**Комплект оценочных материалов по дисциплине**
**«Архитектура информационных систем»**

## Задания закрытого типа

### Задания закрытого типа на выбор правильного ответа

1. Выберите один правильный ответ:
Наиболее эффективной формой контроля результатов освоения обучающимися темы «Основные понятия архитектуры информационных систем» являются:
А) собеседование, анкетирование;
Б) творческое задание, эссе;
В) устный опрос, тестирование.
Г) фронтальный опрос, наблюдение;
Правильный ответ: В
Компетенции: ПК-5

2. Выберите один правильный ответ:
Для определения уровня знаний и общей эрудиции учащихся перед изучением дисциплины «Архитектура информационных систем» целесообразно использовать:
А) индивидуальные консультации;
Б) практические работы, тестирование;
В) беседы, анкетирование, наблюдения;
Г) повторное тестирование.
Правильный ответ: В
Компетенции: ПК-5

3. Выберите один правильный ответ:
Что делает клиент в клиент-серверной архитектуре при работе с веб-приложением?
А) хранит данные;
Б) обрабатывает данные;
В) отображает пользовательский интерфейс;
Г) управляет базой данных.
Правильный ответ: В
Компетенции (индикаторы): УК-1, ПК-8

4. Выберите один правильный ответ:
Что такое «монолитная архитектура»?
А) архитектура, состоящая из множества независимых сервисов;
Б) архитектура, в которой все компоненты приложения объединены в единое целое;
В) архитектура, основанная на облачных технологиях;

Г) архитектура, использующая контейнеризацию.
Правильный ответ: Б
Компетенции (индикаторы): УК-1, ПК-8

**Задания закрытого типа на установление соответствия**

1. Сопоставьте типы архитектур с их преимуществами:

| | |
|---|---|
| 1. Микросервисная архитектура | А) Простота разработки и развертывания для небольших приложений. |
| 2. Монолитная архитектура | Б) Разделение логики приложения между клиентом и сервером. |
| 3. Клиент-серверная архитектура | В) Независимое масштабирование отдельных компонентов. |

Правильный ответ: 1 – В, 2 – А, 3 – Б
Компетенции (индикаторы): УК-1, ПК-8

2. Сопоставьте понятия с их определениями:

| | |
|---|---|
| 1. API | А) Упаковка приложений и зависимостей в контейнеры. |
| 2. Контейнеризация | Б) Способность системы обрабатывать большее количество запросов. |
| 3. Масштабируемость | В) Набор правил для взаимодействия программ. |

Правильный ответ: 1 – В, 2 – А, 3 –Б
Компетенции (индикаторы): УК-1, ПК-8

3. Установите соответствие между принципами архитектуры и их описаниями:

| | |
|---|---|
| 1. Масштабируемость | А) Скорость обработки запросов и выполнения операций. |
| 2. Отказоустойчивость | Б) Способность системы выдерживать нагрузки. |
| 3. Безопасность | В) Способность системы восстанавливаться после сбоев. |
| 4. Производительность | Г) Защита данных и системы от угроз. |

Правильный ответ: 1 – Б, 2 – В, 3 – Г, 4 – А
Компетенции (индикаторы): УК-1, ПК-8

4. Установите соответствие между компонентами архитектуры и их функциями:

| | |
|---|---|
| 1. API (Application Programming Interface) | А) Отображение пользовательского интерфейса и отправка запросов. |
| 2. База данных | Б) Обработка запросов и выполнение бизнес-логики. |
| 3. Сервер приложений | В) Взаимодействие между различными компонентами системы.. |
| 4. Клиент | Г) Хранение и управление данными. |

Правильный ответ: 1 – В, 2 – Г, 3 – Б, 4 – А
Компетенции (индикаторы): УК-1, ПК-8

**Задания закрытого типа на установление правильной последовательности**

1. Расположите этапы развертывания гибридной облачной архитектуры, включающей локальные и облачные ресурсы, в правильном порядке:

А) Настройка VPN-соединения или выделенного канала между локальной сетью и облачным провайдером.

Б) Определение требований к приложениям и выбор подходящих облачных сервисов.

В) Развертывание критически важных приложений в локальной инфраструктуре.

Г) Миграция менее критичных приложений в облако и настройка автоматического масштабирования.

Правильный ответ: Б, А, В, Г.
Компетенции (индикаторы): УК-1, ПК-8

2. Расположите этапы проектирования архитектуры для обработки больших данных, в правильном порядке:

А) Анализ и визуализация данных.

Б) Сбор и хранение данных.

В) Выбор технологий для обработки данных.

Г) Обработка и трансформация данных.

Правильный ответ: В, Б, Г, А.
Компетенции (индикаторы): УК-1

3. Расположите этапы проектирования облачной архитектуры для веб-приложения, в правильном порядке:

А) Настройка масштабирования и оптимизация производительности.

Б) Определение требований к приложению и выбор облачных сервисов.

В) Развертывание приложения в облачной среде.

Г) Проектирование архитектуры и выбор компонентов.

Правильный ответ: Б, Г, В, А.
Компетенции (индикаторы): УК-1, ПК-8

4. Расположите этапы проектирования микросервисной архитектуры приложения, в правильном порядке:

А) Развертывание и мониторинг микросервисов.

Б) Определение границ микросервисов и их взаимодействия.

В) Выбор технологий и инструментов для реализации архитектуры.

Г) Разработка отдельных микросервисов.

Правильный ответ: В, Б, Г, А.
Компетенции (индикаторы): УК-1, ПК-8

**Задания открытого типа**

**Задания открытого типа на дополнение**

1. В клиент-серверной архитектуре_____ может отображать данные, полученные из базы данных.

Правильный ответ: клиент

Компетенции (индикаторы): УК-1, ПК-8

2. В клиент-серверной архитектуре _____ обрабатывает запросы к базе данных и возвращает результаты.

Правильный ответ: сервер

Компетенции (индикаторы): УК-1, ПК-8

3. Масштабируемость в архитектуре информационных систем — это способность системы обрабатывать большее количество _____

Правильный ответ: запросов

Компетенции (индикаторы): УК-1, ПК-8

4. _____ архитектура характеризуется объединением всех компонентов приложения в единое целое.

Правильный ответ: монолитная

Компетенции (индикаторы): УК-1, ПК-8

**Задания открытого типа с кратким свободным ответом**

1. Клиент-серверная архитектура информационных систем состоит из _____.

Правильный ответ: сервера, клиента/клиентов

Компетенции (индикаторы): УК-1, ПК-8

2. В архитектуре информационных систем для обеспечения безопасности данных используются _____.

Правильный ответ: шифрование, аутентификация

Компетенции (индикаторы): УК-1, ПК-8

3. В архитектуре информационных систем для обеспечения отказоустойчивости используются _____.

Правильный ответ: резервирование, балансировка нагрузки

Компетенции (индикаторы): УК-1, ПК-8

4. В архитектуре информационных систем для обеспечения безопасности сетевого трафика используются _____.

Правильный ответ: брандмауэры, файерволы, межсетевые экраны

Компетенции (индикаторы): УК-1, ПК-8

**Задания открытого типа с развернутым ответом**

1. Из предложенного текста выделите основные понятия, которые необходимы для проведения учебного занятия по теме «Клиент-серверная архитектура».

**Что такое клиент-серверная архитектура?**

Клиент-серверная архитектура – это распределенная архитектура, которая разделяет приложение на две основные части: клиент и сервер, взаимодействующие друг с другом по сети. Ключевая идея заключается в том, что разделение ответственности. Клиент и сервер выполняют разные, но взаимодополняющие функции.

Клиент - обычно пользовательское приложение (например, веб-браузер, мобильное приложение, десктопное приложение), которое запрашивает ресурсы или услуги у сервера. Клиент отвечает за отображение интерфейса пользователя, ввод данных и взаимодействие с пользователем.

Сервер - мощный компьютер или программа, которая предоставляет ресурсы или услуги клиентам. Сервер отвечает за хранение данных, обработку запросов, выполнение бизнес-логики и управление доступом к ресурсам. Сервер может обслуживать множество клиентов одновременно.

Клиент и сервер обмениваются сообщениями по сети, используя определенные протоколы (например, HTTP для веб-приложений, SMTP для электронной почты). Клиент отправляет запросы на сервер, а сервер отвечает ответами, которые могут содержать данные, результаты обработки или сообщения об ошибках.

**Основные компоненты клиент-серверной архитектуры:**

- Клиенты: Программные приложения, установленные на компьютерах пользователей или других устройствах (например, мобильные телефоны, планшеты). Клиенты инициируют запросы к серверам.
- Тонкие клиенты: Клиенты, которые выполняют минимальную обработку данных и полагаются на сервер для основной логики (терминалы, веб-интерфейсы).
- Толстые клиенты: Клиенты, которые выполняют значительную часть обработки данных на стороне клиента, снижая нагрузку на сервер (десктопные приложения с богатым функционалом).
- Серверы: Специализированные компьютеры или программное обеспечение, предназначенные для обслуживания запросов клиентов и предоставления ресурсов.

**Преимущества клиент-серверной архитектуры:**

Централизация ресурсов: Серверы позволяют централизованно управлять ресурсами (данными, приложениями, вычислительными мощностями), обеспечивая более эффективное использование ресурсов и упрощая управление и обслуживание.

Безопасность: Централизованное управление серверами упрощает обеспечение безопасности. Меры безопасности могут быть реализованы на сервере, контролируя доступ клиентов к ресурсам и данным.

Масштабируемость: Клиент-серверная архитектура легко масштабируется. Для увеличения производительности можно увеличить мощность серверов (вертикальное масштабирование) или добавить новые серверы (горизонтальное масштабирование), не затрагивая клиентские приложения.

Управляемость: Управление и администрирование системы становится более простым и эффективным, так как основные компоненты и данные находятся на сервере. Обновления и изменения программного обеспечения часто требуется вносить только на сервере, а не на каждом клиентском компьютере.

Доступность данных: Данные, хранящиеся на сервере, становятся доступными для многих клиентов одновременно, обеспечивая совместный доступ и обмен информацией.

Разделение труда: Клиент-серверная архитектура позволяет разделить труд между разработчиками клиентской и серверной частей, упрощая разработку и поддержку сложных приложений.

**Недостатки клиент-серверной архитектуры:**

Зависимость от сервера: Клиенты зависят от доступности и работоспособности сервера. Если сервер выходит из строя, клиенты теряют доступ к сервисам и данным.

Единая точка отказа (Single Point of Failure): Сервер может стать единой точкой отказа. Сбой сервера может привести к неработоспособности всей системы или ее части. Для повышения надежности требуется резервирование серверов.

Перегрузка сервера: При большом количестве клиентов или интенсивных запросах сервер может быть перегружен, что приводит к снижению производительности и времени отклика. Требуется масштабирование серверов и оптимизация нагрузки.

Сетевой трафик: Взаимодействие клиентов и серверов создает сетевой трафик. При большом количестве клиентов или передаче больших объемов данных сеть может быть перегружена, что влияет на производительность. Требуется оптимизация сетевой инфраструктуры и протоколов.

Безопасность сети: Передача данных по сети создает риски безопасности. Необходимо обеспечивать безопасность сетевого трафика (например, с использованием HTTPS, VPN) и защиту серверов от внешних угроз.

Сложность настройки и обслуживания: Настройка и обслуживание серверной инфраструктуры может быть сложным и требовать квалифицированного персонала.

Клиент-серверная архитектура является фундаментальным архитектурным стилем, который обеспечивает эффективное разделение труда, централизацию ресурсов, масштабируемость и управляемость информационных систем. Несмотря на некоторые недостатки, клиент-

серверная архитектура остается доминирующей в большинстве современных IT-систем и является важной концепцией для понимания основ современной информатики.

Время выполнения – 20 мин.

Ожидаемый результат:

Основные понятия для учебного занятия по теме «Клиент-серверная архитектура»:

1) Клиент-серверная архитектура - распределенная архитектура информационной системы, которая разделяет приложение на две основные части: клиент и сервер и взаимодействуют друг с другом по сети.
2) Клиент - программное приложение, установленное на компьютере пользователя или другом устройстве, которое инициирует запросы к серверам для получения ресурсов или услуг.
3) Сервер - специализированный компьютер или программное обеспечение, предназначенное для обслуживания запросов клиентов.
4) Распределенная архитектура - тип архитектуры информационной системы, в которой компоненты системы распределены по нескольким компьютерам, взаимодействующим по сети.
5) Разделение ответственности - ключевой принцип клиент-серверной архитектуры, заключающийся в разделении функций и задач между клиентом и сервером.
6) Запрос-ответ - модель взаимодействия между клиентом и сервером.
7) Протоколы - наборы правил и стандартов, определяющие формат, порядок и правила обмена данными между клиентом и сервером по сети.
8) Типы клиентов (тонкий, толстый) - классификация клиентов по объему обработки данных, выполняемому на стороне клиента.

Критерии оценивания:

Правильный ответ должен содержать минимум три элемента из перечня, представленного в ожидаемом результате.

Компетенции: ПК-4, ПК-8

2. Из предложенного текста выделите основные понятия, которые необходимы для проведения учебного занятия по теме «Монолитная архитектура».

**Что такое монолитная архитектура?**

В контексте информационных систем, монолитная архитектура – это единое, неделимое приложение, построенное как цельная конструкция. Представьте себе крепкий, но цельный каменный блок – монолит. В монолитном приложении все функциональные возможности, компоненты и модули тесно связаны и развернуты как единое целое.

**Ключевые характеристики монолитной архитектуры**

Единая кодовая база: Все компоненты приложения (пользовательский интерфейс, бизнес-логика, доступ к данным и т.д.) разрабатываются и хранятся в едином репозитории кода.

Тесная взаимосвязь компонентов: Модули и компоненты монолитного приложения глубоко интегрированы и взаимозависимы. Изменения в одном компоненте могут затрагивать другие части системы.

Единица развертывания: Монолитное приложение развертывается как единый, неделимый блок. Все компоненты развертываются одновременно и на одних и тех же ресурсах.

Централизованное управление: Управление приложением, как правило, централизованное, и все аспекты, от разработки до развертывания и мониторинга, обычно осуществляются одной командой или в рамках единого процесса.

Типичные компоненты монолитного приложения (хотя они и не явно разделены архитектурно):

- Пользовательский интерфейс (UI): Компонент, отвечающий за взаимодействие с пользователем (веб-интерфейс, десктопное приложение и т.д.).
- Бизнес-логика: Основная функциональность приложения, реализующая бизнес-правила, алгоритмы и процессы.
- Интеграция с данными: Компонент, отвечающий за доступ к базе данных и управление данными (ORM, DAO и т.д.).
- Инфраструктурные сервисы: Общие сервисы, используемые приложением (логирование, аутентификация, авторизация и т.д.).

**Преимущества монолитной архитектуры**

Простота разработки: На начальных этапах и для небольших приложений, монолитная архитектура может быть проще в разработке и понимании. Единая кодовая база и тесная интеграция упрощают процессы разработки и тестирования.

Легкость развертывания: Развертывание монолитного приложения часто более прямолинейно, так как это одна единица развертывания. Обычно требуется развернуть один артефакт (например, WAR-файл для Java веб-приложения).

Производительность (в определенных случаях): В некоторых случаях, за счет тесной интеграции и меньших накладных расходов на взаимодействие между компонентами (по сравнению с распределенными системами), монолитные приложения могут демонстрировать хорошую производительность для определенных задач.

Проще начальная разработка: Для команд, начинающих проект или для простых приложений, монолит может быть более быстрым и легким путем к запуску, поскольку не требуется сложной архитектурной проработки распределенных систем.

**Недостатки монолитной архитектуры**

Ограниченная масштабируемость: Масштабирование монолитного приложения часто означает масштабирование всего приложения целиком, даже

если нагрузка возрастает только на отдельные его части. Горизонтальное масштабирование может быть затруднительным и менее эффективным, чем в микросервисных архитектурах.

Технологическая зависимость (Vendor lock-in): Монолитные приложения часто тесно связаны с определенным технологическим стеком. Внесение изменений в технологический стек или внедрение новых технологий может быть затруднительным и затратным.

Сложность внесения изменений: По мере роста приложения, монолит становится все более сложным и трудным для понимания и изменения. Любые изменения, даже небольшие, могут потребовать пересборки и повторного развертывания всего приложения.

Затруднение внедрения новых технологий: Из-за тесной связанности компонентов, внедрение новых технологий или фреймворков в монолитное приложение может быть рискованным и сложным, требующим переработки значительной части кода.

Проблемы при развертывании крупных приложений: Развертывание больших монолитных приложений может стать долгим и рискованным процессом, увеличивая время простоя при обновлениях.

Связанность и хрупкость: Из-за тесной связанности, ошибки в одном модуле могут каскадно влиять на другие части приложения, повышая риск общих сбоев.

**Когда монолитная архитектура может быть оправдана?**

Монолитная архитектура может быть хорошим выбором для:

Небольших приложений: Для проектов с ограниченным объемом функциональности и небольшой командой разработчиков.

Приложений с простыми бизнес-требованиями: Когда сложность предметной области невелика, и ожидается, что приложение не будет сильно расти и усложняться.

Прототипов и MVP (Minimum Viable Product): Для быстрого старта и проверки гипотез, когда скорость разработки важнее масштабируемости и долгосрочной гибкости.

Приложений, где производительность критична, а распределение нагрузки не является основной проблемой: В случаях, когда важна максимальная производительность в рамках одного сервера, и не ожидается резких скачков нагрузки.

Монолитная архитектура – это классический подход, имеющий как свои преимущества, так и недостатки. Для простых и небольших приложений она может быть эффективной и простой в реализации. Однако, для крупных, сложных и растущих систем, более гибкие и масштабируемые архитектурные подходы, такие как микросервисы, становятся все более предпочтительными. Выбор архитектуры всегда должен зависеть от конкретных требований и контекста проекта.

Время выполнения – 20 мин.

Ожидаемый результат:

Основные понятия для учебного занятия по теме «Монолитная архитектура»:

1) Монолитная архитектура - единый, неделимый подход к построению программного обеспечения, где приложение разрабатывается как цельная конструкция с единой кодовой базой, тесной взаимосвязью компонентов и развертывается как единый блок.
2) Единая кодовая база - ситуация, когда все компоненты приложения (пользовательский интерфейс, бизнес-логика, доступ к данным и т.д.) разрабатываются и хранятся в одном общем репозитории кода.
3) Тесная взаимосвязь компонентов - характеристика монолитных приложений, где модули и компоненты глубоко интегрированы и взаимозависимы, так что изменения в одном месте могут влиять на другие части системы.
4) Единица развертывания - подход к развертыванию, при котором монолитное приложение упаковывается и развертывается как единый, неделимый программный артефакт.
5) Централизованное управление - метод управления приложением, где все аспекты, такие как разработка, развертывание и мониторинг, обычно осуществляются одной командой или в рамках единого организационного процесса.

Критерии оценивания:

Правильный ответ должен содержать минимум три элемента из перечня, представленного в ожидаемом результате.

Компетенции: ПК-4, ПК-8

3. Опишите основные виды архитектур информационных систем.

Пример ответа:

Архитектура информационных систем (ИС) представляет собой основополагающий каркас, определяющий структуру, поведение и взаимодействие компонентов, обеспечивающих функционирование системы. Виды архитектур ИС варьируются в зависимости от масштаба, сложности, требований к производительности, надежности, безопасности и бизнес-целей. Архитектуры ИС можно классифицировать по степени централизации, способу взаимодействия компонентов и используемой парадигме построения. Монолитная архитектура, представляет собой единое, неделимое приложение. Распределенные архитектуры: клиент-серверная, облачная, и микросервисная, предлагающие большую гибкость и масштабируемость.

Клиент-серверная архитектура, одна из наиболее распространенных, разделяет систему на две ключевые роли: клиент, запрашивающий ресурсы или услуги, и сервер, предоставляющий их. Такой подход обеспечивает централизацию ресурсов и упрощает управление, но может создавать точки отказа и ограничения в масштабировании.

Облачные архитектуры выводят распределение на новый уровень, используя инфраструктуру облачных провайдеров для динамического масштабирования ресурсов по требованию.

Микросервисная архитектура фокусируется на декомпозиции приложения на небольшие, независимые сервисы, каждый из которых выполняет определенную функцию и взаимодействует с другими через сеть.

Критерии оценивания:

- Описаны основные виды архитектур ИС (клиент-серверная, монолитная, микросервисная).
- Указаны основные характеристики архитектур.
- Ответ логично структурирован, легко читается и понимается.

Время выполнения – 25 мин.

Компетенции (индикаторы): УК-1, ПК-8

4. Объясните, как облачные вычисления могут помочь в масштабировании информационных систем.

Пример ответа:

Облачные вычисления предоставляют возможность быстрого масштабирования ресурсов по требованию. При увеличении нагрузки на систему можно легко добавить дополнительные вычислительные мощности, хранилище или сетевые ресурсы. При снижении нагрузки ресурсы могут быть автоматически уменьшены, что позволяет оптимизировать затраты.

Облачные вычисления предлагают разнообразные стратегии масштабирования, включая горизонтальное и вертикальное. Горизонтальное масштабирование позволяет легко добавлять новые экземпляры виртуальных серверов, контейнеров или баз данных, распределяя нагрузку между ними и обеспечивая высокую доступность и отказоустойчивость. Вертикальное масштабирование позволяет увеличивать мощность отдельных ресурсов, например, наращивать объем оперативной памяти или процессорную мощность виртуальной машины. Облачные провайдеры предоставляют широкий спектр масштабируемых сервисов, охватывающих вычислительные ресурсы, позволяя комплексно масштабировать различные аспекты информационной системы.

Вклад облачных вычислений в масштабирование ИС усиливается за счет мощных инструментов автоматизации и глобальной инфраструктуры. Автоматическое масштабирование, оркестрация контейнеров и инфраструктура как код позволяют автоматизировать процессы развертывания и управления масштабированием, снижая ручной труд и минимизируя время реакции на изменения нагрузки. Глобальная сеть дата-центров облачных провайдеров обеспечивает географическое масштабирование, позволяя размещать приложения и данные ближе к пользователям по всему миру, улучшая производительность и обеспечивая соответствие нормативным требованиям. В итоге, облачные вычисления предоставляют комплексное решение для эффективного масштабирования информационных систем, позволяя организациям быть гибкими, инновационными и конкурентоспособными в динамичной цифровой среде.

Критерии оценивания:

- Правильно объяснена концепция облачных вычислений и их роль в масштабировании.
- Ответ логично структурирован, объяснение последовательно.

Время выполнения – 25 мин.

Компетенции (индикаторы): УК-1, ПК-8

**Экспертное заключение**

Представленный комплект оценочных материалов по дисциплине «Архитектура информационных систем» соответствует требованиям ФГОС ВО.

Предлагаемые оценочные материалы адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 44.03.04. Профессиональное обучение (по отраслям).

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанные и представленные для экспертизы оценочные материалы рекомендуются к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической комиссии института компьютерных систем и информационных технологий — Н.Н. Ветрова

## Лист изменений и дополнений

| № п/п | Виды дополнений и изменений | Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения | Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами) |
|---|---|---|---|
| 1 | В фонд оценочных средств добавлен комплект оценочных материалов | 26.02.2025 г., №14 | А.И. Горбунов |
| | | | |
| | | | |
| | | | |