Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гриб Владислав Валерьевич
Должность: Ректор
Дата подписания: 06.11.2026 16:26:53
Уникальный программный ключ:
637517d24e103c3db032acf37e839d98ec1c5bb2f5eb89c29abfcd7f4398547

Образовательное частное учреждение высшего образования

«МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ А.С. ГРИБОЕДОВА»

(ИМПЭ им. А.С. Грибоедова)

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
(МЕЖДУНАРОДНЫЙ ДИПЛОМАТИЧЕСКИЙ КОЛЛЕДЖ)

УТВЕРЖДАЮ
Директор Международного
дипломатического колледжа
_____ Н.И. Платонова
«16» февраля 2026г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
Разработка бизнес-приложений

Специальность
09.02.11 Разработка и управление программным обеспечением

Квалификация выпускника:
Программист

на базе основного общего образования

Форма обучения: очная

Москва

Фонд оценочных средств для профессионального модуля «Разработка бизнес-приложений». Специальность 09.02.11 Разработка и управление программным обеспечением /– М.: ИМПЭ им. А. С. Грибоедова.

Фонд оценочных средств является неотъемлемой частью рабочей программы профессионального модуля.

Организация-разработчик: Образовательное частное учреждение высшего образования «Московский университет имени А. С. Грибоедова»

Директор Международного
дипломатического колледжа_____/Н.И. Платонова

1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Настоящий Фонд оценочных средств (ФОС) для профессионального модуля «Разработка бизнес-приложений» является неотъемлемым приложением для профессионального модуля «Разработка бизнес-приложений». На данный ФОС распространяются все реквизиты утверждения, представленные в РПД по данной дисциплине.

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

Для определения качества освоения обучающимися учебного материала по дисциплине используются следующие оценочные средства:

№ п/п	Оценочное средство	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Тестирование	Вид контроля, позволяющий оценить изученный теоретический материал.	Вопросы для проведения тестирования
2	Практические задания	Вид контроля, позволяющий оценить умение обучающегося применять осваиваемую компетенцию в практических ситуациях и при решении производственных задач	Задания к практическому занятию
3	Контрольная работа	Вид контроля, позволяющий определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций	Задания контрольной работы
4	Самостоятельная работа	Вид контроля, позволяющий оценить проработку теоретического материала, изучение рекомендуемой литературы, выполнение практико-ориентированных заданий (заполнение таблиц, проведение сравнительного анализа, составление схем и др.), решение практических задач, создание презентаций, написание рефератов, подборку нормативного и иного материала и выполнение других заданий	Задания самостоятельной работы
5	Курсовая работа	Вид контроля, позволяющий выявить степень владения базовыми знаниями, умениями и навыками, необходимыми для обучения, и определить уровень владения новым материалом	Индивидуальные задания (темы) для курсовой работы
6	Зачет/Зачет с оценкой/Экзамен	Вид контроля, позволяющий выявить степень овладения знаниями, умениями и навыками, необходимыми для дальнейшего освоения образовательной программы подготовки	Вопросы для подготовки к зачету/зачету с оценкой/экзамену

3.ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Темы для подготовки к промежуточной аттестации

МДК.03.01 Проектирование бизнес-приложений

1. Жизненный цикл разработки программного обеспечения (SDLC): этапы, модели, особенности применения для бизнес-приложений.
2. Сравнительный анализ методологий разработки: Waterfall, Agile, Scrum, Kanban.
3. Роль и место этапа проектирования в различных методологиях разработки.
4. Заинтересованные стороны (Stakeholders) в проектах разработки: выявление, анализ, управление ожиданиями.
5. Понятие архитектуры программного обеспечения. Архитектурные стили: файлсерверный, клиент-серверный, трехуровневый.
6. Микросервисная архитектура: принципы, преимущества, сложности внедрения.
7. Техническое задание (ТЗ): структура, требования к оформлению, процесс согласования.
8. Понятие качества программного обеспечения. Модели и стандарты качества.
9. Риски в проектах разработки: идентификация, анализ, методы минимизации.
10. Управление проектами разработки: основные процессы и документы.
11. Команда разработки: роли, зоны ответственности, взаимодействие.
12. Оценка трудоемкости разработки: методы и подходы.
13. Требования к программному обеспечению: определение, классификация, уровни требований.
14. Функциональные и нефункциональные требования: методы выявления и документирования.
15. Бизнес-требования и пользовательские требования: взаимосвязь и способы формализации.
16. Источники требований: внутренние и внешние, первичные и вторичные.
17. Методы сбора требований: интервьюирование, анкетирование, анализ документов.
18. Методы сбора требований: наблюдение, мозговой штурм, прототипирование.
19. Пользовательские истории (User Stories): структура, принципы написания, критерии готовности.
20. Бэклог продукта: формирование, приоритизация, управление изменениями.
21. Спецификация требований к ПО (SRS): структура и содержание.
22. Управление изменениями требований в процессе разработки.
23. Валидация и верификация требований: методы и инструменты.
24. Трассировка требований: понятие, цели, способы реализации.
25. Бизнес-процесс: определение, характеристики, классификация.
26. Нотации моделирования бизнес-процессов: обзор и сравнительная характеристика.
27. Нотация BPMN 2.0: основные элементы, правила построения диаграмм.
28. Объекты потока управления в BPMN: события, действия, логические операторы.
29. Разделение ответственности в BPMN: пулы, дорожки, артефакты.

30. Моделирование процессов AS-IS: цели, методы сбора информации, типовые ошибки.
31. Анализ модели AS-IS: выявление узких мест, дублирования функций, неэффективных операций.
32. Моделирование процессов TO-BE: принципы построения целевого процесса.
33. Реинжиниринг бизнес-процессов: понятие, этапы, подходы.
34. Регламентация бизнес-процессов: структура регламента, порядок утверждения.
35. Унифицированный язык моделирования (UML): назначение, виды диаграмм.
36. Диаграмма вариантов использования (Use Case): элементы, правила построения, примеры.
37. Актеры в Use Case: типы, отношения с вариантами использования.
38. Отношения между вариантами использования: включение, расширение, обобщение.
39. Глоссарий проекта: назначение, структура, процесс ведения.
40. Концептуальное моделирование данных: сущности, атрибуты, связи.
41. Типы связей между сущностями: один-к-одному, один-ко-многим, многие-комногим.
42. Логическая и физическая модели данных: отличия, этапы построения.
43. Основы UX-проектирования: понятие пользовательского опыта, принципы человеко-ориентированного дизайна.
44. Сценарии использования (User Flow): построение, анализ, оптимизация.
45. Карта пути клиента (CJM): структура, этапы построения, применение.
46. Принципы юзабилити: законы Нильсена, эвристики оценки интерфейсов.
47. Визуальное восприятие в интерфейсах: законы гештальта, композиция, типографика, цвет.
48. Прототипирование интерфейсов: виды прототипов, инструменты, методы тестирования.
49. Дизайн-системы: компоненты, принципы построения, преимущества использования.
50. Оценка качества интерфейсов: методы тестирования с пользователями, экспертные оценки.

МДК.03.02 Разработка бизнес-приложений

1. Архитектура бизнес-приложений: компоненты, уровни, взаимодействие.
2. Языки программирования для backend-разработки: сравнительный анализ, критерии выбора.
3. Фреймворки для backend-разработки: назначение, классификация, примеры.
4. Системы контроля версий: Git, основные операции, стратегии ветвления.
5. Работа с удаленными репозиториями: GitHub, GitLab, совместная разработка.
6. Объектно-ориентированное программирование: основные концепции и принципы.
7. Классы и объекты: создание, свойства, методы, конструкторы.
8. Инкапсуляция: модификаторы доступа, свойства, сокрытие реализации.
9. Наследование: иерархии классов, абстрактные классы, виртуальные методы.
10. Полиморфизм: перегрузка методов, переопределение методов, интерфейсы.

11. Обработка исключений: иерархия исключений, конструкция try-catch-finally.
12. Статические члены классов: назначение, особенности использования.
13. Паттерны проектирования: порождающие, структурные, поведенческие.
14. Внедрение зависимостей (Dependency Injection): принципы, преимущества.
15. Логирование в приложениях: уровни логирования, конфигурация, анализ.
16. Системы управления базами данных: реляционные и NoSQL, критерии выбора.
17. Основы SQL: типы данных, создание таблиц, ограничения целостности.
18. Операторы SQL для выборки данных: SELECT, FROM, WHERE, JOIN.
19. Операторы SQL для модификации данных: INSERT, UPDATE, DELETE.
20. Транзакции в базах данных: свойства ACID, уровни изоляции.
21. Индексы в базах данных: назначение, типы, влияние на производительность.
22. Нормализация данных: нормальные формы, цели нормализации.
23. Технологии доступа к данным: ADO.NET, JDBC, ORM.
24. ORM-фреймворки: принципы работы, преимущества, недостатки.
25. Выполнение операций CRUD через ORM.
26. Оптимизация запросов к базе данных: анализ планов запросов, настройка индексов.
27. Веб-технологии: HTML, CSS, JavaScript, их роль и взаимодействие.
28. Основы HTML: структура документа, семантическая верстка, формы.
29. Основы CSS: селекторы, свойства, каскадирование, наследование.
30. Адаптивная верстка: медиа-запросы, резиновые макеты, mobile-first подход.
31. Основы JavaScript: синтаксис, типы данных, функции, события.
32. Работа с DOM: выбор элементов, изменение содержимого, обработка событий.
33. Современные фронтенд-фреймворки: React, Angular, Vue.js - обзор и сравнение.
34. Компонентный подход в разработке интерфейсов.
35. Валидация форм на стороне клиента: встроенные средства, кастомная валидация.
36. Асинхронное взаимодействие с сервером: AJAX, Fetch API, Axios.
37. Протокол HTTP: структура запроса и ответа, методы, коды состояния.
38. RESTful API: принципы проектирования, лучшие практики, идемпотентность.
39. Форматы обмена данными: JSON, XML - синтаксис, области применения.
40. Аутентификация и авторизация: базовые понятия, механизмы реализации.
41. JWT (JSON Web Tokens): структура, создание, проверка, преимущества.
42. Защита веб-приложений: основные уязвимости и методы защиты.
43. HTTPS: принцип работы, необходимость использования.
44. Управление доступом: ролевая модель (RBAC), разграничение прав.
45. Тестирование программного обеспечения: уровни, виды, подходы.
46. Модульное тестирование: принципы, инструменты, организация тестов.
47. Интеграционное тестирование: цели, методы, отличия от модульного.
48. Отладка приложений: методы, инструменты, анализ логов.
49. Развертывание приложений: веб-серверы, хостинг, облачные платформы.
50. Контейнеризация и оркестрация: Docker, Kubernetes, CI/CD процессы.

МДК.03.03 Интеграция и модификация бизнес-приложений

1. Интеграция приложений: понятие, цели, задачи, экономическая целесообразность.
2. Эволюция подходов к интеграции: от файлового обмена к микросервисам.

3. Интеграция "точка-точка": характеристика, преимущества, недостатки.
4. Шина данных (ESB): концепция, функции, место в корпоративной архитектуре.
5. Микросервисная архитектура как подход к интеграции.
6. Уровни интеграции: интеграция данных, приложений, процессов, интерфейсов.
7. Синхронная и асинхронная интеграция: сравнительный анализ, области применения.
8. Связанность систем: понятие, влияние на гибкость и надежность.
9. Топологии интеграции: звезда, шина, сетка.
10. Оркестрация и хореография сервисов: сравнение подходов.
11. Управление транзакциями в распределенных системах: двухфазная фиксация, сага.
12. Интеграция в облачной среде: облачные интеграционные платформы (iPaaS).
13. Протокол HTTP в интеграции: методы, заголовки, коды состояния.
14. RESTful API: принципы проектирования, зрелость по Ричардсону.
15. Идемпотентность и безопасность методов HTTP в REST.
16. SOAP протокол: структура сообщения, WSDL, области применения.
17. Сравнение REST и SOAP: производительность, сложность, гибкость.
18. Формат JSON: синтаксис, типы данных, схемы JSON.
19. Формат XML: синтаксис, пространства имен, схемы XSD.
20. Графовый API: GraphQL, принципы работы, сравнение с REST.
21. gRPC: особенности протокола, преимущества использования.
22. Проектирование API: лучшие практики, документирование (OpenAPI/Swagger).
23. Управление жизненным циклом API: версионирование, депрекация.
24. API Gateway: функции, преимущества, примеры реализации.
25. Безопасность API: аутентификация, авторизация, Rate Limiting, Throttling.
26. Шаблоны интеграции корпоративных приложений (EIP): классификация.
27. Паттерны маршрутизации сообщений: Content-Based Router, Message Filter.
28. Паттерны маршрутизации: Recipient List, Splitter, Aggregator.
29. Паттерны трансформации сообщений: Message Translator, Enricher, Normalizer.
30. Паттерны управления конечными точками: Messaging Gateway, Transactional Client.
31. Паттерны обработки ошибок: Dead Letter Channel, Retry, Circuit Breaker.
32. Паттерны обеспечения надежности: Idempotent Receiver, Compensating Transaction.
33. Применение паттернов интеграции при проектировании решений.
34. Асинхронная интеграция: концепция очередей сообщений.
35. Брокеры сообщений: функции, классификация, обзор решений.
36. Протокол AMQP: концепции, модель взаимодействия.
37. Протокол MQTT: особенности, области применения, QoS.
38. RabbitMQ: архитектура, обменники, очереди, привязки.
39. Apache Kafka: архитектура, топики, партиции, потребители.
40. Гарантии доставки сообщений: at-most-once, at-least-once, exactly-once.
41. Consumer groups: масштабирование потребления сообщений.
42. Подтверждение обработки и повторная обработка сообщений.
43. Работа с унаследованным кодом (legacy code): анализ, документирование, тестирование.
44. Технический долг: причины возникновения, стратегии управления.

45. Рефакторинг: цели, техники, обеспечение безопасности изменений.
46. Миграция данных: стратегии, этапы, валидация результатов.
47. ETL-процессы: Extract, Transform, Load - этапы, инструменты.
48. Мониторинг интеграционных решений: метрики, алерты, дашборды.
49. Распределенная трассировка: концепция, инструменты, анализ производительности.
50. Обеспечение отказоустойчивости: резервирование, балансировка, восстановление после сбоев.

Темы для самостоятельной работы обучающихся

МДК.03.01 Проектирование бизнес-приложений

1. Сравнительный анализ методологий разработки программного обеспечения: выбор оптимальной для различных типов бизнес-приложений.
2. Исследование роли проектировщика (системного аналитика) в командах с различными методологиями разработки.
3. Анализ заинтересованных сторон для типового бизнес-приложения (на примере интернет-магазина, банковской системы, CRM).
4. Изучение и сравнительный анализ стандартов качества программного обеспечения (ISO 25000, IEEE Std 830-1998).
5. Исследование методов оценки трудоемкости разработки: функциональные точки, COCOMO, Planning Poker.
6. Анализ рисков при разработке бизнес-приложений: идентификация, оценка, разработка мер минимизации.
7. Изучение структуры и содержания технического задания для государственных и коммерческих заказчиков.
8. Исследование особенностей проектирования для различных предметных областей (логистика, финансы, здравоохранение, образование).
9. Сравнительный анализ инструментов для моделирования (CASE-средства): Aris, Rational Rose, Visio, Draw.io, Online-сервисы.
10. Изучение нотации ArchiMate для моделирования корпоративной архитектуры.
11. Исследование методов приоритизации требований: MoSCoW, Kano model, стоимостная оценка.
12. Анализ техник сбора требований в условиях удаленной работы распределенных команд.
13. Изучение прототипирования как метода сбора и уточнения требований.
14. Исследование проблем коммуникации между заказчиком и командой разработки.
15. Анализ типичных ошибок при сборе и документировании требований.
16. Разработка глоссария предметной области для выбранного бизнес-приложения.
17. Построение диаграммы вариантов использования для системы управления задачами (Task Manager).
18. Построение диаграммы вариантов использования для системы бронирования гостиниц.
19. Разработка пользовательских историй для мобильного приложения доставки еды.

20. Создание модели бизнес-процесса AS-IS для процесса оформления заказа в розничном магазине.
21. Создание модели бизнес-процесса TO-BE для процесса оформления заказа с предложениями по автоматизации.
22. Моделирование процесса согласования документов на предприятии в нотации BPMN.
23. Построение диаграммы деятельности (Activity Diagram) для сценария регистрации пользователя.
24. Разработка концептуальной модели данных для библиотечной системы.
25. Разработка концептуальной модели данных для системы учета кадров.
26. Исследование и сравнительный анализ современных дизайн-систем (Material Design, Human Interface Guidelines, OpenUI).
27. Анализ пользовательского опыта (UX-аудит) существующего бизнес-приложения.
28. Разработка карты пути клиента (CJM) для интернет-магазина одежды.
29. Построение сценариев использования (User Flow) для процесса оформления подписки.
30. Создание низкодетализированных прототипов (wireframes) для личного кабинета клиента.
31. Разработка высокодетализированных макетов (mockups) для мобильного приложения банка.
32. Исследование принципов accessible design (доступность интерфейсов для людей с ограниченными возможностями).
33. Анализ юзабилити типовых интерфейсов корпоративных информационных систем (1С, SAP, CRM системы).
34. Исследование геймификации в бизнес-приложениях: методы и примеры.
35. Сравнительный анализ инструментов прототипирования (Figma, Sketch, Adobe XD, Balsamiq).
36. Разработка полного пакета проектной документации для системы управления проектами (Project Management System).
37. Проектирование архитектуры интернет-магазина с использованием трехуровневой модели.
38. Разработка технического задания на разработку мобильного приложения для фитнес-клуба.
39. Проектирование базы данных для системы складского учета.
40. Разработка требований к системе лояльности для сети розничных магазинов.
41. Проектирование интеграции новой CRM-системы с существующей бухгалтерской системой.
42. Разработка пользовательской документации (руководство пользователя) для спроектированного приложения.
43. Создание презентации проектного решения для защиты перед заказчиком.
44. Исследование и выбор технологического стека для разрабатываемого приложения.
45. Проектирование системы с учетом требований безопасности и защиты персональных данных.
46. Разработка стратегии тестирования для спроектированного приложения.

47. Проектирование системы с учетом требований к производительности и масштабируемости.
48. Анализ и выбор оптимальной архитектуры для стартапа в сфере электронной коммерции.
49. Разработка плана внедрения спроектированной системы на предприятии.
50. Экономическое обоснование разработки бизнес-приложения: расчет стоимости и оценка эффективности.

МДК.03.02 Разработка бизнес-приложений

1. Сравнительный анализ языков программирования для backend-разработки: производительность, экосистема, сообщество.
2. Исследование современных фреймворков для backend-разработки (Spring, .NET Core, Django, Express.js).
3. Изучение систем контроля версий: сравнительный анализ Git, Mercurial, SVN.
4. Исследование стратегий ветвления в Git: Git Flow, GitHub Flow, Trunk-based development.
5. Анализ принципов SOLID в контексте разработки бизнес-приложений.
6. Изучение паттернов проектирования GoF: классификация, примеры реализации.
7. Исследование архитектурных паттернов: MVC, MVP, MVVM.
8. Сравнительный анализ реляционных и NoSQL баз данных для различных типов приложений.
9. Изучение современных ORM-решений: Entity Framework, Hibernate, Dapper, Prisma.
10. Исследование принципов CQRS и Event Sourcing в разработке приложений.
11. Анализ фронтенд-фреймворков: React, Angular, Vue.js - экосистема, производительность, кривая обучения.
12. Изучение TypeScript: преимущества использования в больших проектах.
13. Исследование современных подходов к стилизации веб-приложений: CSS-in-JS, Tailwind CSS, SCSS.
14. Анализ методов оптимизации производительности фронтенда.
15. Изучение прогрессивных веб-приложений (PWA): возможности, ограничения, примеры.
16. Разработка простого REST API на выбранном языке программирования.
17. Реализация CRUD-операций для сущности "Товар" с хранением в базе данных.
18. Разработка системы аутентификации пользователей с использованием JWT.
19. Реализация ролевой модели доступа (администратор, пользователь, гость).
20. Разработка модуля регистрации и входа с хешированием паролей.
21. Создание middleware для логирования запросов в приложении.
22. Реализация обработки ошибок и глобального обработчика исключений.
23. Разработка сервиса для отправки email-уведомлений.
24. Реализация валидации входящих данных на сервере.
25. Разработка модуля для работы с файлами (загрузка, хранение, скачивание).
26. Верстка адаптивной страницы входа в систему.
27. Разработка формы регистрации с валидацией полей на стороне клиента.

28. Создание компонента "Таблица" для отображения данных с возможностью сортировки.
29. Разработка корзины интернет-магазина с использованием состояния (state).
30. Реализация поиска с автодополнением (поиск по мере ввода).
31. Создание карусели изображений для карточки товара.
32. Разработка дашборда с графиками и диаграммами.
33. Реализация бесконечной прокрутки (infinite scroll) для списка элементов.
34. Создание многошаговой формы с сохранением промежуточных данных.
35. Разработка интерфейса чата с отправкой и отображением сообщений.
36. Разработка клиента для публичного REST API (например, погода, курсы валют).
37. Интеграция с платежной системой (тестовый режим) на выбранном языке.
38. Разработка OAuth-авторизации через социальные сети.
39. Реализация загрузки файлов на сервер и интеграция с облачным хранилищем.
40. Разработка веб-сокетов для реализации уведомлений в реальном времени.
41. Интеграция с картографическим сервисом (отображение точек на карте).
42. Разработка Telegram-бота для взаимодействия с бизнес-приложением.
43. Написание модульных тестов для бизнес-логики приложения.
44. Проведение интеграционного тестирования взаимодействия с базой данных.
45. Настройка логирования в приложении с использованием готовых библиотек.
46. Контейнеризация разработанного приложения с помощью Docker.
47. Развертывание приложения на облачной платформе (Heroku, Railway, AWS).
48. Настройка CI/CD пайплайна для автоматического развертывания.
49. Разработка полного приложения (full-stack) с клиентской и серверной частью.
50. Рефакторинг существующего кода с применением изученных паттернов.

МДК.03.03 Интеграция и модификация бизнес-приложений

1. Исследование эволюции подходов к интеграции: от ETL к микросервисам и сервисным сеткам (Service Mesh).
2. Сравнительный анализ интеграционных платформ (iPaaS): MuleSoft, Dell Boomi, Apache Camel.
3. Изучение концепции Data Fabric и ее роли в современной интеграции данных.
4. Анализ паттернов распределенных транзакций: двухфазная фиксация (2PC), сага (Saga).
5. Исследование проблем консистентности данных в распределенных системах (CAPтеорема, BASE-подход).
6. Сравнительный анализ форматов сериализации данных: JSON, XML, Avro, Protobuf, MessagePack.
7. Изучение стандартов описания API: OpenAPI (Swagger), RAML, API Blueprint.
8. Исследование GraphQL: преимущества, недостатки, сценарии использования.
9. Анализ протоколов асинхронного взаимодействия: AMQP, MQTT, STOMP.
10. Сравнительный анализ брокеров сообщений: RabbitMQ, Apache Kafka, ActiveMQ, Redis.
11. Изучение архитектуры Kafka: брокеры, партиции, репликация, группы потребителей.

12. Исследование паттерна Event-Driven Architecture (событийно-ориентированная архитектура).
13. Анализ подходов к обеспечению отказоустойчивости интеграционных решений.
14. Изучение методов мониторинга распределенных систем: метрики, логи, трассировка (Observability).
15. Исследование инструментов distributed tracing: Jaeger, Zipkin, OpenTelemetry.
16. Разработка REST API с документацией в Swagger/OpenAPI.
17. Создание клиента для SOAP-сервиса (генерация кода по WSDL).
18. Реализация версионирования API различными способами (URI, заголовки).
19. Разработка API Gateway с базовой маршрутизацией и аутентификацией.
20. Реализация Rate Limiting для ограничения количества запросов к API.
21. Разработка сервиса для трансформации данных из XML в JSON.
22. Создание прокси-сервера для агрегации данных из нескольких источников.
23. Реализация паттерна Circuit Breaker для повышения отказоустойчивости.
24. Разработка механизма повторных попыток (retry) с экспоненциальной задержкой.
25. Реализация идемпотентности обработки запросов для предотвращения дублирования.
26. Установка и настройка RabbitMQ (или Kafka) в локальной среде.
27. Разработка producer-приложения для отправки сообщений в очередь.
28. Разработка consumer-приложения для обработки сообщений из очереди.
29. Реализация различных типов обменников в RabbitMQ (direct, topic, fanout).
30. Настройка подтверждений (acknowledgments) и dead letter queue.
31. Разработка приложения для работы с Kafka: producer, consumer, обработка партиций.
32. Реализация гарантированной доставки сообщений с подтверждением от потребителя.
33. Создание сценария с несколькими потребителями в одной группе (конкурирующие потребители).
34. Разработка приложения для пакетной обработки сообщений из Kafka.
35. Интеграция брокера сообщений с существующим бизнес-приложением.
36. Анализ технического долга в предоставленном коде и составление плана рефакторинга.
37. Проведение рефакторинга "большого грязного кода" с выделением слоев (layers).
38. Добавление новой функциональности в существующее приложение с сохранением обратной совместимости.
39. Миграция приложения с монолитной архитектуры на микросервисную (декомпозиция).
40. Разработка стратегии миграции данных с тестированием на копии базы данных.
41. Реализация ETL-процесса для выгрузки данных из одной БД в другую с трансформацией.
42. Валидация корректности мигрированных данных (сверка контрольных сумм, выборочная проверка).
43. Разработка системы для синхронизации данных между двумя независимыми приложениями.

44. Создание интеграционной шины (упрощенный ESB) для маршрутизации сообщений между сервисами.
45. Разработка микросервисного приложения с использованием брокера сообщений для взаимодействия.
46. Проектирование и реализация системы уведомлений на основе событий.
47. Разработка дашборда для мониторинга состояния интеграционных потоков.
48. Создание системы для агрегации данных из нескольких внешних API с кэшированием.
49. Разработка стратегии и проведение миграции данных при смене CRM-системы.
50. Полный цикл интеграции нового сервиса в существующую ИТ-инфраструктуру предприятия (проект).

Примерные варианты тестовых оценочных заданий (ТОЗ) для контрольного рубежа в рамках текущего контроля

МДК.03.01 Проектирование бизнес-приложений

Тестовое задание	**Оцениваемые компетенции**
ВОПРОСЫ ОТКРЫТОГО ТИПА	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Основной стандарт для функционального моделирования бизнеспроцессов называется _____. +: IDEF0	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Программные средства для автоматизации проектирования систем называются _____. +: CASE-средства	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Универсальный язык графического моделирования объектноориентированных систем называется _____. +: UML	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Диаграмма, отображающая структуру данных системы, называется диаграммой _____. +: классов	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Модель жизненного цикла, предполагающая последовательное выполнение этапов, называется _____ моделью. +: каскадной	ОК 01

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Система планирования ресурсов предприятия обозначается аббревиатурой _____. +: ERP	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Диаграмма, которая показывает, как объекты взаимодействуют друг с другом во времени, называется диаграммой _____. +: последовательности	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Модель организации в её текущем состоянии называется моделью _____. +: «как есть»	ПК 3.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Методология моделирования потоков данных называется _____. +: DFD	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Основная цель создания информационной системы — поддержка и автоматизация _____ организации. +: бизнес-процессов	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Диаграмма, показывающая физическое размещение компонентов	ПК 3.1.

системы, называется диаграммой _____. +: развертывания	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс изучения и документирования потребностей заказчика называется сбором _____. +: требований	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Методология описания последовательности работ и сценариев процессов называется _____. +: IDEF3	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Специалист, отвечающий за взаимодействие между пользователем и продуктом, называется _____ дизайнером. +: UX	ПК 3.2.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Визуальный макет или черновик будущего интерфейса называется _____. +: прототипом	ПК 3.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Дизайн, при котором интерфейс подстраивается под разные размеры экранов, называется _____ дизайном. +: адаптивным	ПК 3.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Международная организация по стандартизации имеет аббревиатуру _____. +: ISO	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) В методологии DFD стрелки, обозначающие перемещение информации, называются _____ данных. +: потоками	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Специалист, отвечающий за визуальное оформление интерфейса, называется _____ дизайнером. +: UI	ПК 3.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Совокупность работ по преобразованию системы из текущего состояния в целевое называется _____ системы. +: модернизацией	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Основной документ, формализующий требования к создаваемой системе, — это _____. +: техническое задание	ПК 3.1.

ВОПРОСЫ ЗАКРЫТОГО ТИПА	
К структурным методологиям моделирования бизнес-процессов относится: +: методология IDEF0 -: теория вероятностей -: метод Монте-Карло -: теория игр	ПК 3.1.

К статическим диаграммам языка UML относится: +: диаграмма классов -: диаграмма последовательности -: диаграмма деятельности -: диаграмма состояний	ПК 3.1.
К динамическим диаграммам языка UML относится: -: диаграмма компонентов -: диаграмма развертывания -: диаграмма классов +: диаграмма состояний	ПК 3.1.
К основным элементам методологии DFD относится: +: поток данных -: работа (как блок с номером) -: стрелка управления -: механизм (исполнитель)	ПК 3.1.
К типовым корпоративным информационным системам (ИС) относится: +: ERP-система -: ОС семейства Windows -: система Mathematica -: антивирус Касперского	ПК 3.1.
К основным целям создания информационной системы для бизнеса относится: +: автоматизация бизнес-процессов -: развлечение сотрудников -: увеличение энергопотребления -: усложнение отчетности	ПК 3.1.
К методам выявления требований к информационной системе относится: +: интервью с заинтересованными лицами -: тестирование производительности -: написание исходного кода -: рефакторинг базы данных	ПК 3.1.
К международным стандартам в области качества ПО относится: +: ISO 25010 -: ГОСТ 34 -: IEEE 802.11 -: TCP/IP	ПК 3.1.

К основным принципам UX/UI дизайна интерфейсов относится: +: согласованность и стандарты -: максимальная насыщенность цветами -: использование мелкого шрифта для экономии места -: скрытие основных функций в глубоких меню	ПК 3.2.

МДК.03.02 Разработка бизнес-приложений

Тестовое задание	Оцениваемые компетенции
ВОПРОСЫ ОТКРЫТОГО ТИПА	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Комплекс программных средств для автоматизации проектирования, разработки и сопровождения ПО называется _____. +: CASE-средство	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Программное обеспечение, предоставляющее комплекс инструментов для написания, отладки и сборки кода, называется _____. +: интегрированная среда разработки	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Система для отслеживания изменений в исходном коде и совместной работы называется система контроля _____. +: версий	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Методология разработки, основанная на итерациях и гибком реагировании на изменения, называется _____. +: Agile	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Набор готовых компонентов и библиотек для ускорения разработки приложений называется _____. +: фреймворк	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Набор базовых операций для работы с данными: создание, чтение, обновление, удаление — называется _____. +: CRUD	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Функция информационной системы, позволяющая находить данные по заданным критериям, называется _____. +: поиск	ОК 02

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс отбора данных по одному или нескольким условиям называется _____. +: фильтрация	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ)	ПК 3.3.

Фильтр, который применяет условия к данным по мере ввода пользователем символов, называется _____ фильтр. +: динамический	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Упорядочивание данных по выбранному параметру (например, от А до Я) называется _____. +: сортировка	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Уникальный признак, позволяющий отличить один объект системы от другого, называется _____. +: идентификатор	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс подтверждения заявленной пользователем идентичности (например, с помощью логина и пароля) называется _____. +: аутентификация	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Метод аутентификации с использованием одноразового кода, отправляемого на телефон, называется _____ пароль. +: одноразовый	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Аутентификация по отпечатку пальца или радужной оболочке глаза относится к _____ методам. +: биометрическим	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Механизм проверки, при котором пользователь должен ввести искаженный текст с картинки, называется _____. +: капча	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс определения, какие действия и ресурсы разрешены аутентифицированному пользователю, называется _____. +: авторизация	ПК 3.3.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Механизм управления доступом, основанный на назначении пользователям ролей с определенными правами, называется _____ управление доступом. +: ролевое	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс объединения изменений из разных веток в системе контроля версий называется _____. +: слияние	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Ситуация, когда система контроля версий не может автоматически объединить изменения из двух веток, называется _____. +: конфликт	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ)	ПК 3.3.

Совокупность правил и рекомендаций по построению пользовательского интерфейса относится к _____ интерфейса. +: проектированию	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Структура данных, организованная по принципу «первым пришел — первым вышел», называется _____. +: очередь	ПК 3.3.
ВОПРОСЫ ЗАКРЫТОГО ТИПА	
Какой компонент CASE-средства отвечает за хранение моделей и метаданных проекта? -: Компилятор +: Репозиторий -: Текстовый редактор -: Отладчик	ПК 3.3.
Какой элемент среды разработки предоставляет автодополнение кода и подсветку синтаксиса? -: Командная строка -: Система контроля версий +: Текстовый редактор с поддержкой IDE -: Файловый менеджер	ПК 3.3.
Какой подход к разработке предполагает фиксированные последовательные этапы? -: Agile -: Scrum +: Waterfall -: Kanban	ПК 3.3.

Какой инструмент используется для автоматического отображения объектов приложения на таблицы БД? -: API-шлюз +: ORM-фреймворк -: Веб-фреймворк -: Контейнеризация	ПК 3.3.
Какая из перечисленных операций НЕ входит в базовый набор CRUD? -: Create (Создание) -: Read (Чтение) +: Search (Поиск) -: Delete (Удаление)	ПК 3.3.
Какой метод поиска позволяет находить данные по их точному или частичному совпадению с образцом? +: Контекстный поиск -: Алгоритмический поиск -: Морфологический анализ -: Машинное обучение	ПК 3.3.
Какой тип фильтрации применяет условия отбора к данным в реальном времени по мере ввода символов? -: Статический фильтр	ПК 3.3.
+: Динамический фильтр -: Пакетный фильтр -: Периодический фильтр	
Какой механизм НЕ является стандартным для работы с коллекциями данных в языках программирования? -: Итератор -: Генератор -: Функция высшего порядка +: SQL-инъекция	ПК 3.3.
Какой тип фреймворка используется для создания десктопных приложений с графическим интерфейсом? +: GUI-фреймворк -: ORM-фреймворк -: Unit-тест фреймворк -: Микросервисный фреймворк	ПК 3.3.

МДК.03.03 Интеграция и модификация бизнес-приложений

Тестовое задание	Оцениваемые компетенции
ВОПРОСЫ ОТКРЫТОГО ТИПА	

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс объединения независимых приложений для совместной работы называется _____. +: интеграция	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Набор правил и форматов для обмена данными между системами называется _____. +: протокол	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Архитектурный стиль взаимодействия компонентов в распределенной среде на основе независимых сервисов называется _____. +: микросервисы	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Промежуточный слой, который обеспечивает маршрутизацию и преобразование сообщений между системами, называется _____. +: шина данных	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Формат обмена данными, основанный на парах «ключ — значение» и являющийся альтернативой XML, называется _____. +: JSON	ПК 3.4
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Программный интерфейс приложения, предназначенный для взаимодействия с внешними системами, называется _____. +: API	ПК 3.4

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Принцип проектирования API, при котором каждый запрос к серверу содержит всю информацию, необходимую для его выполнения, называется _____. +: REST	ПК 3.4
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Специальный компонент, который принимает все запросы от клиентов и направляет их к соответствующим микросервисам, называется _____. +: API Gateway	ПК 3.4.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Программное обеспечение, обеспечивающее асинхронную передачу сообщений между приложениями, называется брокером _____. +: сообщений	ПК 3.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Протокол асинхронного обмена сообщениями, широко используемый в системах интернета вещей (IoT), называется _____. +: MQTT	ПК 3.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Тип архитектуры, при котором приложения обмениваются данными через центральный брокер, а не напрямую, называется архитектурой на основе _____. +: событий	ПК 3.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс изменения структуры или кода приложения без изменения его внешнего поведения называется _____. +: рефакторинг	ПК 3.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Проблемы в структуре кода, которые могут замедлить разработку или увеличить риск ошибок, называются _____. +: технический долг	ПК 3.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс переноса данных из одной системы в другую с возможным изменением формата называется _____ данных. +: миграция	ПК 3.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс извлечения, преобразования и загрузки данных обозначается аббревиатурой _____. +: ETL	ПК 3.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Устаревшее, но все еще работающее приложение, которое сложно поддерживать и модернизировать, называется _____. +: унаследованным	ПК 3.5.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Введение новой версии API без прекращения поддержки старой называется обеспечением _____ совместимости. +: обратной	ПК 3.4.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Паттерн, предотвращающий повторные вызовы к отказавшему сервису, называется _____. +: предохранитель	ПК 3.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Очередь для сообщений, которые не удалось обработать, называется _____. +: очередь мертвых писем	ПК 3.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Технология, позволяющая упаковывать приложение и его зависимости в изолированную среду, называется _____. +: контейнеризация	ПК 3.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Технология, позволяющая отслеживать путь запроса через множество микросервисов для выявления узких мест и ошибок, называется распределенной _____. +: трассировкой	ПК 3.4.
ВОПРОСЫ ЗАКРЫТОГО ТИПА	
К стандартным протоколам синхронного взаимодействия при интеграции относится: -: AMQP -: MQTT +: HTTP/REST -: STOMP	ПК 3.4.
К стандартным протоколам асинхронного взаимодействия при интеграции относится: -: HTTP +: AMQP -: SOAP -: WebSocket (только как транспорт)	ПК 3.4.
К популярным брокерам сообщений относится: -: Nginx -: Apache Tomcat +: RabbitMQ -: Docker	ПК 3.4.
К форматам обмена данными в интеграции НЕ относится: -: JSON -: XML +: HTML -: YAML	ПК 3.4.
Что из перечисленного является методом обеспечения безопасности API? -: Увеличение таймаутов +: Rate Limiting (ограничение скорости запросов) -: Отключение кэширования -: Использование только POST-запросов	ПК 3.4.

Какой подход к модификации приложения позволяет добавлять новый функционал, не ломая существующий? -: Переписывание всего кода с нуля +: Использование принципа открытости/закрытости (OCP) -: Удаление устаревших модулей -: Увеличение частоты выхода новых версий	ПК 3.5.
Что является основной целью рефакторинга? -: Добавление новой функциональности -: Увеличение скорости работы приложения любой ценой +: Улучшение структуры и читаемости кода без изменения поведения -: Смена языка программирования	ПК 3.5.
Какой паттерн используется для обработки ошибок в распределенных системах и предотвращения каскадных сбоев? -: Singleton -: Factory Method +: Circuit Breaker -: Adapter	ПК 3.4.
Какой метод аутентификации чаще всего используется для взаимодействия микросервисов? -: Логин и пароль на каждый запрос +: JWT (JSON Web Token) -: Отправка номера кредитной карты -: Доступ по IP-адресу без проверки	ПК 3.4.

Шкала оценивания тестовых заданий

Данный вид контроля, рассчитан на выявление уровня усвоения теоретического и практического материала в рамках изучения дисциплин. Критерии оценивания теста: 1 вопрос = 1 балл.

Оценка «5»	30-27 Баллов
Оценка «4»	26-23 Баллов
Оценка «3»	22-18 Баллов
Оценка «2»	0-17 Баллов