

---

**УТВЕРЖДЕНА**

Решением Ученого совета  
АНО ВО «Центральный университет»  
«24» июня 2025 г.  
Протокол № 2

**Рабочая программа дисциплины (модуля)  
«Архитектура программных систем»**

**Направление подготовки:** 02.04.01 Математика и компьютерные науки

**Направленность (профиль) подготовки:** Backend-разработка

**Квалификация (степень) выпускника:** магистр

**Форма обучения:** очная

**Срок освоения программы:** 2 года

**Год набора:** 2025

**Москва  
2025**

## Содержание

1. Краткая характеристика дисциплины (модуля) .....	3
2. Перечень планируемых результатов обучения.....	5
3. Тематический план.....	7
4. Содержание дисциплины (модуля).....	7
5. Учебно-методическое обеспечение .....	8
6. Материально-техническое обеспечение .....	8
7. Методические и оценочные материалы .....	10

## 1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Архитектура программных систем» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по специальности 02.04.01 Математика и компьютерные науки, профиль Backend-разработка, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Архитектура программных систем» позволяет студентам позволять понять, как функционируют компьютерные системы, включая взаимодействие аппаратного и программного обеспечения, что критично для оптимизации производительности и разработки эффективных приложений. Это знание поможет в профессиональной деятельности эффективно управлять ресурсами, обеспечивая стабильность и безопасность вычислительных систем.

### Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Backend-разработка и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 2 курсе в 4 семестре, доступна для прохождения при условии успешного завершения дисциплин (модулей) «Сетевые технологии», «Системы управления базами данных», «Многопоточное программирование».

**Цель изучения дисциплины (модуля):** формирование знаний и навыков проектирования и оценки архитектуры высоконагруженных сервисов с учетом выбора и интеграции современных компонентов и технологий.

### Задачи изучения дисциплины (модуля):

- освоить фундаментальные принципы и подходы к созданию эффективных высоконагруженных систем;
- изучить состав и роль основных элементов в архитектуре современных сервисов;
- познакомиться с разнообразием типов баз данных и систем хранения для распределенных приложений;
- развить способности к анализу вычислительных ресурсов и подбору оптимальных технологий под конкретные задачи;
- научиться разрабатывать и обосновывать архитектурные решения, включая пользовательские интерфейсы API.

### В результате освоения дисциплины (модуля) обучающийся должен:

#### *знать:*

- основные принципы конструирования высоконагруженных сервисов и подходы к их разработке;
- основные компоненты, из которых состоят современные высоконагруженные сервисы (базы данных, кэши, очереди сообщений, другие виды хранилищ данных);
- разновидности современных баз данных, key-value хранилищ, распределенных файловых систем.

#### *уметь:*

- оценивать количество физических вычислительных ресурсов, необходимых для реализации того или иного сервиса;
- выбирать подходящую версию базы данных, брокера сообщений, объектного хранилища и т. п. для решения конкретной прикладной задачи с учетом бизнес-требований.

***владеть:***

- навыками проектирования архитектуры высоконагруженных сервисов и аргументации использования тех или иных архитектурных компонентов;
- навыками разработки удобного для пользователя и поддерживаемого API разрабатываемых сервисов.

## 2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-2.	Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1.	Знает основные методологии управления проектами, ключевые этапы жизненного цикла проекта и инструменты для планирования и контроля
		УК-2.2.	Умеет разрабатывать проектную документацию, устанавливать цели и задачи проекта, а также эффективно распределять ресурсы и управлять рисками на всех этапах его реализации
		УК-2.3.	Имеет практический опыт в управлении реальными проектами, включая планирование, исполнение и завершение, а также в оценке результатов и проведении анализа успешности проекта
ОПК-1.	Способен находить, формулировать и решать актуальные и значимые проблемы прикладной и компьютерной математики	ОПК-1.1.	Знает основные методы и подходы к решению задач прикладной и компьютерной математики, включая алгоритмы, математическое моделирование и теорию оптимизации, а также современные инструменты и технологии, используемые в этой области
		ОПК-1.2.	Умеет анализировать и формулировать математические задачи, применять соответствующие методы и алгоритмы для их решения, а также интерпретировать и представлять результаты в понятной и доступной форме
		ОПК-1.3.	Имеет практический опыт работы над проектами или исследованиями в области прикладной и компьютерной математики, включая участие в конкурсах, олимпиадах или научных публикациях, где были решены актуальные и значимые задачи

ПК-2.	Способен математически корректно ставить естественнонаучные и прикладные задачи	ПК-2.1.	Знает основные методы и подходы к математическому моделированию, а также теоретические основы естественных и прикладных наук, необходимые для корректной формулировки задач
		ПК-2.2.	Умеет анализировать практические ситуации и формулировать на их основе математические модели, включая выбор адекватных методов решения и формулировку условий задачи
		ПК-2.3.	Имеет практический опыт в разработке и решении математических задач в рамках проектов или научных исследований, где были успешно поставлены и решены естественнонаучные и прикладные задачи

### 3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		<i>Очная форма</i>				
		Аудиторная работа		Контроль	Самостоя тельная работа	
Лекции	Семинары (практические занятия)					
1	Основы моделирования систем и анализ требований	8	8		34	Домашнее задание
2	Основные архитектурные компоненты программных систем	14	14		58	Домашнее задание Контрольная работа
3	Практика проектирования систем	8	8		34	Домашнее задание
	<i>Экзамен</i>			4		
	<b>Итого:</b>	<b>30</b>	<b>30</b>	<b>4</b>	<b>126</b>	
	<b>Объем дисциплины (модуля) (в ак. ч.)</b>	<b>190</b>				
	<b>Объем дисциплины (модуля) (в зач. ед.)</b>	<b>5</b>				

### 4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы моделирования систем и анализ требований	Основные компоненты для проектирования систем. Функциональные и нефункциональные требования. Сбор требований, оценка ресурсов. Патерны проектирования систем. Клиент-серверная модель. CQRS (Command Query Responsibility Segregation) Монолитная и микросервисная архитектура. Преимущества и недостатки микросервисов. Service Discovery. CAP-теорема. Уровни согласованности. Теорема PACELC.
2	Основные архитектурные компоненты программных систем	Типы баз данных. Key-value хранилища, объектные хранилища. Федерации и шардирование. Вертикальное и горизонтальное масштабирование Репликация. Master-slave репликация, Master-master репликация. Очереди сообщений. Модель publisher-subscriber Балансировщики нагрузки. Согласованное хеширование Кеширование. Content Delivery Network Ограничители нагрузки. Обратные прокси
3	Практика проектирования систем	Решение практических задач дизайна систем: проектирование сервиса генерации уникальных идентификаторов Решение практических задач дизайна систем: проектирование ленты новостей или мессенджера Решение практических задач дизайна систем: проектирование сервиса доставки заказов или такси Решение практических задач дизайна систем на других примерах высоконагруженных распределенных систем

## 5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

### **Основная литература:**

1. Site Reliability Engineering. Надежность и безотказность как в Google : практическое руководство / Б. Бейер, К. Джоунс, Дж. Петофф, Н. Р. Мёрфи. - Санкт-Петербург : Питер, 2019. - 592 с. - (Серия «Бестселлеры O'Reilly»). - ISBN 978-5-4461-0976-0. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2139467>.

2. Site Reliability Workbook: практическое применение : научно-популярное издание / сост. Б. Бейер, Н. Р. Мёрфи, Д. Рензин [и др.]. - Санкт-Петербург : Питер, 2021. - 544 с. - (Серия «Бестселлеры O'Reilly»). - ISBN 978-5-4461-1087-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1733732>.

3. Петин, В. А. API Яндекс, Google и других популярных веб-сервисов. Готовые решения для вашего сайта : практическое руководство / В. А. Петин. - Санкт-Петербург : БХВ-Петербург, 2012. - 477 с. - (Профессиональное программирование). - ISBN 978-5-9775-0743-1. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1861011>.

### **Дополнительная литература:**

1. Зыков, С. В. Архитектура информационных систем. Основы проектирования : учебник для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2025. — 260 с. — (Высшее образование). — ISBN 978-5-534-21538-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/575500>.

2. Лакамера, Д. Архитектура встраиваемых систем : разработка защищенных и подключенных устройств с примерами кода на C : практическое пособие / Д. Лакамера ; пер. с англ. В. С. Яценкова ; под науч. ред. А. Ю. Романова. – Москва : ДМК Пресс, 2023. - 334 с. – ISBN 978-5-93700-206-8. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2204236>.

## 6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных: — столами и стульями;

- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	<a href="https://elibrary.ru/defaultx.asp">https://elibrary.ru/defaultx.asp</a>
2.	База данных для IT-специалистов	<a href="https://habr.com">https://habr.com</a>
3.	База данных ScienceDirect	<a href="https://www.sciencedirect.com">https://www.sciencedirect.com</a>
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	<a href="https://minobrnauki.gov.ru/">https://minobrnauki.gov.ru/</a>
5.	Федеральный портал «Российское образование»	<a href="https://www.edu.ru/">https://www.edu.ru/</a>
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	<a href="http://window.edu.ru/">http://window.edu.ru/</a>
7.	Единая коллекция цифровых образовательных ресурсов	<a href="http://school-collection.edu.ru/">http://school-collection.edu.ru/</a>
8.	Федеральный центр информационно - образовательных ресурсов	<a href="http://fcior.edu.ru/">http://fcior.edu.ru/</a>

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
<b>Операционные системы:</b>		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
<b>Браузеры:</b>		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
<b>Офисные приложения:</b>		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
<b>Программное обеспечение для планирования и учета времени:</b>		
Toggle app	зарубежное	свободно распространяемое
<b>Системы управления проектами:</b>		
Microsoft Imagine (Project)	зарубежное	лицензионное
<b>Системы управления базами данных:</b>		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
<b>Системы резервного копирования (backup):</b>		

Acronis Backup Advanced for HyperV	зарубежное	лицензионное
<b>Справочно-правовые системы:</b>		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
<b>Средства антивирусной защиты:</b>		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
<b>Среды разработки:</b>		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
<b>Пакеты программных средств и библиотек:</b>		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
<b>Системы управления библиографической информацией:</b>		
Zotero	зарубежное	свободно распространяемое
<b>Сервисы и службы:</b>		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

## 7. Методические и оценочные материалы

### Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Архитектура программных систем» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, контрольная работа, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

*Лекция* – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

*Семинар* – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

*Домашнее задание* – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с  
Электронный документ

требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

*Контрольная работа* – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы – получить специальные знания по одной или нескольким темам дисциплины и продемонстрировать навыки их практического применения.

*Самостоятельная работа* – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

### **Система оценивания результатов обучения по дисциплине (модулю)**

#### **Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Архитектура программных систем»**

Оценивание уровня учебных достижений обучающихся по дисциплине (модулю) осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

**Промежуточная аттестация** по дисциплине (модулю) осуществляется в форме **экзамена**, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину (модуль). Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	
8	Отлично	
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать,
6	Хорошо	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

Дисциплина (модуль) «Архитектура программных систем» оценивается следующим образом:

Активность	Вес	Описание
Домашние задания	50%	За каждое из заданий можно набрать 10 баллов
Контрольная работа	20%	Письменная работа с набором задач, которые нужно решить за ограниченное время
Экзамен	30%	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

**Формула расчёта итоговой оценки по дисциплине (модулю) «Архитектура программных систем»:** « $0,5 \times$  среднее за домашние задания +  $0,2 \times$  контрольная работа +  $0,3 \times$  экзамен».

#### Текущий контроль успеваемости обучающихся по дисциплине (модулю)

##### Примерные домашние задания

###### Домашнее задание: Основы моделирования систем и анализ требований

1. Дайте определение понятию «система» и приведите три примера различных систем.
2. Опишите два метода моделирования систем и укажите, в каких случаях каждый из них наиболее эффективен.
3. Составьте список требований для простой системы (например, система онлайн-заказа еды) и классифицируйте их на функциональные и нефункциональные.
4. Проведите анализ требований, выявив возможные противоречия или неопределённости в них.

5. Подготовьте документ требований для выбранной системы, включающий описание целей, функциональных требований и ограничений.

### **Домашнее задание: Основные архитектурные компоненты программных систем**

1. Опишите, что такое архитектура программного обеспечения и почему она важна.
2. Назовите и кратко охарактеризуйте основные компоненты типичной программной системы.
3. Выберите один архитектурный шаблон (например, MVC или микросервисы) и опишите его структуру и преимущества.
4. Нарисуйте диаграмму взаимодействия компонентов для выбранной вами системы.
5. Объясните, как осуществляется взаимодействие между компонентами и какие протоколы или интерфейсы могут использоваться.

### **Домашнее задание: Практика проектирования систем**

1. Спроектируйте архитектуру сервиса генерации уникальных идентификаторов (например, на основе Snowflake), опишите компоненты (базы данных, балансировщики), оцените требования к масштабируемости и предложите механизм обработки конфликтов.
2. Разработайте архитектуру ленты новостей для социальной сети, включая компоненты для хранения данных (NoSQL), кэширования и очередей сообщений; объясните, как модель publisher-subscriber обеспечивает доставку контента миллионам пользователей.
3. Спроектируйте систему для сервиса доставки заказов (аналог Uber Eats), укажите интеграцию с геолокацией, балансировку нагрузки и механизмы авторизации; оцените нефункциональные требования, такие как latency и throughput.
4. Создайте архитектурный дизайн мессенджера с поддержкой реального времени, используя WebSockets и очереди сообщений; опишите компоненты для хранения истории чатов и управления доступом пользователей.
5. Выберите пример высоконагруженной системы (например, стриминговый сервис вроде Netflix) и спроектируйте ее архитектуру, фокусируясь на масштабировании, кэшировании и безопасности; сравните с одним из изученных примеров (генерация ID, лента новостей или доставка).

### **Примерные задания для контрольной работы**

#### **Контрольная работа**

1. Для сервиса онлайн-банкинга перечислите:
  - a) 5 функциональных требований;
  - b) 5 нефункциональных требований (SLA/латентность/доступность/безопасность/масштабируемость и т. п.).Укажите, как минимум, по одному измеримому критерию для каждого нефункционального требования.
2. Вам нужно спроектировать B2C-сервис доставки еды. Даны вводные: 1 млн MAU, 100 тыс DAU, пик 5 тыс заказов/мин, хранение истории 3 года.
  - a) Какие вопросы вы зададите заказчику, чтобы уточнить требования? (не менее 10)
  - b) Оцените порядок нагрузки: RPS на чтение/запись, объем данных в месяц, примерный рост.
  - c) Какие метрики и SLO вы предложите?
3. Нарисуйте (текстом) логическую архитектуру системы “интернет-магазин”: веб/мобайл, API, каталог, корзина, заказы, платежи, склад, уведомления. Для каждого компонента укажите ответственность и основной тип взаимодействия (sync HTTP/gRPC, async через очередь и т. д.).
4. Для каждого сценария выберите подходящий паттерн/подход и кратко объясните:
  - a) “повторить запрос без побочных эффектов при сетевой ошибке”;

- b) “обеспечить обработку события хотя бы один раз”;
  - c) “долгая операция, нужно вернуть пользователю статус”;
  - d) “защитить систему от каскадных отказов соседнего сервиса”.
5. Опишите 3 преимущества и 3 ограничения клиент–серверной модели для высоконагруженного приложения.

Дополнительно: какие изменения нужны, чтобы поддержать офлайн-режим клиента?

6. Для системы “личный кабинет оператора связи” предложите, где применить CQRS.

- a) Какие команды и какие запросы разделить?
- b) Что будет с моделью данных (одна БД или разные)?
- c) Как решать проблему согласованности между write/read моделями?

7. Компания запускает новый продукт за 2 месяца, команда 6 разработчиков, ожидается рост  $\times 10$  за год.

a) Выберите: монолит или микросервисы на старте — аргументируйте.

b) Назовите 5 рисков микросервисов и способы их снижения.

c) Когда и по каким сигналам вы бы начали декомпозицию?

8. Объясните, зачем нужен Service Discovery в микросервисах.

- a) Опишите отличие client-side и server-side discovery.
- b) Какой компонент обычно участвует (например, registry/балансировщик/sidecar)?
- c) Что происходит при “прыгающих” инстансах (autoscaling)?

9. Даны 3 сервиса:

- a) чат;
- b) корзина интернет-магазина;
- c) лента новостей.

Для каждого выберите предпочтение в CAP при сетевом разделении (CP или AP) и обоснуйте.

Дополнительно: какой уровень согласованности (strong/eventual/read-your-writes и т. п.) подходит каждому?

10. Для геораспределённой системы с пользователями в ЕС и США ответьте:

- a) Что означает PACELC и чем дополняет CAP?
- b) Для сценариев “поиск по каталогу” и “оплата” выберите приоритет (Latency vs Consistency) в случае отсутствия partition и при наличии partition. Объясните.

11. Подберите тип хранилища и обоснуйте:

- a) сессии пользователей (быстрый доступ по ключу, TTL);
- b) медиафайлы (изображения/видео);
- c) профили пользователей (поиск по атрибутам, частые чтения);
- d) события кликов (поток, аналитика, большие объёмы).

12. Есть таблица заказов, 5 млрд записей, рост 50 млн/день, запросы: по user\_id и по date.

- a) Предложите ключ шардирования и объясните компромиссы.
- b) Чем шардирование отличается от федерации (по смыслу/уровню)?
- c) Приведите пример вертикального и горизонтального масштабирования для этого кейса.

13. Для каждого типа репликации перечислите:

- a) плюсы/минусы;
- b) типичные конфликты и как их решают;
- c) что происходит при падении мастера (failover).

Затем: какой вариант вы выбрали бы для системы бронирования билетов и почему?

14. Спроектируйте обработку событий “создан заказ”: нужно отправить уведомление, списать бонусы, создать задачу на доставку, обновить аналитику.

- a) Где уместна модель pub-sub?
- b) Какие гарантии доставки нужны (at-most-once/at-least-once/exactly-once) для каждого подписчика?
- c) Как обеспечить идемпотентность обработчиков?

15. Для API с 200k RPS на чтение и 20k RPS на запись:

- a) Где поставить балансировщик нагрузки и по каким алгоритмам балансировать?

- b) Зачем согласованное хеширование (пример: кеш/шарды)?
- с) Предложите стратегию кеширования (что кешируем, TTL, инвалидация). Где применить CDN?
- d) Спроектируйте rate limiting: по чему лимитировать (IP/user/token), какие ответы/заголовки.
- е) Объясните роль reverse proxy (TLS termination, маршрутизация, сжатие, WAF и т. п.) и где он стоит в схеме.

### Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какой из следующих шаблонов проектирования используется для создания объектов? A) Singleton B) Observer C) Factory Method D) Strategy	С	ОПК-1
2.	Какой из следующих документов используется для документирования требований? A) Техническое задание B) План проекта C) Отчет о тестировании D) Руководство пользователя	А	УК-2
3.	Что из следующего относится к нефункциональным требованиям? A) Скорость обработки данных B) Возможности интеграции C) Функциональность системы D) Удобство использования	А	ПК-2
4.	Какой документ используется для сбора требований?	Техническое задание/техническое задание/ТЗ	УК-2
5.	Что такое архитектура программного обеспечения?	Структура системы	ПК-2
6.	Какой принцип связан с увеличением нагрузки на систему?	Масштабируемость/масштабируемость	ПК-2
7.	Какой из факторов важен для безопасности системы?	Шифрование	ПК-2
8.	Процесс создания предварительной версии продукта для тестирования и получения обратной связи – это ...	Прототипирование	ОПК-1
9.	Назовите основной принцип проектирования архитектуры программных систем	Модульность/модульность	ОПК-1
10.	Как называется влияние архитектуры на производительность системы?	Эффективность/эффективность	ОПК-1
11.	Назовите теорему, описывающую компромиссы в распределенных системах.	CAP теорема	УК-2
12.	Укажите два типа масштабирования, обсуждаемых в курсе.	вертикальное масштабирование, горизонтальное масштабирование	УК-2
13.	Назовите модель взаимодействия, где клиент запрашивает ресурсы у сервера.	клиент-серверная модель	УК-2

14.	Назовите метод распределения данных в базах данных для повышения производительности.	партиционирование	ОПК-1
15.	Укажите тип хранилища, предназначенного для хранения пар ключ-значение.	key-value хранилище	ОПК-1
16.	Назовите механизм, обеспечивающий согласованное распределение нагрузки в системах.	согласованное хеширование	ПК-2
17.	Укажите два принципа дизайна для управления нагрузкой в системах.	ограничители нагрузки, обратные прокси	ПК-2
18.	Назовите модель асинхронного обмена сообщениями в высоконагруженных системах.	publisher-subscriber модель	ПК-2
19.	Назовите тип клиента, который выполняет большую часть обработки на стороне клиента.	толстый клиент	УК-2
20.	Назовите механизм для обеспечения отказоустойчивости баз данных.	репликация	ОПК-1