**Тестовые вопросы по реферату "Требования к интеграциям и API"**

**Вопрос 1. (Множественный выбор, 2 правильных ответа)**  
Какие две основные проблемы призвана решать качественно разработанная интеграция? Выберите два варианта:  
A. Низкая скорость написания нового программного кода с нуля  
B. Сложность взаимодействия большого количества разнородных систем  
C. Необходимость полного отказа от унаследованных (legacy) систем  
D. Неэффективность и хаотичность point-to-point (точечных) соединений между приложениями  
E. Отсутствие потребности в специалистах по информационной безопасности

**Правильные ответы: B, D**  
**Пояснение:**  
• **B (✓)** — Одна из ключевых целей интеграции — обеспечить взаимодействие систем, разработанных на разных платформах, с разными форматами данных.  
• **D (✓)** — В реферате подчеркивается, что точечные соединения плохо масштабируются и создают "информационный хаос", а современные подходы к интеграции (слабая связанность, API) призваны это исправить.  
• **A (✗)** — Интеграция не ускоряет написание кода "с нуля", она ускоряет и упрощает соединение уже существующих систем.  
• **C (✗)** — Напротив, одна из задач интеграции — включить унаследованные системы в общие бизнес-процессы, а не отказываться от них.  
• **E (✗)** — Безопасность является критически важным требованием к любой интеграции, а не проблемой, которую она устраняет.

**Вопрос 2. (Единственный выбор, 1 правильный ответ)**  
Какой ключевой принцип проектирования интеграций обеспечивает независимость систем друг от друга, позволяя изменять внутреннюю логику одной системы без воздействия на другие?  
A. Стандартизация протоколов  
B. Слабая связанность (Loose Coupling)  
C. Идемпотентность  
D. Синхронность взаимодействия

**Правильный ответ: B**  
**Пояснение:**  
• **B (✓)** — В реферате прямо указано, что слабая связанность — это фундаментальный принцип, при котором системы взаимодействуют, имея минимальные знания друг о друге, что позволяет менять их внутреннюю реализацию.  
• **A (✗)** — Стандартизация важна для совместимости, но сама по себе не гарантирует слабой связанности.  
• **C (✗)** — Идемпотентность важна для надежности, но не отвечает напрямую за независимость систем.  
• **D (✗)** — Синхронность — это стиль взаимодействия, который может даже способствовать большей связанности.

**Вопрос 3. (Множественный выбор, 2 правильных ответа)**  
Какие два аспекта входят в нефункциональные требования (NFR) к API? Выберите два варианта:  
A. Список конечных точек (эндпоинтов) и HTTP-методов  
B. Максимально допустимое время ответа (латентность)  
C. Требования к аутентификации и авторизации  
D. Модели данных и атрибуты ресурсов  
E. Лимиты запросов (Rate Limiting)

**Правильные ответы: B, E**  
**Пояснение:**  
• **B (✓)** — Производительность (время ответа) является классическим нефункциональным требованием.  
• **E (✓)** — Лимиты запросов — это требование, описывающее *как* система должна себя вести под нагрузкой, а не *что* она делает, что относится к NFR.  
• **A (✗)** — Эндпоинты и методы — это часть функциональных требований (что делает система).  
• **C (✗)** — Безопасность (аутентификация и авторизация) в контексте реферата рассматривается как сквозное требование, которое часто выносится отдельно, но логика управления доступом может относиться к функциональности.  
• **D (✗)** — Модели данных относятся к функциональным требованиям, описывающим структуру информации.

**Вопрос 4. (Единственный выбор, 1 правильный ответ)**  
Что из перечисленного является основной целью принципа идемпотентности в проектировании интеграций?  
A. Обеспечить минимальное время отклика системы.  
B. Гарантировать, что повторное выполнение операции не изменит результат сверх первого выполнения.  
C. Скрыть внутреннюю реализацию сервиса от потребителя.  
D. Преобразовать данные из одного формата в другой.

**Правильный ответ: B**  
**Пояснение:**  
• **B (✓)** — Это точное определение идемпотентности, которое приведено в реферате. Оно критически важно для надежности, особенно в асинхронных сценариях, где возможны повторные отправки сообщений.  
• **A (✗)** — Это цель требований к производительности.  
• **C (✗)** — Это цель принципа инкапсуляции.  
• **D (✗)** — Это одна из функций, которую может выполнять интеграционная шина или сам API.

**Вопрос 5. (Единственный выбор, 1 правильный ответ)**  
Какой из перечисленных элементов НЕ является частью функциональных требований к API?  
A. Коды состояния HTTP для различных сценариев  
B. Стратегия версионирования (например, через URL /v1/...)  
C. Набор ресурсов и их атрибуты (модели данных)  
D. Бизнес-логика преобразования данных перед сохранением

**Правильный ответ: B**

**Пояснение:**  
• **B (✓)** — Стратегия версионирования — это аспект управления жизненным циклом API и его эволюции, который относится к нефункциональным или архитектурным требованиям, а не к описанию его непосредственной функциональности "здесь и сейчас".  
• **A (✗)** — Описание кодов ответов прямо указывает, как система функционирует в разных ситуациях.  
• **C (✗)** — Модели данных — ядро функционального описания того, с чем работает API.  
• **D (✗)** — Логика преобразования данных — это часть функционального поведения системы.

**Вопрос 6. (Единственный выбор, 1 правильный ответ)**  
Чем принципиально отличается асинхронный стиль интеграции от синхронного?  
A. Асинхронная интеграция не требует безопасности.  
B. Синхронная интеграция не может использовать REST API.  
C. При асинхронной интеграции система-инициатор не блокируется в ожидании немедленного ответа.  
D. Асинхронная интеграция подразумевает прямые соединения между системами, а синхронная — через посредника.

**Правильный ответ: C**

**Пояснение:**  
• **C (✓)** — Это ключевое отличие, описанное в реферате. Синхронный вызов блокирует клиента до получения ответа, а асинхронный — нет, сообщение ставится в очередь, и отправитель продолжает работу.  
• **A (✗)** — Безопасность обязательна для обоих стилей.  
• **B (✗)** — REST API является классическим примером синхронного взаимодействия.  
• **D (✗)** — Все наоборот: асинхронная интеграция часто использует брокеров сообщений (посредников), в то время как синхронная может быть прямой.

**Система оценивания**

* **Вопрос 1, 3:** 2 балла за оба правильных ответа (1 балл, если выбран только один верный ответ).
* **Вопрос 2, 4, 5, 6:** 1 балл за каждый правильный ответ.
* **Максимум: 8 баллов.**

**Шкала перевода:**

* 8 баллов — **Отлично**
* 7 баллов — **Очень хорошо**
* 5–6 баллов — **Хорошо**
* 3–4 балла — **Удовлетворительно**
* 0–2 балла — **Неудовлетворительно**

**Пример:**  
Если студент выбрал:

* Вопрос 1: B, D → 2 балла
* Вопрос 2: B → 1 балл
* Вопрос 3: B, E → 2 балла
* Вопрос 4: B → 1 балл
* Вопрос 5: B → 1 балл
* Вопрос 6: C → 1 балл  
  **Итого: 8 баллов (максимум).**