

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Узунов Федор Владимирович

Должность: Ректор

Дата подписания: 24.09.2026 14:09:48

Уникальный программный ключ:

fd935d10451b860e912264c0378f8448453b607f94788908a20877a6bcb5f

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
«ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ»
«УНИВЕРСИТЕТ ЭКОНОМИКИ И УПРАВЛЕНИЯ»
Факультет экономики, управления и юриспруденции**

Кафедра управления и бизнес-информатики



УТВЕРЖДАЮ

Проректор по учебно-методической работе

/ Г.П. Узунова

«02» февраля 2026 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
АРХИТЕКТУРА ИНФОРМАЦИОННЫХ СИСТЕМ**

Направление подготовки

09.03.02 «Информационные системы и технологии»

Профиль

«Специалист информационных систем и технологий»

Квалификация выпускника: бакалавр

Для всех
форм обучения

Симферополь, 2026 г.

1. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

Средства оценивания в ходе текущего контроля:

- устные опросы в ходе семинарских занятий;
- рефераты;
- тестирование;
- практические задания, выполняемые в ходе семинарского (практического) занятия или рекомендуемые для самостоятельной работы.

Коды компетенции	Результаты Освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ОПК-1	Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1. Знать: основы математики, физики, вычислительной техники и программирования. ОПК-1.2. Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования. ОПК-1.3. Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, использовать их для решения задач профессиональной деятельности	ОПК-2.1. Знать: современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. ОПК-2.2. Уметь: выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. ОПК-2.3. Владеть: навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности

1.1 Вопросы к текущему контролю

1. Определение: Дайте определение информационной системы (ИС) и её архитектуры. В чем разница между архитектурой ИС и ИТ-инфраструктурой?
2. Компоненты: Перечислите основные функциональные подсистемы типовой АИС (подсистема ввода, обработки, хранения, вывода).
3. Классификация: Классификация ИС по масштабу (персональные, групповые, корпоративные) и по сфере применения (ERP, CRM, BI).
4. Жизненный цикл: Основные этапы жизненного цикла разработки информационной системы (модели Waterfall, Spiral, Agile).
5. Требования: Функциональные и нефункциональные требования к архитектуре ИС (производительность, масштабируемость, надежность).
6. Монолит vs Микросервисы: Сравнение монолитной и микросервисной архитектур. Преимущества и недостатки каждого подхода.

7. Трехзвенная архитектура (*Three-tier*): Описание уровней представления (*presentation*), бизнес-логики (*business logic*) и данных (*data tier*).
8. SOA (*Service-Oriented Architecture*): Принципы сервисно-ориентированной архитектуры. Роль шины предприятия (*ESB*).
9. REST API: Принципы построения RESTful сервисов. Различие методов *GET*, *POST*, *PUT*, *DELETE*.
10. Модели данных: Иерархическая, сетевая и реляционная модели данных. Их историческое значение и современное применение.
11. Реляционный подход: Понятие сущности, атрибута, связи. Ключевые поля (первичный, внешний, кандидатный ключи). Нормализация баз данных.
12. SQL vs NoSQL: Критерии выбора между реляционными БД (*PostgreSQL*, *Oracle*) и нереляционными хранилищами (*MongoDB*, *Redis*).
13. Распределенные БД: Стратегии шардирования (*sharding*) и репликации данных в высоконагруженных системах.
14. Хранилища данных (*DWH*): Отличие операционной базы данных от корпоративного хранилища данных. Концепция OLAP-кубов.
15. UML: Назначение языка UML. Диаграммы вариантов использования (*Use Case*) и классов (*Class Diagram*).
16. ER-моделирование: Построение инфологической и даталогической моделей с помощью ER-диаграмм (сущности, связи, атрибуты).
17. Паттерны проектирования: Применение паттернов Singleton, Factory, DAO (*Data Access Object*) при построении архитектуры ПО.
18. Интеграция систем: Методы интеграции корпоративных приложений (*Point-to-Point*, *Hub-and-Spoke*, общие типы файлов, общая база данных).
19. Cloud-native архитектура: Особенности проектирования систем для облачных сред (контейнеризация Docker, оркестрация Kubernetes).
20. Безопасность архитектуры: Модель AAA (Authentication, Authorization, Accounting). Принцип наименьших привилегий и защита поверхности атаки (*Attack Surface*).

1.2 Темы рефератов:

1. Эволюция архитектур ИС: от централизованных мейнфреймов к распределенным облачным вычислениям.
2. Многоуровневые архитектуры: сравнительный анализ двухзвенной (*Client-Server*) и трехзвенной (*3-tier*) архитектур.
3. Сервис-ориентированная архитектура (SOA): принципы построения, роль корпоративной сервисной шины (*ESB*).
4. Микросервисная архитектура: декомпозиция системы, паттерны коммуникации (*API Gateway*, *Service Mesh*), проблемы распределенных транзакций.
5. Событийно-ориентированная архитектура (EDA): асинхронное взаимодействие компонентов через брокеры сообщений (*Kafka*, *RabbitMQ*).
6. Реляционная модель данных в архитектуре ИС: нормализация, целостность данных, проектирование схем БД.
7. NoSQL базы данных в высоконагруженных системах: выбор между документо-ориентированными (*MongoDB*), колоночными (*Cassandra*) и графовыми (*Neo4j*) СУБД.
8. Корпоративные хранилища данных (*DWH*) и озера данных (*Data Lake*): архитектурные различия, концепция *Lambda Architecture*.
9. Визуальное моделирование архитектуры: эволюция нотаций от DFD и ER-диаграмм к UML и ArchiMate.
10. Язык UML в проектировании ИС: назначение основных диаграмм (классов, вариантов использования, последовательности) на разных этапах жизненного цикла.

11. Модель Захмана (*Zachman Framework*): матричный подход к описанию архитектуры предприятия.
12. Облачные-native архитектуры: использование контейнеризации (*Docker*) и оркестрации (*Kubernetes*) для построения масштабируемых систем.
13. Граничные вычисления (*Edge Computing*): перенос обработки данных ближе к источнику их возникновения (IoT устройства).
14. Мультиоблачные стратегии (*Multi-cloud*): методы обеспечения отказоустойчивости при использовании нескольких провайдеров одновременно.
15. Headless CMS и Jamstack: современная архитектура веб-приложений с разделением бэкенда и фронтенда.
16. Нефункциональные требования: методы обеспечения производительности, масштабируемости и доступности (*High Availability*) в архитектуре.
17. Безопасность архитектурного уровня: принцип наименьших привилегий, управление доступом (*RBAC/ABAC*), защита поверхности атаки.
18. Паттерны интеграции корпоративных приложений: сравнение подходов Point-to-Point, Hub-and-Spoke и файловых шлюзов.
19. Проектирование API-first систем: спецификация OpenAPI (Swagger) как основа контракта между компонентами.
20. Цифровые двойники (*Digital Twins*): создание виртуальных копий физических объектов для мониторинга и предиктивного анализа.

1.3 Тестовые задания

1. Какой уровень архитектуры описывает бизнес-процессы организации, цели создания системы и требования стейкхолдеров?

- а) Технологическая архитектура
- б) Бизнес-архитектура (уровень требований)
- в) Архитектура программных средств
- г) Информационная архитектура (модель данных)

Правильный ответ: б) Бизнес-архитектура

2. Какая модель представления данных лежит в основе реляционных СУБД (SQL)?

- а) Иерархическая
- б) Сетевая
- в) Реляционная (таблицы, ключи, связи)
- г) Объектно-ориентированная

Правильный ответ: в) Реляционная

3. Главный недостаток файл-серверной архитектуры при многопользовательской работе:

- а) Высокая стоимость сервера
- б) Высокий сетевой трафик из-за передачи файлов целиком на клиента для обработки
- в) Невозможность работы без интернета
- г) Сложность написания кода

Правильный ответ: б) Высокий сетевой трафик...

4. В классической двухзвенной архитектуре *Client-Server*, где выполняется основная часть бизнес-логики?

- а) На стороне клиента (толстый клиент)
- б) На сервере баз данных (хранимые процедуры)
- в) Равномерно распределена
- г) На отдельном веб-сервере

Правильный ответ: а) На стороне клиента (толстый клиент)

5. Какую роль выполняет промежуточный слой («сервер приложений») в трехзвенной архитектуре?

- а) Хранит графические файлы интерфейса
- б) Реализует бизнес-правила и управляет соединениями с БД
- в) Выполняет только функции маршрутизации пакетов
- г) Заменяет собой базу данных

Правильный ответ: б) Реализует бизнес-правила...

6. Что является основным преимуществом многоуровневой архитектуры перед трехзвенной?

- а) Меньшее количество компьютеров
- б) Горизонтальная масштабируемость каждого слоя независимо друг от друга
- в) Отсутствие необходимости в администрировании
- г) Использование только одного языка программирования

Правильный ответ: б) Горизонтальная масштабируемость...

7. Основной принцип сервис-ориентированной архитектуры — это:

- а) Жесткая связанность компонентов через общие переменные
- б) Использование слабосвязанных сервисов со стандартизированными интерфейсами
- в) Отказ от использования сетей
- г) Централизованное хранение всего кода в одном модуле

Правильный ответ: б) Использование слабосвязанных сервисов...

8. Что НЕ является характеристикой микросервисной архитектуры?

- а) Автономное развертывание каждого сервиса
- б) Наличие собственной базы данных у каждого сервиса
- в) Единая общая база данных для всех функций системы
- г) Коммуникация через легковесные протоколы (*API/REST*)

Правильный ответ: в) Единая общая база данных...

9. Для чего используется паттерн *API Gateway* в распределенных системах?

- а) Для хранения логов
- б) Как единая точка входа для внешних клиентов, скрывающая внутреннюю структуру микросервисов
- в) Для шифрования жестких дисков
- г) Для замены SQL на NoSQL

Правильный ответ: б) Как единая точка входа...

10: Почему в микросервисах не рекомендуется использовать классические распределенные транзакции (*ACID*)?

- а) Они слишком дешевые
- б) Они снижают доступность системы и создают тесную связанность (согласно теореме CAP)
- в) Они запрещены законом
- г) Они работают только под Windows

Правильный ответ: б) Они снижают доступность...

11: Согласно теореме CAP, в распределенной системе можно одновременно обеспечить только два из трех свойств: Согласованность (*Consistency*), Доступность (*Availability*), Устойчивость к разделению (*Partition tolerance*). Если сеть работает нестабильно (*P* неизбежно), какой выбор обычно делают современные веб-сервисы между *C* и *A*?

- а) Всегда выбирают *Consistency*

- б) Выбирают Availability (AP-системы), жертвуя мгновенной согласованностью
 - в) Останавливают работу всей системы
 - г) Игнорируют Partition tolerance
- Правильный ответ: б) Выбирают Availability...

12: Как называется компонент в событийно-ориентированной архитектуре, который принимает события от источников и передает их потребителям, не зная их имен?

- а) Брокер сообщений (Message Broker / Bus)
- б) Веб-браузер
- в) Текстовый редактор
- г) Антивирус

Правильный ответ: а) Брокер сообщений

13: Для визуального моделирования структуры программного обеспечения (классов, объектов) чаще всего используется язык:

- а) LaTeX
- б) UML (Unified Modeling Language)
- в) HTML
- г) Python

Правильный ответ: б) UML

14: Какой архитектурный паттерн отделяет высокоуровневый бизнес-код от низкоуровневого кода доступа к базе данных?

- а) Singleton
- б) Factory
- в) DAO (Data Access Object)
- г) Observer

Правильный ответ: в) DAO

15: Особенность *Headless CMS* или *Headless eCommerce* заключается в том, что:

- а) Система не имеет административного интерфейса
- б) Бэкенд полностью отделен от фронтенда и предоставляет данные только через *API*
- в) Система работает без электричества
- г) Используется только текстовый интерфейс в консоли

Правильный ответ: б) Бэкенд полностью отделен от фронтенда...

16. Уровни архитектуры

Вопрос: В классической трехзвенной архитектуре (Three-tier) промежуточный слой, отвечающий за выполнение бизнес-логики и управление соединениями с базой данных, называется сервером _____.

Ответ: приложений

17. Модели данных

Вопрос: Реляционная модель данных основана на математическом понятии отношения; данные представляются в виде двумерных таблиц, а для манипуляции ими используется язык _____ запросов.

Ответ: структурированных (или SQL)

18. Файл-серверная архитектура

Вопрос: Главный недостаток файл-серверной архитектуры заключается в высоком сетевом трафике, так как обработка данных происходит на стороне клиента, а по сети передаются целые _____ базы данных.

Ответ: файлы

19. Масштабируемость

Вопрос: Способность информационной системы увеличивать производительность путем добавления новых аппаратных ресурсов (например, серверов), называется горизонтальным _____.

Ответ: масштабированием

20. Сервис-ориентированный подход (SOA)

Вопрос: В сервис-ориентированной архитектуре (SOA) независимые программные единицы со стандартизированными интерфейсами называются веб-_____.

Ответ: сервисы

21. Микросервисы

Вопрос: Ключевое отличие микросервисной архитектуры от монолита — принцип «Database per service», означающий, что каждый микросервис управляет своим собственным, приватным хранилищем _____.

Ответ: данных

22. Паттерны интеграции

Вопрос: Посредник, который выступает единой точкой входа для всех внешних клиентов к набору микросервисов, скрывая внутреннюю структуру системы, называется API-_____.

Ответ: Gateway (шлюз)

23. Теорема CAP

Вопрос: Согласно теореме CAP, при возникновении сетевого разделения (*Partition*) в распределенной системе приходится выбирать между согласованностью данных (*Consistency*) и их _____ (*Availability*).

Ответ: доступностью

24. Событийно-ориентированная архитектура (EDA)

Вопрос: В событийно-ориентированной архитектуре асинхронная передача сообщений между компонентами обычно осуществляется через специальное программное обеспечение — брокер _____.

Ответ: сообщений (или шины событий)

25. Документирование архитектуры

Вопрос: Для визуального моделирования структуры программного обеспечения, классов и связей между ними чаще всего используется унифицированный язык моделирования — _____.

Ответ: UML (Unified Modeling Language)

26. Установить соответствие между термином и его определением:

1. Верстка сайта	А) Процесс создания структуры и визуального оформления веб-страницы с помощью HTML и CSS
2. Адаптивный дизайн	Б) Подход, при котором сайт корректно отображается на устройствах с любым размером экрана
3. CMS	В) Система управления контентом сайта

4. Прототипирование	Г) Создание черновой модели сайта для демонстрации структуры и логики работы
---------------------	--

27. Установить соответствие между свойством CSS и его описанием:

1. display	А) Управляет внешним видом курсора
2. cursor	Б) Определяет, как элемент отображается на странице
3. box-shadow	В) Добавляет тень к тексту
4. text-shadow	Г) Добавляет тень к блоку элемента

28. Установить соответствие между этапом разработки сайта и его результатом:

1. Проектирование	А) Готовый сайт на хостинге
2. Дизайн	Б) Черно-белая схема структуры страниц
3. Верстка	В) Красивый макет с цветами, шрифтами и изображениями
4. Запуск	Г) Работающий интерактивный сайт в браузере

29. Установить соответствие между тегом HTML и его назначением:

1. <header>	А) Контактная информация, авторство, ссылки
2. <footer>	Б) Заголовок раздела или всей страницы
3. <nav>	В) Основное содержимое страницы
4. <main>	Г) Навигационное меню

30. Установить соответствие между типом позиционирования элемента в CSS и его описанием:

1. static	А) Элемент позиционируется отдельно своего ближайшего позиционированного предка
2. relative	Б) Элемент «вырывается» из потока документа и располагается относительно окна браузера
3. absolute	В) Элемент остается в обычном потоке документа, его положение не меняется
4. fixed	Г) Элемент смещается относительно своего обычного положения, но место в потоке сохраняется

1.4 Практические задания

Практическое задание № 1.

Создание стилового оформления сайта.

Создайте сайт, используя каскадные таблицы стилей. Реализуйте адаптивный дизайн для мобильных устройств и компьютеров.

Практическое задание № 2.

Компоновка страниц сайта.

Разработайте макеты трех основных страниц сайта: главная страница, страница продукта и контактная страница. Используйте сетки и флексбокс для компоновки.

Практическое задание № 3.

Формы и элементы пользовательского интерфейса.

Создайте форму обратной связи с полями для имени, электронной почты и сообщения. Добавьте кнопки и выпадающие списки.

Практическое задание № 4.

Создание динамических элементов.

Реализуйте сценарий на JavaScript, который меняет цвет фона страницы при нажатии на кнопку.

Практическое задание № 5.

Проектирование и разработка интерфейса пользователя.

Разработайте интерфейс для мобильного приложения доставки еды. Уделите внимание удобству навигации и интуитивному дизайну.

Практическое задание № 6.

Создание, использование и оптимизация изображений.

Подготовьте изображения для веб-приложения, оптимизируйте их размеры и качество для быстрой загрузки.

Практическое задание № 7.

Графический дизайн.

Разработать логотип для стартапа, занимающегося доступной цифровой средой.

Практическое задание № 8.

Выбор дизайнерского решения.

Проведите анализ рынка и выберите наиболее подходящее дизайнерское решение для сайта фитнес-центра.

Практическое задание № 9.

Проектирование дизайна с промежуточными эскизами.

Создайте серию эскизов для будущего лендинга онлайн-курсов. Включите требования к эргономике и технической эстетике.

Практическое задание № 10.

Подготовка графической информации.

Подберите иконки и иллюстрации для раздела «Часто задаваемые вопросы» на сайте медицинской клиники.

Практическое задание № 11.

Создание GIF-анимаций.

Сделайте короткую анимацию для приветственного баннера на главной странице сайта интернет-магазина.

Практическое задание № 12.

Подготовка мультимедиа для сайта.

Запишите и смонтируйте короткое промо-видео длительностью до 30 секунд для продвижения нового продукта.

Практическое задание № 13.

3D-моделирование.

Создайте простую трехмерную модель стула с использованием Blender или другого программного обеспечения.

Практическое задание № 14.

Визуализация.

Подготовьте рендеры интерьера гостинной комнаты с различными вариантами освещения.

Практическое задание № 15.

Рендеринг сцены.

Настройте сцену с автомобилем и создайте финальный рендер с высоким разрешением для рекламного материала.

2. КРИТЕРИИ ОЦЕНИВАНИЯ ПРИ ПРОВЕДЕНИИ ТЕКУЩЕГО КОНТРОЛЯ

Вид контроля	Наименование работы	Наименование оценочных средств	Шкала оценивания
Текущий контроль	- Вопросы для обсуждения на занятиях; - Устные опросы по ранее изученному материалу; - Письменные работы: рефераты, тестовые задания; - Практические задания; - Рефераты и доклады по темам (вопросам), вынесенным на самостоятельную работу.	Оценка выступлений на практическом (семинарском) занятии, проверка заданий и аудиторных работ, устный опрос, оценивание докладов, рефератов	отлично хорошо удовлетворительно неудовлетворительно

Критерии оценивания устных ответов обучающихся

Шкала оценивания	Характеристика оценивания
отлично	оценивается ответ, который показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа.
хорошо	оценивается ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.
удовлетворительно	оценивается ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в

	содержании ответа.
неудовлетворительно	оценивается ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа.

Критерии оценивания работы обучающихся на практических и семинарских занятиях

Шкала оценивания	Показатели	Критерии
Отлично	1. Полнота выполнения практического и тестового задания (полнота ответа); 2. Своевременность выполнения задания; 3. Последовательность и рациональность выполнения практического задания (логичность и четкость ответа);	Задание решено самостоятельно. При этом составлен правильный алгоритм решения задания, в логических рассуждениях, в выборе формул и решении нет ошибок, получен верный ответ, задание решено рациональным способом. Дан правильный и исчерпывающий ответ на поставленные теоретические и тестовые вопросы, в которых обучающийся показал всестороннее системное знание программного материала, усвоение основной и дополнительной литературы, четкое владение понятийным аппаратом.
Хорошо	4. Правильность ответов на вопросы; 5. Самостоятельность решения (владение дополнительным материалом); 6. Знание нормативно-законодательной базы и терминологии курса	Задание решено с помощью преподавателя. При этом составлен правильный алгоритм решения задания, в логическом рассуждении и решении нет существенных ошибок; правильно сделан выбор формул для решения; есть объяснение решения, но задание решено нерациональным способом или допущено не более двух несущественных ошибок, получен верный ответ. На поставленные теоретические и тестовые вопросы, при которых обучающийся показал достаточный уровень знаний основного программного материала: освоение информации лекционного курса и учебных пособий, овладение понятийным аппаратом, методикой исследований при попытке анализа различных ситуаций.
Удовлетворительно		Задание решено с подсказками преподавателя. Задание решено в общем виде. Обучающийся показал средний уровень знаний основного программного материала, но не мог убедительно аргументировать свой ответ, ошибся в использовании понятийного аппарата, показал недостаточные знания литературных источников.
Неудовлетворительно		Задание не решено. Обучающийся продемонстрировал значительные пробелы в знаниях основного программного материала, не аргументировал свой ответ, показал неудовлетворительные знания понятийного аппарата и специальной литературы.

Критерии оценивания рефератов

Средство контроля	Критерии оценивания	Шкала оценивания
Реферат	Выполнены все требования к написанию и защите реферата: обозначена проблема и обоснована её актуальность, сделан краткий анализ различных точек зрения на рассматриваемую проблему и логично изложена собственная позиция, сформулированы выводы, тема раскрыта полностью, выдержан объём, соблюдены требования к внешнему оформлению, даны правильные ответы на дополнительные вопросы. Реферат раскрывает поднятую проблематику в полном объеме.	отлично
	Основные требования к реферату и его защите выполнены, но при этом допущены недочёты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объём реферата; имеются упущения в оформлении; на дополнительные вопросы при защите даны неполные ответы. В реферате имеются неточности и предметная область выступления раскрыта не в полной мере.	хорошо
	Имеются существенные отступления от требований к реферированию. В частности: тема освещена лишь частично; допущены фактические ошибки в содержании реферата или при ответе на дополнительные вопросы; во время защиты отсутствует вывод. В реферате не в полной степени раскрыт понятийный аппарат, имеются существенные неточности в процессе формирования выводов.	удовлетворительно
	Тема реферата не раскрыта, обнаруживается существенное непонимание проблемы. Тема реферата не раскрыта или выполнена не по существу ранее поставленного вопроса. Реферат не сдан / доклад не сделан.	неудовлетворительно

Критерии оценивания тестов

Средство контроля	Критерии оценивания – процент положительных ответов	Шкала оценивания
Тестирование	90-100	отлично
	70-89	хорошо
	40-69	удовлетворительно
	< 39	неудовлетворительно

3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Средства оценивания в ходе промежуточной аттестации:

- вопросы для зачета;
- тестовые задания к зачету.

Коды компетенции	Результаты Освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ОПК-1	Способен применять естественнонаучные и инженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1. Знать: основы математики, физики, вычислительной техники и программирования. ОПК-1.2. Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и инженерных знаний, методов математического анализа и моделирования. ОПК-1.3. Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, использовать их для решения задач профессиональной деятельности	ОПК-2.1. Знать: современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. ОПК-2.2. Уметь: выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. ОПК-2.3. Владеть: навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности

3.1. Вопросы к зачету:

- Определение:** Дайте определение информационной системы (ИС) и её архитектуры. В чем разница между архитектурой ИС и ИТ-инфраструктурой?
- Компоненты:** Перечислите основные функциональные подсистемы АИС (ввод, обработка, хранение, вывод).
- Классификация:** Классификация ИС по масштабу (персональные, групповые, корпоративные) и по сфере применения (ERP, CRM, BI).
- Жизненный цикл:** Основные этапы жизненного цикла разработки ИС. Сравнение каскадной модели (*Waterfall*) и спиральной модели.
- Требования:** Функциональные и нефункциональные требования к архитектуре (производительность, масштабируемость, надежность, безопасность).
- Файл-серверная архитектура:** Принцип работы, достоинства и критические недостатки при многопользовательской работе.
- Клиент-сервер (2-tier):** Распределение функций между клиентом и сервером СУБД. Понятие «толстого» и «тонкого» клиента.
- Трехзвенная архитектура (3-tier):** Роль сервера приложений как посредника между интерфейсом и данными.
- Многоуровневая архитектура (N-tier):** Отличие от трехзвенной. Преимущества для крупных распределенных предприятий.
- Интернет-ориентированная архитектура:** Особенности построения систем через веб-браузер (Stateless-протокол *HTTP*, использование промежуточного ПО).
- SOA (Service-Oriented Architecture):** Принципы слабой связанности сервисов. Назначение корпоративной сервисной шины (*ESB*).

12. **Микросервисная архитектура:** Декомпозиция монолита на мелкие сервисы. Паттерны коммуникации (*API Gateway, Service Mesh*).
13. **EDA (Event-Driven Architecture):** Использование брокеров сообщений (*Kafka, RabbitMQ*) для асинхронного взаимодействия.
14. **Cloud-native архитектура:** Контейнеризация (*Docker*) и оркестрация (*Kubernetes*) как основа современного развертывания.
15. **Serverless вычисления:** Концепция *Function as a Service (FaaS)*. Плюсы и минусы отсутствия управления серверами.
16. **Реляционная модель:** Понятия сущности, атрибута, связи. Нормализация баз данных (до какой формы обычно нормализуют промышленные БД?).
17. **NoSQL базы данных:** Причины появления. Краткая характеристика четырех типов: ключ-значение, документы, колонки, графы.
18. **CAP-теорема:** Формулировка теоремы. Какой тип согласованности обычно жертвуется в высокодоступных системах (*AP-системы*)?
19. **DWH vs Data Lake:** Различия в структуре хранения данных. Когда целесообразно строить озеро данных вместо классического хранилища?
20. **Шардирование и репликация:** Методы распределения нагрузки и обеспечения отказоустойчивости в кластерах баз данных.
21. **Уровни описания архитектуры:** Логический, физический и концептуальный уровни представления системы.
22. **UML (Unified Modeling Language):** Назначение диаграмм классов (*Class Diagram*) и вариантов использования (*Use Case Diagram*).
23. ****ER-моделирование:** Инфологическая модель предметной области. Правила преобразования ER-диаграммы в реляционную схему.
24. **DAO (Data Access Object):** Суть паттерна. Какую проблему он решает при смене типа СУБД?
25. **Паттерн MVC (Model-View-Controller):** Распределение обязанностей между компонентами в архитектуре пользовательских интерфейсов.
26. **Методы интеграции приложений:** Сравнение подходов Point-to-Point («звезда») и Hub-and-Spoke (шина предприятия). Проблемы спагетти-кода.
27. **REST API:** Принципы RESTful архитектуры. Различие методов *GET, POST, PUT, DELETE*.
28. **Безопасность уровня приложения:** Аутентификация, авторизация и аудит (AAA). Принцип наименьших привилегий.
29. **Межсистемное взаимодействие:** Что такое *Middleware*? Приведите примеры связующего программного обеспечения.
30. **Документирование архитектуры:** Языки описания архитектуры предприятия (*ArchiMate, TOGAF*). Для чего нужен технический проект АИС?

3.2. Задания для зачета:

Задача 1. Файл-сервер vs Клиент-сервер

Условие: В компании работает 5 бухгалтеров, которые одновременно редактируют общий файл Excel с финансами через сетевую папку. Часто возникают ошибки сохранения («файл занят»). Какую архитектуру вы предложите внедрить для решения проблемы конфликта правок? Обоснуйте выбор.

Ключевые понятия: Файл-серверная архитектура, СУБД, блокировки записей.

Задача 2. Трехзвенная архитектура

Условие: Опишите путь прохождения запроса в трехзвенной архитектуре (*Web-браузер -> Сервер приложений -> База данных*) при оформлении заказа в интернет-магазине. Какие функции выполняет каждый слой?

Ключевые понятия: Презентационный уровень, бизнес-логика, уровень данных.

Задача 3. Монолит или Микросервисы

Условие: Стартап запускает MVP (минимально жизнеспособный продукт) из 5 функций. Команда состоит из 3 разработчиков. Стоит ли им сразу использовать микросервисную архитектуру? Аргументируйте ответ с точки зрения затрат на поддержку инфраструктуры. Ключевые понятия: Сложность развертывания, DevOps, масштабируемость команды.

Задача 4. Слабая связанность (Loose Coupling)

Условие: В системе SOA сервис «Склад» упал. Как поведет себя система оформления заказов, если она построена по принципу слабой связанности через шину (ESB)? А как — если они связаны напрямую через вызов метода в коде?

Ключевые понятия: Отказоустойчивость, синхронное/асинхронное взаимодействие.

Задача 5. Serverless (Бессерверные вычисления)

Условие: У компании есть функция генерации ежемесячных PDF-отчетов, которая запускается всего один раз в месяц и работает 5 минут. Почему классическая архитектура выделенного сервера (VPS) будет экономически невыгодной по сравнению с FaaS (AWS Lambda)?

Ключевые понятия: Оплата за время выполнения, idle time (простой).

Задача 6. EDA (Событийно-ориентированная архитектура)

Условие: При регистрации нового пользователя нужно отправить приветственное письмо, начислить бонусные баллы и обновить статистику CRM. Предложите архитектурный паттерн, который позволит выполнить эти действия независимо друг от друга, чтобы задержка отправки письма не тормозила регистрацию.

Ключевые понятия: Брокер сообщений, топик, продюсер, консьюмер.

Задача 7. Нормализация

Условие: Дана таблица Orders: (OrderID, CustomerName, CustomerPhone, ProductID, ProductPrice). В одном заказе может быть много товаров. Выявите аномалию обновления: что произойдет, если клиент сменит номер телефона, а у него 100 старых заказов в этой таблице?

Ключевые понятия: Избыточность данных, 3НФ (Третья нормальная форма).

Задача 8. NoSQL выбор

Условие: Вам нужно хранить профили пользователей социальной сети, где у каждого человека разное количество полей (у кого-то заполнено хобби, у кого-то нет). Какой тип NoSQL базы данных (Документно-ориентированная или Колоночная) лучше подойдет и почему?

Ключевые понятия: Гибкая схема (Schema-less), MongoDB/Cassandra.

Задача 9. CAP-теорема (Практика)

Условие: Банковская система переводов между регионами. Произошел разрыв связи между дата-центрами Москвы и Новосибирска. Что важнее обеспечить согласно теореме CAP: мгновенную согласованность баланса (Consistency) или доступность приложения (Availability)? К какому типу (CP или AP) относится такая система?

Ключевые понятия: Согласованность, Доступность, Устойчивость к разделению.

Задача 10. Шардирование (Sharding)

Условие: Таблица сообщений мессенджера выросла до петабайта данных. Запросы стали медленными. Предложите стратегию разделения таблицы по полю user_id. Как сервер поймет, на какой физический узел отправить запрос конкретного пользователя?

Ключевые понятия: Хеширование ключа, маршрутизатор запросов.

Задача 11. Диаграмма классов

Условие: Нарисуйте (опишите текстом) структуру классов для системы «Библиотека»: сущности Книга, Читатель, ВыдачаКниги. Укажите атрибуты и связь «многие-ко-многим».

Ключевые понятия: Ассоциация, кратность (Multiplicity), агрегация.

Задача 12. Диаграмма последовательности (Sequence Diagram)

Условие: Опишите последовательность действий на диаграмме для сценария «Оплата картой на сайте». Участники: Пользователь, Браузер, Веб-сервер, Банк-эквайер.

Ключевые понятия: Lifeline, Message (синхронное/асинхронное).

Задача 13. Use Case (Варианты использования)

Условие: Для онлайн-кинотеатра определите одного Актера (Actor) и три его Цели (Use Cases), включая альтернативный поток событий (например, «Попытка оплаты с недостатком средств»).

Ключевые понятия: Преусловия (Preconditions), Постусловия (Postconditions).

Задача 14. MVC на примере

Условие: В приложении «Список дел» пользователь нажимает кнопку «Удалить задачу». Какой компонент архитектуры MVC перехватывает клик мыши, какой удаляет запись из памяти, а какой перерисовывает экран без этой задачи?

Ключевые понятия: Model, View, Controller.

Задача 15. RESTful принципы

Условие: Проанализируйте эндпоинт `/api/getAllUsers`. Соответствует ли он принципам *REST*? Если нет, то как должен выглядеть правильный путь?

Ключевые понятия: Именованное ресурс (существительные), использование HTTP-глаголов.

Задача 16. API Gateway

Условие: Мобильное приложение обращается к 10 разным микросервисам. Разработчики хотят добавить проверку токена авторизации (*JWT*) перед каждым вызовом. Где логичнее разместить эту проверку, чтобы не дублировать код в каждом из 10 сервисов?

Ключевые понятия: Единая точка входа, кросс-функциональные задачи.

Задача 17. Паттерн Bulkhead (Переборка)

Условие: Ваша система вызывает сторонний сервис доставки. Этот сервис начал отвечать очень медленно (5 секунд вместо 100 мс), подвешивая потоки вашего приложения. Какой паттерн устойчивости поможет изолировать этот сбой, чтобы остальная часть вашей системы продолжала работать быстро?

Ключевые понятия: Таймауты, изоляция ресурсов.

Задача 18. GraphQL vs REST

Условие: Мобильному приложению для экрана профиля нужны только имя и фото пользователя. Однако *REST*-эндпоинт `/profile` возвращает всю информацию (адрес, историю заказов, настройки). Какое решение оптимизирует трафик мобильного интернета?

Ключевые понятия: Избыточность данных (Over-fetching), гибкость запроса.

Задача 19. AAA (Authentication, Authorization, Accounting)

Условие: Объясните разницу между аутентификацией и авторизацией на примере доступа сотрудника к папке "Зарплаты".

Ключевые понятия: Проверка личности vs проверка прав доступа.

Задача 20. Принцип наименьших привилегий

Условие: Веб-приложение подключается к базе данных под учетной записью root (суперпользователь). Хакер нашел уязвимость *SQL Injection*. Какова область поражения? Как изменить архитектуру подключения, чтобы минимизировать ущерб?

Ключевые понятия: Ограничение прав сервисного аккаунта.

Задача 21. Circuit Breaker (Предохранитель)

Условие: Сервис А постоянно ждет ответа от падающего Сервиса Б, исчерпывая все свои ресурсы (поток). Как автоматизировать процесс «отключения» связи с Сервисом Б на 1 минуту после серии неудач, чтобы дать ему восстановиться?

Ключевые понятия: Открытое/Закрытое состояние предохранителя.

Задача 22. Rate Limiting (Ограничение частоты)

Условие: Ваш публичный API атакуют боты, делая 1000 запросов в секунду. Как ограничить нагрузку так, чтобы честные пользователи могли делать не более 10 запросов в минуту?

Ключевые понятия: Token Bucket, IP-based limits.

Задача 23. ATAM (Architecture Tradeoff Analysis Method)

Условие: Вы проектируете систему управления беспилотником. Выберите два конфликтующих атрибута качества (например, Производительность vs Энергопотребление) и опишите архитектурный компромисс.

Ключевые понятия: Trade-offs (жертвование одним ради другого).

Задача 24. CQRS (Command Query Responsibility Segregation)

Условие: В высоконагруженной системе аналитики чтение данных происходит в 100 раз чаще, чем запись. Как разделить модели чтения и записи, чтобы оптимизировать базу данных под тяжелые аналитические запросы (SELECT), не мешая быстрым вставкам (INSERT)?

Ключевые понятия: Разделение ответственности, Read Model / Write Model.

Задача 25. SAGA Pattern

Условие: Транзакция бронирования тура включает: списание денег, бронирование отеля, бронирование авиабилета. Если бронирование отеля провалилось, как отменить уже списанные деньги, используя цепочку локальных транзакций?

Ключевые понятия: Компенсирующая транзакция (Compensating Transaction).

Задача 26. Monorepo vs Polyrepo

Условие: У вас 50 микросервисов. В каком случае хранение кода всех 50 сервисов в одном гигантском репозитории (*Monorepo*) будет удобнее, чем в разных (*Polyrepo*)?

Ключевые понятия: Управление зависимостями, версионирование.

Задача 27. Strangler Fig Pattern

Условие: У вас есть старый монолит на языке PHP. Вы хотите переписать его на Go, но полная замена займет 2 года. Какой паттерн позволит постепенно перенаправлять трафик со старого модуля на новый внутри одной URL-адресации?

Ключевые понятия: Постепенная миграция legacy-систем.

Задача 28. Data Mesh

Условие: Традиционное центральное хранилище данных (*DWH*) стало узким местом, так как отдел маркетинга долго ждет очистки данных от отдела продаж. Как концепция Data Mesh меняет владение данными?

Ключевые понятия: Доменно-ориентированное владение данными.

Задача 29. Twelve-Factor App

Условие: Согласно методологии «12 факторов», где должно храниться секретное значение пароля от базы данных в контейнеризированном приложении: жестко прописанным в коде, в конфиг-файле рядом с кодом или в переменных окружения (*environment variables*)?

Ключевые понятия: Отделение конфигурации от кода.

Задача 30. Документирование (ADR)

Условие: Через год после разработки новой системы никто не помнит, почему был выбран протокол gRPC вместо REST для внутренней коммуникации. Какой легковесный документ должен вестись командой для фиксации таких архитектурных решений?

Ключевые понятия: Architecture Decision Record (Запись архитектурного решения).

4. ОСНОВНЫЕ КРИТЕРИИ ОЦЕНИВАНИЯ ПРИ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Шкала оценивания уровня сформированности универсальной компетенций (зачет)

Шкала оценивания	Уровень освоение компетенции	Критерии оценивания
Зачет	Базовый уровень освоения компетенции	Дан правильный и исчерпывающий ответ на вопрос. Обучающийся демонстрирует знание теоретического материала, изложено правильное понимание вопроса, дано достаточно подробное описание предмета ответа, приведены и раскрыты в тезисной форме основные понятия, относящиеся к предмету ответа. Имеется базовый уровень овладения практическими умениями и навыками по данной дисциплине в соответствии с ФГОС .
Незачет	Неудовлетворительный уровень	Отсутствует ответ или в ответе есть грубые ошибки, свидетельствующие о отсутствии знаний соответствующего программного материала; отсутствие умений и навыков по данной дисциплине в соответствии с ФГОС и/или фрагментарные знания основного учебно-программного материала.

Текущий контроль и промежуточная аттестация осуществляются в соответствии с «Положением о текущей и промежуточной аттестации обучающихся в Автономной некоммерческой организации «Образовательная организация высшего образования» «Университет экономики и управления».

Вид промежуточной аттестации – зачет.

Форма проведения промежуточной аттестации – письменный зачет.