конфигурируемых сервисов. Развертывание сервисов с помощью пакетов для отдельных языков. Развертывание сервисов в виде виртуальных машин. Развертывание сервисов в виде контейнеров. Бессерверное развертывание сервисов. Стороннее ПО для разработки и обслуживания микросервисов. Изучение Docker, Gateway, Continuous delivery. изучение CI/CD, изучение систем мониторинга. AMQP. Socket. HTTP. HTTPS. Mercury. FTP.

Тема 7. Тестирование микросервисов

Стратегии тестирования микросервисных архитектур. Обзор методик тестирования. Трудности тестирования микросервисов. Процесс развертывания. Написание модульных тестов для сервиса. Написание интеграционных тестов. Разработка компонентных тестов. Написание сквозных тестов.



5.2. Учебно – тематический план

Заочная форма обучения

п/ п	Наименование тем (разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоятельная работа
			Общая, в т.ч.:	Лекции	Семинары, практические занятия	
1	Введение в микросервисную архитектуру	8	2	1	1	6
2	Стратегии декомпозиции	12	1	0	1	11
3	Межпроцессное взаимодействие в микросервисной архитектуре	21	1	0	1	20
4	Управление транзакциями в микросервисной архитектуре	15	2	1	1	13
5	Проектирование микросервисов и систем на основе микросервисной архитектуры	19	1	0	1	18
6	Разработка микросервисов	19	1	0	1	18
7	Тестирование микросервисов	14	2	0	2	12
В целом по дисциплине		108	10	2	8	98



5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Введение в микросервисную архитектуру	Понятие микросервиса и микросервисной архитектуры.	Занятия в интерактивной форме в виде дискуссий.
Стратегии декомпозиции	Декомпозиция программного продукта на микросервисы.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений.
Межпроцессное взаимодействие в микросервисной архитектуре	Обзор межпроцессного взаимодействия в микросервисной архитектуре.	Занятия в интерактивной форме в виде дискуссий.
Управление транзакциями в микросервисной архитектуре	Микросервисная архитектура и необходимость в распределенных транзакциях.	Занятия в интерактивной форме в виде дискуссий.
Проектирование микросервисов и систем на основе микросервисной архитектуры	Построение архитектуры программного продукта с применением стороннего ПО. Разработка простейшего микросервиса. Создание Gateway для программного продукта на основе микросервисной архитектуры. Обмен информацией между микросервисами по протоколу HTTP. Очередь сообщений как способ асинхронного взаимодействия между микросервисами. Логирование в микросервисах. Организация непрерывной доставки для микросервисов. Авторизация в программных продуктах на основе микросервисной архитектуры.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений.



Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Разработка микросервисов	Разработка микросервиса. AMQP. Socket. HTTP. HTTPS. Mercury. FTP. Тестирование ПО с микросервисной архитектурой. Docker. Gateway. Continuous delivery.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений.
Тестирование микросервисов	Тестирование ПО с микросервисной архитектурой.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений.



6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Введение в микросервисную архитектуру	Переход на микросервисы.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Стратегии декомпозиции	Определение API сервисов.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Межпроцессное взаимодействие в микросервисной архитектуре	Использование асинхронного обмена сообщениями для улучшения доступности.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Управление транзакциями в микросервисной архитектуре	Координация повествований.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Проектирование микросервисов и систем на основе микросервисной архитектуры	Организации взаимодействия между компонентами сложных систем.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Разработка микросервисов	Развертывание сервисов в виде виртуальных машин. Развертывание сервисов в виде контейнеров. Бессерверное развертывание сервисов.	Работа с учебной литературой. Разбор вопросов по теме занятия.



Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Тестирование микросервисов	Написание сквозных тестов.	Работа с учебной литературой. Разбор вопросов по теме занятия.



6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерный вариант контрольной работы

Требуется реализовать систему, состоящую из нескольких взаимодействующих друг с другом сервисов.

Общие требования:

1. Каждый сервис имеет свое собственное хранилище, если оно ему нужно. Для учебных целей можно использовать один instance базы данных, но каждый сервис работает только со своей логической базой. Запросы между базами запрещены.
2. Для межсервисного взаимодействия использовать HTTP (придерживаться RESTful). Допускается использовать и другие протоколы, например, grpc, но это требуется согласовать с преподавателем.
3. Выделить Gateway Service как единую точку входа и межсервисной коммуникации. Горизонтальные запросы между сервисами делать нельзя.
4. Код хранить на Github, для сборки использовать Github Actions.
5. Gateway Service должен запускаться на порту 8080, остальные сервисы запускать на портах 8050, 8060, 8070.
6. Каждый сервис должен быть завернут в docker.
7. В docker-compose.yml прописать сборку и запуск docker контейнеров.
8. В classroom.yml дописать шаги на сборку и прогон unit-тестов.

Пример системы для контрольной работы:

1. Система предоставляет пользователю возможность поиска и покупки билетов. При покупке билетов пользователю начисляются баллы, которые он может использовать для оплаты.

Покупка билета:

1. Запрос к Flight Service для проверки, что такой рейс существует. Если Flight Service недоступен, то запрос завершается с ошибкой.
2. Выполняется запрос к Ticket Service на создание записи о билете. Если сервис недоступен, то запрос завершается с ошибкой.
3. Если при покупке билета указан флаг paidFromBalance, то в Bonus Service выполняется запрос на списание бонусов. Иначе, выполнится запрос на пополнение бонусного счета на 10% от стоимости заказа. В любом случае в Bonus Service будет создана запись в таблице privilege_history.
4. Если запрос к Bonus Service завершился неудачей (500 ошибка или сервис недоступен), то выполняется откат операции создания заказа в Ticket Service.

Возврат билета:

1. Выполняется запрос к Ticket Service для обновления статуса билета. Если этот сервис недоступен, то весь запрос завершается ошибкой.



2. После этого выполняется запрос к Bonus Service на откат изменений в бонусном счете. Если этот сервис недоступен, то пользователю все равно отдается информация, что операция завершилась успешно, а на Gateway Service запрос ставится в очередь и повторяется пока не завершится успехом (timeout 10 секунд).

Дополнительная информация

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.



7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. «Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине».

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ПКП-3. Способен выполнять интеграцию программных модулей и компонентов и проверять работоспособность выпусков программного продукта	Разрабатывает процедуры интеграции программных модулей	Знать: принципы микросервисной архитектуры и способы взаимодействия сервисов; подходы к интеграции программных модулей через REST API, gRPC, брокеры сообщений; методы проектирования контрактов взаимодействия между микросервисами;	Задание 1 Разработать спецификацию REST API для взаимодействия двух микросервисов (заказы и склад): описать эндпоинты, форматы запросов/ответов, коды ошибок. Задание 2 Составить план интеграции микросервисов через брокер сообщений (Kafka/RabbitMQ): описать топики, схемы сообщений, очередность подключения сервисов. Задание 3 По заданной архитектурной схеме системы разработать процедуру поэтапного развёртывания сервисов с учётом их зависимостей (health-check, порядок запуска, откат при сбое). Задание 4



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		способы версионирования API и управления совместимостью сервисов; принципы организации CI/CD-процессов для интеграции микросервисных компонентов; методы обеспечения отказоустойчивости интеграционных сценариев (retry, circuit breaker, saga). Уметь: проектировать процедуры интеграции микросервисов в распределенной системе; разрабатывать спецификации API и правила обмена сообщениями	Описать процедуру версионирования API при обратно-несовместимых изменениях: стратегию миграции потребителей, сосуществование версий, сроки deprecation.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		между сервисами; настраивать маршруты взаимодействия между программными модулями; разрабатывать сценарии автоматизированной интеграции в CI/CD-конвейере; выбирать инструменты оркестрации и мониторинга интеграционных процессов; документировать процедуры интеграции и требования к взаимодействию компонентов.	
	Выполняет интеграцию программных модулей и компонен-	Знать: технологии контейнеризации и оркестрации микро-	Задание 1 Реализовать интеграцию сервиса аутентификации с API-шлюзом: настроить передачу JWT-токена, проверку прав доступа и обработку ошибок авторизации.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
	тов и проверку работоспособности выпусков программного продукта	сервисов; методы интеграционного, контрактного и end-to-end тестирования; средства автоматической сборки, развертывания и выпуска микросервисных приложений; методы мониторинга, логирования и трассировки распределенных систем; критерии оценки работоспособности релизов программного продукта; подходы к rollback и blue-green/canary deployment. Уметь: выполнять	Задание 2 Написать набор интеграционных тестов (например, на Testcontainers) для проверки совместной работы двух микросервисов: сценарии нормального потока и отказов зависимостей. Задание 3 Настроить CI/CD-пайплайн (GitHub Actions / GitLab CI) с этапами сборки, интеграционного тестирования и выкатки релиза в тестовую среду; устранить ошибки при сбое пайплайна. Задание 4 По описанию дефекта, обнаруженного после релиза, провести smoke-тестирование развёрнутого окружения, локализовать проблему и предложить исправление или откат версии.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		интеграцию микросервисов в единую программную систему; использовать CI/CD-инструменты для автоматической сборки и развертывания релизов; проводить интеграционное и регрессионное тестирование взаимодействующих сервисов; проверять корректность работы API, очередей сообщений и сервисных зависимостей; анализировать логи, метрики и трассировки для выявления ошибок интеграции; вы-	



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		полнять выпуск программного продукта и подтверждать его работоспособность в тестовой и продуктивной среде.	
ПКП-8. Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению информационных систем (ИС), автоматизирующих задачи организационного управле-	Разрабатывает модели бизнес-процессов заказчика в рамках проекта создания (модификации) ИС	Знать: нотации моделирования бизнес-процессов: BPMN 2.0, UML (Use Case, Activity), EPC; методологии описания процессов «as-is» / «to-be» и принципы их декомпозиции; подходы к выявлению требований: интервью, наблюдение, анализ документов, прототипирование; способы документи-	Задание 1 По текстовому описанию работы интернет-магазина построить BPMN-диаграмму процесса «Оформление и обработка заказа» (as-is), выделив участников, развилки и исключения. Задание 2 На основе модели as-is разработать модель to-be с учётом автоматизации через микросервисы: определить, какие шаги автоматизируются, и обозначить границы сервисов. Задание 3 По готовой BPMN-модели составить перечень user stories для команды разработки; расставить приоритеты с обоснованием по критерию бизнес-ценности. Задание 4 Обнаружить ошибки и противоречия в предложенной BPMN-модели (тупики, отсутствующие события, дублирующиеся процессы) и предложить исправленный вариант с пояснениями.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ния и бизнес-процессы		рования требований: user stories, use cases, глоссарий, матрица трассировки. Уметь: строить модели процессов «as-is» на основе собранных требований и выявлять узкие места; проектировать модели «to-be» с учётом автоматизации и ограничений микросервисной архитектуры; транслировать бизнес-процессы в функциональные требования к сервисам и их границам (Bounded Context); согласовы-	



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		вать модели с заказчиком: презентовать, фиксировать правки, версионировать артефакты.	
	Разрабатывает архитектуру ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	Знать: основные понятия архитектуры информационных систем (монолитная, микросервисная, клиент-серверная, сервис-ориентированная, событийно-ориентированная); типовые архитектурные стили и шаблоны (layered, pipe-and-filter, MVC, CQRS, Event Sourcing, Hexagonal и др.); нотации и методы опи-	Задание 1. Проектирование архитектуры под заданные требования На основе заказного кейса (например, «интернет-магазин с пиковой нагрузкой в 10 000 запросов/с и требованием 99,99% доступности») разработать архитектуру ИС, включая: выбор архитектурного стиля, схему развёртывания, основные компоненты и их взаимодействие, обоснование решений по масштабированию и отказоустойчивости. Результат представить в виде диаграмм (C4: контекст, контейнеры) и краткого пояснения. Задание 2. Анализ и рефакторинг «плохой» архитектуры Выдаётся описание и схема текущей архитектуры гипотетической ИС с явными проблемами (монолит с «богом-объектом», жёсткие интеграции, узкие места производительности). Студент должен выявить не менее 5 архитектурных недостатков, предложить способы их устранения, разработать целевую архитектуру (с помощью диаграмм) и оценить необходимые трудозатраты на переход. Задание 3. Сравнительный анализ и выбор архитектурного подхода Постановка задачи: разработать систему управления заказами для логистической компании. Даны профили нагрузки (утром низкая активность, вечером пик), требования к



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		сания архитектуры (UML, C4 model, ArchiMate, BPMN); нефункциональные требования (производительность, масштабируемость, надежность, безопасность, сопровождаемость) и способы их учёта в архитектуре; принципы проектирования (SOLID, DRY, KISS, YAGNI, separation of concerns); подходы к интеграции систем (REST, gRPC, message brokers, ESB, GraphQL); жизненный цикл создания и	надёжности и стоимости владения. Студент сравнивает два подхода (например, микросервисный и модульный монолит) по критериям: сложность разработки, масштабируемость, затраты на сопровождение, скорость развёртывания. Результат – таблица «плюсов/минусов» и обоснованная рекомендация с архитектурной схемой выбранного варианта. Задание 4. Разработка Architecture Decision Record (ADR) Студент получает описание проблемной ситуации в ходе сопровождения ИС (например, необходимость заменить монолитную базу данных на распределённую из-за роста объёмов). Он должен сформулировать архитектурное решение, оформить его по шаблону ADR (контекст, драйверы, рассмотренные варианты, выбранное решение, последствия), а также нарисовать простую диаграмму взаимодействия компонентов до и после изменения. Дополнительно – смоделировать риск-анализ выбранного решения.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		сопровождения ИС (ГОСТ 34, DevOps, Agile); стандарты оформления архитектурной документации (ADR – Architecture Decision Record, техническое задание на архитектуру); инструменты моделирования и анализа архитектуры (Draw.io, Enterprise Architect, PlantUML, Miro). Уметь: анализировать функциональные и нефункциональные требования к ИС; выбирать и обосновывать архитектурный стиль и технологиче-	



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		ский стек под заданный контекст; проектировать высокоуровневую и детальную архитектуру ИС с выделением компонентов, модулей и их взаимодействий; создавать архитектурные диаграммы (контекста, контейнеров, компонентов, последовательностей); документировать архитектурные решения с помощью ADR и других форматов; оценивать влияние изменений требований на архитектуру и предлагать варианты мо-	



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		дификации; представлять и защищать архитектурные решения перед стейкхолдерами (заказчик, разработчики, тестировщики); использовать инструменты для совместной работы над архитектурой (например, Git + Markdown + диаграммы as code).	



Примеры практико-ориентированных заданий

Задание 1.

Студенту предлагается предметная область (например, система управления заказами), для которой необходимо выделить набор микросервисов, описать их интерфейсы и реализовать взаимодействие между сервисами с использованием REST или асинхронного обмена сообщениями.

Результат выполнения:

- схема микросервисной архитектуры;
- описание API микросервисов;
- реализованные сервисы с настроенным межсервисным взаимодействием.

Задание 2.

На основе готового набора микросервисов студент выполняет контейнеризацию компонентов и организует их развертывание в тестовой среде с использованием оркестратора контейнеров.

Результат выполнения:

- конфигурационные файлы контейнеризации и развертывания;
- развернутое микросервисное приложение;
- отчет о процессе развертывания и проверке работоспособности.

Примеры вопросов для подготовки к промежуточной аттестации

Примерные вопросы для подготовки к зачету

1. Понятие микросервиса и микросервисной архитектуры
2. Свойства, характерные для микросервисной архитектуры
3. Компонентное представление через сервисы
4. Гетерогенность
5. Куб масштабирования и микросервисы
6. Микросервисы как разновидность модульности
7. Микросервисная архитектура для FTGO
8. Сравнение микросервисной и сервис-ориентированной архитектур
9. Достоинства и недостатки микросервисной архитектуры
10. Язык шаблонов микросервисной архитектуры
11. Организация и процессы
12. Переход на микросервисы
13. Определение микросервисной архитектуры приложения
14. Определение системных операций
15. Разбиение на сервисы по бизнес-возможностям
16. Разбиение на сервисы по проблемным областям



17. Декомпозиция программного продукта на микросервисы
18. Рекомендации по декомпозиции
19. Трудности при разбиении приложения на сервисы
20. Определение API сервисов
21. Обзор межпроцессного взаимодействия в микросервисной архитектуре
22. Стили взаимодействия
23. Описание API в микросервисной архитектуре
24. Взаимодействие на основе удаленного вызова процедур
25. Использование REST
26. Использование gRPC
27. Обнаружение сервисов
28. Взаимодействие с помощью асинхронного обмена сообщениями
29. Использование асинхронного обмена сообщениями для улучшения доступности
30. Управление транзакциями с помощью повествований
31. Микросервисная архитектура и необходимость в распределенных транзакциях
32. Проблемы с распределенными транзакциями
33. Координация повествований
34. Шаблоны организации бизнес-логики
35. Изучение основ проектирования микросервисов
36. Декомпозиции проекта на микросервисы
37. Организации взаимодействия между компонентами сложных систем
38. Изучение инструментов и подходов к разработке микросервисов
39. Настройка взаимодействия между микросервисами
40. Реализация хранилища событий
41. Совместное использование повествований и порождения событий
42. Шаблоны внешних API
43. Проблемы с проектированием внешних API
44. Реализация API-шлюза
45. Разработка безопасных сервисов
46. Проектирование конфигурируемых сервисов
47. Развертывание сервисов с помощью пакетов для отдельных языков
48. Развертывание сервисов в виде виртуальных машин
49. Развертывание сервисов в виде контейнеров
50. Бессерверное развертывание сервисов
51. Стороннее ПО для разработки и обслуживания микросервисов
52. Изучение Docker, Gateway, Continuous delivery



- 53. Изучение CI/CD, изучение систем мониторинга
- 54. AMQP. Socket. HTTP. HTTPS. Mercury. FTP
- 55. Стратегии тестирования микросервисных архитектур
- 56. Обзор методик тестирования
- 57. Трудности тестирования микросервисов
- 58. Процесс развертывания
- 59. Написание модульных тестов для сервиса
- 60. Написание интеграционных тестов
- 61. Разработка компонентных тестов
- 62. Написание сквозных тестов

**Промежуточная аттестация обучающихся проводится в формате тестирования*



8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Гусев, К. В. Системная и программная инженерия [Электронный ресурс] : методические указания по выполнению практических работ / Гусев К. В., Воронцов Ю. А., Михайлова Е. К., входят: Москва : РТУ МИРЭА, 2021 30 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/182487>. [БИК ID: RU-LAN-BOOK-182487]
2. Водяхо, А. И. Архитектурные решения информационных систем [Электронный ресурс] / Водяхо А. И., Выговский Л. С., Дубенецкий В. А., Цехановский В. В. 3-е изд., стер. Санкт-Петербург : Лань, 2022 356 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/254624> ISBN 978-5-507-44710-7. [БИК ID: RU-LAN-BOOK-254624]
3. Миронов, А. Н. Системная и программная инженерия [Электронный ресурс] / Миронов А. Н., Воронцов Ю. А., Михайлова Е. К., Трушин С. М. Москва : РТУ МИРЭА, 2022 129 с. Книга из коллекции РТУ МИРЭА - Психология. Педагогика <https://e.lanbook.com/book/310997>. [БИК ID: RU-LAN-BOOK-310997]

Дополнительная литература:

1. Андрианова, Е. Г. Проектная практика [Электронный ресурс] : учебно-методическое / Андрианова Е. Г., Полторак А. В. Москва : РТУ МИРЭА, 2021 166 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/218432>. [БИК ID: RU-LAN-BOOK-218432]
2. Баланов, А. Н. Построение микросервисной архитектуры и разработка высоконагруженных приложений [Электронный ресурс] : учебное пособие для вузов / Баланов А. Н. 2-е изд., стер. Санкт-Петербург : Лань, 2025 244 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/456920> ISBN 978-5-507-52652-9. [БИК ID: RU-LAN-BOOK-456920]

Нормативные правовые акты:

-

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/> (<http://library.fa.ru/files/elibfa.pdf>)



2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Электронно-библиотечная система "Университетская библиотека ОНЛАЙН"
<http://biblioclub.ru/>
4. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
5. Информационно-образовательный портал Финуниверситета: <https://org.fa.ru>
6. • Электронно-библиотечная система Znanium <http://www.znanium.ru/>
7. • Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
8. • Электронно-библиотечная система издательства Проспект
<http://ebs.prospekt.org/books>
9. СПАРК <https://spark-interfax.ru/>
10. Электронно-библиотечная система издательства "Лань" <https://e.lanbook.com/>



10. Методические указания для обучающихся по освоению дисциплины

Основные этапы работы студента по дисциплине **«Микросервисная архитектура»**:

1. Предварительная ориентировка в подлежащем изучению учебном материале по программе.
2. Ознакомление с рекомендованной учебной литературой.
3. Слушание и конспектирование лекций, а также выполнение других видов учебной работы.
4. Планирование самостоятельной работы.
5. Обобщение и систематизация информации, почерпнутой из лекций и прочитанной литературы.
6. Выполнение контрольной работы.

Студентам при подготовке контрольной работы следует использовать нормативные документы Финансового университета, Методические рекомендации по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете, утвержденные приказом Финуниверситета от 11.05.2021 г. № 1040/о (см. сайт Финансового Университета: на главной странице раздел "Наш университет" → "Единая правовая база Финуниверситета" → "Организация учебного процесса" → "Нормативные документы по самостоятельной работе").

Рекомендации по работе с учебным материалом:

1. Осознавайте наличный уровень полученных вами знаний.
2. В ситуации непонимания нужно выявить тот первичный уровень и факторы непонимания, которые стали препятствием понимания последующего.
3. Задавайте сами себе вопросы и пытайтесь ответить на них.

Рекомендации по работе на лекции и с лекционным материалом:

1. Основная задача на лекции – осмысление излагаемого в ней материала. Для этого необходимо слушать лекцию с самого начала, не упуская общих, ориентирующих в материале рассуждений и установок лектора.
2. Ведение записей на лекции важно и полезно для лучшего осмысливания материала, для сохранения информации, с целью ее дальнейшего использования.
3. Для облегчения записи рекомендуется применять сокращения повторяющихся терминов или хорошо известных понятий.

Рекомендации по работе с литературой:

1. Если возникли затруднения при разыскивании материала, по какому-либо конкретному вопросу, следует обратиться к предметному указателю, напечатанному,



как правило, в конце каждого литературного источника.

2. Предметный указатель – это алфавитный список основных научных понятий (терминов), содержание которых раскрыто в книге, рядом с термином стоят числа, обозначающие номера страниц, на которых изложен материал, относящийся к данному понятию.



11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Microsoft Office (Windows)
2. Антивирус Kaspersky

Современные профессиональные базы данных и информационные справочные системы

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»

Сертифицированные программные и аппаратные средства защиты информации

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. Учебная аудитория для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)