

**Федеральное государственное образовательное бюджетное
учреждение высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Департамент анализа данных и машинного обучения
Факультета информационных технологий и анализа больших данных**

УТВЕРЖДАЮ

Проректор по учебной и
методической работе

_____ Е.А. Каменева

25.04.2023 г.

Абашин В.Г.

Микросервисная архитектура

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки

01.03.02 – Прикладная математика и информатика,

ОП «Прикладная математика и информатика»,

ОП «Прикладное машинное обучение»

*Рекомендовано Ученым советом
Факультета информационных технологий и анализа больших данных
(протокол №31 от 18.04.2023 г.)*

*Одобрено Советом учебно-научного
Департамента анализа данных и машинного обучения
(протокол №2 от 29.03.2023 г.)*

Москва 2023

Содержание

3
3
4
5
5
5
7
8
10
10
11
12
19
19
20
21
22

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
ОП «Прикладная математика и информатика»			
ОПК-4	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	1. Использует информационно-коммуникационные технологии при поиске необходимой информации, сбора, визуализации и обработки данных.	Уметь выполнять предварительное (архитектурное) и окончательное (алгоритмическое) проектирование микросервисов Знать теоретические основы проектирования и реализации микросервисов
		2. Осуществляет рациональный выбор программного продукта в зависимости от поставленной задачи.	Уметь использовать системное мышление при проектировании архитектуры и планировании работ на проекте Знать принципы построения микросервисных архитектур, методы проектирования и реализации отказоустойчивых, масштабируемых, производительных сервисов
		3. Владеет навыками обеспечения информационной безопасности автоматизированных систем.	Уметь разрабатывать микросервисы и интегрировать их в единую систему Знать особенности интеграции микросервисов в единую систему
ПКП-4	Способность применять математический аппарат при разработке вычислительных алгоритмов для решения задач в	1. Владеет математическим аппаратом, необходимым для разработки вычислительных алгоритмов.	Уметь применять математический аппарат для разработки вычислительных алгоритмов Знать математический аппарат для разработки вычислительных алгоритмов

	области экономики и финансов	2.Разрабатывает вычислительные алгоритмы для решения задач в области экономики и финансов.	Уметь разрабатывать вычислительные алгоритмы для микросервисов Знать вычислительные алгоритмы для микросервисов
ОП «Прикладное машинное обучение»			
ПКП-6	Способность выполнять сборку модулей и компонент программной реализации моделей машинного обучения и развертывания реализации моделей машинного обучения	1. Демонстрирует знание архитектуры современных инструментальных средств реализации моделей машинного обучения и подходов к их интеграции.	Уметь выполнять предварительное (архитектурное) и окончательное (алгоритмическое) проектирование микросервисов Знать теоретические основы проектирования и реализации микросервисов
		2. Владеет практическим навыком интеграции компонент моделей машинного обучения.	Уметь использовать системное мышление при проектировании архитектуры и планировании работ на проекте Знать принципы построения микросервисных архитектур, методы проектирования и реализации отказоустойчивых, масштабируемых, производительных сервисов
		3. Владеет практическим навыком сборки и развертывания многокомпонентных моделей машинного обучения	Уметь разрабатывать микросервисы и интегрировать их в единую систему Знать особенности интеграции микросервисов в единую систему

3. Место дисциплины в структуре образовательных программ

Дисциплина «Микросервисная архитектура» относится к Циклу профиля (элективный) по направлению подготовки 01.03.02 - Прикладная математика и информатика, ОП «Прикладная математика и информатика», ОП «Прикладное машинное обучение».

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

2021, 2022 год приема / 2023 год приема и т.д.

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 7 (в часах)
Общая трудоемкость дисциплины	3/108	108
<i>Контактная работа – Аудиторные занятия</i>	<i>50</i>	<i>50</i>
<i>Лекции</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>34</i>	<i>34</i>
Самостоятельная работа	58	58
Вид текущего контроля	контрольная работа	контрольная работа
Вид промежуточной аттестации	экзамен / зачет	экзамен / 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

№ п/ п	Наименование тем (разделов) дисциплины	Трудоёмкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самостоя- тельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		
1	Введение в микросервисную архитектуру	10	4	2	2	6	Опрос, выполнение индивидуальных заданий
2	Стратегии декомпозиции	10	4	2	2	6	Опрос, выполнение индивидуальных заданий
3	Межпроцессное взаимодействие в микросервисной архитектуре	10	4	2	2	6	Опрос, выполнение индивидуальных заданий
4	Управление транзакциями в микросервисной архитектуре	12	4	2	2	8	Опрос, выполнение индивидуальных заданий
5	Проектирование микросервисов и систем на основе микросервисной архитектуры	20	12	2	10	8	Опрос, выполнение индивидуальных заданий

6	Разработка микросервисов	32	16	4	12	16	Опрос, выполнение индивидуальных заданий, контрольная работа
7	Тестирование микросервисов	14	6	2	4	8	Обсуждение, опрос контрольная работа
	В целом по дисциплине	108	50	16	34	58	Согласно учебному плану: контрольная работа
	Итого в %		46	32	68	54	

5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
Введение в микросервисную архитектуру	Понятие микросервиса и микросервисной архитектуры <i>Рекомендуемые источники: 8.[1],[2],[3]</i>	Занятия в интерактивной форме в виде дискуссий.
Стратегии декомпозиции	Декомпозиция программного продукта на микросервисы <i>Рекомендуемые источники: 8.[1],[2],[3]</i>	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Межпроцессное взаимодействие в микросервисной архитектуре	Обзор межпроцессного взаимодействия в микросервисной архитектуре <i>Рекомендуемые источники: 8.[1],[2],[3]</i>	Занятия в интерактивной форме в виде дискуссий.

Управление транзакциями в микросервисной архитектуре	Микросервисная архитектура и необходимость в распределенных транзакциях <i>Рекомендуемые источники: 8.[1],[2],[3]</i>	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 <i>Рекомендуемые источники: 8.[1],[2],[3]</i>	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Разработка микросервисов	Разработка микросервиса. AMQP. Socket. HTTP. HTTPS. Mercury. FTP. Тестирование ПО с микросервисной архитектурой. Docker. Gateway. Continuous delivery. <i>Рекомендуемые источники: 8.[1],[2],[3]</i>	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Тестирование микросервисов	Тестирование ПО с микросервисной архитектурой <i>Рекомендуемые источники: 8.[1],[2],[3]</i>	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. Выполнение и защита домашней контрольной работы
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	Развёртывание сервисов в виде виртуальных машин. Развёртывание сервисов в виде контейнеров. Бессерверное развёртывание сервисов.	Работа с учебной литературой. Разбор вопросов по теме занятия.
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 своё 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 счёте. Если этот сервис недоступен, то пользователю все равно отдаётся информация, что операция завершилась успешно, а на Gateway Service запрос ставится в очередь и повторяется пока не завершится успехом (timeout 10 секунд).

Критерии бальной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Департамента анализа данных и машинного обучения 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, знаний и умений**

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесённые с индикаторами достижения компетенции	Типовые контрольные задания
ОП «Прикладная математика и информатика»			
ОПК-4. Способность понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	1. Использует информационно-коммуникационные технологии при поиске необходимой информации, сбора, визуализации и обработки данных.	Уметь выполнять предварительное (архитектурное) и окончательное (алгоритмическое) проектирование микросервисов Знать теоретические основы проектирования и реализации микросервисов	Декомпозиция программного продукта на микросервисы Сколько должно быть микросервисов в проекте?
	2. Осуществляет рациональный выбор программного продукта в зависимости от поставленной задачи.	Уметь использовать системное мышление при проектировании архитектуры и планировании работ на проекте Знать принципы построения микросервисных архитектур, методы проектирования и реализации отказоустойчивых, масштабируемых, производительных сервисов	Какие виды тестирования используются при разработке приложения на основе микросервисной архитектуры? Для чего используется Docker?
	3. Владеет навыками обеспечения информационной безопасности автоматизированных систем.	Уметь разрабатывать микросервисы и интегрировать их в единую систему Знать особенности интеграции микросервисов в единую систему	Разработать сервис, запрашивающий с консоли ввод имени пользователя и затем отображающий на консоли приветствие данного пользователя. Разработать спецификацию в формате OpenAPI для набора веб-сервисов, реализующего следующую функциональность: Первый веб-сервис должен осуществлять управление коллекцией объектов. Второй веб-сервис должен располагаться на URL

			/demography, и реализовывать ряд дополнительных операций, связанных с вызовом API первого сервиса: /nationality/{nationality}/hair-color : вывести количество людей с заданным цветом волос в пределах указанной национальности /nationality/{nationality}/hair-color/{hair-color}/percentage/{hair-color} : вывести долю людей с заданным цветом волос в пределах указанной национальности (в процентах)
ПКП-4. Способность применять математический аппарат при разработке вычислительных алгоритмов для решения задач в области экономики и финансов	1. Владеет математическим аппаратом, необходимым для разработки вычислительных алгоритмов.	Уметь применять математический аппарат для разработки вычислительных алгоритмов Знать математический аппарат для разработки вычислительных алгоритмов	Построение архитектуры программного продукта с применением стороннего ПО Что необходимо предусмотреть при организации логирования в приложении на основе микросервисной архитектуры?
	2. Разрабатывает вычислительные алгоритмы для решения задач в области экономики и финансов.	Уметь разрабатывать вычислительные алгоритмы для микросервисов Знать вычислительные алгоритмы для микросервисов	Что такое Gateway? Для чего используется Docker Compose?
ОП «Прикладное машинное обучение»			
ПКП-6. Способность выполнять сборку модулей и компонент программной реализации моделей машинного обучения и развёртывания реализации моделей машинного обучения	1. Демонстрирует знание архитектуры современных инструментальных средств реализации моделей машинного обучения и подходов к их интеграции.	Уметь выполнять предварительное (архитектурное) и окончательное (алгоритмическое) проектирование микросервисов Знать теоретические основы проектирования и реализации микросервисов	1. Декомпозиция программного продукта на микросервисы 2. Построение архитектуры программного продукта с применением стороннего ПО 3. Что необходимо предусмотреть при организации логирования в приложении на основе микросервисной архитектуры? Сколько должно быть микросервисов в проекте?

	2. Владеет практическим навыком интеграции компонент моделей машинного обучения.	Уметь использовать системное мышление при проектировании архитектуры и планировании работ на проекте Знать принципы построения микросервисных архитектур, методы проектирования и реализации отказоустойчивых, масштабируемых, производительных сервисов	Какие виды тестирования используются при разработке приложения на основе микросервисной архитектуры? Что такое Gateway? Для чего используется Docker? Для чего используется Docker Compose?
	3. Владеет практическим навыком сборки и развёртывания многокомпонентных моделей машинного обучения	Уметь разрабатывать микросервисы и интегрировать их в единую систему Знать особенности интеграции микросервисов в единую систему	Разработать сервис, запрашивающий с консоли ввод имени пользователя и затем отображающий на консоли приветствие данного пользователя. Разработать спецификацию в формате OpenAPI для набора веб-сервисов, реализующего следующую функциональность: Первый веб-сервис должен осуществлять управление коллекцией объектов. Второй веб-сервис должен располагаться на URL /demography, и реализовывать ряд дополнительных операций, связанных с вызовом API первого сервиса: /nationality/{nationality}/hair-color : вывести количество людей с заданным цветом волос в пределах указанной национальности /nationality/{nationality}/hair-color/{hair-color}/percentage/{hair-color} : вывести долю людей с заданным цветом волос в пределах указанной национальности (в процентах)

Примерные вопросы для подготовки к экзамену / 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 удалённого 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. Развёртывание сервисов с помощью пакетов для отдельных языков
48. Развёртывание сервисов в виде виртуальных машин
49. Развёртывание сервисов в виде контейнеров
50. Бессерверное развёртывание 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 развёртывания
59. Написание модульных тестов для сервиса
60. Написание интеграционных тестов
61. Разработка компонентных тестов
62. Написание сквозных тестов

Пример экзаменационного билета

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

Департамент анализа данных и машинного обучения

Дисциплина: «Микросервисная архитектура»

Факультет информационных технологий и анализа больших данных

Форма обучения: очная

Семестр: 7

Направление подготовки: 01.03.02 – Прикладная математика и информатика.

БИЛЕТ № __

1. Координация повествований **(20 баллов)**
2. Развёртывание сервисов в виде виртуальных машин **(20 баллов)**
3. AMQP. Socket. HTTP. HTTPS. Mercury. FTP **(20 баллов)**

Подготовил

В.Г. Абашин

Утверждаю:

Заместитель руководителя департамента

В.А. Малекова

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

Основная литература:

1. Архитектурные решения информационных систем : учебник для вузов / А. И. Водяхо, Л. С. Выговский, В. А. Дубенецкий, В. В. Цехановский. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 356 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/254624> (дата обращения: 04.04.2023). — Текст: электронный.

Дополнительная литература:

2. Цифровые платформы. Методологии. Применение в бизнесе : монография / под общей редакцией Б. Б. Славина [и др.]. — Москва : Прометей, 2019. — 228 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/126738> (дата обращения: 04.04.2023). — Текст : электронный.

3. Андрианова, Е. Г. Проектная практика : учебно-методическое пособие / Е. Г. Андрианова, А. В. Полторак. — Москва : РТУ МИРЭА, 2021. — 166 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/218432> (дата обращения: 04.04.2023). — Текст: электронный.

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

1. Информационно-образовательный портал Финансового университета при Правительстве Российской Федерации <http://portal.ufrf.ru/>

2. Сайт департамента анализа данных и машинного обучения.

3. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>

4. Электронно-библиотечная система BOOK.RU <http://www.book.ru>

5. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>

6. Электронно-библиотечная система Znanium <http://www.znanium.com>

7. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>

8. Электронно-библиотечная система издательства Проспект
<http://ebs.prospekt.org/books>
9. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com/>
10. Электронная библиотека Издательского дома «Гребенников»
<https://grebennikon.ru/>
11. Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>
12. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
13. Национальная электронная библиотека <http://нэб.пф/>
14. СПАРК <https://spark-interfax.ru/>

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.

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

11.1. Комплект лицензионного программного обеспечения:

Astra Linux, Libre Office

Антивирус Kaspersky

11.2. Современные профессиональные базы данных и информационные справочные системы

Информационно-правовая система «Гарант»

Информационно-правовая система «Консультант Плюс»

Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>

Система комплексного раскрытия информации «СКРИН» -<http://www.skrin.ru/>

11.3. Сертифицированные программные и аппаратные средства защиты информации - не используются

11.4. Python 3.

11.5. StarUML.

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

Лекционные занятия проводятся в мультимедийных аудиториях, а семинарские занятия – в компьютерных классах.