

УТВЕРЖДЕНА

Приказом Ректора АНО ВО
«Центральный университет»
Е.В. Ивашкевич
от «26» июня 2025 г. № 0626.32

**Рабочая программа дисциплины (модуля)
«Архитектура программных систем»
дополнительной профессиональной программы – программы
профессиональной переподготовки «Академия data science»**

Траектория: Продуктовый менеджмент

**Москва
2025**

Содержание

| | |
|--|---|
| 1. Краткая характеристика дисциплины (модуля) | 3 |
| 2. Тематический план | 4 |
| 3. Содержание дисциплины (модуля) | 4 |
| 4. Учебно-методическое обеспечение | 5 |
| 5. Материально-техническое обеспечение | 5 |
| 6. Методические и оценочные материалы | 7 |

1. Краткая характеристика дисциплины (модуля)

Изучение дисциплины (модуля) «Архитектура программных систем» позволяет слушателям 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.

Цель изучения дисциплины (модуля): формирование у слушателей фундаментальных знаний об устройстве работы компьютера и операционных систем, которые необходимы для эффективной разработки программного обеспечения, а также для понимания более сложных аспектов компьютерных наук, таких как распределенные системы, виртуализация и кибербезопасность.

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. 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 | 8 | 8 | | 23 | Домашнее задание |
| 2 | Основные архитектурные компоненты программных систем | 8 | 8 | | 23 | Домашнее задание |
| 3 | Основные принципы дизайна высоконагруженных систем | 8 | 9 | | 24 | Домашнее задание |
| 4 | Практика проектирования систем | 9 | 9 | | 24 | Домашнее задание Проект |
| | <i>Зачет</i> | | | 4 | | |
| | Итого: | 33 | 34 | 4 | 94 | |
| | Объем дисциплины (модуля) (в ак. ч.) | 165 | | | | |

3. Содержание дисциплины (модуля)

| №п/п | Наименование раздела дисциплины (модуля) | Содержание дисциплины (модуля) по темам |
|------|--|--|
| 1 | Основы моделирования систем и анализ требований | Определение системы. Методы моделирования. Сбор и анализ требований. Документирование требований |
| 2 | Основные архитектурные компоненты программных систем | Архитектура программного обеспечения. Компоненты системы. Шаблоны проектирования. Взаимодействие компонентов |
| 3 | Основные принципы дизайна высоконагруженных систем | Масштабируемость. Отказоустойчивость. Производительность. Безопасность |
| 4 | Практика проектирования систем | Прототипирование. Тестирование и валидация. Документация и поддержка. Методологии разработки |

4. Учебно-методическое обеспечение

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

Каждый слушатель 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.

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

Основная литература:

1. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебник для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2025. — 176 с. — (Высшее образование). — ISBN 978-5-534-14383-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567946>.

2. Керриск Майкл. Linux API. Исчерпывающее руководство. — СПб.: Питер, 2019. — 1248 с. — ISBN 978-5-4461-0985-2.

Дополнительная литература:

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2022. — 1120 с. — ISBN 978-5-4461-1155-8.

5. 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 |
| Системы управления базами данных: | | |
| 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 |

6. 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, 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.

Участие в семинаре (аудиторная работа) – активная работа слушателя на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре слушателям рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

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.

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

В процессе самостоятельной работы слушатели взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи слушателя 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)

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 *зачета*.

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

| Десятибалльная оценка | Пятибалльная оценка | Оценка за зачет | Общая характеристика результата обучения по дисциплине (модулю) |
|-----------------------|---------------------|-----------------|--|
| 10 | Отлично | Зачтено | Слушатель 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. Слушатель хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами. |
| 9 | Отлично | Зачтено | |
| 8 | Отлично | Зачтено | |
| 7 | Хорошо | Зачтено | Слушатель 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. |
| 6 | Хорошо | Зачтено | |

| Десятибалльная оценка | Пятибалльная оценка | Оценка за зачет | Общая характеристика результата обучения по дисциплине (модулю) |
|-----------------------|---------------------|-----------------|---|
| | | | Слушатель хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами. |
| 5 | Удовлетворительно | Зачтено | Слушатель 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. Слушатель способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования. |
| 4 | Удовлетворительно | Зачтено | |
| 3 | Не сдан | Не зачтено | Слушатель не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы. |
| 2 | Не сдан | Не зачтено | |
| 1 | Не сдан | Не зачтено | |

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

| Активность | Вес | Описание |
|------------------|-----|--|
| Домашние задания | 50% | За каждое из заданий можно набрать 10 баллов |
| Проект | 30% | Исследовательская работа по дисциплине (модулю) и презентация результатов |
| Зачет | 20% | 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»: $\langle 0,5 \times \text{среднее за домашние задания} + 0,3 \times \text{проект} + 0,2 \times \text{зачет} \rangle$.

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: Основные принципы дизайна высоконагруженных систем

1. Опишите механизм работы ограничителя нагрузки (rate limiter) в высоконагруженной системе, приведите пример реализации с использованием алгоритма sliding window и объясните, как он предотвращает перегрузку сервиса.
2. Сравните функции обратного прокси и балансировщика нагрузки в архитектуре распределенной системы, нарисуйте схему их взаимодействия и укажите преимущества для обработки HTTP-запросов.
3. Проанализируйте модель publisher-subscriber для очередей сообщений, опишите сценарий использования в системе уведомлений и объясните, как она обеспечивает асинхронную обработку в условиях высокой нагрузки.
4. Разработайте схему авторизации и аутентификации для веб-сервиса с использованием JWT-токенов, укажите роли пользователей и механизм управления доступом на основе ролей (RBAC).
5. Предложите архитектурное решение для интеграции ограничителя нагрузки и очереди сообщений в системе, где publisher-subscriber модель используется для обработки запросов; обоснуйте выбор компонентов для обеспечения отказоустойчивости.

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).

Примерное описание и критерии оценивания к проекту

Описание проекта:

Проект представляет собой комплексную работу, в которой обучающийся должен спроектировать архитектуру высоконагруженного распределенного сервиса (например, системы генерации уникальных идентификаторов, ленты новостей, мессенджера, сервиса доставки заказов или такси), применяя знания из изученных тем. Проект включает анализ

требований (функциональных и нефункциональных), оценку ресурсов, выбор архитектурных компонентов (базы данных, хранилища, балансировщики, очереди сообщений), принципы масштабирования и безопасности, а также обоснование решений с учетом CAP-теоремы, моделей клиент-сервер и publisher-subscriber. Работа должна быть представлена в виде документа (техническое описание с диаграммами, обоснованиями и анализом альтернатив) и устной презентации, демонстрирующими способность к практическому применению теоретических знаний для решения реальных задач проектирования высоконагруженных систем.

Критерии оценивания:

- Корректность анализа требований и моделирования системы, включая применение CAP-теоремы и оценку вычислительных ресурсов;
 - Обоснованный выбор и интеграция архитектурных компонентов (базы данных, хранилища, балансировщики нагрузки, очереди сообщений);
 - Применение принципов дизайна высоконагруженных систем, таких как масштабирование, кэширование, ограничители нагрузки и механизмы безопасности;
 - Качество решения практических задач проектирования, включая сравнение альтернатив и учет нефункциональных требований;
6. Структура и ясность презентации проекта, включая диаграммы, обоснования решений и демонстрацию инновационных подходов.

Примерные вопросы для подготовки к зачету

1. Что такое система и какие основные характеристики она имеет?
2. Перечислите и опишите основные методы моделирования систем.
3. Каковы этапы сбора требований для разработки системы?
4. В чем разница между функциональными и нефункциональными требованиями?
Приведите примеры.
5. Как документировать требования к системе? Какие элементы должны быть включены в документ?
6. Какие техники можно использовать для анализа требований?
7. Каковы основные проблемы, возникающие при сборе требований?
8. Что такое диаграмма прецедентов и как она используется в моделировании систем?
9. Как можно проверить полноту и согласованность требований?
10. Каковы основные принципы управления изменениями в требованиях?
11. Что такое архитектура программного обеспечения и почему она важна?
12. Перечислите основные компоненты программной системы и их функции.
13. Что такое шаблон проектирования и какие преимущества он предоставляет?
14. Опишите различия между монолитной архитектурой и микросервисной архитектурой.
15. Как осуществляется взаимодействие между компонентами системы?
16. Что такое REST и как он используется для взаимодействия компонентов?
17. Какие факторы следует учитывать при выборе архитектурного шаблона для системы?
18. Какова роль интерфейсов в архитектуре программных систем?
19. Что такое слой абстракции и как он помогает в проектировании системы?
20. Как можно оценить качество архитектуры программного обеспечения?
21. Что такое масштабируемость и какие подходы существуют для её реализации?
22. Каковы основные принципы отказоустойчивости в высоконагруженных системах?
23. Как можно оптимизировать производительность системы? Приведите примеры.
24. Какие меры безопасности необходимо учитывать при проектировании высоконагруженных систем?
25. Что такое балансировка нагрузки и как она работает?

26. Как можно использовать кэширование для повышения производительности системы?
27. Каковы последствия недостаточной безопасности в высоконагруженных системах?
28. В чем разница между горизонтальным и вертикальным масштабированием?
29. Что такое прототипирование и какие его типы существуют?
30. Каковы ключевые этапы тестирования и валидации системы?