

Рабочая программа дисциплины

Архитектура программного обеспечения

<i>Направление подготовки</i>	Информатика и вычислительная техника
<i>Код</i>	09.03.01
<i>Направленность (профиль)</i>	Автоматизированные системы обработки информации и управления
<i>Квалификация выпускника</i>	бакалавр

1. Перечень кодов компетенций, формируемых дисциплиной в процессе освоения образовательной программы

Группа компетенций	Категория компетенций	Код
Профессиональные	-	ПК-2

2. Компетенции и индикаторы их достижения

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
ПК-2	ПК-2. Способен анализировать информационное обеспечение систем среднего и крупного масштаба сложности, реализовать инновационные решения по их интеграции, комплексному развитию и функциональной направленности	ПК-2.1. Знает методы подбора, обработки и анализа научно-технической и патентной информации по тематике исследования с использованием специализированных баз данных и информационных технологий ПК-2.2. Умеет осуществлять выбор эффективных средств и способов обеспечения качества разработки программного обеспечения, его нагрузочное, конфигурационное и интеграционное тестирование ПК-2.3. Владеет навыками создания программных продуктов, сетевых решений, в результате экспериментального исследования информационных систем, их математического описания, цифровых технологий

3. Описание планируемых результатов обучения по дисциплине

3.1. Описание планируемых результатов обучения по дисциплине

Планируемые результаты обучения по дисциплине представлены дескрипторами (знания, умения, навыки).

Дескрипторы по дисциплине	Знать	Уметь	Владеть
Код компетенции	ПК-2		
	Основы операционных систем: Знать архитектуру и основные компоненты операционных систем (например, Windows,	Конфигурировать операционные системы: Уметь устанавливать и настраивать операционные системы для веб-разработки.	Практическими навыками конфигурирования: Владеть навыками установки и

	Linux). Знать принципы работы файловых систем и управления процессами. Сетевые протоколы: Знать основные сетевые протоколы (TCP/IP, HTTP, HTTPS, FTP) и их функции. Знать принципы работы сетевых устройств (маршрутизаторов, коммутаторов, брандмауэров). Инструменты конфигурирования: Знать инструменты и утилиты для конфигурирования операционных систем и сетевых устройств (например, командная строка, PowerShell, SSH). Безопасность: Знать основные принципы и методы обеспечения безопасности операционных систем и сетевых устройств.	Уметь управлять пользователями и правами доступа в операционных системах. Настраивать сетевые устройства: Уметь конфигурировать маршрутизаторы и коммутаторы для обеспечения сетевой связности. Уметь настраивать брандмауэры для защиты веб-приложений. Использовать инструменты командной строки: Уметь использовать командную строку для выполнения задач администрирования и конфигурирования. Уметь применять скрипты для автоматизации процессов конфигурирования. Обеспечивать безопасность систем: Уметь применять методы шифрования и аутентификации для защиты данных. Уметь проводить аудит безопасности и выявлять уязвимости в системах.	настройки операционных систем в средах веб-разработки. Владеть навыками конфигурирования сетевых устройств для оптимизации работы веб-приложений. Работой с документацией: Владеть умением читать и интерпретировать техническую документацию по операционным системам и сетевым устройствам. Решением проблем: Владеть навыками диагностики и устранения неполадок в операционных системах и сетевых устройствах. Работой в команде: Владеть навыками взаимодействия с другими специалистами (разработчиками, системными администраторами) для эффективного конфигурирования и поддержки веб-приложений.
--	---	--	--

4. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Архитектура программного обеспечения» является дисциплиной по выбору учебного плана ОПОП.

Данная дисциплина взаимосвязана с другими дисциплинами, такими как:

«Алгоритмизация и программирование», «Математическая логика и дискретная математика», «Объектно-ориентированное программирование», «Теория вероятностей и математическая статистика», «Базы данных», «Вычислительные системы, сети, телекоммуникации».

В рамках освоения программы бакалавриата выпускники готовятся к решению задач профессиональной деятельности следующих типов:

Профиль (направленность) программы установлена путем ее ориентации на сферу профессиональной деятельности выпускников: Автоматизированные системы обработки информации и управления

5. Объем дисциплины

<i>Виды учебной работы</i>	<i>Формы обучения</i>
	<i>Очно-заочное</i>
Общая трудоемкость: зачетные единицы/часы	4/144
Контактная работа:	
Занятия лекционного типа	12
Занятия семинарского типа	16
Промежуточная аттестация: зачет	0,1
Самостоятельная работа (СРС)	115,9

6. Содержание дисциплины (модуля), структурированное по темам / разделам с указанием отведенного на них количества академических часов и видов учебных занятий

6.1. Распределение часов по разделам/темам и видам работы

6.1.1. Очно-заочная форма обучения

№ п/п	Раздел/тема	Виды учебной работы (в часах)							Самосто ятельная работа
		Контактная работа							
		Занятия лекционного типа		Занятия семинарского типа					
		Лекции	Иные учебные занятия	Практи ческие занятия	Семи нары	Лабора торные работы	Иные		
1.	Введение в архитектуру ПО: понятия, цели, уровни, представления	2			2				19
2.	Архитектурные стили и паттерны: обзор и критерии выбора	2			2				19
3.	Модульность.	2			3				19

	декомпозиция и границы системы							
4.	Распределённые системы и взаимодействие компонентов	2			3			19
5.	Хранение данных и интеграция в архитектуре	2			3			19
6.	Архитектурные качества, оценка и эволюция системы	2			3			20,9
	Промежуточная аттестация	0,1						
	Итого	12			16			115,9

6.2 Программа дисциплины, структурированная по темам / разделам

6.2.1 Содержание лекционного курса

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1.	Введение в архитектуру ПО: понятия, цели, уровни, представления	– Понятие архитектуры ПО, её роль в жизненном цикле системы; отличие от проектирования и кодирования. – Ключевые цели архитектуры: функциональность, производительность, масштабируемость, отказоустойчивость, поддерживаемость, безопасность. – Уровни архитектуры (бизнес-уровень, концептуальный, логический, физический) и архитектурные представления (4+1 модель Kruchten). – Архитектурные артефакты: диаграммы, ADR (Architecture Decision Record), спецификации интерфейсов.
2.	Архитектурные стили и паттерны: обзор и критерии выбора	– Основные стили: монолит, клиент-сервер, многоуровневая (N-tier), микросервисы, событийно-ориентированная, бессерверная. – Паттерны проектирования в контексте архитектуры: слои, порты и адаптеры (Hexagonal), CQRS, Saga, Circuit Breaker. – Критерии выбора стиля: требования к масштабируемости, задержкам, отказоустойчивости, стоимости владения, скорости вывода на рынок. – Формализация выбора: применение многокритериальных методов (в т. ч.

		критериев из «Алгоритмизация процессов принятия решений»). – Практические ограничения: влияние ОС, сетей и инфраструктуры на реализуемость выбранного стиля.
3.	Модульность, декомпозиция и границы системы	– Принципы модульности: связность (cohesion) и связанность (coupling); метрики для оценки. – Подходы к декомпозиции: по бизнес-доменам (DDD), по функциям, по данным, по времени жизни. – Определение границ сервисов и контрактов API; версионирование, backward compatibility. – Управление зависимостями: циклические зависимости, транзитивные зависимости, стратегии изоляции. – Инструменты: графы зависимостей, матрицы зависимостей, диаграммы компонентов и пакетов.
4.	Распределённые системы и взаимодействие компонентов	– Синхронное и асинхронное взаимодействие; протоколы (HTTP/REST, gRPC, MQ, WebSocket). – Обеспечение согласованности и отказоустойчивости: eventual consistency, идемпотентность, повторные попытки, таймауты. – Компенсация сбоев: паттерны Saga, Retry, Circuit Breaker, Bulkhead. – Влияние сетевых задержек и потерь (из «Вычислительные системы, сети, телекоммуникации») на выбор стратегий взаимодействия. – Мониторинг и трассировка распределённых вызовов (OpenTelemetry, Zipkin, Jaeger).
5.	Хранение данных и интеграция в архитектуре	– Стратегии хранения: одна БД на всё, database per service, shared database; компромиссы. – Модели данных и их соответствие архитектуре: реляционные, документные, графовые, key-value; денормализация для производительности. – Интеграционные паттерны: API Gateway, BFF, агрегация, кэширование. – Консистентность и транзакции: локальные транзакции, распределённые транзакции (2PC, SAGA), стратегии компенсации.
6.	Архитектурные качества, оценка и эволюция системы	– Нефункциональные требования (NFR) как основа архитектурных решений: производительность, надёжность, безопасность, тестируемость, развёртываемость.

		– Методы оценки архитектуры: ATAM (Architecture Tradeoff Analysis Method), чек-листы, моделирование нагрузки. – Эволюция архитектуры: рефакторинг, постепенная миграция, стратегии деплоя (blue/green, canary). – Документация архитектуры и управление изменениями; ADR как инструмент фиксации решений и альтернатив. – Связь с «Информационной безопасностью» (границы доверия, контроль доступа) и «Методологиями разработки» (CI/CD, управление релизами).
--	--	--

6.2.2 Содержание практических занятий

№ п/п	Наименование раздела дисциплины	Содержание практических занятий
1.	Введение в архитектуру ПО: понятия, цели, уровни, представления	Анализ требований и выбор архитектурных представлений: на примере учебного кейса выделить функциональные и нефункциональные требования; построить 2–3 представления (контекст-диаграмма, диаграмма контейнеров C4, диаграмма развёртывания); заполнить шаблон ADR для одного ключевого решения. Оценка по шкале 0/1/2: полнота требований, соответствие представлений целям, корректность ADR.
2.	Архитектурные стили и паттерны: обзор и критерии выбора	Сравнение архитектурных стилей для заданного сценария: выбрать 2–3 стиля (монолит, микросервисы, N-tier) и сравнить их по критериям (стоимость, масштабирование, отказоустойчивость, скорость поставки). Оформить в виде таблицы с оценкой 0/1/2 по каждому критерию и итоговым обоснованием. Использовать подходы из «Алгоритмизация процессов принятия решений».
3.	Модульность, декомпозиция и границы системы	Декомпозиция системы и управление зависимостями: на основе предметной области выделить 4–6 модулей/сервисов; определить границы и контракты API; построить граф зависимостей и выявить циклы; предложить способ устранения. Практическое задание: описать стратегию версионирования API и правила backward compatibility.
4.	Модульность, декомпозиция и границы системы	Оценка модульности и связности: рассчитать метрики cohesion/coupling для предложенной декомпозиции; обосновать, насколько границы модулей соответствуют бизнес-доменам или техническим слоям. Работа с инструментами: PlantUML/IDE для визуализации зависимостей. Оценка: корректность метрик, обоснованность границ.

5.	Распределённые системы и взаимодействие компонентов	Проектирование взаимодействия компонентов: выбрать протокол (REST/gRPC/MQ) для каждого типа взаимодействия; описать обработку ошибок, таймаутов, повторных попыток; применить паттерн Circuit Breaker или Saga для сценария сбоя. Практикум: описать последовательность сообщений и условия срабатывания компенсаций.
6.	Хранение данных и интеграция в архитектуре	Проектирование слоя хранения и интеграции: выбрать стратегию хранения (одна БД/per service/гибрид) и модель данных под архитектуру; описать схему шардирования/репликации; выбрать паттерн интеграции (API Gateway/BFF). Практическое задание: спроектировать 2–3 таблицы/коллекции и соответствующие репозитории/адаптеры.
7.	Архитектурные качества, оценка и эволюция системы	Оценка архитектуры по АТАМ-подходу: выделить ключевые сценарии использования, определить чувствительные точки и риски; построить матрицу «сценарий — атрибут качества — риск». Практическое задание: сформулировать 3–5 архитектурных рисков и меры по их снижению.
8.	Архитектурные качества, оценка и эволюция системы	Оценка архитектуры по АТАМ-подходу: выделить ключевые сценарии использования, определить чувствительные точки и риски; построить матрицу «сценарий — атрибут качества — риск». Практическое задание: сформулировать 3–5 архитектурных рисков и меры по их снижению.

6.2.3 Содержание самостоятельной работы

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1.	Введение в архитектуру ПО: понятия, цели, уровни, представления	– Понятие архитектуры ПО, её роль в жизненном цикле системы; отличие от проектирования и кодирования. – Ключевые цели архитектуры: функциональность, производительность, масштабируемость, отказоустойчивость, поддерживаемость, безопасность. – Уровни архитектуры (бизнес-уровень, концептуальный, логический, физический) и архитектурные представления (4+1 модель Kruchten). – Архитектурные артефакты: диаграммы, ADR (Architecture Decision Record), спецификации интерфейсов.
2.	Архитектурные стили и паттерны: обзор и критерии выбора	– Основные стили: монолит, клиент-сервер, многоуровневая (N-tier), микросервисы, событийно-ориентированная, бессерверная.

		– Паттерны проектирования в контексте архитектуры: слои, порты и адаптеры (Hexagonal), CQRS, Saga, Circuit Breaker. – Критерии выбора стиля: требования к масштабируемости, задержкам, отказоустойчивости, стоимости владения, скорости вывода на рынок. – Формализация выбора: применение многокритериальных методов (в т. ч. критериев из «Алгоритмизация процессов принятия решений»). – Практические ограничения: влияние ОС, сетей и инфраструктуры на реализуемость выбранного стиля.
3.	Модульность, декомпозиция и границы системы	– Принципы модульности: связность (cohesion) и связанность (coupling); метрики для оценки. – Подходы к декомпозиции: по бизнес-доменам (DDD), по функциям, по данным, по времени жизни. – Определение границ сервисов и контрактов API; версионирование, backward compatibility. – Управление зависимостями: циклические зависимости, транзитивные зависимости, стратегии изоляции. – Инструменты: графы зависимостей, матрицы зависимостей, диаграммы компонентов и пакетов.
4.	Распределённые системы и взаимодействие компонентов	– Синхронное и асинхронное взаимодействие; протоколы (HTTP/REST, gRPC, MQ, WebSocket). – Обеспечение согласованности и отказоустойчивости: eventual consistency, идемпотентность, повторные попытки, таймауты. – Компенсация сбоев: паттерны Saga, Retry, Circuit Breaker, Bulkhead. – Влияние сетевых задержек и потерь (из «Вычислительные системы, сети, телекоммуникации») на выбор стратегий взаимодействия. – Мониторинг и трассировка распределённых вызовов (OpenTelemetry, Zipkin, Jaeger).
5.	Хранение данных и интеграция в архитектуре	– Стратегии хранения: одна БД на всё, database per service, shared database; компромиссы. – Модели данных и их соответствие архитектуре: реляционные, документные, графовые, key-value; денормализация для производительности. – Интеграционные паттерны: API Gateway, BFF, агрегация, кэширование.

		– Консистентность и транзакции: локальные транзакции, распределённые транзакции (2PC, SAGA), стратегии компенсации.
6.	Архитектурные качества, оценка и эволюция системы	– Нефункциональные требования (NFR) как основа архитектурных решений: производительность, надёжность, безопасность, тестируемость, развёртываемость. – Методы оценки архитектуры: ATAM (Architecture Tradeoff Analysis Method), чек-листы, моделирование нагрузки. – Эволюция архитектуры: рефакторинг, постепенная миграция, стратегии деплоя (blue/green, canary). – Документация архитектуры и управление изменениями; ADR как инструмент фиксации решений и альтернатив. – Связь с «Информационной безопасностью» (границы доверия, контроль доступа) и «Методологиями разработки» (CI/CD, управление релизами).

7. Текущий контроль по дисциплине (модулю) в рамках учебных занятий

В рамках текущего контроля преподаватель самостоятельно может проводить следующие мероприятия:

№ п/п	Контролируемые разделы (темы)	Наименование оценочного средства
1.	Введение в архитектуру ПО: понятия, цели, уровни, представления	Опрос, информационный проект.
2.	Архитектурные стили и паттерны: обзор и критерии выбора	Опрос, информационный проект, тестирование.
3.	Модульность, декомпозиция и границы системы	Опрос, информационный проект.
4.	Модульность, декомпозиция и границы системы	Опрос, информационный проект.
5.	Распределённые системы и взаимодействие компонентов	Опрос, информационный проект, тестирование.
6.	Хранение данных и интеграция в архитектуру	Опрос, информационный проект.
7.	Архитектурные качества, оценка и эволюция системы	Опрос, Опрос, информационный проект.
8.	Архитектурные качества, оценка и эволюция системы	Опрос, тестирование.

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

8.1. Основная литература:

1. Техническое и программное обеспечение вычислительных машин и систем : учебное пособие / О. В. Конюхова, Э. А. Кравцова, П. В. Лукьянов, А. Ю. Ужаринский. — Москва, Вологда : Инфра-Инженерия, 2023. — 200 с. — ISBN 978-5-9729-1186-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/132885.html>
2. Долженко, А. И. Технологии командной разработки программного обеспечения информационных систем : учебное пособие / А. И. Долженко. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 300 с. — ISBN 978-5-4497-2486-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/133985.html>
3. Фоминых, Е. И. Арифметико-логические основы вычислительной техники : учебное пособие / Е. И. Фоминых, Т. Е. Фоминых, Ю. Л. Пархоменко. — 2-е изд. — Минск : Республиканский институт профессионального образования (РИПО), 2022. — 224 с. — ISBN 978-985-895-027-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/134131.html>
4. Полетайкин, А. Н. Конфигурирование, управление и мониторинг ИТ-инфраструктуры. DevOps : учебное пособие для СПО / А. Н. Полетайкин, Д. А. Демчук. — Саратов : Профобразование, 2026. — 162 с. — ISBN 978-5-4488-3049-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/161959.html>
5. Извозчикова, В. В. Организация ЭВМ и систем : учебное пособие / В. В. Извозчикова. — Оренбург : Оренбургский государственный университет, ЭБС АСВ, 2024. — 159 с. — ISBN 978-5-7410-3188-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/153061.html> (дата обращения: 30.06.2026). — Режим доступа: для авторизир. Пользователей

1.

2. 8.2. Дополнительная литература:

1. Архитектура ЭВМ и систем : учебное пособие / Ю. Ю. Громов, О. Г. Иванова, М. Ю. Серегин [и др.]. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2012. — 200 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/64069.html>
2. Архитектура и технологии IBM @Server zSeries : учебное пособие / В. А. Варфоломеев, Э. К. Лецкий, М. И. Шамров, В. В. Яковлев ; под редакцией Э. К. Лецкого, В. В. Яковлева. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 637 с. — ISBN 978-5-4497-3468-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/142264.html>
3. Пьявченко, А. О. Архитектура, основы программирования и применения AVR-микроконтроллеров и ARM-микросистем. В 3 частях. Ч.3 : учебное пособие / А. О.

Пьявченко, В. Н. Пуховский. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2022. — 151 с. — ISBN 978-5-9275-4102-7 (ч.3), 978-5-9275-3429-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/125697.html>

10. Методические указания для обучающихся по освоению дисциплины (модуля)

Успешное освоение данного курса базируется на рациональном сочетании нескольких видов учебной деятельности – лекций, семинарских занятий, самостоятельной работы. При этом самостоятельную работу следует рассматривать одним из главных звеньев полноценного высшего образования, на которую отводится значительная часть учебного времени.

Самостоятельная работа студентов складывается из следующих составляющих:

1. Работа с основной и дополнительной литературой, с материалами интернета и конспектами лекций;
2. Внеаудиторная подготовка к контрольным работам, выполнение докладов, рефератов и курсовых работ;
3. Выполнение самостоятельных практических работ;
4. Подготовка к экзаменам (зачетам) непосредственно перед ними.

Для правильной организации работы необходимо учитывать порядок изучения разделов курса, находящихся в строгой логической последовательности. Поэтому хорошее усвоение одной части дисциплины является предпосылкой для успешного перехода к следующей. Задания, проблемные вопросы, предложенные для изучения дисциплины, в том числе и для самостоятельного выполнения, носят междисциплинарный характер и базируются, прежде всего, на причинно-следственных связях между компонентами окружающего нас мира. Важным составляющим в изучении данного курса является решение ситуационных задач и работа над проблемно-аналитическими заданиями, что предполагает знание соответствующей научной терминологии и т.д.

Для лучшего запоминания материала целесообразно использовать индивидуальные особенности и разные виды памяти: зрительную, слуховую, ассоциативную. Успешному запоминанию также способствует приведение ярких свидетельств и наглядных примеров. Учебный материал должен постоянно повторяться и закрепляться.

При выполнении докладов, творческих, информационных, исследовательских проектов особое внимание следует обращать на подбор источников информации и методику работы с ними.

Для успешной сдачи экзамена (зачета) рекомендуется соблюдать следующие правила:

1. Подготовка к экзамену (зачету) должна проводиться систематически, в течение всего семестра.
2. Интенсивная подготовка должна начаться не позднее, чем за месяц до экзамена.
3. Время непосредственно перед экзаменом (зачетом) лучше использовать таким образом, чтобы оставить последний день свободным для повторения курса в целом, для систематизации материала и доработки отдельных вопросов.

На экзамене высокую оценку получают студенты, использующие данные, полученные в процессе выполнения самостоятельных работ, а также использующие собственные выводы на основе изученного материала.

Учитывая значительный объем теоретического материала, студентам рекомендуется регулярное посещение и подробное конспектирование лекций.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при

необходимости)

1. Microsoft Windows Server;
 2. Семейство ОС Microsoft Windows;
 3. Libre Office свободно распространяемый офисный пакет с открытым исходным кодом;
 4. Информационно-справочная система: Система КонсультантПлюс (КонсультантПлюс);
 5. Информационно-правовое обеспечение Гарант: Электронный периодический справочник «Система ГАРАНТ» (Система ГАРАНТ);
- Перечень используемого программного обеспечения указан в п.12 данной рабочей программы дисциплины.

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

12.1. Учебная аудитория для проведения учебных занятий, предусмотренных образовательной программой, оснащенная оборудованием и техническими средствами обучения.

Специализированная мебель:

Комплект учебной мебели (стол, стул) по количеству обучающихся; комплект мебели для преподавателя; доска (маркерная).

Технические средства обучения:

Компьютер в сборе для преподавателя, проектор, экран, колонки

Перечень лицензионного программного обеспечения, в том числе отечественного производства:

Windows 10, КонсультантПлюс, Kaspersky Endpoint Security.

Перечень свободно распространяемого программного обеспечения:

Яндекс Браузер, LibreOffice, МТС Линк.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду ММУ.

12.2. Помещение для самостоятельной работы обучающихся.

Специализированная мебель:

Комплект учебной мебели (стол, стул) по количеству обучающихся; комплект мебели для преподавателя; доска (маркерная).

Технические средства обучения:

Компьютер в сборе для преподавателя; компьютеры в сборе для обучающихся; колонки; проектор, экран.

Перечень лицензионного программного обеспечения, в том числе отечественного производства:

Windows Server 2016, Windows 10, Microsoft Office, КонсультантПлюс, Kaspersky Endpoint Security.

Перечень свободно распространяемого программного обеспечения:

Яндекс Браузер, LibreOffice, МТС Линк, Gimp, Paint.net, AnyLogic, Inkscape.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду ММУ.

13. Образовательные технологии, используемые при освоении дисциплины

Для освоения дисциплины используются как традиционные формы занятий – лекции (типы лекций – установочная, вводная, текущая, заключительная, обзорная; виды лекций – проблемная, визуальная, лекция конференция, лекция консультация); и

семинарские (практические) занятия, так и активные и интерактивные формы занятий - деловые и ролевые игры, решение ситуационных задач и разбор конкретных ситуаций.

На учебных занятиях используются технические средства обучения мультимедийной аудитории: компьютер, монитор, колонки, настенный экран, проектор, микрофон, пакет программ Microsoft Office для демонстрации презентаций и медиафайлов, видеопроектор для демонстрации слайдов, видеосюжетов и др. Тестирование обучаемых может осуществляться с использованием компьютерного оборудования университета.

13.1. В освоении учебной дисциплины используются следующие традиционные образовательные технологии:

- чтение проблемно-информационных лекций с использованием доски и видеоматериалов;
- семинарские занятия для обсуждения, дискуссий и обмена мнениями;
- контрольные опросы;
- консультации;
- самостоятельная работа студентов с учебной литературой и первоисточниками;
- подготовка и обсуждение рефератов (проектов), презентаций (научно-исследовательская работа);
- тестирование по основным темам дисциплины.

13.2. Активные и интерактивные методы и формы обучения

Из перечня видов: («мозговой штурм», анализ НПА, анализ проблемных ситуаций, анализ конкретных ситуаций, инциденты, имитация коллективной профессиональной деятельности, разыгрывание ролей, творческая работа, связанная с освоением дисциплины, ролевая игра, круглый стол, диспут, беседа, дискуссия, мини-конференция и др.) используются следующие:

- диспут;
- анализ проблемных, творческих заданий, ситуационных задач;
- ролевая игра;
- круглый стол;
- мини-конференция;
- дискуссия;
- беседа.

13.3. Особенности обучения инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ)

При организации обучения по дисциплине учитываются особенности организации взаимодействия с инвалидами и лицами с ограниченными возможностями здоровья (далее – инвалиды и лица с ОВЗ) с целью обеспечения их прав. При обучении учитываются особенности их психофизического развития, индивидуальные возможности и при необходимости обеспечивается коррекция нарушений развития и социальная адаптация указанных лиц.

Выбор методов обучения определяется содержанием обучения, уровнем методического и материально-технического обеспечения, особенностями восприятия учебной информации студентами с инвалидностью и студентами с ограниченными возможностями здоровья и т.д. В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе.

При обучении лиц с ограниченными возможностями здоровья электронное обучение и дистанционные образовательные технологии предусматривают возможность приема-передачи информации в доступных для них формах.

Обучающиеся из числа лиц с ограниченными возможностями здоровья обеспечены печатными и электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

Автономная некоммерческая организация высшего образования
«МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ»

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ**

Архитектура программного обеспечения

<i>Направление подготовки</i>	Информатика и вычислительная техника
<i>Код</i>	09.03.01
<i>Направленность (профиль)</i>	Автоматизированные системы обработки информации и управления
<i>Квалификация выпускника</i>	бакалавр

1. Перечень кодов компетенций, формируемых дисциплиной в процессе освоения образовательной программы

Группа компетенций	Категория компетенций	Код
Профессиональные	-	ПК-2

2. Компетенции и индикаторы их достижения

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
ПК-2	ПК-2. Способен анализировать информационное обеспечение систем среднего и крупного масштаба сложности, реализовать инновационные решения по их интеграции, комплексному развитию и функциональной направленности	ПК-2.1. Знает методы подбора, обработки и анализа научно-технической и патентной информации по тематике исследования с использованием специализированных баз данных и информационных технологий ПК-2.2. Умеет осуществлять выбор эффективных средств и способов обеспечения качества разработки программного обеспечения, его нагрузочное, конфигурационное и интеграционное тестирование ПК-2.3. Владеет навыками создания программных продуктов, сетевых решений, в результате экспериментального исследования информационных систем, их математического описания, цифровых технологий

3. Описание планируемых результатов обучения по дисциплине

3.1. Описание планируемых результатов обучения по дисциплине

Планируемые результаты обучения по дисциплине представлены дескрипторами (знания, умения, навыки).

Дескрипторы по дисциплине	Знать	Уметь	Владеть
Код компетенции	ПК-2		
	Основы операционных систем: Знать архитектуру и основные компоненты операционных систем (например, Windows, Linux).	Конфигурировать операционные системы: Уметь устанавливать и настраивать операционные системы для веб-разработки. Уметь управлять	Практическими навыками конфигурирования: Владеть навыками установки и настройки

	Знать принципы работы файловых систем и управления процессами. Сетевые протоколы: Знать основные сетевые протоколы (TCP/IP, HTTP, HTTPS, FTP) и их функции. Знать принципы работы сетевых устройств (маршрутизаторов, коммутаторов, брандмауэров). Инструменты конфигурирования: Знать инструменты и утилиты для конфигурирования операционных систем и сетевых устройств (например, командная строка, PowerShell, SSH). Безопасность: Знать основные принципы и методы обеспечения безопасности операционных систем и сетевых устройств.	пользователями и правами доступа в операционных системах. Настраивать сетевые устройства: Уметь конфигурировать маршрутизаторы и коммутаторы для обеспечения сетевой связности. Уметь настраивать брандмауэры для защиты веб-приложений. Использовать инструменты командной строки: Уметь использовать командную строку для выполнения задач администрирования и конфигурирования. Уметь применять скрипты для автоматизации процессов конфигурирования. Обеспечивать безопасность систем: Уметь применять методы шифрования и аутентификации для защиты данных. Уметь проводить аудит безопасности и выявлять уязвимости в системах.	операционных систем в средах веб-разработки. Владеть навыками конфигурирования сетевых устройств для оптимизации работы веб-приложений. Работой с документацией: Владеть умением читать и интерпретировать техническую документацию по операционным системам и сетевым устройствам. Решением проблем: Владеть навыками диагностики и устранения неполадок в операционных системах и сетевых устройствах. Работой в команде: Владеть навыками взаимодействия с другими специалистами (разработчиками, системными администраторами и) для эффективного конфигурирования и поддержки веб-приложений.
--	--	---	--

3.2. Критерии оценки знаний студентов

Шкала оценивания	Индикаторы достижения	Показатели оценивания результатов обучения
------------------	-----------------------	--

ОТЛИЧНО/Зачтено	Знает:	- студент глубоко и всесторонне усвоил материал, уверенно, логично, последовательно и грамотно его излагает, опираясь на знания основной и дополнительной литературы, - на основе системных научных знаний делает квалифицированные выводы и обобщения, свободно оперирует категориями и понятиями.
	Умеет:	- студент умеет самостоятельно и правильно решать учебно-профессиональные задачи или задания, уверенно, логично, последовательно и аргументировано излагать свое решение, используя научные понятия, ссылаясь на нормативную базу.
	Владеет:	- студент владеет рациональными методами (с использованием рациональных методик) решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.; При решении продемонстрировал навыки - выделения главного, - связкой теоретических положений с требованиями руководящих документов, - изложения мыслей в логической последовательности, - самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.
ХОРОШО/Зачтено	Знает:	- студент твердо усвоил материал, достаточно грамотно его излагает, опираясь на знания основной и дополнительной литературы, - затрудняется в формулировании квалифицированных выводов и обобщений, оперирует категориями и понятиями, но не всегда правильно их верифицирует.
	Умеет:	- студент умеет самостоятельно и в основном правильно решать учебно-профессиональные задачи или задания, уверенно, логично, последовательно и аргументировано излагать свое решение, не в полной мере используя научные понятия и ссылки на нормативную базу.
	Владеет:	- студент в целом владеет рациональными методами решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.; При решении смог продемонстрировать достаточность, но не глубинность навыков - выделения главного, - изложения мыслей в логической последовательности. - связки теоретических положений с требованиями руководящих документов, - самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.
УДОВЛЕТВОРИТЕЛЬНО/Зачтено	Знает:	- студент ориентируется в материале, однако затрудняется в его изложении; - показывает недостаточность знаний основной и дополнительной литературы; - слабо аргументирует научные положения; - практически не способен сформулировать выводы и обобщения;

		- частично владеет системой понятий.
	Умеет:	- студент в основном умеет решить учебно-профессиональную задачу или задание, но допускает ошибки, слабо аргументирует свое решение, недостаточно использует научные понятия и руководящие документы.
	Владеет:	- студент владеет некоторыми рациональными методами решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.; При решении продемонстрировал недостаточность навыков - выделения главного, - изложения мыслей в логической последовательности. - связки теоретических положений с требованиями руководящих документов, - самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.
НЕУДОВЛЕТВОРИТЕЛЬНО/Не зачтено	Знает:	- студент не усвоил значительной части материала; - не может аргументировать научные положения; - не формулирует квалифицированных выводов и обобщений; - не владеет системой понятий.
	Умеет:	студент не показал умение решать учебно-профессиональную задачу или задание.
	Владеет:	не выполнены требования, предъявляемые к навыкам, оцениваемым “удовлетворительно”.

При ответе на вопросы в рамках прохождения промежуточной аттестации (зачет/ зачет с оценкой/ экзамен) допускается вольная формулировка ответа, по смыслу раскрывающая содержание ответа, указанного в фонде оценочных средств, в качестве верного ответа.

4. Типовые контрольные задания (закрытого, открытого и иного типа) для проведения промежуточной аттестации, необходимые для оценки достижения компетенции, соотнесенной с результатами обучения по дисциплине

ТЕСТИРОВАНИЕ

6 СЕМЕСТР

ПК-2

1. Что в первую очередь отражает архитектура ПО (в отличие от детального проектирования и кода)?

А) Синтаксис языка программирования.

Б) Алгоритмы отдельных функций.

В) Высокоуровневую структуру системы, ключевые решения и компромиссы.

Г) Детали реализации классов и методов.

2. Модель «4+1» Крухтена включает представления: логическое, процессное, физическое, представление разработки и...

А) представление тестирования.

Б) сценарное (use case) представление.

В) представление документации.

Г) представление маркетинга.

3. ADR (Architecture Decision Record) нужен для:

А) фиксации принятого архитектурного решения, альтернатив, критериев выбора и последствий.

Б) хранения исходного кода модулей.

В) описания пользовательского интерфейса.

Г) составления технического задания.

4. Высокая связанность (coupling) между модулями обычно считается:

А) положительным признаком, повышающим производительность.

Б) негативным признаком, снижающим поддерживаемость и усложняющим изменения.

В) нейтральным фактором, не влияющим на архитектуру.

Г) обязательным требованием для микросервисов.

5. Для сценария с частыми релизами, независимой разработкой команд и возможностью масштабировать отдельные функции лучше подходит:

А) монолит.

Б) микросервисная архитектура.

В) событийно-ориентированная архитектура без сервисов.

Г) бессерверная архитектура для всех компонентов без исключений.

6. Архитектурный стиль «клиент-сервер» предполагает:

А) разделение на клиентскую часть (интерфейс/логика взаимодействия) и серверную часть (бизнес-логика/данные).

Б) отсутствие сервера, все вычисления выполняются на стороне клиента.

В) только веб-интерфейс без бэкенда.

Г) использование исключительно REST API без других протоколов.

7. Паттерн «Hexagonal Architecture» (порты и адаптеры) позволяет:

А) отделить бизнес-логику от инфраструктуры и внешних фреймворков, упрощая тестирование и замену технологий.

Б) автоматически масштабировать приложение в облаке.

В) полностью исключить зависимости между модулями.

Г) ускорить выполнение отдельных запросов за счёт кэширования.

8. При выборе архитектурного стиля для ИТ-системы в первую очередь учитывают:
- А) требования к функциональности, производительности, масштабируемости, отказоустойчивости и стоимости владения.**
 - Б) личные предпочтения разработчиков.
 - В) количество строк кода в прототипе.
 - Г) популярность фреймворка в социальных сетях.
9. В подходе Domain-Driven Design (DDD) границы сервисов чаще всего определяют по:
- А) бизнес-доменам и ограниченным контекстам (bounded contexts).**
 - Б) техническим слоям (контроллеры, сервисы, репозитории).
 - В) языкам программирования, на которых написан код.
 - Г) количеству разработчиков в команде.
10. Циклические зависимости между модулями:
- А) считаются антипаттерном и требуют устранения (например, через выделение общего модуля или инверсию зависимостей).**
 - Б) являются нормой и не влияют на поддерживаемость.
 - В) ускоряют сборку проекта.
 - Г) обязательны для микросервисной архитектуры.
11. Версионирование API необходимо для:
- А) обеспечения backward compatibility и возможности независимого развития клиентов и сервисов.**
 - Б) увеличения количества запросов к серверу.
 - В) автоматического масштабирования системы.
 - Г) шифрования трафика.
12. Метрика «связность модуля» (cohesion) отражает:
- А) насколько задачи внутри модуля связаны общей целью (высокая связность — положительный признак).**
 - Б) количество внешних зависимостей модуля
 - В) скорость выполнения модуля.
 - Г) объём кода в модуле.
13. Для сценариев с высокой нагрузкой и необходимостью гарантированной доставки сообщений чаще используют:
- А) синхронный HTTP/REST без очередей.
 - Б) асинхронную передачу через брокер сообщений (MQ) с гарантиями доставки.**
 - В) WebSocket для всех типов запросов.
 - Г) gRPC без повторных попыток.
14. Паттерн Circuit Breaker применяется для:
- А) предотвращения каскадных отказов: при сбоях зависимого сервиса запросы временно блокируются или перенаправляются на запасной вариант.**
 - Б) ускорения обработки запросов за счёт параллельных вызовов.
 - В) шифрования данных между сервисами.
 - Г) автоматического масштабирования контейнеров.
15. Eventual consistency (согласованность в конечном счёте) означает, что:

- А) данные могут временно различаться на разных узлах, но со временем становятся одинаковыми.**
Б) все узлы всегда содержат одинаковые данные в каждый момент времени.
В) данные никогда не синхронизируются между узлами.
Г) согласованность обеспечивается только на уровне базы данных, а не сервисов.

16. Для трассировки распределённых вызовов (distributed tracing) применяют:

- А) OpenTelemetry, Zipkin, Jaeger.**
Б) Git, GitHub Actions
В) Docker, Kubernetes.
Г) Selenium, Cypress.

17. Стратегия «database per service» в микросервисах:

- А) повышает автономность сервисов, но усложняет обеспечение согласованности данных между ними.**
Б) упрощает транзакции между сервисами и снижает риски рассинхронизации.
В) обязательна для всех микросервисов без исключений.
Г) исключает необходимость в кэшировании.

18. Паттерн Saga используется для:

- А) координации распределённых транзакций через последовательность локальных транзакций и компенсационных действий при сбоях.**
Б) шифрования данных в базе.
В) кэширования результатов запросов.
Г) балансировки нагрузки между серверами.

19. API Gateway в распределённой архитектуре чаще всего отвечает за:

- А) маршрутизацию запросов, аутентификацию, ограничение частоты, агрегацию ответов.**
Б) хранение данных пользователей.
В) выполнение бизнес-логики всех сервисов.
Г) физическое размещение серверов

20. ATAM (Architecture Tradeoff Analysis Method) — это:

- А) метод оценки архитектуры через сценарии использования, чувствительные точки и риски по атрибутам качества (производительность, надёжность и т.д.).**
Б) инструмент автоматического тестирования кода.
В) фреймворк для написания микросервисов.
Г) система контроля версий