

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

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

УТВЕРЖДАЮ

Проректор по учебной и  
методической работе

\_\_\_\_\_ Е.А. Каменева

23.04.2024 г.

**Абашин В.Г.**

**Микросервисная архитектура**

**Рабочая программа дисциплины**

для студентов, обучающихся по направлению подготовки:

09.03.03 – Прикладная информатика,

ОП «Инженерия данных»,

ОП «Прикладные информационные системы в экономике и финансах»,

ОП «Цифровые платформы управления предприятиями»

*Рекомендовано Ученым советом  
Факультета информационных технологий и анализа больших данных  
(протокол № 43 от 16.04.2024 г.)*

*Одобрено заседанием  
Кафедры анализа данных и машинного обучения  
(протокол № 03 от 21.03.2024 г.)*

**Москва 2024**

## 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 образовательных программ.....	5
4. Объем дисциплины(модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся.....	5
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий .....	6
5.1. Содержание дисциплины.....	6
5.2. Учебно – тематический план.....	8
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 .....	14
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	15
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины .....	23
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	24
10. Методические указания для обучающихся по освоению дисциплины .....	25
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем .....	26
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине .....	27

## 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
<b>ОП «Инженерия данных», ОП «Прикладные информационные системы в экономике и финансах»</b>			
ПКН-3	Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней	1. Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов.	<b>Уметь</b> выполнять предварительное (архитектурное) и окончательное (алгоритмическое) проектирование микросервисов  <b>Знать</b> теоретические основы проектирования и реализации микросервисов
		2. Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе.	<b>Уметь</b> использовать системное мышление при проектировании архитектуры и планировании работ на проекте  <b>Знать</b> принципы построения микросервисных архитектур, методы проектирования и реализации отказоустойчивых, масштабируемых, производительных сервисов
		3. Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии.	<b>Уметь</b> разрабатывать микросервисы и интегрировать их в единую систему  <b>Знать</b> особенности интеграции микросервисов в единую систему

ОП «Инженерия данных»			
ПКП-2	Способность разрабатывать, согласовывать и управлять исполнением технического задания и технического проекта с использованием технологий больших данных	1.Работает со стандартами, в том числе адаптирует стандарты для специфических требований больших данных.	<b>Уметь</b> работать со стандартами для специфических требований больших данных  <b>Знать</b> стандарты для специфических требований больших данных
		2.Разрабатывает технические задания и технические проекты для технологий больших данных.	<b>Уметь</b> разрабатывать технические задания и технические проекты для технологий больших данных  <b>Знать</b> принципы создания технических заданий и технических проектов
		3.Реализует управление рабочими проектами технологической инфраструктуры больших данных.	<b>Уметь</b> реализовывать управление рабочими проектами технологической инфраструктуры больших данных  <b>Знать</b> технологическую инфраструктуру больших данных
ОП «Цифровые платформы управления предприятиями»			
ПКП-4	Способность принимать участие в создании ИТ-инфраструктуры, реализовать процесс внедрения программного обеспечения и информационных систем различного уровня сложности и масштабов с использованием стандартов и технологий управления проектами	1.Демонстрирует знания стандартов и технологий управления проектами внедрений программного обеспечения и информационных систем различного уровня сложности и масштаба	<b>Уметь</b> работать со стандартами и технологиями управления проектами внедрений программного обеспечения и информационных систем  <b>Знать</b> стандарты и технологии управления проектами внедрений программного обеспечения и информационных систем
		2.Организовывает ИТ-инфраструктуру и реализует процесс внедрения программного обеспечения и информационных систем в экономике и финансах	<b>Уметь</b> организовывать ИТ-инфраструктуру и реализует процесс внедрения программного обеспечения и информационных систем в экономике и финансах  <b>Знать</b> процесс внедрения программного обеспечения и информационных систем в экономике и финансах

		3. Владеет навыками внедрения, тестирования и оценки качества программного обеспечения и информационных систем в экономике и финансах	<b>Уметь</b> внедрять, тестировать и оценивать качество программного обеспечения и информационных систем в экономике и финансах  <b>Знать</b> принципы тестирования и оценки качества программного обеспечения и информационных систем в экономике и финансах
--	--	---	---

### 3. Место дисциплины в структуре образовательных программ

Дисциплина «Микросервисная архитектура» относится к Циклу профиля (элективный) по направлению подготовки 09.03.03 – Прикладная информатика, ОП «Инженерия данных», ОП «Прикладные информационные системы в экономике и финансах», ОП «Цифровые платформы управления предприятиями».

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

ОП «Инженерия данных», ОП «Цифровые платформы управления предприятиями»

очная форма обучения

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

**ОП «Цифровые платформы управления предприятиями»**

очно-заочная, очно-заочная (ИОО) формы обучения

<b>Вид учебной работы по дисциплине</b>	<b>Всего (в з/е и часах)</b>	<b>Семестр 8 (в часах)</b>
<b>Общая трудоемкость дисциплины</b>	<b>3/108</b>	<b>108</b>
<i>Контактная работа – Аудиторные занятия</i>	<i>34</i>	<i>34</i>
<i>Лекции</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>18</i>	<i>18</i>
<b>Самостоятельная работа</b>	<b>74</b>	<b>74</b>
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет

**ОП «Прикладные информационные системы в экономике и финансах»**

заочная форма обучения (ИОО)

<b>Вид учебной работы по дисциплине</b>	<b>Всего (в з/е и часах)</b>	<b>Семестр 8 (в часах)</b>
<b>Общая трудоемкость дисциплины</b>	<b>3/108</b>	<b>108</b>
<i>Контактная работа – Аудиторные занятия</i>	<i>12</i>	<i>12</i>
<i>Лекции</i>	<i>4</i>	<i>4</i>
<i>Семинары, практические занятия</i>	<i>8</i>	<i>8</i>
<b>Самостоятельная работа</b>	<b>96</b>	<b>96</b>
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

ОП «Инженерия данных», ОП «Цифровые платформы управления предприятиями»

очная форма обучения

№ п/ п	Наименование тем (разделов) дисциплины	Трудоемкость 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,
7	Тестирование микросервисов	14	6	2	4	8	Обсуждение, опрос
	В целом по дисциплине	108	50	16	34	58	Согласно учебному плану: контрольная работа
	Итого в %		46	32	68	54	

ОП «Цифровые платформы управления предприятиями»  
очно-заочная, очно-заочная (ИОО) формы обучения

№ п/ п	Наименование тем (разделов) дисциплины	Трудоемкость 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	12	4	2	2	8	Опрос, выполнение индивидуальных заданий

2	Стратегии декомпозиции	12	4	2	2	8	Опрос, выполнение индивидуальных заданий
3	Межпроцессное взаимодействие в микросервисной архитектуре	12	4	2	2	8	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	6	2	4	14	Опрос, выполнение индивидуальных заданий
6	Разработка микросервисов	24	8	4	4	16	Опрос, выполнение индивидуальных заданий
7	Тестирование микросервисов	16	4	2	2	12	Обсуждение, опрос
	В целом по дисциплине	108	34	16	18	74	Согласно учебному плану: контрольная работа
	Итого в %		31	47	53	69	

ОП «Прикладные информационные системы в экономике и финансах»  
заочная форма обучения (ИОО)

№ п/ п	Наименование тем (разделов) дисциплины	Трудоемкость 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	14	4	2	2	10	Опрос, выполнение индивидуальных заданий
2	Стратегии декомпозиции	20	-	-	-	20	Опрос, выполнение индивидуальных заданий
3	Межпроцессное взаимодействие в микросервисной архитектуре	20	-	-	-	20	Опрос, выполнение индивидуальных заданий
4	Управление транзакциями в микросервисной архитектуре	20	-	-	-	20	Опрос, выполнение индивидуальных заданий
5	Проектирование микросервисов и систем на основе микросервисной архитектуры	14	4	2	2	10	Опрос, выполнение индивидуальных заданий
6	Разработка микросервисов	10	2	-	2	8	Опрос, выполнение индивидуальных заданий
7	Тестирование микросервисов	10	2	-	2	8	Обсуждение, опрос
	В целом по дисциплине	108	12	4	8	96	Согласно учебному плану: контрольная работа
	Итого в %		11	33	67	89	

\* объем контактной работы в очно-заочной/заочной формах обучения и индивидуальных учебных планах определяется соответствующими учебными планами. Темы, реализуемые в виде контактной работы, определяются преподавателем самостоятельно, исходя из уровня их сложности.

### 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:</i> 8.[1],[2],[5]; 9.[1]	Занятия в интерактивной форме в виде дискуссий.
Стратегии декомпозиции	Декомпозиция программного продукта на микросервисы <i>Рекомендуемые источники:</i> 8.[1],[2],[4],[5]; 9.[1], [7]	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:</i> 8.[1],[2],[5]; 9.[1], [11]	Занятия в интерактивной форме в виде дискуссий.
Управление транзакциями в микросервисной архитектуре	Микросервисная архитектура и необходимость в распределенных транзакциях <i>Рекомендуемые источники:</i> 8.[1],[2],[5]; 9.[1], [11]	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:</i> 8.[3],[4]; 9.[1], [11]	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:</i> 8.[1],[5]; 9.[1].	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Тестирование микросервисов	Тестирование ПО с микросервисной архитектурой <i>Рекомендуемые источники:</i> 8.[1],[2],[5]; 9.[1], [11]	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	Развертывание сервисов в виде виртуальных машин. Развертывание сервисов в виде контейнеров. Бессерверное развертывание 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 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).

*Критерии бальной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры 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*

# ОП «Инженерия данных»

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ПКН-3. Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней	1. Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, умение оценивать сложность алгоритмов.	<b>Уметь</b> выполнять предварительное (архитектурное) и окончательное (алгоритмическое) проектирование микросервисов  <b>Знать</b> теоретические основы проектирования и реализации микросервисов	Произвести рефакторинг уже существующей реализации алгоритма подбора близкой для числа дроби  Произвести рефакторинг уже существующей реализации алгоритма игры, которая предлагает игроку различные примеры на арифметические действия.
	2. Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе.	<b>Уметь</b> использовать системное мышление при проектировании архитектуры и планировании работ на проекте  <b>Знать</b> принципы построения микросервисных архитектур, методы проектирования и реализации отказоустойчивых, масштабируемых, производительных сервисов	Создайте файл Dockerfile, дополняющий существующий файл Docker NGINX.  Создать сервис, состоящий из 2 различных контейнеров: 1 - веб, 2 - БД
	3. Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии.	<b>Уметь</b> разрабатывать микросервисы и интегрировать их в единую систему  <b>Знать</b> 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:



			<p>Первый веб-сервис должен осуществлять управление коллекцией объектов.</p> <p>Второй веб-сервис должен располагаться на URL /demography, и реализовывать ряд дополнительных операций, связанных с вызовом API первого сервиса:</p> <p>/nationality/{nationality}/hair-color : вывести количество людей с заданным цветом волос в пределах указанной национальности</p> <p>/nationality/{nationality}/hair-color/{hair-color}/percentage/{hair-color} : вывести долю людей с заданным цветом волос в пределах указанной национальности (в процентах)</p>
<p>ПКП-2.</p> <p>Способность разрабатывать, согласовывать и управлять исполнением технического задания и технического проекта с использованием технологий больших данных</p>	<p>1.Работает со стандартами, в том числе адаптирует стандарты для специфических требований больших данных.</p>	<p><b>Уметь</b> работать со стандартами для специфических требований больших данных</p> <p><b>Знать</b> стандарты для специфических требований больших данных</p>	<p>Декомпозиция программного продукта на микросервисы</p> <p>Сколько должно быть микросервисов в проекте?</p>
	<p>2.Разрабатывает технические задания и технические проекты для технологий больших данных.</p>	<p><b>Уметь</b> разрабатывать технические задания и технические проекты для технологий больших данных</p> <p><b>Знать</b> принципы создания технические заданий и технических проектов</p>	<p>Что необходимо предусмотреть при организации логирования в приложении на основе микросервисной архитектуры?</p> <p>Дана маленькая библиотека, вычисляющая некоторые тригонометрические функции.</p> <p>Инициализируйте репозиторий в папке mat_lib. Создайте коммит и просмотрите объект коммита, найдите хэш объекта-дерева корня репозитория</p>

	3.Реализует управление рабочими проектами технологической инфраструктуры больших данных.	<b>Уметь</b> реализовывать управление рабочими проектами технологической инфраструктуры больших данных  <b>Знать</b> технологическую инфраструктуру больших данных	Какие виды тестирования используются при разработке приложения на основе микросервисной архитектуры?  Для чего используется Docker?
--	--	--	---

### ОП «Цифровые платформы управления предприятиями»

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соответствующие с индикаторами достижения компетенции	Типовые контрольные задания
ПКП-4. Способность принимать участие в создании ИТ-инфраструктуры, реализовать процесс внедрения программного обеспечения и информационных систем различного уровня сложности и масштабов с использованием стандартов и технологий управления проектами	1.Демонстрирует знания стандартов и технологий управления проектами внедрений программного обеспечения и информационных систем различного уровня сложности и масштаба	<b>Уметь</b> работать со стандартами и технологиями управления проектами внедрений программного обеспечения и информационных систем  <b>Знать</b> стандарты и технологии управления проектами внедрений программного обеспечения и информационных систем	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. Организует ИТ-инфраструктуру и реализует процесс внедрения программного обеспечения и информационных систем в экономике и финансах	<b>Уметь</b> организовывать ИТ-инфраструктуру и реализует процесс внедрения программного обеспечения и информационных систем в экономике и финансах  <b>Знать</b> процесс внедрения программного обеспечения и информационных систем в экономике и финансах	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 навыками внедрения, тестирования и оценки качества программного обеспечения и информационных систем в экономике и финансах	<p><b>Уметь</b> внедрять, тестировать и оценивать качество программного обеспечения и информационных систем в экономике и финансах</p> <p><b>Знать</b> принципы тестирования и оценки качества программного обеспечения и информационных систем в экономике и финансах</p>	<p>Разработать сервис, запрашивающий с консоли ввод имени пользователя и затем отображающий на консоли приветствие данного пользователя.</p> <p>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)</p>
--	---	--	--

### ОП «Прикладные информационные системы в экономике и финансах»

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соответствующие с индикаторами достижения компетенции	Типовые контрольные задания
ПKN-3. Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом	1. Демонстрирует знание основных алгоритмов и структур данных, умение использовать на практике простые структуры данных, умение оценивать	<b>Уметь</b> выполнять предварительное (архитектурное) и окончательное (алгоритмическое) проектирование микросервисов	1. Декомпозиция программного продукта на микросервисы 2. Построение архитектуры программного продукта с применением стороннего ПО 3. Что необходимо предусмотреть при организации

задачи и требования к ней	сложность алгоритмов.	<i><b>Знать</b></i> теоретические основы проектирования и реализации микросервисов	логирования в приложении на основе микросервисной архитектуры?  Сколько должно быть микросервисов в проекте?
	2.Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирать архитектурные решения на их основе.	<i><b>Уметь</b></i> использовать системное мышление при проектировании архитектуры и планировании работ на проекте  <i><b>Знать</b></i> 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.Создает объектно-ориентированный код, инкапсулирующий условия задачи, произвести декомпозицию задачи и спроектировать систему в пределах одной платформы или технологии.	<i><b>Уметь</b></i> разрабатывать микросервисы и интегрировать их в единую систему  <i><b>Знать</b></i> 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 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

## **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: 13.03.2024). — Текст : электронный.

2. Системная и программная инженерия : учебное пособие / А. Н. Миронов, Ю. А. Воронцов, Е. К. Михайлова, С. М. Трушин. — Москва : РТУ МИРЭА, 2022. — 129 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/310997> (дата обращения: 13.03.2024). — Текст : электронный.

### **Дополнительная литература:**

3. 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: 13.03.2024). — Текст : электронный.

4. 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: 13.03.2024). — Текст : электронный.

5. Баланов, А. Н. Построение микросервисной архитектуры и разработка высоконагруженных приложений : учебное пособие для вузов / А. Н. Баланов. — Санкт-Петербург : Лань, 2024. — 244 с. — Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/394538> (дата обращения: 13.03.2024). — 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.рф/>
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.