
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол № 2

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

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Управление командами
разработки

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2026

**Москва
2026**

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	9
6. Материально-техническое обеспечение	9
7. Методические и оценочные материалы	11

1. Краткая характеристика дисциплины (модуля)

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

Изучение дисциплины (модуля) «Архитектура программных систем» позволяет сформировать целостное понимание построения сложных программных решений, способных эффективно работать в условиях высокой нагрузки и масштаба, выбирать и интегрировать подходящие технологии хранения, обработки и передачи данных, а также проектировать архитектуры.

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 Управление командами разработки и входит в обязательную часть Блока 1.

Дисциплина (модуль) доступна к изучению на 1 курсе во 2 семестре.

Цель изучения дисциплины (модуля): формирование у обучающихся системного мышления и практических навыков проектирования масштабируемых, отказоустойчивых и эффективных программных систем, способных функционировать в условиях высокой нагрузки и изменяющихся бизнес-требований.

Задачи изучения дисциплины (модуля):

- изучить архитектурные принципы построения современных высоконагруженных сервисов;
- освоить состав и назначение ключевых компонентов распределённых систем: баз данных, кэшей, очередей, объектных хранилищ и файловых систем;
- научиться оценивать потребность в вычислительных ресурсах на основе нагрузки и сценариев использования;
- развить умение выбирать и обосновывать использование конкретных технологий, в зависимости от требований к производительности, доступности и согласованности;
- овладеть практиками проектирования удобных, стабильных и поддерживаемых 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, распределённых 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;
- разрабатывать удобное для пользователя и поддерживаемое API разрабатываемых сервисов.

владеть:

- навыком проектирования архитектуры высоконагруженных сервисов;
- навыком аргументации использования тех или иных архитектурных компонентов.

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)
		Очная форма				
		Аудиторная работа		Контроль	Самостояте льная работа	
		Лекции	Семинары (практичес кие занятия)			
1	Основы моделирования систем и анализ требований	10	10		42	Домашние задания
2	Основные архитектурные компоненты программных систем	10	10		42	Домашние задания
3	Практика проектирования систем	10	10	2	42	Контрольная работа
	Экзамен			2		
Итого:		30	30	4	126	
Объем дисциплины (модуля) (в ак. ч.)		190				
Объем дисциплины (модуля) (в зач. ед.)		5				

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. Разделение ответственности команд и запросов. 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. Ключ-Значение хранилища, объектные хранилища. Федерация и шардирование. Вертикальное и горизонтальное масштабирование. Репликация. ведущий-ведомый репликация, ведущий-ведущий репликация. Очереди сообщений. Модель публикатор-подписчик. Балансировщики нагрузки. Согласованное хеширование. Кеширование. Сеть доставки контента. 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. Немцова Т. И., Голова С. Ю., Абрамова И. В. Программирование на языке высокого уровня. Программирование на языке Object Pascal / под ред. Л. Г. Гагариной. — Москва : Издательский Дом ФОРУМ, 2023. <https://znanium.ru/catalog/document?id=420771>.
2. Бейер Б., Джоунс К., Петофф Д., Мёрфи Н. Р. Site Reliability Engineering. Надежность и безотказность как в Google. — СПб. : Питер, 2019. <https://znanium.ru/catalog/document?id=442057>.

Дополнительная литература:

1. Архитектура ЭВМ и системное программное обеспечение : учебник для вузов. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт.
2. Чернышев С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебник для СПО. — Москва : Издательство Юрайт. — ISBN 978-5-534-18705-2.

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	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

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
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Архитектура программных систем:		

Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
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
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
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

Электронный документ

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 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, если иное не указано в силлабусе дисциплины:

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
9	Отлично	
8	Отлично	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
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, 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.
6	Хорошо	
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	Количество	Описание
Домашние задания	50%	4	Набор задач по темам недели
Контрольная работа	20%	1	Письменная работа с набором задач, которые нужно решить за ограниченное время
Экзамен	30%	1	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»: $0,5 \times \text{среднее за домашние задания} + 0,2 \times \text{контрольная работа} + 0,3 \times \text{экзамен}$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1

1. Проведите сбор требований для системы электронной очереди в поликлинику: выделите 5 функциональных и 5 нефункциональных требований.
2. Оцените примерное количество пользователей в пик, объём данных в день, необходимые ресурсы (CPU, RAM, дисковое пространство).
3. Выберите архитектурный стиль — монолит или микросервисы — и обоснуйте выбор.
4. Постройте диаграмму компонентов системы с указанием основных сервисов и их взаимодействия.
5. Определите, какой уровень согласованности (из CAP) приоритетен для данной системы, и объясните, почему.

Домашнее задание 2

1. Спроектируйте систему хранения аватарок пользователей: выберите тип хранилища (база данных, объектное хранилище, CDN) и обоснуйте выбор.
2. Реализуйте схему масштабирования: опишите, как будете масштабировать систему при росте нагрузки (вертикально/горизонтально).
3. Настройте репликацию базы данных: выберите тип (ведущий-ведомый или ведущий-ведущий) и опишите сценарий отказоустойчивости.
4. Добавьте кэширование: укажите, какие данные будут кэшироваться, где (Redis, CDN), и с какой стратегией инвалидации.
5. Настройте балансировщик нагрузки: опишите алгоритм распределения запросов и использование согласованного хеширования, если необходимо.

Домашнее задание 3

1. Спроектируйте сервис генерации уникальных ID (аналог Twitter Snowflake): опишите формат ID, укажите компоненты системы.
2. Реализуйте отказоустойчивость: как система будет работать при падении одного из генераторов?
3. Спроектируйте ленту новостей (типа Facebook или VK): выберите стратегию — push, pull или гибридную — и обоснуйте.
4. Определите, где будут храниться данные (базы, кэши, очереди), как будет реализована пагинация и актуальность.
5. Оцените задержки и пропускную способность системы при 1 млн активных пользователей.

Примерные задания для контрольной работы

1. Для сервиса такси оцените количество запросов в секунду при 100 000 активных водителях, если каждый отправляет своё местоположение раз в 5 секунд.
2. Выберите тип базы данных для хранения чатов в мессенджере с 50 млн пользователей. Обоснуйте выбор (SQL/NoSQL, репликация, шардирование).

- Опишите, как будет работать система, если один из микросервисов недоступен: какие механизмы отказоустойчивости можно применить?
- Настройте кэширование для API каталога товаров: укажите, какие данные кэшировать, срок жизни, стратегию обновления.
- Спроектируйте балансировку нагрузки между 10 серверами: выберите алгоритм (round-robin, least connections, consistent hashing) и объясните выбор.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какой архитектурный стиль предполагает разделение приложения на независимые сервисы, взаимодействующие через API?	микросервисная архитектура	ПК-2
2.	Какие требования описывают характеристики системы, такие как производительность, безопасность и масштабируемость?	нефункциональные	ОПК-1
3.	Какая модель репликации предполагает, что запись принимает только один узел, а остальные копируют данные?	ведущий-ведомый	ПК-2
4.	Какой механизм позволяет клиенту обнаруживать доступные экземпляры сервисов в распределённой системе?	Service Discovery	УК-2
5.	Какой компонент теоремы CAP описывает возможность системы оставаться доступной при сетевом разбиении?	доступность (A из CAP)	ПК-2
6.	Какой тип хранилища предназначен для хранения больших бинарных объектов, например изображений и видео?	объектное хранилище	ОПК-1
7.	Какой метод масштабирования подразумевает добавление новых серверов для распределения нагрузки?	горизонтальное	ПК-2
8.	Какая технология используется для ускорения доставки статического контента с помощью географически распределённых серверов?	CDN	УК-2
9.	Какое in-memory хранилище чаще всего используется для кэширования в распределённых системах?	Redis	ПК-2
10.	Какой паттерн архитектуры разделяет операции чтения и записи на разные модели данных?	CQRS	ОПК-1
11.	Какой механизм ограничивает частоту запросов от одного клиента для защиты от перегрузки?	rate limiter	ПК-2
12.	Какой тип прокси-сервера размещается перед приложением для маршрутизации и фильтрации входящих запросов?	обратный прокси	УК-2
13.	Какая модель взаимодействия используется, когда издатели публикуют сообщения, а подписчики их получают?	публикатор-подписчик	ПК-2

14.	Какой алгоритм балансировки нагрузки распределяет запросы по круговому принципу?	round-robin	ОПК-1
15.	Какой компонент распределяет входящий трафик между несколькими серверами для повышения доступности?	балансировщик нагрузки	ПК-2
16.	Какая теорема расширяет CAP, учитывая компромисс между задержкой и согласованностью при отсутствии сбоев?	PACELC	УК-2
17.	Какой метод разделения данных используется для распределения строк таблицы между узлами?	шардирование	ПК-2
18.	Какой принцип лежит в основе клиент-серверной архитектуры?	разделение ответственности	ОПК-1
19.	Какой архитектурный стиль описывает систему как единое, централизованное приложение?	монолитная архитектура	ПК-2
20.	Какая технология используется для асинхронного обмена сообщениями между микросервисами?	очередь сообщений	УК-2