Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гриб Владислав Валерьевич
Должность: Ректор
Дата подписания: 23.09.2026 19:01:43
Уникальный программный ключ:
637517d24e103c3db032acf37e839d98ec1c5bb2f5eb89c29abfcd7f43985447

Образовательное частное учреждение высшего образования
«МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ А.С. ГРИБОЕДОВА»
(ИМПЭ им. А.С. Грибоедова)

МЕЖДУНАРОДНЫЙ ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И БИЗНЕС-ИНФОРМАТИКИ

УТВЕРЖДАЮ
Директор Международного института информационных технологий и бизнес-информатики
_____ Т.В. Новикова
«17» декабря 2025г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по учебной дисциплине:
ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Направление подготовки
09.03.03 Прикладная информатика
(уровень бакалавриат)

Направленность (профиль):
«Анализ данных»

Форма обучения: **очная, заочная**

Москва

Фонд оценочных средств для дисциплины «Проектирование информационных систем». Направление подготовки/специальность 09.03.03 Прикладная информатика, направленность (профиль/специализация): Анализ Данных / И.В. Торицын – М.: ИМПЭ им. А.С. Грибоедова.

Фонд оценочных средств является неотъемлемой частью рабочей программы дисциплины.

Разработчик: И.В. Торицын

Заведующий кафедрой: _____/ Н. Н. Загускин

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

Настоящий Фонд оценочных средств (ФОС) по дисциплине «Проектирование информационных систем» является неотъемлемым приложением к рабочей программе дисциплины (РПД) «Проектирование информационных систем». 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. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЕ

3.1. Сопроводительная информация.

Разработчик	И.В. Торицын
Кафедра	Прикладной информатики и информационных технологий
Наименование дисциплины	Проектирование информационных систем
Факультет / институт	Институт международной экономики, лидерства и менеджмента
Направление подготовки / специальность	09.03.03 Прикладная информатика
Количество вопросов в оценочных заданиях (диапазон)	От 36 до 72
Общее время тестирования (мин)	90
Общее количество вопросов/заданий в ФОС	72
Размещенность на сайте Университета примерного перечня вопросов, заданий ФОС – для подготовки обучающихся к прохождению оценки (да / нет)	Да

3.2. Характеристика оцениваемых компетенций.

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
ПК-6	Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов
ПК-7	Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов

4. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ТЕКУЩЕГО КОНТРОЛЯ

4.1. ТИПОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ.

Тесты содержат набор вопросов, в полном объеме охватывающие изученный теоретический материал по указанной теме. Выполнение тестов позволяет определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций. Индивидуальный тестовый сеанс для каждого обучающегося формируется по специальному алгоритму, обеспечивающему заданную тематическую структуру и пропорциональное наличие вопросов разного типа и сложности.

При формировании тестов необходимо использовать задания следующих типов:

Тип задания 1. Задания закрытого типа на установление соответствия.

Тип задания 2. Задания закрытого типа на установление последовательности.

Тип задания 3. Задания комбинированного типа, предполагающие выбор одного правильного ответа из предложенных

Тип задания 4. Задания комбинированного типа, предполагающие выбор нескольких ответов из предложенных

Тип задания 5. Задания открытого типа с развернутым ответом.

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td>Тема 1. Проектирование информационных систем</td><td>ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения</td><td>ИПК-6.1 Знать методы и средства проектирования баз данных</td><td>
Задания закрытого типа на установление соответствия
Задание 1
Установите соответствие между этапами проектирования базы данных и их основными результатами:
<table>
<tr><td>Этап проектирования</td><td>Результат этапа</td></tr>
<tr><td>1. Инфологическое (концептуальное) проектирование</td><td>А. Нормализованная логическая структура таблиц с указанием первичных и внешних ключей</td></tr>
<tr><td>2. Даталогическое (логическое) проектирование</td><td>Б. Диаграмма «сущность-связь» (ER-диаграмма) без привязки к СУБД</td></tr>
<tr><td>3. Физическое проектирование</td><td>В. Оптимизированные индексы, выбор типов данных и параметров хранения с учётом СУБД</td></tr>
</table>
Правильное соответствие:
1 → Б
2 → А
3 → В

Задание 2
Установите соответствие между типом связи в ER-модели и способом её реализации в реляционной базе данных:
<table>
<tr><td>Тип связи</td><td>Реализация в реляционной БД</td></tr>
<tr><td>1. Один ко многим (1:N)</td><td>А. Создание отдельной связующей таблицы с двумя внешними ключами</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td></td></tr>
<tr><td>2. Многие ко многим (M:N)</td><td>Б. Добавление первичного ключа одной сущности как внешнего ключа в таблицу другой сущности</td></tr>
<tr><td>3. Один к одному (1:1)</td><td>В. Объединение атрибутов обеих сущностей в одну таблицу либо размещение внешнего ключа с уникальным ограничением (UNIQUE)</td></tr>
</table>
Правильное соответствие:
1 → Б
2 → А
3 → В
</td></tr>
<tr><td>Тема 2. Жизненный цикл информационной системы</td><td>ПК-7 Способен осуществлять проектирование программных интерфейсов</td><td>ИПК-7.1 Знать методы и средства проектирования программных интерфейсов</td><td>
Задания закрытого типа на установление соответствия
Задание 1
Установите соответствие между этапами жизненного цикла информационной системы и задачами проектирования программных интерфейсов (API), характерными для каждого этапа:
<table>
<tr><td>Этап жизненного цикла ИС</td><td>Задача проектирования API</td></tr>
<tr><td>1. Анализ и проектирование</td><td>А. Настройка rate limiting, мониторинг доступности эндпоинтов и логирование запросов</td></tr>
<tr><td>2. Реализация и интеграция</td><td>Б. Определение контракта API: эндпоинты, форматы данных (JSON/CSV), методы аутентификации</td></tr>
<tr><td>3. Тестирование</td><td>В. Версионирование API (v1, v2) и поддержка backward compatibility при модификации схемы</td></tr>
<tr><td>4. Сопровождение и эксплуатация</td><td>Г. Написание интеграционных тестов для проверки корректности возврата аналитических данных</td></tr>
</table>
Правильное соответствие:
</td></tr>
</table>

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td></td><td></td><td></td><td>1 → Б
2 → Г
3 → В
4 → А

Задание 2
Установите соответствие между моделью жизненного цикла ИС и подходом к проектированию и эволюции программных интерфейсов:
<table><tr><td>Модель жизненного цикла</td><td>Подход к проектированию и развитию API</td></tr><tr><td>1. Каскадная (водопадная) модель</td><td>А. Постепенное расширение функционала API через новые эндпоинты с сохранением старых версий</td></tr><tr><td>2. Итеративная модель</td><td>Б. Чёткое определение полного контракта API на этапе проектирования с минимальными изменениями в будущем</td></tr><tr><td>3. Гибкие методологии (Agile/DevOps)</td><td>В. Непрерывная доставка обновлений API с автоматизированным тестированием и мониторингом в продакшене</td></tr><tr><td>4. Спиральная модель</td><td>Г. Прототипирование API на каждой итерации с оценкой рисков интеграции источников данных</td></tr></table>Правильное соответствие:
1 → Б
2 → А
3 → В
4 → Г</td></tr>
</table>

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td>Тема 3. Стандартизация разработки программных средств</td><td>ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения</td><td>ИПК-6.1 Знать методы и средства проектирования баз данных</td><td>
Задания закрытого типа на установление соответствия
Задание 1
Установите соответствие между стандартами и методологиями проектирования баз данных и их основным назначением в контексте разработки информационных систем для анализа данных:
<table>
<tr><th>Стандарт / Методология</th><th>Основное назначение</th></tr>
<tr><td>1. ГОСТ 34.601-90</td><td>А. Определение трёхуровневой архитектуры БД (внешний, концептуальный, внутренний уровни)</td></tr>
<tr><td>2. Модель ANSI/SPARC</td><td>Б. Регламентация этапов жизненного цикла автоматизированных систем (включая проектирование БД)</td></tr>
<tr><td>3. Нотация IDEF1X</td><td>В. Формализованное описание семантики данных и метаданных для обеспечения их повторного использования в аналитических системах</td></tr>
<tr><td>4. ГОСТ Р ИСО/МЭК 11179</td><td>Г. Графическое моделирование структуры БД с чёткими правилами отображения сущностей, атрибутов и связей</td></tr>
</table>
Правильное соответствие:
1 → Б
2 → А
3 → Г
4 → В

Задание 2
Установите соответствие между правилами стандартизации при проектировании баз данных и их практическим применением в аналитических информационных системах:
<table>
<tr><th>Правило стандартизации</th><th>Практическое применение в аналитической ИС</th></tr>
<tr><td>1. Единые соглашения об</td><td>А. Применение префиксов dim для</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Тест по теме</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>именовании (naming conventions)</td><td>таблиц измерений и fact_ для таблиц фактов в схеме «звезда»</td></tr>
<tr><td>2. Нормализация до 3НФ</td><td>Б. Разделение таблицы «Клиенты» на отдельные таблицы «Клиенты», «Адреса», «Контакты» для устранения избыточности</td></tr>
<tr><td>3. Денормализация для OLAP-систем</td><td>В. Создание единого глоссария бизнес-терминов с привязкой к техническим атрибутам таблиц для согласования метрик между аналитиками и разработчиками</td></tr>
<tr><td>4. Стандарты документирования моделей данных</td><td>Г. Объединение таблиц измерений с таблицей фактов в единую плоскую структуру для ускорения агрегации данных</td></tr>
</table>
Правильное соответствие:
1 → А
2 → Б
3 → Г
4 → В
</td></tr>
<tr><td>Тема 4. Модели жизненного цикла</td><td>ПК-7 Способен осуществлять проектирование программных интерфейсов</td><td>ИПК-7.1 Знать методы и средства проектирования программных интерфейсов</td><td>Задания закрытого типа на установление последовательности
Задание 1
Расположите этапы проектирования и внедрения RESTful API для аналитической информационной системы в соответствии с каскадной моделью жизненного цикла (ГОСТ 34.601-90):
А. Разработка технического задания на API: определение эндпоинтов, форматов данных (JSON/CSV), требований к аутентификации
Б. Написание программного кода эндпоинтов, реализация логики агрегации и фильтрации аналитических данных
В. Составление отчётности по результатам приёмочных испытаний: проверка корректности возврата метрик, производительности запросов
Г. Проведение модульного и интеграционного тестирования эндпоинтов с использованием инструментов (Postman, Swagger)</td></tr>
</table>

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			Д. Анализ предметной области: выявление источников данных, бизнес-метрик и требований заинтересованных лиц к аналитическим отчётам Правильная последовательность: Д → А → Б → Г → В Задание 2 Расположите фазы итерации в методологии Agile/Scrum при разработке версии 2.0 API для системы бизнес-аналитики, начиная с планирования и заканчивая ретроспективой: А. Демонстрация заинтересованным лицам новых эндпоинтов для получения аналитических дашбордов в реальном времени Б. Адаптация бэклога продукта: приоритизация задач по улучшению производительности API на основе метрик мониторинга В. Проведение спринт-планирования: формирование бэклога спринта с задачами «реализовать пагинацию», «добавить фильтрацию по времени» Г. Ежедневные стендапы команды: синхронизация прогресса по реализации эндпоинтов, выявление блокеров интеграции с источниками данных Д. Проведение ретроспективы: анализ эффективности процесса разработки API, выработка улучшений для следующей итерации Правильная последовательность: Б → В → Г → А → Д
Тема 5. Прототипы	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	**Задания закрытого типа на установление последовательности** Задание 1 Расположите этапы создания прототипа базы данных для аналитической системы в правильной последовательности от концептуального проектирования до верификации: А. Формирование таблиц измерений (измерения «Время», «Клиент», «Товар») и таблицы фактов с внешними ключами Б. Построение концептуальной модели «сущность-связь» с выделением бизнес-сущностей и их атрибутов В. Преобразование ER-модели в логическую реляционную схему с нормализацией до 3НФ Г. Денормализация логической схемы в модель «звезда» для оптимизации аналитических запросов Д. Верификация прототипа: выполнение типовых OLAP-запросов (агрегация, срезы,

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			сводные отчёты) и оценка производительности Правильная последовательность: Б → В → А → Г → Д Задание 2 Расположите действия при итеративном прототипировании витрины данных (data mart) для отдела маркетинга в порядке их выполнения в рамках спиральной модели жизненного цикла: А. Реализация ETL-скрипта для загрузки данных из источников в прототип витрины (таблицы фактов продаж и измерений) Б. Сбор обратной связи от маркетологов: анализ корректности метрик, полноты измерений, удобства формирования отчётов В. Определение границ прототипа: выбор ключевых метрик (выручка, конверсия) и измерений (период, канал привлечения, сегмент клиента) Г. Оценка рисков: проверка качества исходных данных, доступности источников, соответствия бизнес-требованиям Д. Визуализация результатов: подключение прототипа к инструменту (Power BI / Tableau) и демонстрация интерактивных дашбордов Правильная последовательность: В → Г → А → Д → Б
Тема 6. Анализ и моделирование функциональной области внедрения ИС	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	**Задания закрытого типа на установление последовательности** Задание 1 Расположите этапы анализа функциональной области и проектирования API для внедрения системы бизнес-аналитики в порядке их выполнения: А. Декомпозиция бизнес-процессов «Продажи» и «Маркетинг» с выделением операций, генерирующих данные для анализа (оформление заказа, клик по баннеру, оплата) Б. Формирование требований к контракту API: определение эндпоинтов /api/sales/aggregate, /api/customers/segment, форматов ответа (JSON с метриками), параметров фильтрации В. Сбор требований от заинтересованных лиц: определение ключевых метрик (выручка, конверсия, LTV) и сценариев использования аналитических отчётов Г. Создание диаграммы потоков данных (DFD): отображение источников данных

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			(CRM, веб-аналитика), процессов агрегации и потребителей (дашборды, отчётность) Д. Моделирование интерфейсов взаимодействия с внешними системами: определение точек интеграции (API CRM, Google Analytics) и форматов обмена данными Правильная последовательность: В → А → Г → Д → Б Задание 2 Расположите действия при моделировании функциональной области «Управление клиентскими данными» для построения единого профиля клиента (Customer 360°) в правильной последовательности: А. Проектирование интерфейса агрегации данных: разработка эндпоинта /api/customer/{id}/profile с объединением данных из всех источников в единый JSON-ответ Б. Анализ источников данных: выявление систем-доноров (CRM, колл-центр, мобильное приложение) и их текущих методов экспорта данных (API, файлы, БД) В. Определение функциональных требований: составление списка атрибутов единого профиля (персональные данные, история покупок, поведенческие метрики, жалобы) Г. Моделирование бизнес-объекта «Клиент»: создание атрибутивной модели с классификацией атрибутов на идентификаторы, измерения и факты Д. Спецификация требований к внешним интерфейсам: формирование технических заданий на адаптацию API источников (добавление полей, изменение форматов, обеспечение частоты обновления) Правильная последовательность: В → Б → Г → Д → А
Тема 7. Полная бизнес-модель компании	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	**Задания комбинированного типа (выбор одного ответа)** Задание 1 При проектировании хранилища данных для полной бизнес-модели компании «Ритейл+», включающей операционные процессы (продажи, логистика, закупки) и стратегические метрики (рентабельность, доля рынка), архитектор БД должен определить структуру таблиц измерений. Какой подход к моделированию измерения «Продукт» обеспечит максимальную гибкость для кросс-функционального анализа (продажи по категориям, маржинальность по поставщикам, оборачиваемость по складам)?

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			Варианты ответов: А. Создать единую плоскую таблицу dim_product со всеми атрибутами: product_id, name, category, supplier_id, warehouse_id, cost_price, selling_price. Б. Создать иерархическую структуру измерений: dim_product → dim_category → dim_product_group с внешними ключами, а атрибуты поставщика и склада вынести в отдельные измерения dim_supplier, dim_warehouse. В. Денормализовать модель, продублировав атрибуты категорий и поставщиков непосредственно в таблице фактов fact_sales для ускорения запросов. Г. Использовать документную модель (JSONB в PostgreSQL) для хранения всех атрибутов продукта в одном поле с индексацией по ключевым полям. Правильный ответ: Б Объяснение: Иерархическая структура измерений (Б) обеспечивает нормализацию справочников, исключает избыточность данных и поддерживает сквозную аналитику по разным бизнес-функциям. Для полной бизнес-модели критично разделение контекстов: категория относится к маркетингу, поставщик — к закупкам, склад — к логистике; их выделение в отдельные измерения позволяет строить гибкие срезы без дублирования бизнес-логики в таблице фактов. Задание 2 В полной бизнес-модели производственной компании выделены ключевые бизнес-объекты: «Заказ», «Партия», «Оборудование», «Сотрудник». При концептуальном проектировании БД возникает вопрос: как корректно отразить связь «Сотрудник участвует в производстве нескольких Партий, а одна Партия производится бригадой из нескольких Сотрудников». Какой тип связи и её реализация в реляционной модели соответствует бизнес-семантике и поддерживает аналитику трудозатрат? Варианты ответов: А. Связь «многие-ко-многим» (M:N) между employee и batch с созданием связующей таблицы employee_batch с атрибутами hours_worked, role, shift_date. Б. Связь «один-ко-многим» (1:N) с добавлением внешнего ключа employee_id в таблицу batch, предполагая ответственного за партию. В. Связь «один-к-одному» (1:1) через объединение атрибутов сотрудника и партии в единую таблицу production_unit.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			Г. Отказ от явной связи, хранение списка сотрудников в текстовом поле workers_list таблицы batch с последующим парсингом при анализе. Правильный ответ: А Объяснение: Связь M:N (А) точно отражает бизнес-семантику коллективного участия сотрудников в производстве партий. Связующая таблица employee_batch с атрибутами hours_worked, role позволяет анализировать трудозатраты на микроуровне (кто, сколько, в какой роли), что невозможно при упрощённых связях 1:N или 1:1 и критично для расчёта себестоимости в полной бизнес-модели.
Тема 8. Спецификация функциональных требований к ИС	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	**Задания комбинированного типа (выбор одного ответа)** Задание 1 При спецификации функциональных требований к системе бизнес-аналитики для онлайн-ритейла определено требование: «Аналитик должен иметь возможность получать агрегированные метрики продаж за произвольный период с разбивкой по регионам, категориям товаров и каналам продаж». Какой формат спецификации эндпоинта API наиболее полно и корректно отразит данное требование с учётом гибкости аналитических запросов? Варианты ответов: А. /api/sales/report — POST-запрос с телом в формате JSON, содержащим структуру фильтров (date_from, date_to, regions[], categories[], channels[]) и список запрашиваемых метрик (revenue, units_sold, avg_check). Б. /api/sales/2024/01/01/to/2024/12/31/by/region/category/channel — GET-запрос с жёстко закодированным периодом и измерениями в пути URL. В. /api/sales — GET-запрос без параметров, возвращающий полный дамп всех продаж за последний месяц в формате CSV для последующей обработки аналитиком вручную. Г. /api/report1, /api/report2, /api/report3 — набор фиксированных эндпоинтов для заранее определённых отчётов («продажи по регионам», «продажи по категориям» и т.д.). Объяснение: Вариант А обеспечивает декларативную спецификацию требований: структура

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			JSON-тела точно отражает бизнес-логику аналитического запроса (произвольный период, множественные измерения, выбор метрик) без жёсткой привязки к конкретным значениям. Такой подход соответствует принципам проектирования аналитических API — поддерживает динамические сценарии анализа, минимизирует количество эндпоинтов и позволяет валидировать сложные комбинации фильтров на уровне схемы запроса (например, через OpenAPI Specification). Задание 2 В спецификации функциональных требований к ИС указано: «Система должна предоставлять данные о клиентах для сегментации в режиме реального времени с задержкой не более 5 секунд». При проектировании интерфейса интеграции с системой сегментации возникает вопрос выбора паттерна взаимодействия. Какой подход к проектированию API корректно отразит данное требование к временным характеристикам и обеспечит его верифицируемость? Варианты ответов: А. Синхронный REST API с заголовком X-Expected-Latency: 5000, где сервер обязан возвращать ответ в течение 5 секунд, иначе клиент получает статус 504 Gateway Timeout. Б. Асинхронный паттерн: эндпоинт /api/segmentation/jobs создаёт задачу с возвратом job_id, а клиент опрашивает статус через /api/segmentation/jobs/{id} до завершения обработки. В. Вебхуки (webhooks): система сегментации подписывается на события «новый клиент», «изменение поведения» и получает данные по push-механизму без запросов. Г. Потоковая передача (Server-Sent Events): эндпоинт /api/segmentation/stream устанавливает постоянное соединение и передаёт обновления клиентских данных в реальном времени. Правильный ответ: А Объяснение: Требование «режим реального времени с задержкой ≤5 сек» описывает

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			синхронный сценарий взаимодействия, где аналитик ожидает немедленного результата для принятия решений. Вариант А транслирует бизнес-требование в верифицируемую техническую спецификацию: SLA фиксируется в контракте API (заголовок или документация), а превышение лимита приводит к однозначному статусу ошибки, что позволяет автоматизировать тестирование производительности. Асинхронные паттерны (Б, В, Г) не соответствуют семантике «режима реального времени» для интерактивной аналитики, так как вводят неопределённость времени получения результата.
Тема 9. Структурные и объектные модели предметной области	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	**Задания комбинированного типа (выбор одного ответа)** Задание 1 При анализе предметной области «Электронная коммерция» построены две модели: • Структурная модель (DFD): отражает потоки данных между процессами «Оформление заказа», «Оплата», «Доставка» с хранилищами «Корзина», «Заказы», «Платежи». • Объектная модель (диаграмма классов UML): выделяет бизнес-объекты Customer, Order, OrderItem, Product, Payment с атрибутами и связями. При проектировании операционной БД (OLTP) для обработки транзакций и хранилища данных (OLAP) для анализа продаж необходимо трансформировать эти модели в реляционные структуры. Какой подход к проектированию таблиц корректно учитывает различия между моделями и специфику аналитической нагрузки? Варианты ответов: А. Использовать единую нормализованную схему 3НФ для обеих систем, так как объектная модель уже обеспечивает структурную целостность данных. Б. Для OLTP реализовать нормализованную схему на основе объектной модели (таблицы orders, order_items, products), а для OLAP построить денормализованную модель «звезда» на основе структурной модели потоков данных (факты = события потока: оформление, оплата, доставка). В. Игнорировать структурную модель и проектировать обе БД исключительно по объектной модели, так как она содержит полное описание бизнес-сущностей. Г. Применить документную модель NoSQL для обеих систем, чтобы сохранить иерархию объектов из UML-диаграммы без трансформации. Правильный ответ: Б Объяснение:

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			OLTP-система оптимизирована под транзакции и требует нормализации (объектная модель → 3НФ) для обеспечения целостности операционных данных. OLAP-система оптимизирована под агрегацию, поэтому факты моделируются как события потока данных из структурной модели (оформление = факт продажи, оплата = факт поступления средств), а измерения берутся из объектной модели — это позволяет анализировать динамику бизнес-процессов, а не только статику сущностей. Задание 2 В предметной области «Банковский риск-менеджмент» построена объектная модель с классами Client, Loan, Collateral, Guarantor и связями «один ко многим» между ними. При проектировании БД для анализа кредитного портфеля выявлено требование: «Рассчитывать динамику обеспечения по каждому кредиту с учётом истории изменений залогов (добавление, изъятие, переоценка)». Какой подход к проектированию структуры хранения обеспечит корректное отражение временной семантики из предметной области? Варианты ответов: А. Создать таблицу collateral с полями loan_id, asset_type, value и внешним ключом на loan, обновляя запись при изменении залога (UPDATE). Б. Добавить в таблицу collateral поля valid_from и valid_to для хранения периода действия каждой версии залога (модель Slowly Changing Dimension Type 2). В. Хранить историю изменений в отдельной таблице collateral_audit с полями old_value, new_value, changed_at, не используя её для аналитических расчётов. Г. Отказаться от реляционной модели и использовать графовую БД для отображения динамических связей «кредит — залог». Правильный ответ: Б Объяснение: Простое обновление записей (вариант А) уничтожает историю изменений, делая невозможным расчёт корректной динамики обеспечения на произвольную дату в прошлом — критически важного сценария для риск-анализа. Модель SCD Type 2 (Б) сохраняет все версии залога с периодами действия, что позволяет строить «срезы» кредитного портфеля на любую дату и точно рассчитывать метрики вроде «покрытие залогом на 01.01.2024», что напрямую отражает временную семантику предметной области в структуре БД.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
Тема 10. Унифицированный процесс разработки	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	**Задания комбинированного типа (выбор нескольких ответов)** Задание 1 В фазе уточнения (Elaboration) Унифицированного процесса разработки для системы бизнес-аналитики необходимо зафиксировать архитектурно значимые решения по проектированию API. Какие из перечисленных действий обязательны на этой фазе для обеспечения стабильности контракта интерфейсов при последующих итерациях? Варианты ответов (выберите 2–3 правильных): А. Разработка полного технического задания на все эндпоинты с указанием точных форматов ответа для каждого атрибута аналитической метрики. Б. Определение архитектурного стиля API (REST, GraphQL, gRPC) и принципов версионирования (в пути, заголовках, параметрах). В. Создание «скелета» критических эндпоинтов (например, /api/analytics/aggregate) с заглушками и валидацией параметров без реализации бизнес-логики. Г. Фиксация базового набора измерений и метрик, которые будут доступны через API (например: выручка, конверсия, период, регион), с описанием их семантики в глоссарии. Д. Написание нагрузочных тестов для всех эндпоинтов с эмуляцией 10 000 одновременных запросов от аналитиков. Правильные ответы: Б, В, Г Объяснение: Фаза уточнения направлена на снижение архитектурных рисков: выбор стиля и версионирования (Б) задаёт основу стабильности контракта, «скелет» эндпоинтов (В) позволяет проверить интегрируемость интерфейсов до реализации логики, а фиксация семантики метрик (Г) предотвращает разночтения между аналитиками и разработчиками. Полная спецификация форматов (А) и нагрузочное тестирование (Д) относятся к фазе конструирования, когда архитектура уже утверждена и ведётся детальная реализация. Задание 2 При применении Унифицированного процесса к проекту создания единой платформы аналитики для розничной сети выделены дисциплины «Требования» и «Анализ и проектирование». Какие практики проектирования программных интерфейсов корректно распределены между этими дисциплинами в рамках одной итерации фазы

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			конструирования? Варианты ответов (выберите 2 правильных): А. Дисциплина «Требования»: описание пользовательских сценариев использования API («Аналитик формирует отчёт по продажам за квартал с разбивкой по категориям»); Дисциплина «Анализ и проектирование»: проектирование структуры запроса эндпоинта /api/sales/aggregate с параметрами period=Q1_2024, dimensions=["category"]. Б. Дисциплина «Требования»: выбор инструмента для документирования API (Swagger/OpenAPI); Дисциплина «Анализ и проектирование»: написание спецификации OpenAPI 3.0 для всех эндпоинтов. В. Дисциплина «Требования»: определение нефункционального требования «Время ответа эндпоинта агрегации ≤ 3 сек при объёме данных 100 млн записей»; Дисциплина «Анализ и проектирование»: выбор стратегии кэширования и индексирования для выполнения данного требования. Г. Дисциплина «Требования»: проектирование схемы базы данных для хранения аналитических метрик; Дисциплина «Анализ и проектирование»: реализация SQL-запросов для эндпоинтов. Д. Дисциплина «Требования»: определение формата аутентификации (OAuth 2.0); Дисциплина «Анализ и проектирование»: реализация механизма выдачи токенов и мидлвара проверки прав доступа к эндпоинтам. Правильные ответы: А, В Объяснение: Дисциплина «Требования» фокусируется на что система должна делать (бизнес-сценарии, метрики SLA), а «Анализ и проектирование» — на как это реализовать технически (структура эндпоинтов, архитектурные паттерны). Вариант А корректно разделяет сценарий использования и техническую спецификацию запроса; вариант В — требование к производительности и техническое решение для его достижения. Варианты Б, Г, Д смешивают инструментальную/реализационную деятельность с определением требований, что нарушает принцип разделения ответственности дисциплин в UP.
Тема 11. Унифицированный язык	ПК-6 Способен применять методы и	ИПК-6.1 Знать методы и средства проектирования баз данных	**Задания комбинированного типа (выбор нескольких ответов)** Задание 1 При проектировании хранилища данных для анализа клиентского поведения в

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
моделирование (UML)	средства проектирования компьютерного программного обеспечения		розничной сети используется диаграмма классов UML для моделирования измерения «Клиент». Какие элементы диаграммы классов корректно отражают специфику аналитического измерения с поддержкой истории изменений атрибутов (например, смена адреса, обновление сегмента лояльности)? Варианты ответов (выберите 2 правильных): А. Ассоциация «один ко многим» между классами Customer и CustomerAddress с атрибутом is_current: boolean в CustomerAddress. Б. Композиция между Customer и CustomerVersion, где каждый экземпляр CustomerVersion содержит полный снимок атрибутов клиента на определённую дату. В. Зависимость (dependency) от класса Address к классу Customer с пометкой «<<history>>» в стереотипе. Г. Атрибуты valid_from и valid_to непосредственно в классе Customer с ограничением {temporal} в заметке. Д. Обобщение (наследование) класса Customer от абстрактного класса SCD_Type2_Entity. Правильные ответы: А, Г Объяснение: Вариант А отражает паттерн хранения истории через выделение сущности-истории (CustomerAddress) с признаком актуальности — стандартный подход для атрибутов с частыми изменениями. Вариант Г моделирует медленно меняющееся измерение типа 2 (SCD Type 2) через временные метки непосредственно в классе — корректно для атрибутов с редкими изменениями. Вариант Б (композиция с полным снимком) избыточен и нарушает нормализацию; вариант В (зависимость) не передаёт семантику владения данными; вариант Д (наследование) некорректен, так как SCD Type 2 — это стратегия хранения, а не тип сущности. Задание 2 При проектировании операционной БД и хранилища данных для системы электронной коммерции необходимо выбрать типы диаграмм UML для разных этапов проектирования. Какие соответствия между задачей проектирования и типом диаграммы UML являются методически корректными? Варианты ответов (выберите 3 правильных): А. Моделирование бизнес-сущностей «Заказ», «Товар», «Клиент» и их связей на этапе

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			концептуального проектирования → Диаграмма классов. Б. Описание последовательности шагов преобразования данных из операционной БД в таблицу фактов при загрузке витрины → Диаграмма последовательностей. В. Отображение потоков данных между источниками (CRM, веб-сайт) и процессами (очистка, агрегация) на этапе анализа требований → Диаграмма деятельности. Г. Моделирование жизненного цикла заказа («создан» → «оплачен» → «отгружен» → «доставлен») с триггерами переходов → Диаграмма состояний. Д. Спецификация контракта REST API для получения аналитических отчётов → Диаграмма компонентов. Е. Визуализация архитектуры системы: веб-сервер, приложение аналитики, СУБД, кэш → Диаграмма развёртывания. Правильные ответы: А, Г, Е Объяснение: Диаграмма классов (А) — основной инструмент концептуального моделирования сущностей и связей, напрямую транслируемый в структуру таблиц. Диаграмма состояний (Г) корректно отражает бизнес-логику жизненного цикла сущностей, влияющую на проектирование атрибутов статуса и триггеров. Диаграмма развёртывания (Е) необходима для проектирования физической архитектуры БД (мастер-реплика, шардинг). Диаграмма последовательностей (Б) фокусируется на взаимодействии объектов во времени, а не на преобразовании данных; для ETL-процессов уместнее диаграмма деятельности. Диаграмма деятельности (В) может дополнять анализ, но основной инструмент для потоков данных — DFD, а не UML. Диаграмма компонентов (Д) описывает модульную структуру ПО, а не контракт API — для этого используется спецификация OpenAPI.
Тема 12. Определение требований в рамках унифицированного процесса	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	**Задания комбинированного типа (выбор нескольких ответов)** Задание 1 В дисциплине «Требования» Унифицированного процесса для системы бизнес-аналитики разработана карта прецедентов. Какие из перечисленных прецедентов корректно специфицируют функциональные требования к API с учётом специфики профиля «Анализ данных» и принципов трассируемости требований? Варианты ответов (выберите 2 правильных): А. «Аналитик получает агрегированные метрики продаж» с основным потоком: «Система возвращает сумму выручки, количество заказов и средний чек за указанный

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			период, сгруппированные по выбранным измерениям (регион, категория)». Б. «Система обрабатывает 10 000 запросов в секунду» как отдельный прецедент с потоком событий. В. «Аналитик формирует когортный анализ удержания» с альтернативным потоком: «Если в запросе указан параметр cohort_type=date_of_first_purchase, система рассчитывает retention rate по неделям с момента первой покупки». Г. «Разработчик вызывает эндпоинт /api/v1/sales» как прецедент с актёром «Внешняя система». Д. «Система возвращает данные в формате JSON» как отдельный прецедент с постусловием «Ответ содержит поля data, metadata, pagination». Правильные ответы: А, В Объяснение: Прецеденты А и В корректны, так как описывают цели актёра-аналитика в бизнес-терминах (агрегация метрик, когортный анализ) с детализацией потоков, что обеспечивает трассируемость от бизнес-требования к контракту API. Вариант Б описывает нефункциональное требование (производительность), которое фиксируется в спецификации, а не в прецеденте; вариант Г ошибочно делает актёром разработчика вместо конечного пользователя; вариант Д специфицирует техническую деталь реализации, а не бизнес-цель, что нарушает абстракцию уровня прецедентов. Задание 2 При определении требований к API системы аналитики в рамках итерации фазы уточнения (Elaboration) необходимо зафиксировать не только функциональные, но и архитектурно значимые нефункциональные требования. Какие из перечисленных требований следует включить в спецификацию на этом этапе, чтобы обеспечить стабильность контракта интерфейсов при масштабировании? Варианты ответов (выберите 3 правильных): А. «Эндпоинт агрегации должен поддерживать пагинацию результатов с параметрами page и page_size для обработки выборок свыше 10 000 записей». Б. «Все эндпоинты должны использовать аутентификацию по токену OAuth 2.0 с сроком действия 1 час». В. «Система должна возвращать данные о продажах за 2024 год». Г. «При изменении схемы ответа эндпоинта должна поддерживаться обратная

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			совместимость в течение 6 месяцев через механизм версионирования в заголовке API-Version». Д. «Цветовая схема дашборда должна соответствовать корпоративному стилю». Е. «Время ответа эндпоинта /api/analytics/aggregate при объёме данных до 100 млн записей не должно превышать 5 секунд для 95% запросов». Правильные ответы: А, Б, Г, Е (выбрать любые 3 из 4) Объяснение: Требования А (пагинация), Б (аутентификация), Г (версионирование) и Е (производительность) определяют контракт API и его масштабируемость — их изменение на поздних этапах потребует переделки клиентских приложений. Вариант В описывает конкретные данные, а не поведение интерфейса, и относится к наполнению системы; вариант Д касается UI/UX дашборда, а не программного интерфейса. Архитектурно значимые требования фиксируются на этапе уточнения для снижения рисков изменения контракта в фазе конструирования.
Тема 13. Анализ структуры в рамках унифицированного процесса	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	**Задания открытого типа с развёрнутым ответом** Задание 1 Вопрос: В рамках какой фазы Унифицированного процесса выполняется анализ структуры предметной области для проектирования хранилища данных, и чем принципиально отличается этот анализ от структурного анализа операционной базы данных? Эталон ответа: Анализ структуры для хранилища данных выполняется в фазе уточнения (Elaboration), так как именно здесь закладывается архитектурная основа системы и снижаются ключевые риски, включая выбор модели данных. Для операционной БД (OLTP) анализ структуры ориентирован на нормализацию (3НФ), обеспечение целостности транзакций и минимизацию избыточности — артефакты: диаграмма классов с детализацией атрибутов и связей. Для хранилища данных (OLAP) анализ фокусируется на выделении измерений (устойчивые бизнес-контексты) и фактов (измеримые события), критерий качества — производительность агрегации и гибкость срезов. Артефакты: концептуальная модель «звезда/снежинка» с указанием гранулярности фактов. Принципиальное отличие: OLTP-структуры моделируют текущее состояние бизнес-процессов, OLAP-структуры — историю событий для анализа трендов. Это

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			требует разных подходов к анализу: для OLTP — акцент на связи и ограничения; для OLAP — на семантику измерений и аддитивность метрик. Задание 2 Вопрос: Почему при анализе структуры в дисциплине «Анализ и проектирование» Унифицированного процесса для систем анализа данных недостаточно ограничиться только диаграммой классов, и какие дополнительные структурные артефакты необходимо разработать? Эталон ответа: Диаграмма классов недостаточна для аналитических систем, так как фокусируется на статической структуре сущностей и не отражает ключевые аспекты анализа: временную динамику, гранулярность фактов и иерархии измерений. Необходимо дополнить анализ тремя артефактами. Во-первых, матрица «измерения-метрики» — определяет, какие агрегаты допустимы для каждой комбинации измерений (например, сумма выручки по регионам допустима, а средняя дата заказа — нет). Во-вторых, спецификация гранулярности фактовой таблицы — фиксирует атомарное событие («продажа товара в заказе»), что критично для корректной агрегации. В-третьих, модель медленно меняющихся измерений (SCD) — описывает стратегию хранения истории изменений атрибутов (например, смена сегмента клиента) через диаграмму состояний с указанием триггеров переходов. Эти артефакты обеспечивают проектирование структур, пригодных не только для хранения данных, но и для выполнения сложных аналитических запросов без искажения бизнес-семантики.
Тема 14. Проектирование в рамках унифицированного процесса	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	**Задания открытого типа с развёрнутым ответом** Задание 1 Почему проектирование контракта программного интерфейса (спецификация эндпоинтов, форматов данных, правил версионирования) должно быть завершено к концу фазы уточнения (Elaboration) Унифицированного процесса, а не отложено до фазы конструирования? Эталон ответа: Контракт API является архитектурно значимым решением, так как определяет точки интеграции между компонентами системы и внешними потребителями (дашборды,

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			отчётные инструменты). Его изменение после начала массовой разработки клиентских приложений ведёт к каскадному переделыванию кода и нарушению обратной совместимости — критический риск для аналитических систем, где интерфейсы потребляются десятками пользователей-аналитиков. Фаза уточнения предназначена именно для снижения таких рисков: к её завершению должна быть подтверждена жизнеспособность архитектуры через прототипирование ключевых сценариев (например, агрегация данных за период). Для аналитики особенно важно зафиксировать семантику метрик и измерений в контракте на этом этапе, так как их изменение (например, переопределение «выручки» с НДС на без НДС) делает исторические данные несопоставимыми. Версионирование как механизм обратной совместимости также проектируется в фазе уточнения — это архитектурное решение, а не деталь реализации. Отсрочка проектирования контракта до фазы конструирования превращает архитектурный риск в операционную проблему, требующую дорогостоящей миграции клиентских приложений. Задание 2 Как итеративная природа Унифицированного процесса влияет на подход к проектированию программных интерфейсов для аналитических систем? В чём состоит баланс между ранним утверждением контракта (для стабильности) и гибкостью при уточнении требований к метрикам и измерениям? Эталон ответа: Итеративность UP требует разделения контракта API на стабильные и эволюционирующие части. На ранних итерациях фазы уточнения фиксируется «каркас» интерфейса: структура запроса (параметры фильтрации по периоду, измерениям), формат ответа (вложенная структура {"data": [...], "metadata": {...}}), правила версионирования и аутентификации. Эти элементы обеспечивают стабильность интеграции клиентских приложений. Гибкость достигается через параметризацию содержимого: список доступных метрик и измерений выносится в отдельный эндпоинт /api/metadata/schema, а не жёстко кодируется в контракте основных эндпоинтов. При добавлении новой метрики (например, «коэффициент удержания») изменяется только содержимое справочника метаданных, а контракт эндпоинта агрегации /api/analytics/aggregate остаётся неизменным. Для критичных изменений схемы применяется стратегия параллельного развёртывания версий (v1 и v2

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			работают одновременно 6 месяцев). Такой подход соответствует духу итеративной разработки: бизнес-требования к метрикам уточняются в каждой итерации фазы конструирования, но технический контракт взаимодействия остаётся предсказуемым, что критично для аналитиков, строящих отчёты на основе стабильных интерфейсов.
Тема 15. Документирование процесса	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	**Задания открытого типа с развёрнутым ответом** Задание 1 Почему в проектировании баз данных для аналитических систем (хранилищ данных, витрин) документирование бизнес-семантики метрик и измерений критически важно на этапе концептуального проектирования, и чем грозит отсутствие такого документа при эксплуатации системы? Эталон ответа: Документирование бизнес-семантики (глоссарий метрик и измерений) критично, так как аналитические системы потребляются пользователями из разных подразделений, имеющими разное понимание терминов. Например, метрика «выручка» может трактоваться как сумма заказов (без учёта возвратов), как сумма оплаченных заказов или как сумма отгруженных товаров. Без единой спецификации в документе «Бизнес-глоссарий» аналитики строят отчёты на основе разных интерпретаций, что приводит к противоречивым выводам и ошибкам в принятии решений. На этапе концептуального проектирования фиксация семантики позволяет корректно спроектировать структуру измерений и аддитивность фактов. При отсутствии документации при изменении источников данных (например, переход на новую платёжную систему) невозможно определить, изменилась ли бизнес-логика расчёта метрики или только техническая реализация. Это ведёт к «размыванию» доверия к данным, росту трудозатрат на поддержку и невозможности аудита исторических отчётов — ключевые риски для зрелости аналитической культуры организации. Задание 2 Какие особенности документирования процесса проектирования баз данных обусловлены итеративной природой Унифицированного процесса, и почему «живая документация» предпочтительнее единовременно созданного технического задания для аналитических систем? Эталон ответа: Итеративная природа Унифицированного процесса требует, чтобы документация эволюционировала параллельно с моделью данных: на ранних итерациях фиксируется

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			концептуальная модель с ключевыми измерениями и фактами, на последующих — уточняется гранулярность, добавляются атрибуты измерений, специфицируются правила обработки медленно меняющихся измерений (SCD). «Живая документация» (например, спецификация в Confluence или DataHub с версионированием) позволяет отслеживать историю изменений схемы и обоснование архитектурных решений. Для аналитических систем это критично: при добавлении нового канала продаж необходимо документировать не только техническое изменение (новое значение в dim_channel), но и бизнес-контекст («маркетплейс Яндекс.Маркет»), правила атрибуции продаж и влияние на расчёт метрик. Единовременное ТЗ становится устаревшим после первой итерации конструирования, тогда как живая документация служит источником истины для аналитиков, инженеров данных и бизнес-заказчиков. Она обеспечивает трассируемость от бизнес-требования к структуре таблицы и позволяет быстро оценить влияние изменений на существующие отчёты — ключевой фактор снижения рисков при развитии аналитической платформы.
Тема 16. Организация и планирование работ	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	**Задания закрытого типа на установление последовательности** Задание 1 Расположите этапы планирования проектирования аналитического API в правильной последовательности в рамках итерации Унифицированного процесса: А. Спецификация контракта эндпоинта: параметры фильтрации, структура ответа, правила пагинации Б. Сбор бизнес-требований: определение метрик (выручка, конверсия) и измерений (период, регион, категория) В. Оценка трудозатрат на реализацию эндпоинта и интеграцию с источниками данных Г. Анализ источников данных: выявление систем-доноров и их текущих интерфейсов экспорта Д. Формирование бэклога спринта: включение задачи «Реализовать эндпоинт /api/analytics/aggregate» с критериями приёмки Правильная последовательность: Б → Г → А → В → Д Задание 2 Расположите действия при организации работ по внедрению версионирования API для системы бизнес-аналитики в порядке их выполнения:

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			А. Разработка механизма параллельной поддержки версий v1 и v2 в течение 6 месяцев Б. Уведомление потребителей API (аналитиков, владельцев дашбордов) о предстоящих изменениях В. Фиксация стратегии версионирования в архитектурной документации (в пути URL: /api/v1/...) Г. Анализ воздействия: выявление всех клиентских приложений, использующих текущую версию API Д. Проведение пилотного тестирования новой версии с группой опытных аналитиков Правильная последовательность: В → Г → Б → Д → А
Тема 17. Оценка стоимости разработки программного продукта	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	**Задания закрытого типа на установление последовательности** Задание 1 Расположите этапы оценки трудозатрат на проектирование витрины данных для аналитики продаж в правильной последовательности: А. Расчёт трудозатрат на проектирование измерений: нормализация справочников, реализация SCD Type 2 для клиентских сегментов Б. Определение функционального размера: подсчёт логических файлов данных (измерения «Клиент», «Товар», «Время») и групп транзакций (агрегация продаж) В. Корректировка оценки с учётом сложности источников: интеграция 5 разнородных систем (CRM, ERP, веб-аналитика) с разными форматами выгрузки Г. Формирование итоговой сметы: умножение функционального размера на норматив трудозатрат (20 чел.-часов на функциональную точку) с учётом корректирующих коэффициентов Д. Анализ требований: выделение бизнес-метрик (выручка, конверсия) и измерений (период, регион, категория) из спецификации заказчика Правильная последовательность: Д → Б → А → В → Г Задание 2 Расположите действия при экспертной оценке стоимости проектирования хранилища данных методом «планирование покер» в порядке их выполнения: А. Коллективное обсуждение расхождений: архитектор БД поясняет сложность реализации истории изменений сегментов клиентов, аналитик — требования к

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			гранулярности фактов Б. Индивидуальная оценка каждым экспертом: присвоение «очков сложности» (по шкале Фибоначчи) задаче «Проектирование схемы витрины маркетинговой аналитики» В. Формирование итоговой оценки: медиана индивидуальных оценок после достижения консенсуса умножается на норматив скорости команды (15 чел.-часов на 1 очко) Г. Декомпозиция задачи: выделение подзадач «концептуальное моделирование», «логическое проектирование измерений», «спецификация ETL-потоков» Д. Предоставление контекста: описание бизнес-требований (когортный анализ, атрибуция) и ограничений (срок 3 месяца, 2 источника данных) Правильная последовательность: Д → Г → Б → А → В
Тема 18. Инструментальные средства разработки программ	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	**Задания закрытого типа на установление последовательности** Задание 1 Расположите этапы применения инструментальных средств при проектировании аналитического API в правильной последовательности от спецификации до публикации документации: А. Генерация интерактивной документации (Swagger UI) из OpenAPI-спецификации для предоставления аналитикам Б. Валидация OpenAPI-спецификации в редакторе (Swagger Editor) на соответствие стандарту и семантическую корректность В. Прототипирование ответов эндпоинтов в Postman с использованием mock-сервера для демонстрации заказчику до реализации бэкенда Г. Создание черновика спецификации в формате OpenAPI YAML/JSON с описанием эндпоинтов, параметров и схем ответов Д. Экспорт коллекции запросов из Postman в формат OpenAPI для синхронизации прототипа со спецификацией Правильная последовательность: Г → Б → В → Д → А Задание 2 Расположите действия при настройке CI/CD-конвейера для обеспечения качества

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Тест по теме
			программных интерфейсов в порядке их выполнения: А. Настройка автоматического тестирования контракта API в Postman/Newman при каждом пуше в ветку develop Б. Интеграция линтера Spectral в конвейер для проверки OpenAPI-спецификации на соответствие корпоративным стандартам именования эндпоинтов В. Развёртывание новой версии API на тестовом окружении после успешного прохождения всех проверок Г. Конфигурация Git-репозитория: размещение OpenAPI-спецификации в каталоге /spec и коллекции Postman в /tests Д. Добавление шага автоматической публикации обновлённой документации на портал (Redocly/ReadMe) после мержа в main Правильная последовательность: Г → Б → А → В → Д

Критерии оценивания тестового задания:

Оценка	Критерии оценивания
отлично	от 90 до 100 % правильно выполненных заданий
хорошо	от 70 до 89 % правильно выполненных заданий
удовлетворительно	от 50 до 69 % правильно выполненных заданий
неудовлетворительно	менее 50 % правильно выполненных заданий

4.2 ТИПОВЫЕ ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Практические задания должны отражать умение обучающегося применять осваиваемую компетенцию в практических ситуациях и при решении производственных задач.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
Тема 1. Проектирование информационных систем	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Название: Разработка структуры базы данных информационной системы предприятия торговли бытовой техникой. Цель: Применение методов проектирования баз данных и структур данных для хранения информации о товарах, поставщиках, заказчиках и продажах. Задания: 1. Анализ предметной области Составьте список сущностей, участвующих в процессе торговли бытовой техники. Определите атрибуты каждой сущности. 2. Проектирование ER-диаграммы Разработайте концептуальную модель данных (ER-модель). Представьте связи между сущностями (отношения). 3. Нормализация Приведите таблицы базы данных к третьей нормальной форме (3NF). 4. Разработка физической модели Создайте физическую структуру таблиц, включая типы полей, ключи и индексы. Предложите оптимальные стратегии индексирования для ускорения запросов. 5. Описание архитектуры БД Опишите основные компоненты будущей информационной системы, необходимые для реализации хранилища данных. Практическое задание №2 Название: Проектирование клиент-серверного приложения информационной системы учета рабочего времени сотрудников. Цель: Применить принципы проектирования программного обеспечения и разработки программных интерфейсов для автоматизации процесса учёта рабочего времени сотрудников. Задания: 1. Сбор требований

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			Провести интервью с представителями отдела кадров и руководителями подразделений организации. Записать функциональные требования к приложению. 2. Выбор технологий Подберите инструменты и технологии для реализации проекта (например, Python/Django, Java/Spring Boot, PHP/Laravel, Node.js + Express.js и др.). Назовите преимущества выбранного решения. 3. Проектирование архитектурных компонентов Спроектируйте общую архитектуру приложения, включающую клиентскую и серверную части. Описать взаимодействия клиента и сервера. 4. Реализация основных функций Реализовать регистрацию пользователей, аутентификацию и авторизацию. Предоставить интерфейс для ввода рабочих часов сотрудниками. Автоматизировать процесс формирования отчетов. 5. Документирование API-интерфейса Подготовьте документацию по API-приложения с описанием всех эндпоинтов, передаваемых данных и возвращаемых результатов.
Тема 2. Жизненный цикл информационной системы	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Название: Проектирование жизненного цикла информационной системы управления запасами продукции на предприятии розничной торговли. Цель: Освоение этапов жизненного цикла информационной системы и применение соответствующих методов и инструментов на каждом этапе. Порядок выполнения задания: 1. Инициация проекта: Определите цели проекта, сформулируйте бизнес-требования и ограничения. Продемонстрируйте понимание целей заказчика и специфики бизнеса. 2. Анализ требований: Составьте перечень функциональных и нефункциональных требований к системе. Используйте диаграмму Use Case для описания вариантов использования системы. 3. Проектирование системы:

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			Выполните проектирование архитектуры системы, спроектируйте базу данных и определите структуру классов. Представьте UML-диаграммы (классовые, последовательностные, компонентные). 4. Разработка прототипа: Разработайте рабочий прототип системы, демонстрирующий ключевые функции (управление складскими остатками, оформление заказов). 5. Тестирование и верификация: Организуйте тестирование разработанного прототипа, используя Unit-тесты и интеграционные тесты. Оформите отчёт о результатах тестирования. 6. Эксплуатация и сопровождение: Спланируйте процессы эксплуатации и сопровождения системы. Разработайте рекомендации по внедрению и обучению персонала работе с системой. Практическое задание №2 Название: Управление жизненным циклом веб-портала образовательной организации. Цель: Изучение методологии управления проектом, направленного на создание и поддержку веб-ресурса учебного заведения. Порядок выполнения задания: 1. Определение целей и требований: Установите четкую цель проекта, выявите потребности целевой аудитории (студенты, преподаватели, администрация вуза). Зафиксируйте функциональные и нефункциональные требования к ресурсу. 2. Планирование работ: Постройте детальный график выполнения проекта с использованием методик планирования (метод критического пути, сетевое моделирование). Выделите контрольные точки и риски. 3. Организация команды: Распределите роли участников проекта (аналитик, разработчик, тестировщик, администратор). Опишите критерии оценки качества работы каждого члена команды. 4. Управление качеством: Выберите метрики и показатели качества продукта (нагрузка сайта, скорость загрузки

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			страниц, безопасность данных). Опишите стратегию улучшения показателей. 5. Обеспечение внедрения: Создайте подробный план запуска ресурса, включив этапы подготовки инфраструктуры, тестирования, обучения пользователей и технической поддержки. 6. Оценка эффективности: Оцените достигнутые результаты проекта, проанализировав удовлетворенность пользователей и выполнение поставленных задач. Оформите итоговый отчет о проекте.
Тема 3. Стандартизация разработки программных средств	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Название: Проектирование реляционной базы данных туристической фирмы Цель: Отработать навыки стандартизации разработки программных средств, ориентируясь на проектирование баз данных, применяя стандарты нормализации и принципов объектно-реляционного проектирования. Задания: 1.Анализ предметной области Провести исследование предметной области туристических услуг и выделить главные объекты, события и процессы. Выявить ключевые элементы данных, необходимых для регистрации туров, клиентов, бронирования мест и оплаты услуг. 2. Создание концепции модели данных Спроектировать ER-диаграмму (Entity Relationship Diagram) и определить взаимосвязи между объектами. Использовать общепринятые обозначения стандартов для отображения связей ("один-к-одному", "один-ко-многим", "многие-ко-многим"). 3. Нормализация схемы базы данных Преобразовать схему данных в третью нормальную форму (3NF). Проанализировать и устранить избыточность и аномалии данных, возникающие при вставке, обновлении и удалении записей. 4. Физический дизайн базы данных Осуществить преобразование нормализованной логической модели в физическую модель данных, включая определение типов данных, первичных ключей, индексов и

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			ограничений целостности. Реализовать использование триггеров и хранимых процедур для поддержания согласованности данных. 5. Оптимизация производительности Произвести оценку операций над базой данных и разработать рекомендации по улучшению производительности путем оптимизации запросов и индексации. Практическое задание №2 Название: Создание корпоративной базы данных отдела закупок крупной торговой сети Цель: Формирование навыков стандартизированного подхода к созданию схем баз данных с соблюдением отраслевых стандартов и подходов к документированию проектов. Задания: 1. Формулировка требований Перечислить функциональные и нефункциональные требования к проекту, основанные на стандарте ISO/IEC 29148 (Software and systems engineering — Life cycle processes — Requirements engineering). Подробно описать желаемые характеристики и функциональность системы закупки товаров. 2. Моделирование данных Создать подробную ERD (entity relationship diagram) для представленной системы, обозначив сущности, атрибуты и отношения. Используя правила стандарта IDEF1X, представить систему понятным образом, учитывающим стандарты проектирования и интеграции с существующими системами компании. 3. Стандартизация документов Составить техническую спецификацию базы данных согласно ГОСТ Р ИСО/МЭК 12207-2010 («Информационная технология. Процессы жизненного цикла программных средств»). Документально оформить выбранные алгоритмы обработки данных и условия

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			транзакций, используемые при взаимодействии модулей. 4. Генерация документации Оформить полные SQL-скрипты для создания всех объектов базы данных, следуя стандартам написания комментариев и оформления документации SQL-запросов. Сформировать описание всех созданных объектов и подготовить инструкцию по развертыванию базы данных. 5. Автоматизация процессов миграции данных Разработать сценарии миграционных скриптов для переноса данных из существующих устаревших систем в новую базу данных, соблюдая совместимость данных и целостность всей информации.
Тема 4. Модели жизненного цикла	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Название: Проектирование RESTful API для системы отслеживания активности фитнес-клуба Цель: Развитие навыков проектирования программных интерфейсов в рамках каскадной модели жизненного цикла информационной системы. Ход выполнения задания: 1. Анализ требований Провести сбор требований и постановку задачи, основываясь на анализе потребностей клиентов и руководства фитнес-клуба. Фиксировать функциональную и нефункциональную составляющую требований к разрабатываемому API. 2. Проектирование API Использовать методологию проектирования RESTful API, определяя ресурсные маршруты и поддерживаемые HTTP-методы. Описать контракт API, включающий JSON-структуры запросов и ответов, коды ошибок и возможные исключения. 3. Документация Оформить полную документацию API с примерами запросов и примеров данных. Включить разделы по аутентификации, обработке ошибок и поддержке версий API. 4. Кодировка и тестирование Реализовать минимальный MVP (Minimum Viable Product) версии API на одном из

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			современных фреймворков (Django Rest Framework, Flask-RESTful, Spring Boot и т.п.). Написать автотесты для проверки соответствия поведения API спецификации и требованиям. 5. Интеграция Интегрировать разработанный API с фронтенд-приложением фитнес-клуба, обеспечивая надежную передачу данных о посетителях, занятиях и тренировках. 6. Оценка результата Проверить работоспособность API и качество взаимодействий между клиентом и сервером. Оценить полноту покрытия требований и эффективность принятой модели жизненного цикла. Практическое задание №2 Название: Проектирование мобильного приложения с графическим интерфейсом для образовательного учреждения Цель: Овладение навыками проектирования программных интерфейсов мобильных устройств в рамках гибких моделей жизненного цикла (Agile/Scrum). Ход выполнения задания: 1. Постановка задачи Определение ключевых требований к мобильному приложению, предназначенному для учащихся и преподавателей университета. Постановка конкретных задач (просмотр расписания занятий, получение уведомлений, управление учебными материалами). 2. Разработка прототипа интерфейса Использование Sketch/Figma/Miro для создания wireframe и mockup графического интерфейса. Организация сессий Agile-группового мозгового штурма для уточнения деталей интерфейса. 3. Пользовательские истории Работа с командой разработчиков по формулированию User Stories и Backlog'a

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			итераций (sprints). Назначение приоритетов задачам и оценке трудозатрат. 4. Реализация Программирование интерфейса мобильного приложения с использованием React Native или Flutter. Проведение ежедневных Scrum-встреч для контроля прогресса и устранения проблем. 5. Тестирование и улучшение Тщательное тестирование интерфейса с участием конечных пользователей. Сбор обратной связи и проведение ретроспектив для внесения улучшений. 6. Результат Презентация готового продукта команде заказчиков и участникам проекта. Итоговая оценка успешности проекта по критериям достижения заявленных целей и соблюдения сроков.
Тема 5. Прототипы	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Название: Разработка низкоуровневого прототипа информационной системы управления проектами. Цель: Научиться создавать прототип информационной системы, способствующий быстрому пониманию функциональных возможностей программы перед началом полномасштабной разработки. Основные шаги: 1. Анализ предметной области Исследование базовых характеристик проектной деятельности (учёт ресурсов, сроки исполнения, мониторинг состояния задач). Формулирование списка минимальных функциональных требований к системе. 2. Подготовка технического задания Составление чёткого технического задания на разработку прототипа. Выбор подходящей технологической платформы (Python/Django, Ruby on Rails, Vue.js, Angular и т.д.). 3. Создание макета интерфейса Нарисовать low-fi wireframes экранов системы с основными элементами (таблицы задач, графики нагрузки, формы редактирования проектов).

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			Инструментальная поддержка: Figma, Balsamiq Mockups, Adobe XD. 4. Низкоуровневая реализация Кодирование базовой логики приложения с минимальным набором контроллеров и шаблонов представлений. Простое хранение данных в памяти (без полноценной базы данных). 5. Демонстрация и тестирование Демонстрация работоспособности интерфейса и логики основной функциональности. Корректировка недостатков, выявленных в ходе демонстрации. 6. Заключение Подготовка отчета о проделанной работе, содержащего выводы о целесообразности дальнейшего развития прототипа. Практическое задание №2 Название: Высокоуровневый прототип сервиса для онлайн-обучения иностранным языкам. Цель: Развитие навыков высокоуровневой разработки прототипов для быстрой проверки гипотез относительно востребованности будущих продуктов. Основные шаги: 1. Исследование спроса Проведение исследования потенциальной аудитории (школьники, студенты, взрослые пользователи). Сбор информации о текущих продуктах-конкурентах (Duolingo, Lingualeo и т.д.). 2. Формулировка идеи Четкое изложение идеи будущего сервиса, описание уникального торгового предложения (USP). Планирование минимальной жизнеспособной версии продукта (MVP). 3. Высокий уровень детализации прототипа Детальное проектирование экрана главной страницы, профиля пользователя, курсов и уроков. Работа с инструментами высокого уровня детализации (Figma Prototype Mode, InVision Studio).

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			4. Интерфейсное прототипирование Связывание отдельных экранов в единый прототип, симулируя работу реального приложения. Настройка анимаций переходов и взаимодействия с элементами интерфейса. 5. Проведение тестов и фокус-группы Тестирование прототипа среди потенциальных пользователей. Аналитика реакции пользователей на каждый элемент интерфейса и взаимодействия. 6. Отчет о проделанной работе Оформление отчета с выводами и рекомендациями по дальнейшему развитию продукта.
Тема 6. Анализ и моделирование функциональной области внедрения ИС	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Название: Проектирование информационной системы управления логистическими операциями производственной компании. Цель: Научиться проводить функциональный анализ и моделирование конкретной предметной области, выявить базовые процессы и построить их модели. Ход выполнения задания: 1. Изучение предметной области Ознакомьтесь с процессом производства и доставки готовой продукции, исследовав роль поставщиков сырья, производственных цехов, склада готовой продукции и транспортных компаний. 2. Выделение ключевых процессов Определите границы системы и выделите основные процессы, подлежащие автоматизации: закупка материалов, производство, упаковка, доставка. 3. Диаграммы потоков данных (DFD) Постройте DFD первого уровня, отражающего общие потоки данных внутри компании и внешние взаимодействия. Расширьте каждую подсистему до второго уровня детализации, показывая внутренние операции и обмен информацией. 4. Моделирование бизнес-процессов (BPMN) Построить диаграммы BPMN для каждого ключевого процесса (закупка, производство, транспортировка), фиксируя последовательности шагов, принимаемые

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			решения и исполнителей. 5. Анализ полученной модели Проверьте правильность построенных моделей, убедитесь в отсутствии разрывов или противоречий в потоках данных и процессах. Докажите достаточность предложенной структуры для удовлетворения потребностей производственного предприятия. 6. Заключение Подготовьте отчет о проделанной работе, содержащий полученные модели и пояснения к ним. Практическое задание №2 Название: Моделирование информационной системы для клиники медицинского центра диагностики заболеваний. Цель: Овладеть навыком анализа и моделирования сложной функциональной области в сфере медицины, выявляя потребности пациентов и врачей. Ход выполнения задания: 1. Исследования и выявление процессов Проводится обследование текущего состояния клиники: регистрация пациента, запись на прием, назначение анализов, диагностика, лечение, выписка рецепта. 2. Сбор и систематизация требований Формулируются конкретные функциональные и нефункциональные требования к создаваемой информационной системе, такие как ведение электронной медицинской карты, интеграция с лабораториями, уведомление врача о готовности результатов исследований. 3. Функциональный анализ Структурируется деятельность клиника в функциональные блоки, определяются подпроцессы и роли участников. 4. Построение UML-моделей Строятся use case-диаграммы, class-диаграммы и activity-диаграммы для наглядного представления общей картины функционирования клиники. 5. Применение нотации ARIS/EPC

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			Для лучшего понимания внутренних процессов строится event-driven process chain (EPC)-диаграмма, отражающая последовательность и логику происходящих действий. 6. Финальная проверка и представление результатов Готовится отчет с моделью информационной системы, результатами анализа и рекомендациями по дальнейшей реализации системы.
Тема 7. Полная бизнес-модель компании	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Название: Проектирование реляционной базы данных для сервисной компании по обслуживанию компьютерной техники. Цель: Разработать полноценную реляционную модель базы данных, охватывающую всю необходимую информацию о клиентах, услугах, оборудовании и сотрудниках компании. Задания: 1. Анализ предметной области Провести исследование предметной области обслуживания компьютеров, проанализировать рабочие процессы компании и классифицировать основные виды предоставляемых услуг. Выделить ключевые объекты данных: клиенты, оборудование, сотрудники, заказы, услуги. 2. Концептуальное проектирование Спроектировать сущность-отношение (ER) модель данных, показывающую связи между объектами. Выбрать типы отношений (1:N, N:M, 1:1) и задать свойства сущностей. 3. Логическое проектирование Преобразовать ER-модель в реляционную модель данных. Нормализовать модель до 3-й нормальной формы (3NF), устранив повторяющиеся данные и зависимые поля. 4. Физическое проектирование Подобрать физические структуры таблиц, выбрать типы полей, установить первичные и вторичные ключи. Реализовать внешние ключи и ограничить значения, обеспечить консистентность

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			данных с помощью индексов и триггеров. 5. Оптимизация и тестирование Оценить производительность выбранной структуры данных, составить запросы и протестировать быстродействие выборок и модификаций. Сделать вывод о качестве и пригодности предложенной модели для полного охвата деятельности компании. Практическое задание №2 Название: Проектирование OLAP-куба для маркетингового анализа крупного продуктового ритейлера. Цель: Овладеть навыками проектирования многомерных баз данных, используемых для комплексного анализа больших объёмов данных. Задания: 1. Изучение особенностей предметной области Понимание специфики маркетинга продуктовых сетей, выделение важных факторов, влияющих на продажи (категория товара, регион, сезонность, скидки и акции). Идентифицировать измеряемые показатели (продажи, прибыль, остатки на складе, выручка) и измерения (регион, категория товара, временной интервал). 2. Определение мер и измерений Составить перечень важнейших мер (суммарные продажи, средняя цена единицы товара, доход) и измерений (время, товар, магазин, регион). Решить, какие меры являются ключевыми для принятия управленческих решений руководством компании. 3. Проектирование OLAP-куба Разработать куб данных с иерархиями измерений и разрезами для проведения многомерного анализа. Продумать структуру куба с учётом задач оперативного анализа, отчетности и выявления закономерностей. 4. Использование витрины данных (Data Mart) Подготовить витрину данных, агрегированную по важным срезам, облегчающую быстрое извлечение необходимой информации для руководителей.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			Построить ETL-процедуры для регулярной синхронизации данных из оперативной базы данных в витрину. 5. Представление результатов Оформить отчёт о выполненных работах, включить диаграммы OLAP-куба, показать возможности многомерного анализа. Обосновать значимость предложенной структуры данных для повышения конкурентоспособности компании на рынке.
Тема 8. Спецификация функциональных требований к ИС	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Название: Разработка спецификаций функциональных требований для информационно-аналитической системы прогнозирования спроса на товары массового потребления. Цель: Освоить навыки составления спецификаций функциональных требований, грамотно выражающих ожидания пользователей и владельцев системы. Основные этапы: 1. Анализ предметной области Провести исследование и изучить рынок массовых потребительских товаров, выявить факторы, влияющие на спрос (ценообразование, сезонность, рекламные кампании). Сформулировать главную проблему, которую решает система (предсказание изменения спроса для оптимального заказа товаров). 2. Собеседование с заинтересованными сторонами Организовать встречу с менеджерами по снабжению, маркетологами и специалистами по данным. Получить от них информацию о требованиях к точности прогноза, скорости обновления данных и удобстве интерфейса. 3. Классификация требований Разделить требования на функциональные и нефункциональные группы. Функциональные требования включают алгоритм прогнозирования, ввод данных, настройки интерфейса, экспорт отчетов. Нефункциональные требования связаны с производительностью, надежностью, поддержкой разных платформ и масштабируемостью. 4. Спецификация требований

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			Оформить техническое задание в виде документа, содержащее описание функциональных требований и приоритеты их реализации. Пример: Система должна автоматически загружать данные о продажах каждые сутки и выдавать прогнозы спроса на следующую неделю с точностью не менее 90%. 5. Контроль качества спецификации Выполнить проверку правильности и полноты написанных требований, убедиться, что каждая функция имеет четкий смысл и однозначную трактовку. Согласовать готовую спецификацию с заказчиком и всеми участниками проекта. Практическое задание №2 Название: Проектирование технических требований для информационной системы учета энергоэффективности предприятий малого и среднего бизнеса. Цель: Овладеть методом постановки и описания функциональных требований к комплексной автоматизированной системе сбора и анализа данных по энергопотреблению. Этапы выполнения: 1. Постановка проблемы Изучить сферу энергосбережения, выявить проблемы малых и средних предприятий, касающиеся неэффективного расходования энергии. Поставить задачу: создать инструмент для постоянного мониторинга энергопотребления, позволяющий оперативно снижать затраты. 2. Интервью с пользователями Провести встречи с инженерами и экономистами предприятий, выяснить, какая информация необходима для эффективной работы с данными. Уточнить требуемый уровень детализации (час, день, месяц), типы оборудования, чьё потребление важно учитывать. 3. Структура функциональных требований Классифицировать требования на группы: учет показаний приборов, обработка аварийных ситуаций, автоматизация вычислений экономии. Например: Система должна обеспечивать возможность автоматического подключения новых датчиков без перезагрузки основного ПО.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			4. Оформление требований Создать спецификацию требований, оформленную в табличной форме с указанием типа требования, важности, срока реализации и исполнителя. Утвердить спецификацию совместно с ответственными лицами предприятия-заказчика. Просчитать экономический эффект от внедрения предложенных решений.
Тема 9. Структурные и объектные модели предметной области	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Проектирование информационной системы с использованием объектных моделей предметной области. Цель задания: Разработать объектную модель предметной области автоматизированной информационной системы управления персоналом предприятия, включая описание классов объектов, атрибутов и связей между ними. Задания: 1. Опишите основные объекты, процессы и отношения, характерные для системы управления персоналом организации. 2. Разработайте классы и укажите их атрибуты, соответствующие следующим сущностям: Сотрудник Подразделение Должность Компетенция Контракт Навык Отчет сотрудника 3. Установите связи между классами согласно отношениям между объектами в реальной жизни. Определите типы ассоциаций (например, многие-к-многим, один-к-одному). 4. Создать диаграмму классов с указанием всех классов, атрибутов и связей. Используйте нотацию UML. 5. Написать пример реализации одного из классов (например, "Сотрудник") на

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			выбранном вами языке программирования (Python, JavaScript, C#). Пример должен включать конструктор, свойства и метод отображения сведений о сотруднике. Практическое задание №2 Тема: Разработка структурной модели базы данных для предметной области "Управление проектами" Цель задания: Создать структуру реляционной базы данных для автоматизации процессов управления проектами в проектной организации. Задания: 1. Анализ требований и определение сущности. Проведите анализ бизнес-процессов и определите ключевые сущности и их атрибуты для хранения информации о проектах, участниках проектов, ресурсах проекта, сроках исполнения этапов и статусах выполнения работ. 2. Создание ER-диаграммы. Нарисуйте сущность-отношение (ER) диаграмму с описанием сущностей, ключевых полей, взаимосвязей и типов данных каждого атрибута. 3. Нормализация схемы базы данных. Привести схему базы данных к третьей нормальной форме (3NF). 4. SQL-запросы. Составьте SQL-запросы для выполнения типичных операций над базой данных, таких как создание таблиц, вставка записей, выборка данных по критериям и обновление статусов проектных задач.
Тема 10. Унифицированный процесс разработки	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных,	Практическое задание №1 Тема: Применение унифицированного процесса разработки (Unified Process) для проектирования корпоративной информационной системы управления запасами товаров. Цель задания: Применяя методологию Rational Unified Process (RUP), разработать требования и начальные этапы жизненного цикла проекта по созданию корпоративной

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
		программных интерфейсов	информационной системы управления складскими остатками продукции. Задания: 1. Идентификация заинтересованных сторон и целей бизнеса. Определите основных пользователей будущей системы и сформулируйте цели проекта с точки зрения повышения эффективности логистического подразделения. 2. Сбор и анализ требований. Используя технику интервьюирования и анкетирования, соберите функциональные и нефункциональные требования будущих пользователей. Заполните шаблон Use Case для основного сценария взаимодействия пользователей с системой ("создание заказа"). 4. Моделирование вариантов использования (Use Cases). Постройте диаграмму вариантов использования (use case diagram) для описания функциональности системы, включив сценарии обработки заказов, инвентаризации запасов и выдачи товара клиентам. 5. Архитектура решения. Разработайте архитектурную концепцию системы, используя инструмент анализа рисков (Risk Management Framework) и шаблоны проектирования (Design Patterns). Выберите подходящую технологию и инструменты для разработки веб-приложения. 6. Документирование результатов этапа. Подготовьте отчет по итогам проделанной работы, содержащий краткое описание всех выполненных шагов и полученных артефактов (документация по требованиям, use cases, архитектура). Практическое задание №2 Тема: Реализация проекта автоматизации учета рабочего времени сотрудников с применением подходов унифицированного процесса. Цель задания: Используя методологии RUP и Scrum, спроектировать систему учета рабочего времени сотрудников компании с интеграцией в существующую ERP-систему предприятия. Задания: 1. Планирование итерационного подхода. Спланируйте проект, разделив его на

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			несколько коротких итераций (sprints). Определите сроки каждой итерации и поставленные цели. 2. Исследование требований и разработка бэклога продукта. Совместно с заказчиком разработайте перечень функциональных и нефункциональных требований. Оформите их в виде элементов бэклога (Product Backlog Items). 3. Формулировка user stories. Формализуйте конкретные потребности пользователей системы в виде историй пользователей (User Stories), каждая из которых должна отвечать на вопросы: кто заказчик, какую функциональность ожидает увидеть и зачем эта функциональность необходима? 4. Выбор инструментов и технологий. Оцените доступные технологии и выберите наиболее подходящие инструменты для реализации проекта (например, фреймворки Python/Django, Node.js + MongoDB). 5. Итерация разработки. Выполните первую итерацию, реализовав одну или две истории пользователей. Протестируйте полученные результаты и получите обратную связь от заказчика. 6. Отчёт по результатам первой итерации. Докажите успешность выбранного подхода и подтвердите правильность решений по развитию проекта, представив отчёт с результатами проведённой итерации.
Тема 11. Унифицированный язык моделирование (UML)	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Проектирование базы данных информационной системы учёта заявок клиентов туристической фирмы с использованием языка UML. Цель задания: Научиться разрабатывать концептуальную модель данных средствами UML и преобразовывать её в физическую модель данных для последующей реализации базы данных. Задания: 1. Анализ предметной области. Описать основные понятия и отношения предметной области «Туризм»: клиент, заявка, тур, отель, экскурсия, рейс и др., определить главные характеристики каждого объекта. 2. Создание ER-модели. Разработать ERD (Entity Relationship Diagram) — диаграмму сущностей и отношений для заданной предметной области, обозначив сущности,

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			атрибуты и связи между ними. 3. Трансформация в модель UML. Преобразовать ER-модель в UML-классовые диаграммы (Class Diagrams): представить каждую сущность как класс, выделить ассоциации, наследования и композиции между классами. 4. Нормализация и оптимизация структуры БД. Нормализовать структуру данных до 3-й нормальной формы (3NF). 5. Реализация физической модели. Представить физические таблицы для СУБД MySQL на основании подготовленной модели UML-классов, составить SQL-код для создания таблиц и первичного наполнения тестовыми данными. Практическое задание №2 Тема: Использование UML для проектирования информационной системы учета продаж недвижимости. Цель задания: Освоение методов построения UML-диаграмм и способов трансформации UML-моделей в реляционные структуры данных. Задания: 1. Постановка задачи и сбор требований. Описать процессы и объекты предметной области: продавец, покупатель, недвижимость, сделка купли-продажи, договор аренды и т.п.; выявить необходимые операции и запросы к данным. 2. Классовые диаграммы UML. Разработать классовые диаграммы (Class Diagrams) для отражения структуры данных системы учета сделок с недвижимостью. Включить необходимые классы, атрибуты и отношения между ними. 3. Последовательностные диаграммы (Sequence Diagrams). Создать последовательностные диаграммы для базовых сценариев взаимодействия пользователей с системой («оформление сделки», «поиск недвижимости»). 4. Переход к физическим структурам данных. Перевести UML-классы в таблицу реляционной базы данных (MySQL, PostgreSQL), описать поля таблиц, ключи и внешние связи. 5. Генерация SQL-скриптов. Подготовить SQL-команды для создания базы данных и представления основной структуры таблиц на основе полученной UML-модели.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
Тема 12. Определение требований в рамках унифицированного процесса	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Проектирование API-интерфейса сервиса аналитики Big Data с использованием RESTful архитектуры. Цель задания: Получить опыт формулирования требований и проектирования API-сервисов в рамках унифицированного процесса (Rational Unified Process, RUP). Задания: 1. Определите целевую аудиторию и конечных пользователей API. Кто будут основными пользователями вашего API и каковы их потребности? 2. Опишите требуемые возможности API: Какие данные и операции необходимы пользователям (получение статистики по метрикам, фильтрация данных, агрегированные отчёты)? 3. Разработайте спецификацию RESTful API, учитывая лучшие практики проектирования API (REST-конвенции именования ресурсов, HTTP-методы CRUD, форматы передачи данных JSON/XML). 4. Спроектируйте API-ресурсы и маршруты (endpoints) для сервисов аналитики данных. Продемонстрируйте полное покрытие функций, указанных в пункте 2. 5. Оформите документацию API. Предоставьте описание формата запросов и ответов, примерами данных и возможных ошибок. 6. Проверьте согласованность API с ключевыми требованиями (надежность, производительность, безопасность, масштабируемость). Практическое задание №2 Тема: Проектирование GUI-интерфейса интерактивной панели визуализации данных (Dashboard) в рамках унифицированного процесса. Цель задания: Практиковать навыки проектирования графических интерфейсов приложений в процессе сбора и анализа требований. Задания:

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			1. Собрать требования к интерфейсной части приложения. Провести опрос целевой аудитории и зафиксировать ожидания относительно удобства пользования панелью Dashboard. 2. Проанализировать конкурентные продукты. Найти аналогичные интерфейсы и оценить их сильные стороны и недостатки. 3. Создать wireframes (макеты интерфейса). Использовать программы прототипирования (Figma, Adobe XD) для рисования схем расположения элементов интерфейса и кнопок навигации. 4. Предложить дизайн визуальных компонентов: палитру цветов, шрифты, иконки и элементы управления (таблицы, графики, фильтры). 5. Продемонстрировать взаимодействие с интерфейсом. Предусмотреть анимацию переходов между экранами и подсказки для улучшения юзабилити.
Тема 13. Анализ структуры в рамках унифицированного процесса	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Анализ и проектирование структуры информационной системы учебного заведения с использованием метода UML (Unified Modeling Language) Цель задания: Изучить организацию структуры учебной среды вуза и спроектировать базу данных для эффективного управления образовательным процессом, а также создать программу на языке Python для заполнения учебных групп студентами. Порядок выполнения: 1. Проведите анализ структуры учебного заведения: факультеты, кафедры, преподаватели, учебные группы, дисциплины, расписание занятий, успеваемость учащихся. 2. Разработайте общую схему информационных потоков внутри образовательной среды и определите ключевые элементы данных. 3. Применяйте унифицированный язык моделирования (UML) для составления общей модели предметной области и диаграммы классов (class diagrams). 4. Сформулируйте требования к структуре базы данных, соответствующей предметной области учебного заведения. 5. Рассчитайте объем памяти, необходимой для хранения всей совокупности данных,

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			оценивая среднее количество преподавателей, студентов, предметов и экзаменов. 6. Осуществите проектирование структуры базы данных (создавая таблицы и определяя связи между ними). 7. Создайте простой скрипт на языке Python, позволяющий добавлять новых студентов в учебный класс, получать список текущих учеников, распределять студентов по учебным дисциплинам и формировать расписание занятий. 8. По завершении напишите отчет, поясняющий ваши выводы и принятые проектные решения. Практическое задание №2 Тема: Проекты базы данных медицинской клиники с использованием унифицированных методов проектирования (UML) Цель задания: Исследовать структуру медицинского учреждения и разработать единую информационно-аналитическую систему управления медицинскими услугами с возможностью автоматического распределения пациентов по врачам и назначения лечения. Порядок выполнения: 1. Изучите организационную структуру медицинского учреждения: врачи, пациенты, отделения, процедуры диагностики и терапии, назначение лекарственных препаратов. 2. Проведите SWOT-анализ текущего состояния информационных систем в клинике и выявите проблемы и перспективы развития. 3. Создайте модель предметной области с помощью UML-диаграмм (включая диаграммы классов, диаграммы последовательности действий, компонентные диаграммы). 4. Разработайте техническое задание на разработку единой информационной системы управления пациентами и медицинским обслуживанием. 5. Определите структуру реляционной базы данных для регистрации пациентов, врачей, назначаемых процедур и медикаментов. 6. Сделайте расчет объема данных, хранящихся в каждом элементе структуры базы данных, исходя из средней численности персонала и количества приемов ежедневно.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			7. Разработайте простой web-интерфейс для администраторов клиники, обеспечивающий регистрацию нового пациента, ведение медицинских карт и автоматический подбор врача на основе загруженности специалиста. 8. Проверьте работу системы путем тестирования отдельных модулей и оценки производительности.
Тема 14. Проектирование в рамках унифицированного процесса	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Проектирование информационной системы CRM (Customer Relationship Management) с использованием принципов унифицированного процесса (RUP). Цель задания: Применение методики RUP для проектирования системы управления взаимоотношениями с клиентами для малого бизнеса. Порядок выполнения: 1. Проведите анализ потребностей клиента в управлении взаимодействием с потребителями (записи звонков, регистрация обращений, управление заказами, отслеживание лояльности клиентов). 2. Разработайте диаграмму вариантов использования (Use Case Diagram) и перечислите основные функции и сценарии поведения системы. 3. Создайте UML-диаграмму классов (Class Diagram), выделив основные сущности и отношения между ними. 4. Сформируйте модель состояний системы (State Machine Diagram), описывая поведение системы в зависимости от стадий взаимодействия с клиентом. 5. Определите требования к интерфейсу пользователя и проекту архитектуры, включая модульную структуру и механизмы интеграции с внешним оборудованием (телефон, e-mail, мессенджеры). 6. Реализуйте прототип простейшего UI-интерфейса для ввода и вывода данных (на примере любого инструмента проектирования интерфейсов). 7. Настройте инфраструктуру развертывания приложения, спланировав шаги установки сервера и базы данных. 8. Проверьте корректность выполненного проектирования и напишите итоговый отчет, содержащий выводы и рекомендации по дальнейшей разработке.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			Практическое задание №2 Тема: Проектирование и реализация онлайн-магазина электронной коммерции в рамках унифицированного процесса. Цель задания: Изучение и применение методик RUP для проектирования инфраструктуры электронного магазина с управлением товарами, корзиной покупок и платежами. Порядок выполнения: 1. Определите круг заинтересованных лиц (клиенты, продавцы, администрация сайта) и изучите их нужды и пожелания относительно функционала магазина. 2. Постройте Use Case Диаграмму, подробно раскрывающую сценарий покупки товара, добавления в корзину, оформления заказа и оплаты. 3. Разработайте Class Diagram для идентификации классов и их взаимоотношений, отражающих сущность предметной области. 4. Разработайте Activity Diagram, показывающий последовательность действий при покупке товара и обработке заказа. 5. Выявите требования к архитектуре и инфраструктуре, включая серверное оборудование, программное обеспечение и хранилище данных. 6. Создайте макет страницы интерфейса пользователя, используя инструменты прототипирования дизайна (например, Figma, Sketch). 7. Определите базовые модули приложения и способы их взаимодействия, включая безопасность и обработку транзакций. 8. Проверить выполнение требований, провести тестирование интерфейса и архитектуры, написать подробный отчет о результатах исследования и проектирования.
Тема 15. Документирование процесса	ПК-6 Способен применять методы и средства проектирования компьютерного программного	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур	Практическое задание №1 Тема: Документирование процесса проектирования базы данных для информационной системы учета товарных остатков склада торговой сети. Цель задания: Закрепить навыки документирования процесса проектирования и разработки базы данных с целью оптимизации учета товарных остатков на складе крупной розничной

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
	обеспечения	данных, баз данных, программных интерфейсов	сети. Задания: 1. Исследуйте структуру предметной области: проанализируйте поток данных, происходящих в учетной системе торговых точек, особенности работы склада и роль базы данных в этом процессе. 2. Разработайте логическую модель данных с использованием диаграмм Entity-Relationship (ERD), описывающих сущности, атрибуты и связи между ними. 3. Произведите нормализацию структуры данных до 3-й нормальной формы (3NF). 4. Разработайте физический проект базы данных, детализируя представление таблиц, индексов, ключей и ограничений целостности. 5. Создайте документацию, содержащую описание процесса проектирования, требования к базе данных, используемым инструментам и технологиям, логическую и физическую модели данных, инструкции по установке и эксплуатации. 6. Охарактеризуйте риски и угрозы для устойчивости и безопасности базы данных, разработайте меры по их минимизации. 7. Организуйте контроль качества документа: обеспечьте наличие рецензии коллег и руководителя проекта, внесите исправления, обновите версию документа. Практическое задание №2 Тема: Документирование процесса проектирования базы данных службы такси для крупного города. Цель задания: Приобрести навыки полного документирования процесса проектирования базы данных на примере конкретной информационной системы — служб такси крупного российского города. Задания: 1. Подробно исследуйте специфику функционирования диспетчерской службы такси: изучение маршрутов, пассажиров, водителей, автомобилей, тарифов и платежей. 2. Разработайте концептуальную модель данных, использовав подход "сущность—атрибут—связь" (ER), детально отразив все значимые компоненты системы.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			3. Приведите структуру данных к третьей нормальной форме (3NF). 4. Представьте физическое проектирование базы данных, описав структуру таблиц, индексы, первичные и внешние ключи, правила контроля целостности данных. 5. Оформите техническую документацию, охватывающую вводную часть, требования к системе, модели данных, порядок инсталляции и настройки базы данных, инструкцию администратора и оператора. 6. Оцените потенциальные риски и уязвимости системы, предусмотрите мероприятия по обеспечению надежности и безопасности данных. 7. Организуете процедуру проверки и утверждения документа: ознакомьтесь с комментариями коллеги, скорректируйте текст, зафиксируйте новую версию документа.
Тема 16. Организация и планирование работ	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Планирование и организация работ по проектированию мобильного приложения для анализа спортивных показателей спортсменов. Цель задания: Организовать и спрогнозировать деятельность команды разработчиков, занимающихся созданием мобильного приложения для мониторинга и анализа спортивной активности пользователей. Последовательность выполнения: 1. Проведите исследование рынка аналогичных продуктов и определите требования к мобильному приложению. 2. Составьте план-график работ по этапам разработки приложения, распределяя роли среди членов команды (аналитик, дизайнер, фронтенд-разработчик, бекенд-разработчик, тестировщик). 3. Распределите обязанности и временные рамки выполнения каждого этапа, создав детализированный сетевой график работ. 4. Разработайте бюджет проекта, учитывающий затраты на персонал, закупку оборудования и лицензий, хостинг и продвижение приложения. 6. Зафиксируйте риски, влияющие на успех проекта, и разработайте стратегию их минимизации. 6. Сформируйте финальный отчет, включающий планы-графики, распределение

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			обязанностей, оценку бюджета и описание стратегии риск-менеджмента. Практическое задание №2 Тема: Организация и планирование работ по созданию веб-портала по изучению иностранного языка. Цель задания: Изучить принципы организации труда и планирования проекта по созданию образовательного портала, предназначенного для изучения английского языка. Последовательность выполнения: 1. Проанализируйте рынок образовательных платформ и создайте техническое задание на разработку веб-сайта, включающее функциональность, интерфейс и требования к дизайну. 2. Разработайте сетевой график проекта, включающий подготовку технических документов, проектирование архитектуры, написание кода, тестирование и внедрение. 3. Назначьте исполнителей и руководителей команд, установите сроки выполнения задач и ожидаемые результаты. 4. Определите стоимость проекта, принимая во внимание оплату услуг подрядчиков, покупку лицензии на использование CMS-платформы и рекламные расходы. 5. Прогнозируйте вероятные препятствия и трудности, возникающих в ходе реализации проекта, разработайте контрмеры и планы резервных действий. 6. Подготовьте полный пакет документации, включающей план-график, бюджет, должностные инструкции и регламент действий при возникновении форс-мажорных обстоятельств.
Тема 17. Оценка стоимости разработки программного продукта	ПК-6 Способен применять методы и средства проектирования компьютерного программного	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур	Практическое задание №1 Тема: Расчет стоимости разработки корпоративной информационной системы управления финансами. Цель задания: Оценить предварительную стоимость разработки специализированной

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
	обеспечения	данных, баз данных, программных интерфейсов	информационной системы для бухгалтерского учета и финансового менеджмента небольшой коммерческой организации. Порядок выполнения: 1. Проведите анализ требований и особенностей предметной области финансовой отчетности. 2. Определите объемы предстоящих работ по каждому этапу жизненного цикла разработки программного обеспечения (проектирование, кодирование, тестирование, внедрение, сопровождение). 3. Составьте калькуляцию затрат, учитывая трудозатраты сотрудников разных категорий (аналитики, разработчики, дизайнеры, тестировщики), необходимое аппаратное и программное обеспечение, аренду помещений и прочие накладные расходы. 4. Примените один из методов стоимостной оценки разработки (метод COCOMO II, метод аналогий, Delphi-метод и т.д.) и рассчитайте приблизительные затраты на весь цикл разработки. 5. Выделите факторы риска, способные повлиять на увеличение затрат (непредвиденная сложность проекта, задержки в сдаче промежуточных этапов, изменение требований заказчиков и т.д.). 6. Разработайте сводную таблицу затрат и сделайте вывод о целесообразности реализации проекта. Практическое задание №2 Тема: Оценка стоимости разработки веб-сервиса для анализа больших данных (Big Data). Цель задания: Рассчитать ориентировочную стоимость создания веб-сервиса для предварительной обработки и анализа больших массивов данных. Порядок выполнения: 1. Исследование существующих аналогов на рынке, выявление основных характеристик и возможностей подобных решений.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			2. Сбор требований к будущему сервису: доступность большого объёма хранимых данных, высокая скорость обработки запросов, поддержка масштабируемых вычислений, интеграция с облачными системами хранения. 3. Детальное разделение работ на стадии жизненного цикла разработки ПО: анализ требований, проектирование, разработка, тестирование, релиз и обслуживание. 4. Выбор подходящей методологии оценки стоимости (COCOMO II, Function Point Analysis, Parametric Cost Estimation и т.д.). Расчет трудозатрат и материальных затрат по каждому этапу разработки. 5. Анализ факторов риска и неопределенностей, которые могут увеличить общие затраты на проект. 6. Создание таблицы расчета общей стоимости проекта, оценка выгод и рисков запуска такого сервиса.
Тема 18. Инструментальные средства разработки программ	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Проектирование и разработка простого клиент-серверного приложения с использованием современных инструментов разработки. Цель задания: Освоить инструменты и техники разработки простых клиент-серверных приложений, используя языки программирования высокого уровня и специализированные библиотеки. Порядок выполнения: 1. Выберите подходящий стек технологий (например, Python+Flask+Django или NodeJS+Express+MongoDB) и определите среду разработки (IDE, редакторы кода, утилиты компиляции и сборки). 2. Поставьте цель разработки — создание простого клиент-серверного приложения для демонстрации списка книг с возможностями просмотра, редактирования и удаления записей. 3. Проектируйте приложение, следуя паттернам MVC/MVVM, используя встроенные инструменты выбранного языка программирования и среды разработки. 4. Реализуйте механизм аутентификации и авторизации пользователей, используя стандартные библиотеки или плагины выбранного стека технологий. 5. Интегрируйте в проект автоматизированные тесты, используя популярные

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Задания к практическому занятию
			фреймворки тестирования (pytest, Jest, Mocha и др.). 6. Разработайте простой пользовательский интерфейс с минимальным дизайном и проверенными элементами доступности. 7. Завершите проект демонстрацией работоспособности приложения и представлением всех используемых инструментов и библиотек. Практическое задание №2 Тема: Автоматизация процессов тестирования и деплоя с использованием DevOps-инструментов. Цель задания: Познакомиться с инструментами непрерывной интеграции и доставки (CI/CD), применяемыми для ускорения и упрощения разработки программного обеспечения. Порядок выполнения: 1. Проведите анализ и сравнительный обзор популярных инструментов CI/CD (GitLab CI, Jenkins, Travis CI, CircleCI и др.) 2. Создайте простое веб-приложение на языке PHP/Laravel или Ruby on Rails с простым набором страниц и базовой функциональностью. 3. Настройте репозиторий проекта на GitHub/GitLab, настроив триггеры для автоматической сборки и тестирования проекта при коммите изменений. 4. Подготовьте Docker-образ приложения и настройте доставку образа в контейнеры Kubernetes или Docker Swarm. 5. Добавьте возможность автоматически запускать тесты (unit-тесты, end-to-end тесты) при сборке и отправке изменений. 6. Реализуйте систему уведомлений и оповещения о статусе сборки и тестов. 7. После завершения разработки продемонстрируйте работу конвейера сборки и доставки, а также протестированное приложение.

Критерии оценивания практических занятий:

Оценка	Критерии оценивания
отлично	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи, формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, обосновывает свои суждения и даёт правильные ответы на вопросы преподавателя
хорошо	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи и формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, но содержание ответов имеют некоторые неточности и требуют уточнения и комментария со стороны преподавателя
удовлетворительно	Выставляется, если обучающийся знает и понимает материал по заданной теме, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, студент не может обосновать свои ответы на уточняющие вопросы преподавателя
неудовлетворительно	Выставляется, если обучающийся допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал. Делает ошибки в ответах на уточняющие вопросы преподавателя

4.3. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ КОНТРОЛЬНЫХ РАБОТ

Контрольные работы содержат несколько практических заданий по индивидуальным вариантам, в полном объеме охватывающих изученный материал по указанной теме. Выполнение контрольных работ позволяет определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций.

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Типовой вариант контрольной работы</th></tr>
<tr><td>Тема 7. Полная бизнес-модель компании</td><td>ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения</td><td>ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов
ИПК-6.3 Владеть навыками проектирования баз данных</td><td>Задание 1 Установите соответствие между функциональными областями полной бизнес-модели ритейлера и типом таблицы в хранилище данных, предназначенной для анализа эффективности этой области:
<table>
<tr><th>Функциональная область</th><th>Тип таблицы в хранилище данных</th></tr>
<tr><td>1. Продажи (транзакции, чеки, возвраты)</td><td>А. Измерение с медленно меняющимися атрибутами (SCD Type 2)</td></tr>
<tr><td>2. Клиенты (сегменты, лояльность, демография)</td><td>Б. Фактовая таблица с аддитивными метриками</td></tr>
<tr><td>3. Товары (категории, поставщики, ассортимент)</td><td>В. Измерение с иерархией (рекурсивная связь)</td></tr>
<tr><td>4. Маркетинговые кампании (бюджет, каналы)</td><td>Г. Фактовая таблица с полуаддитивными метриками</td></tr>
</table>
Правильное соответствие:
1 → Б
2 → А
3 → В
4 → Г

Задание 2. При проектировании витрины данных для анализа полной бизнес-модели производственной компании выделены бизнес-объекты: «Заказ», «Партия», «Оборудование», «Сотрудник». Какие из перечисленных решений по проектированию структуры таблиц корректно отражают бизнес-логику полной модели и обеспечивают поддержку кросс-функционального анализа?
Варианты ответов (выберите 2 правильных):
А. Создание единой фактовой таблицы fact_production с гранулярностью «партия» и метриками: количество брака, время цикла, загрузка оборудования.</td></tr>
</table>

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			Б. Выделение измерения dim_equipment с атрибутами: тип оборудования, цех, дата ввода в эксплуатацию, текущий статус (в работе/ремонт). В. Объединение атрибутов заказа и партии в одну таблицу для упрощения запросов аналитиков. Г. Создание отдельной фактовой таблицы fact_order_to_cash с гранулярностью «заказ» и метриками: срок исполнения, просрочка, себестоимость, маржинальность. Д. Хранение списка сотрудников, участвовавших в производстве партии, в текстовом поле workers_list таблицы fact_production. Правильные ответы: А, Г Эталон обоснования: Варианты А и Г создают две взаимосвязанные фактовые таблицы с разной гранулярностью (партия vs заказ), что позволяет анализировать цепочку «производство → финансы» через общие измерения (dim_order, dim_date). Вариант В нарушает нормализацию и затрудняет анализ по отдельным сущностям; вариант Д уничтожает структуру данных, делая невозможным анализ трудозатрат по сотрудникам. Задание 3. Расположите этапы трансформации полной бизнес-модели компании в структуру хранилища данных для аналитики в правильной последовательности: А. Денормализация нормализованной модели в схему «звезда» с выделением измерений и фактов Б. Построение концептуальной модели «сущность-связь» на основе бизнес-процессов и объектов полной модели В. Преобразование ER-модели в нормализованную логическую схему (3НФ) для обеспечения целостности Г. Определение гранулярности фактовых таблиц и аддитивности метрик для каждой функциональной области Д. Анализ бизнес-требований: выделение функциональных областей (продажи, логистика, финансы) и ключевых метрик Правильная последовательность: Д → Б → В → А → Г Задание 4. В полной бизнес-модели банка выделено измерение «Клиент» со следующими атрибутами:

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			client_id, full_name, segment («новый», «активный», «лояльный», «дормант»), risk_score, assigned_manager_id. Бизнес-правило: сегмент клиента пересчитывается ежемесячно по результатам активности, но для анализа удержания необходимо знать, к какому сегменту клиент относился на момент совершения транзакции. Задание: Опишите структуру таблицы измерения dim_client с учётом бизнес-правила. Укажите: а) необходимые дополнительные атрибуты; б) тип медленно меняющегося измерения (SCD); в) как будет выглядеть связь с фактовой таблицей транзакций. Эталон ответа: а) Дополнительные атрибуты: valid_from (дата начала действия версии), valid_to (дата окончания действия), is_current (признак актуальной версии). б) Тип SCD: Type 2 (создание новой версии записи при изменении сегмента). в) Связь с фактом: в таблице fact_transaction хранится client_key (surrogate key), который ссылается на конкретную версию клиента в момент транзакции, а не на client_id (business key). Задание 5. Сценарий: Для полной бизнес-модели онлайн-ритейла требуется спроектировать витрину данных «Анализ клиентской ценности», объединяющую данные из функциональных областей: Маркетинг: взаимодействия с кампаниями, стоимость привлечения (CAC) Продажи: транзакции, выручка, возвраты Сервис: обращения в поддержку, время решения Логистика: сроки доставки, количество возвратов Ключевые бизнес-требования: Расчёт пожизненной ценности клиента (LTV) с разбивкой по каналу привлечения Анализ корреляции между качеством сервиса и повторными покупками Оценка влияния сроков доставки на уровень возвратов Задание: Предложите архитектуру витрины данных. В ответе (200–250 слов) укажите:

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Типовой вариант контрольной работы</th></tr>
<tr><td></td><td></td><td></td><td>а) структуру фактовых таблиц (гранулярность, ключевые метрики);
б) состав измерений;
в) как спроектированная структура поддерживает расчёт метрики LTV.
Эталон ответа:
Витрина включает две фактовые таблицы. fact_customer_acquisition с гранулярностью «первое взаимодействие клиента с каналом» содержит метрики: cac_amount, acquisition_date_key, channel_key. fact_customer_activity с гранулярностью «событие в жизненном цикле клиента» (покупка, обращение, доставка) содержит метрики: revenue, support_tickets_count, delivery_days, is_return, event_date_key. Обе таблицы связаны с измерениями: dim_customer (SCD Type 2 для сегментов), dim_date, dim_channel, dim_product_category, dim_support_reason. Для расчёта LTV используется агрегация revenue из fact_customer_activity по customer_key за весь период минус cac_amount из fact_customer_acquisition. Связь «один клиент — много событий» позволяет строить когорты по дате привлечения (из fact_customer_acquisition) и анализировать их поведение во времени через fact_customer_activity. Корреляция сервиса и покупок достигается через общее измерение dim_customer и фильтрацию событий по типу (покупка vs обращение) в рамках одного временного окна.</td></tr>
<tr><td>Тема 13. Анализ структуры в рамках унифицированного процесса</td><td>ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения</td><td>ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов
ИПК-6.3 Владеть навыками проектирования баз данных</td><td>Задание 1. Установите соответствие между фазой Унифицированного процесса и основным артефактом анализа структуры базы данных для аналитической системы:
<table>
<tr><th>Фаза Унифицированного процесса</th><th>Артефакт анализа структуры</th></tr>
<tr><td>1. Инициация (Inception)</td><td>А. Детальная спецификация гранулярности фактовых таблиц и правил обработки медленно меняющихся измерений (SCD)</td></tr>
<tr><td>2. Уточнение (Elaboration)</td><td>Б. Концептуальная модель «сущность-связь» с выделением ключевых бизнес-объектов и их атрибутов</td></tr>
<tr><td>3. Конструирование (Construction)</td><td>В. Матрица «измерения-метрики» с указанием аддитивности каждой метрики по каждому измерению</td></tr>
<tr><td>4. Внедрение (Transition)</td><td>Г. Диаграмма развёртывания с распределением</td></tr>
</table></td></tr>
</table>

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			<table><tr><td></td><td>таблиц по физическим узлам (мастер, реплики, шарды)</td></tr></table> Правильное соответствие: 1 → Д 2 → Б 3 → А 4 → Г Задание 2. При анализе структуры предметной области «Управление запасами» в фазе уточнения Унифицированного процесса выявлены бизнес-требования: • Отслеживание движения товара по складам (приход, перемещение, списание) • Анализ оборачиваемости по категориям и поставщикам • Учёт истории изменения стоимости закупки при переоценке Какие из перечисленных решений по структурному анализу являются архитектурно значимыми и должны быть зафиксированы до начала фазы конструирования? Варианты ответов (выберите 3 правильных): А. Выбор СУБД (PostgreSQL vs ClickHouse) на основе прогнозируемого объёма данных. Б. Определение гранулярности фактовой таблицы как «движение товара» (приход/расход одной партии в один момент времени). В. Решение о применении паттерна SCD Type 2 для измерения «Товар» с атрибутом «стоимость закупки». Г. Проектирование индекса по полю transaction_date в фактовой таблице. Д. Выделение измерения «Склад» с иерархией (регион → город → склад). Е. Написание SQL-запроса для расчёта оборачиваемости по формуле «продажи / средний остаток». Правильные ответы: Б, В, Д Эталон обоснования: Решения Б, В, Д определяют фундаментальную структуру хранилища (гранулярность фактов, стратегия хранения истории, иерархия измерений) — их изменение на поздних этапах потребует полной перестройки схемы и пересчёта исторических данных. Выбор СУБД (А) и индексы (Г) — технические решения фазы

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			конструирования; расчёт метрик (E) — логика приложения, не влияющая на структуру БД. Задание 3. Расположите действия при анализе структуры источников данных для построения хранилища данных в правильной последовательности в рамках итерации фазы уточнения: А. Фиксация ограничений источников: отсутствие истории изменений сегмента клиента в операционной БД Б. Сопоставление атрибутов источника с бизнес-метриками из требований (например, «payment_amount» → «выручка») В. Анализ структуры операционной БД: изучение схемы таблиц orders, customers, payments Г. Определение стратегии компенсации ограничений: реализация логики расчёта сегмента в процессе ETL Д. Выявление расхождений семантики: в источнике «выручка» включает НДС, в требованиях — без НДС Правильная последовательность: В → Б → Д → А → Г Задание 4. В ходе анализа структуры для витрины данных «Анализ эффективности колл-центра» в фазе уточнения выявлено бизнес-правило: «Оператор может одновременно работать с несколькими обращениями клиентов, но каждое обращение в каждый момент времени обслуживается только одним оператором». Задание: Определите тип связи между сущностями «Оператор» и «Обращение» в концептуальной модели. Укажите, как эта связь будет реализована в реляционной структуре фактовой таблицы. Эталон ответа: Тип связи в концептуальной модели: «многие-ко-многим» (M:N) с атрибутами периода действия (дата/время начала и окончания обслуживания). Обоснование: связь меняется во времени (оператор может смениться), и в каждый момент времени соблюдается ограничение «один оператор на обращение», но исторически одно

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			обращение может быть связано с несколькими операторами, а один оператор — с множеством обращений. В реляционной структуре связь реализуется через фактовую таблицу fact_interaction с гранулярностью «факт передачи обращения» или «период обслуживания», содержащую surrogate-ключи оператора и обращения, а также временные метки start_datetime, end_datetime. Задание 5 В фазе уточнения Унифицированного процесса проводится анализ структуры для создания витрины данных «Клиентская аналитика» в банке. Предметная область включает: Операционные данные: транзакции, продукты (кредиты, вклады, карты), офисы Требования аналитиков: анализ кросс-продаж (покупка нескольких продуктов одним клиентом), расчёт комплексной доходности клиента (сумма маржинальности по всем продуктам), отслеживание миграции клиента между сегментами Ключевая сложность: в операционной системе продукты хранятся раздельно (таблицы loans, deposits, cards), а для анализа требуется единая точка зрения на «портфель клиента». Задание: Опишите подход к анализу структуры в рамках дисциплины «Анализ и проектирование» Унифицированного процесса. В ответе (200–250 слов) укажите: а) какие артефакты анализа структуры необходимо создать на фазе уточнения; б) как будет решена проблема интеграции разнородных источников в единую структуру; в) почему предложенный подход минимизирует архитектурные риски при переходе к фазе конструирования. Эталон ответа: На фазе уточнения создаются три ключевых артефакта анализа структуры: 1) концептуальная модель с бизнес-объектом «Клиентский портфель» как агрегирующей сущностью над продуктами; 2) матрица интеграции источников с сопоставлением атрибутов (loan_amount → product_value, deposit_rate → product_rate); 3) спецификация гранулярности фактовой таблицы fact_client_portfolio как «актуальное состояние портфеля клиента на дату» с полуаддитивной метрикой «количество активных продуктов». Проблема интеграции решается через создание единого измерения dim_product с атрибутом product_type (кредит/вклад/карта) и

<table>
<tr><th>Тема</th><th>Код и формулировка компетенции</th><th>Индикаторы достижения компетенции</th><th>Типовой вариант контрольной работы</th></tr>
<tr><td></td><td></td><td></td><td>унифицированными атрибутами (value, start_date, end_date), а данные из операционных таблиц трансформируются в единую структуру на этапе ETL. Такой подход минимизирует риски, так как: а) концептуальная модель «портфель» утверждается бизнесом до реализации, исключая расхождения в интерпретации метрик; б) спецификация гранулярности предотвращает ошибки агрегации (например, двойной учёт клиента при кросс-продажах); в) матрица интеграции фиксирует правила преобразования семантики источников, что упрощает тестирование и аудит данных на фазе конструирования.</td></tr>
<tr><td>Тема 18. Инструментальные средства разработки программ</td><td>ПК-7 Способен осуществлять проектирование программных интерфейсов</td><td>ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов
ИПК-7.3 Владеть навыками проектирования программных интерфейсов</td><td>Задание 1. Установите соответствие между задачей проектирования аналитического API и инструментальным средством, наиболее подходящим для её решения:
<table>
<tr><th>Задача проектирования API</th><th>Инструментальное средство</th></tr>
<tr><td>1. Создание спецификации контракта эндпоинта агрегации метрик до реализации бэкенда</td><td>А. Newman</td></tr>
<tr><td>2. Автоматизированное тестирование корректности ответов эндпоинта при каждом коммите</td><td>Б. Swagger Editor / Stoplight Studio</td></tr>
<tr><td>3. Демонстрация заказчику поведения API без готовой реализации бэкенда</td><td>В. Mockoon / Prism</td></tr>
<tr><td>4. Публикация интерактивной документации для аналитиков с возможностью выполнения запросов из браузера</td><td>Г. Swagger UI / Redoc</td></tr>
</table>
Правильное соответствие:
1 → Б
2 → А
3 → В</td></tr>
</table>

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			4 → Г Задание 2. При настройке конвейера разработки аналитического API необходимо обеспечить контроль качества контракта интерфейса. Какие из перечисленных инструментов и практик следует внедрить на этапе проектирования (до написания кода бэкенда) для предотвращения ошибок в спецификации? Варианты ответов (выберите 3 правильных): А. Запуск нагрузочного тестирования эндпоинта с помощью k6 для проверки производительности агрегации 10 млн записей. Б. Валидация OpenAPI-спецификации линтером Spectral на соответствие правилам: единообразие параметров даты (date_from/date_to), обязательность поля metadata.request_id. В. Генерация клиентской библиотеки на Python из OpenAPI-спецификации с помощью OpenAPI Generator для тестирования интеграции. Г. Проведение сессии «проектирование за круглым столом» с использованием Stoplight Studio для совместной работы над спецификацией с бизнес-аналитиками. Д. Настройка автоматической публикации документации на внутренний портал при пуше в ветку main. Е. Прототипирование ответов эндпоинта в Mockoon с демонстрацией типовых сценариев: агрегация по регионам, фильтрация по периоду. Правильные ответы: Б, Г, Е Эталон обоснования: Инструменты Б, Г, Е применяются до написания кода: линтер (Б) проверяет корректность спецификации, совместный редактор (Г) обеспечивает согласование контракта с бизнесом, мок-сервер (Е) позволяет протестировать сценарии использования. Нагрузочное тестирование (А) и генерация клиентской библиотеки (В) требуют реализованного бэкенда; публикация документации (Д) относится к этапу доставки. Задание 3. Расположите этапы работы с инструментами при реализации подхода «дизайн-первым» (API-first) для создания эндпоинта /api/analytics/sales в правильной последовательности: А. Генерация серверного шаблона кода (например, на Node.js) из валидной OpenAPI-

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			спецификации с помощью OpenAPI Generator Б. Создание черновика спецификации в Stoplight Studio с описанием эндпоинта, параметров фильтрации и схемы ответа В. Запуск мок-сервера на основе спецификации для демонстрации аналитикам поведения эндпоинта без реализации бэкенда Г. Валидация спецификации линтером Spectral и корректировка нарушений правил именования и структуры Д. Интеграция реализованного эндпоинта в конвейер CI: запуск тестов коллекции Postman через Newman при пуше в репозиторий Правильная последовательность: Б → Г → В → А → Д Задание 4. При проектировании аналитического API для системы бизнес-аналитики необходимо обеспечить обратную совместимость при изменении схемы ответа эндпоинта /api/analytics/customers/segmentation. В спецификации добавлен новый атрибут rfm_score в объект сегмента клиента. Задание: Опишите, какие действия следует выполнить в инструментах до развёртывания новой версии API в продакшен, чтобы минимизировать риски для потребителей (аналитиков, дашбордов). Укажите минимум 3 конкретных действия с названиями инструментов или практик. Эталон ответа: Обновить OpenAPI-спецификацию с добавлением поля rfm_score как опционального (необязательного) атрибута, чтобы существующие клиенты не сломались при отсутствии поля. Запустить мок-сервер (Mockoon/Prism) с новой спецификацией и предоставить его аналитикам для тестирования своих запросов до развёртывания. Добавить в коллекцию Postman тест-кейс, проверяющий корректность ответа как со старой схемой (без rfm_score), так и с новой, и запустить его через Newman в конвейере CI. Опубликовать предварительную версию документации (Swagger UI) с пометкой «Новое поле доступно с 01.06.2024» для информирования потребителей.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			Задание 5. Сценарий: Команда проектирует аналитический API для единой платформы данных ритейлера. Требования: • Поддержка агрегации метрик по произвольным измерениям (регион, категория, период) • Версионирование контракта с параллельной поддержкой двух версий в течение 6 месяцев • Автоматизированная проверка соответствия реализации спецификации при каждом изменении кода • Документация, доступная аналитикам в интерактивном режиме с примерами запросов Задание: Предложите архитектуру инструментального стека для проектирования и сопровождения API. В ответе (200–250 слов) опишите: а) набор инструментов для каждой фазы жизненного цикла спецификации (проектирование → реализация → тестирование → документирование); б) как инструменты интегрируются в единый рабочий процесс (например, через Git-репозиторий); в) как предложенная архитектура снижает риски нарушения контракта при изменении бизнес-требований к метрикам. Эталон ответа: Архитектура стека: а) Проектирование — Stoplight Studio для совместной работы над OpenAPI-спецификацией с бизнес-аналитиками; Реализация — генерация серверного шаблона через OpenAPI Generator; Тестирование — коллекции Postman с запуском в CI через Newman и валидация соответствия ответа схеме с помощью Prism; Документирование — автоматическая публикация через Redocly при мерже в main. б) Все артефакты хранятся в едином Git-репозитории: /spec/openapi.yaml — спецификация, /tests/postman-collection.json — тесты, /docs/ — сгенерированная документация. При пуше в любую ветку запускается пайплайн: валидация Spectral → запуск мок-сервера для просмотра → прогон тестов Newman. в) Риски снижаются за счёт: 1) единого источника истины (спецификация в Git) —

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовой вариант контрольной работы
			все изменения проходят ревью; 2) автоматической валидации в CI — нарушение контракта блокирует мерж; 3) мок-сервера для тестирования изменений до реализации — аналитики подтверждают корректность новых метрик на этапе проектирования, а не после развёртывания.

Критерии оценивания контрольной работы:

Оценка	Критерии оценивания
отлично	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи, формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, обосновывает свои суждения и даёт правильные ответы на вопросы преподавателя
хорошо	Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи и формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, но содержание ответов имеют некоторые неточности и требуют уточнения и комментария со стороны преподавателя.
удовлетворительно	Выставляется, если обучающийся знает и понимает материал по заданной теме, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, студент не может обосновать свои ответы на уточняющие вопросы преподавателя
неудовлетворительно	Выставляется, если обучающийся допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал. Делает ошибки в ответах на уточняющие вопросы преподавателя

4.4. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа включает в себя 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, подборка нормативного и иного материала и выполнение других заданий.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
Тема 1. Проектирование информационных систем	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	Темы для рефератов: 1. Методы проектирования реляционных баз данных 2. Средства проектирования баз данных CASE-технологии 3. Модели данных и их влияние на проектирование информационных систем 4. Особенности проектирования OLAP-систем (OnLine Analytical Processing) 5. Современные подходы к проектированию NoSQL баз данных 6. Оптимизация производительности баз данных при проектировании информационных систем 7. Безопасность данных при проектировании информационных систем 8. Проблемы совместимости и миграции данных при проектировании информационных систем
Тема 2. Жизненный цикл информационной системы	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	Темы для рефератов: 1. История развития методологий жизненного цикла информационных систем 2. Модель каскадного жизненного цикла информационной системы (водопадная модель) 3. Методология спирального жизненного цикла информационных систем 4. Agile-подходы в проектировании информационных систем 5. Процесс проектирования пользовательских интерфейсов в жизненном цикле информационной системы 6. Основные фазы внедрения и сопровождения информационных систем 7. Использование DevOps-практик в современном жизненном цикле информационных систем 8. Ключевые тенденции эволюции жизненного цикла информационных систем в цифровую эпоху
Тема 3. Стандартизация разработки программных средств	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	Темы для рефератов: 1. Международные стандарты ISO/IEC в сфере разработки программных средств и баз данных 2. Стандарт IEEE 829 — спецификация требований к документам по тестированию программного обеспечения 3. Методология Waterfall и стандарт ANSI/IEEE Std 1074 — руководство по процессу жизненного цикла программного обеспечения 4. Стандарты ISO 9126 и ISO/IEC 25010 — качество программного обеспечения и баз данных

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			5. Standard SQL: общая концепция и эволюция стандарта 6. ISO/IEC TR 19760 — Руководящие указания по выбору методологий и методов разработки информационных систем 7. Практическое значение стандартов ГОСТ Р ИСО/МЭК в отечественной разработке баз данных и информационных систем 8. Инженерные нормы ГОСТ Р 54593-2011 и их влияние на практику проектирования баз данных
Тема 4. Модели жизненного цикла	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	Темы для рефератов: 1. Водопадная модель жизненного цикла информационных систем 2. Спиральная модель жизненного цикла: принципы и применение 3. Гибкая методология разработки (Agile): принципы и методологии 4. Модели инкрементальной и итеративной разработки 5. Традиционная V-модель жизненного цикла программного обеспечения 6. Каскадная модель с возвратами (Iterative Waterfall Model) 7. Радужная модель жизненного цикла (Rainbow Model) 8. Интеграционная модель жизненного цикла (RAD model — Rapid Application Development)
Тема 5. Прототипы	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	Темы для рефератов: 1. Прототипирование как основа для быстрого проектирования баз данных 2. Классификация методов прототипирования баз данных 3. Этап создания прототипа в жизненном цикле проектирования информационных систем 4. Инструменты и средства для создания прототипов баз данных 5. Эффективность прототипирования при проектировании сложных баз данных 6. Типичные ошибки при создании прототипов баз данных и пути их устранения 7. Опыт прототипирования баз данных в отечественном бизнесе 8. Будущие направления развития методов прототипирования баз данных
Тема 6. Анализ и моделирование функциональной области	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	Темы для рефератов: 1. Основы функционального анализа и моделирования информационных систем 2. Моделирование бизнес-процессов в Notation BPMN 3. Применение IDEF0 в анализе и моделировании информационных систем 4. DFD-диаграммы и их применение в проектировании информационных систем 5. Принципы проектирования пользовательских интерфейсов (UI/UX) на основе анализа функциональной области

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
внедрения ИС			6. Практика анализа и моделирования деятельности предприятий в SAP ARIS 7. Сценарный подход к анализу и моделированию функциональной области информационных систем 8. Анализ и моделирование функциональной области при внедрении CRM-системы
Тема 7. Полная бизнес-модель компании	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Сравнительный анализ №1: Классическая и цифровая полная бизнес-модель компании Цель анализа: сравнить традиционную (классическую) и современную (цифровую) полную бизнес-модель компаний, работающих в сфере торговли потребительскими товарами. Объект сравнения: 1. Классическая бизнес-модель: Ориентация на офлайн-продажи через розничные магазины. Линейная цепочка поставок: производитель → дистрибьютор → магазин → покупатель. Минимальное использование цифровых каналов коммуникации с покупателями. Ограниченность ассортимента и отсутствие персонализации обслуживания покупателей. Управление информацией преимущественно вручную или с использованием устаревших информационных систем. 2. Цифровая бизнес-модель: Широкий ассортимент товаров, продажа через цифровые каналы (интернет-магазины, маркетплейсы). Гибридная цепочка поставок: прямой выход производителя на покупателя, доставка через курьеров, пункты самовывоза. Активное использование цифровых коммуникаций (социальные сети, email-маркетинг, чат-боты). Высокая персонализация обслуживания благодаря анализу данных о поведении потребителей. Использование современных информационных систем (ERP, CRM, BI-системы) для комплексного управления бизнесом. Выводы: Цифровые бизнес-модели обеспечивают большую гибкость, прозрачность и адаптивность бизнеса.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			Переход к цифровой модели позволяет компаниям значительно повысить лояльность клиентов и снизить издержки. Однако цифровая модель требует значительных инвестиций в информационные технологии и квалифицированный персонал. Сравнительный анализ №2: Полная бизнес-модель двух компаний с разным уровнем зрелости цифровых технологий Цель анализа: проанализировать и сопоставить полные бизнес-модели двух компаний, одна из которых активно развивает цифровые технологии, вторая находится на начальном уровне цифровой трансформации. Объект сравнения: 1. Компания А (высокий уровень цифровой зрелости): Комплексное использование цифровых технологий: мобильные приложения, персональные рекомендации, виртуальные помощники. Централизованная система управления данными (data lake, data warehouse). Высокий уровень автоматизации бизнес-процессов и принятие решений на основе данных. Постоянный мониторинг удовлетворённости клиентов и быстрая реакция на отзывы. Регулярное проведение экспериментов и внедрение инновационных решений. 2. Компания В (низкий уровень цифровой зрелости): Частичное использование цифровых технологий: сайт с ограниченными возможностями, минимальная автоматизация процессов. Данные хранятся фрагментировано, отсутствует централизованное хранилище данных. Низкий уровень автоматизации, преобладают ручные операции. Редкое использование аналитики для принятия управленческих решений. Медленная реакция на изменения внешней среды и потребности клиентов. Выводы: Компания А демонстрирует высокую конкурентоспособность и устойчивость к изменениям рынка благодаря глубокой цифровизации. Компании с низким уровнем цифровой зрелости сталкиваются с проблемами

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			неэффективности процессов, низкой скорости реакции на изменения и снижением привлекательности для клиентов. Для достижения устойчивого роста компания В нуждается в комплексной программе цифровой трансформации, направленной на повышение операционной эффективности и качества обслуживания клиентов.
Тема 8. Спецификация функциональных требований к ИС	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Сравнительный анализ №1: Традиционное и Agile-способы спецификации функциональных требований к ИС Цель анализа: сравнить традиционный подход к спецификации функциональных требований и современный Agile-подход, выявить их преимущества и недостатки. Объект сравнения: 1. Традиционный подход (Waterfall): Фиксированные требования, собранные заранее и задокументированные в полном объеме. Основной инструмент — детальные технические задания и спецификации требований (SRS, BRD). Длительное время согласования и утверждение требований. Сложность внесения изменений после начала разработки. Подходит для проектов с чёткими границами и предсказуемостью. 2. Agile-подход: Функциональные требования формируются постепенно, уточняются в процессе разработки. Основная форма фиксации требований — User Story, Epics, Acceptance Criteria. Быстрая обратная связь от заказчика и пользователей, частые релизы и короткие итерации. Легкость внесения изменений и реагирования на изменения условий. Оптимален для проектов с высоким уровнем неопределенности и динамично меняющейся средой. Выводы: Традиционный подход обеспечивает высокую степень формализации и снижает риски недопонимания, однако менее гибок и плохо подходит для быстро меняющихся условий. Agile-подход повышает вовлечённость заказчика и ускоряет реакцию на изменения,

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			однако предъявляет повышенные требования к коммуникационным навыкам участников проекта. Сравнительный анализ №2: Спецификация требований с использованием UML vs BPMN Цель анализа: сравнить два широко распространенных подхода к спецификации функциональных требований — UML и BPMN, выявить их сходства и различия. Объект сравнения: 1. Спецификация требований с использованием UML (Unified Modeling Language): Основное внимание уделяется внутренним структурам и объектам системы. Используется широкий спектр диаграмм: Use Case, Class, Sequence, State Machine и др. Хорошо подходит для проектирования внутренней структуры и архитектуры информационных систем. Чаще всего применяется техническими специалистами и архитекторами ПО. 2. Спецификация требований с использованием BPMN (Business Process Model and Notation): Основное внимание уделяется бизнес-процессам и операционному уровню. Диаграммы BPMN сосредоточены на потоке работ, ролях, событиях и правилах принятия решений. Удобно использовать менеджерами среднего звена, бизнес-аналитиками и руководителям подразделений. Лучше отражает реальные бизнес-процессы и помогает выявлять узкие места и избыточные операции. Выводы: UML больше подходит для глубокого проектирования внутреннего устройства информационных систем и взаимодействия компонентов. BPMN эффективнее для анализа и моделирования бизнес-процессов, выявления слабых мест и оптимизации рабочих процессов. Наиболее эффективный подход — совместное использование обоих методов, когда UML дополняет BPMN, позволяя связать бизнес-задачи с техническим исполнением.
Тема 9.	ПК-6 Способен	ИПК-6.2 Уметь	Сравнительный анализ №1: Структурная и объектная модели предметной области в

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
Структурные и объектные модели предметной области	применять методы и средства проектирования компьютерного программного обеспечения	применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	проектировании баз данных Цель анализа: сравнить структурную и объектную модели предметной области при проектировании баз данных, выявить их преимущества и недостатки. Объект сравнения: 1. Структурная модель (реляционная модель): Основана на табличном представлении данных. Ключевой элемент — таблица с фиксированной структурой столбцов и строк. Связи между таблицами выражаются через внешние ключи. Простота понимания и широкая распространённость. Поддерживается всеми популярными СУБД (Oracle, MS SQL Server, PostgreSQL, MySQL). Идеально подходит для стандартных задач хранения и обработки данных. 2. Объектная модель (объектно-ориентированная модель): Базируется на объектах, состоящих из свойств и методов. Объекты объединяются в классы, поддерживающие наследование и полиморфизм. Прямая поддержка сложных структур данных и поведения объектов. Лучшая поддержка сложных предметных областей с множеством взаимосвязанных сущностей. Часто реализуется в объектно-ориентированных языках программирования (Java, C#, Python). Может использоваться совместно с объектно-ориентированными базами данных (ObjectDB, db4o). Выводы: Структурная модель проще в понимании и поддержке, идеально подходит для классических задач хранения и обработки данных. Объектная модель обладает большей выразительностью и гибкостью, эффективно поддерживает сложные предметные области и сложные взаимосвязи. Наибольшего эффекта можно достичь путём сочетания обеих моделей, например, при использовании ORM (Object Relational Mapping) технологий. Сравнительный анализ №2: Структурный и объектный подходы в проектировании

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			пользовательских интерфейсов Цель анализа: сравнить структурный и объектный подходы при проектировании пользовательских интерфейсов, выявить их преимущества и недостатки. Объект сравнения: 1. Структурный подход (процедурный подход): Интерфейс строится на основе набора экранов и окон, организованных линейно или древовидно. Каждый экран представляет собой отдельную форму или страницу с жестко заданной структурой. Программирование основано на процедурах и функциях, обрабатывающих события интерфейса. Хорошая управляемость и простота реализации простых интерфейсов. Трудности возникают при усложнении структуры и расширении функционала. 2. Объектный подход (компонентный подход): Пользовательский интерфейс формируется из повторно используемых компонентов (кнопки, списки, окна сообщений и т.д.). Каждый компонент является объектом с собственными свойствами и поведением. Возможность повторного использования и лёгкого расширения интерфейса. Повышенная гибкость и адаптивность к изменениям требований. Широко используется в современных средах разработки (React, Angular, Vue.js, WPF, WinForms). Выводы: Структурный подход удобен для небольших и простых интерфейсов, легко осваиваем и понятен начинающим разработчикам. Объектный подход предпочтителен для сложных и масштабируемых интерфейсов, способствует повышению продуктивности разработки и снижению затрат на поддержку и расширение системы. Современная тенденция — широкое использование объектного подхода практически во всех сферах разработки пользовательских интерфейсов.
Тема 10.	ПК-7 Способен	ИПК-7.2 Уметь	Сравнительный анализ №1: Унифицированный процесс разработки (RUP) против

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
Унифицированный процесс разработки	осуществлять проектирование программных интерфейсов	применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	традиционного каскадного подхода (Waterfall) Цель анализа: сравнить Унифицированный Процесс Разработки (Rational Unified Process, RUP) и классический каскадный подход (Waterfall), выявить их преимущества и недостатки. Объект сравнения: 1. Каскадный подход (Waterfall): Линейная последовательность этапов: анализ требований → проектирование → реализация → тестирование → внедрение. Четкое завершение каждого этапа перед началом следующего. Подходит для проектов с хорошо известными требованиями и стабильной средой. Недостатки: низкая гибкость, невозможность оперативно реагировать на изменения требований. 2. Унифицированный процесс разработки (RUP): Итеративный и инкрементальный подход, состоящий из четырёх фаз: начало, уточнение, конструирование, передача. Повторяющиеся циклы разработки, позволяющие регулярно проверять прогресс и вносить изменения. Использование UML-диаграмм и других унифицированных инструментов для моделирования и документирования. Подходит для сложных и неопределённых проектов, где требования могут меняться. Выводы: Водопадный подход хорош для простых и стабильных проектов, но малопригоден для сложных и динамически развивающихся систем. RUP обеспечивает большую гибкость и адаптивность, позволяет своевременно обнаруживать и устранять дефекты, сокращает риски провала проекта. Сравнительный анализ №2: Унифицированный процесс разработки (RUP) против гибких методологий (Agile) Цель анализа: сравнить Унифицированный Процесс Разработки (RUP) и гибкие методологии семейства Agile (Scrum, Kanban и др.), выявить их сходства и различия.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			Объект сравнения: 1. Унифицированный процесс разработки (RUP): Является полугибким методом, сочетающим элементы итеративности и формализации. Включает обширную документацию и формализованный подход к управлению проектом. Рекомендуется для средних и крупных проектов, где важна формализация и регламентация процессов. 2. Гибкие методологии (Agile): Максимально итеративный и инкрементальный подход, с короткими циклами (спринтами) и регулярной обратной связью. Минимум бюрократии и документации, упор делается на живое общение и быстрое внесение изменений. Особенно подходят для малых и средних команд, работающих в условиях частых изменений требований. Выводы: RUP сочетает преимущества формализации и гибкости, подходит для проектов, где важны соблюдение стандартов и регламентов. Agile-методологии обладают максимальной гибкостью и скоростью реакции на изменения, эффективны в условиях высокой неопределённости и быстрых изменений.
Тема 11. Унифицированный язык моделирование (UML)	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Сравнительный анализ №1: UML-диаграммы классов и UML-диаграммы вариантов использования (Use Case) Цель анализа: сравнить UML-диаграммы классов и UML-диаграммы вариантов использования (Use Case), выявить их преимущества и области применения. Объект сравнения: 1. UML-диаграммы классов: Показывают статическую структуру системы: классы, их атрибуты, методы и взаимосвязи. Помогают визуально представлять внутреннюю структуру системы, облегчая понимание и обсуждение архитектуры.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			Используются на начальных этапах проектирования для определения структуры данных и зависимостей между компонентами. 2. UML-диаграммы вариантов использования (Use Case): Демонстрируют высокоуровневые сценарии взаимодействия пользователей с системой. Позволяют анализировать и документировать функциональные требования системы с точки зрения внешнего наблюдателя. Применимы на ранних этапах проектирования для выяснения и документирования требований пользователей. Выводы: Диаграммы классов полезны для общего понимания структуры системы и проектирования данных. Диаграммы вариантов использования незаменимы для анализа и документирования функциональных требований и сценариев взаимодействия пользователей с системой. Оба вида диаграмм взаимно дополняют друг друга и используются совместно на протяжении всего процесса проектирования. Сравнительный анализ №2: UML и альтернативные языки моделирования (BPMN, Archimate) Цель анализа: сравнить UML с альтернативными языками моделирования (BPMN и ArchiMate), выявить их отличительные особенности и области применения. Объект сравнения: 1. UML (Unified Modeling Language): Универсальный язык моделирования, предназначенный для проектирования широкого спектра информационных систем. Включает богатый набор диаграмм (структур и поведения), позволяющих всесторонне описывать систему. Наиболее популярен среди разработчиков программного обеспечения и инженеров информационных систем. 2. BPMN (Business Process Model and Notation): Специализированный язык для моделирования бизнес-процессов.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			Упрощённая и интуитивно понятная нотация, удобная для менеджеров и бизнес-аналитиков. Не охватывает внутренние детали реализации, фокусируется исключительно на процессах и взаимодействии ролей. 3. ArchiMate: Архитектурный язык моделирования, разработанный для описания архитектуры предприятия. Позволяет моделировать всю экосистему предприятия: бизнес-процессы, приложения, данные, инфраструктура и стратегия. Менее распространён в чисто программной разработке, чаще встречается в крупных компаниях для комплексного моделирования архитектуры предприятия. Выводы: UML универсален и подходит для большинства задач проектирования информационных систем. BPMN рекомендуется для моделирования бизнес-процессов и взаимодействия ролей в организации. ArchiMate оптимален для комплексного моделирования архитектуры предприятия, интегрирующей бизнес, приложения и инфраструктуру. Выбор языка зависит от специфики задачи и состава участников проекта.
Тема 12. Определение требований в рамках унифицированного процесса	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Сравнительный анализ №1: Традиционные и Agile-подходы к определению требований в рамках унифицированного процесса Цель анализа: сравнить традиционные и Agile-подходы к определению требований в рамках унифицированного процесса разработки программного обеспечения. Объект сравнения: 1. Традиционный подход: Предварительно собирается исчерпывающий объём требований. Формируется детальное техническое задание (Technical Specification Document). Требования утверждаются до начала разработки. Изменения требований допускаются редко и связаны с существенными изменениями планов.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			Преимущественно применяются формальные методы сбора и документирования требований. 2. Agile-подход: Требования собираются и уточняются постепенно, в течение итераций. Используются User Stories и Epics для описания требований. Допускается постоянное изменение и дополнение требований в процессе разработки. Важна постоянная обратная связь от заказчика и пользователей. Применяется активное вовлечение команды разработки в процесс сбора и уточнения требований. Выводы: Традиционный подход обеспечивает высокую степень формализации и стабильности требований, но слабо приспособлен к быстрым изменениям. Agile-подход отличается гибкостью и способностью оперативно реагировать на изменения, но требует активного участия заказчика и команды. Выбор подхода зависит от специфики проекта, степени неопределённости и готовности заказчика к активному участию в проекте. Сравнительный анализ №2: Методы сбора требований в рамках унифицированного процесса (Интервью vs Наблюдение) Цель анализа: сравнить методы интервью и наблюдения при сборе требований в рамках унифицированного процесса разработки. Объект сравнения: 1. Метод интервью: Проводится личное собеседование с представителями заказчика и пользователями. Позволяет собрать подробную информацию непосредственно от заинтересованных сторон. Требуется подготовка вопросов и протоколирование ответов. Подходит для сбора абстрактных и стратегических требований. 2. Метод наблюдения: Заключается в непосредственном наблюдении за работой пользователей и сотрудниками.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			Позволяет выявить скрытые и неочевидные требования, которые пользователи сами не осознают. Требует длительного погружения исследователя в рабочую обстановку. Подходит для сбора тактических и повседневных требований. Выводы: Интервью удобно для сбора общих и стратегических требований, позволяет задать прямые вопросы и получить точные ответы. Наблюдение полезно для выявления тонких нюансов и неочевидных деталей повседневной работы пользователей. На практике часто комбинируются оба метода для получения целостной картины требований.
Тема 13. Анализ структуры в рамках унифицированного процесса	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Анализ структуры предметной области «Библиотека» в рамках унифицированного процесса проектирования базы данных. Цель задания: Научиться проводить анализ структуры предметной области и проектировать реляционную базу данных на основе унифицированного процесса. Задания: 1. Проведите анализ предметной области «Библиотека». Определите основные сущности (объекты реального мира), участвующие в функционировании библиотеки (например, книга, читатель, библиотека, библиотечный фонд, категория книг и т.д.). 2. Создайте концептуальную модель данных (ER-модель) с помощью унифицированного языка моделирования (UML). Определите атрибуты и связи между сущностями. 3. Приведите модель к третьей нормальной форме (3NF), выполняя поэтапную нормализацию. 4. Разработайте логическую модель базы данных (DDL-описание таблиц и связей). 5. Создайте физическую модель базы данных в выбранной СУБД (например, MySQL, PostgreSQL, Microsoft SQL Server). 6. Подготовьте отчет, содержащий описание процесса анализа структуры, обоснование принятых решений и описание получившейся модели.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			Практическое задание №2 Тема: Анализ структуры предметной области «Клиника» в рамках унифицированного процесса проектирования базы данных. Цель задания: Овладеть навыками анализа структуры предметной области и проектирования эффективной базы данных для клинической информационной системы. Задания: 1. Проведите анализ предметной области «Медицинская клиника». Определите основные сущности (пациент, врач, медицинская карта, диагноз, лечение, приём, отделение, медицинский персонал и т.д.). 2. Создайте концептуальную модель данных (ER-модель) с помощью UML-диаграмм. Определите атрибуты и связи между сущностями. 3. Выполните нормализацию структуры данных до 3-й нормальной формы (3NF). 4. Разработайте логическую модель базы данных (DDL-описание таблиц и связей). 5. Реализуйте физическую модель базы данных в выбранной СУБД. 6. Подготовьте отчет, содержащий подробное описание процесса анализа структуры, обоснование принятых решений и описание готовой модели.
Тема 14. Проектирование в рамках унифицированного процесса	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Проектирование пользовательского интерфейса информационной системы управления проектами в рамках унифицированного процесса. Цель задания: Овладеть навыками проектирования пользовательских интерфейсов в рамках унифицированного процесса разработки программного обеспечения. Порядок выполнения: 1. Проведите анализ предметной области «Информационная система управления проектами». Определите основные категории пользователей (менеджеры, исполнители, руководители высшего уровня).

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			2. Разработайте Use Case-диаграммы, описывающие основные сценарии взаимодействия пользователей с системой. 3. Создайте Wireframe (эскизный макет) интерфейса главной страницы системы, используя инструменты проектирования интерфейсов (например, Figma, Axure RP, Adobe XD). 4. Разработайте прототип интерфейса, демонстрирующий основные экраны и навигацию между ними. 5. Проведите тестирование прототипа на группе пользователей, запишите и проанализируйте обратную связь. 6. Улучшите интерфейс на основе собранной обратной связи и подготовьте итоговую версию макетов. 7. Подготовьте отчет, содержащий описание процесса проектирования, эскизы интерфейсов, результаты тестирования и выводы по доработкам. Практическое задание №2 Тема: Проектирование API-интерфейса информационной системы анализа данных в рамках унифицированного процесса. Цель задания: Получить навыки проектирования программных интерфейсов (API) в рамках унифицированного процесса разработки. Порядок выполнения: 1. Проведите анализ предметной области «Система анализа данных». Определите основные сервисы и функции, которые будут предоставлены через API. 2. Разработайте спецификацию API, используя OpenAPI/Swagger-нотацию, описав основные эндпоинты, методы, параметры и возвращаемые значения. 3. Создайте прототип API-интерфейса, используя инструменты быстрой разработки (например, Postman, Insomnia). 4. Проведите нагрузочное тестирование прототипа API, зафиксируйте результаты и выявите слабые места. 5. Разработайте улучшенную версию спецификации API с учетом результатов тестирования и анализа нагрузки.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			6. Подготовьте отчет, содержащий описание процесса проектирования, спецификацию API, результаты тестирования и рекомендации по дальнейшим действиям.
Тема 15. Документирование процесса	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Документирование процесса проектирования базы данных для информационной системы управления кадрами предприятия. Цель задания: Научиться документировать процесс проектирования базы данных, создавая необходимую техническую документацию. Задания: 1. Проведите анализ предметной области «управление кадрами предприятия». Определите основные сущности и их атрибуты. 2. Разработайте концептуальную модель данных (ER-модель) и приведите её к третьей нормальной форме (3NF). 3. Создайте логическую модель базы данных (DDL-описание таблиц и связей). 4. Разработайте физическую модель базы данных (физическое размещение данных, индексы, ограничения целостности). 5. Подготовьте технический паспорт базы данных, содержащий следующую информацию: Описание предметной области и целей проектирования. Список сущностей и их атрибутов. Описание структуры таблиц и связей. Правила поддержания целостности данных. Алгоритм загрузки и обновления данных. Инструкция по установке и настройке базы данных. 6. Подготовьте отчет, содержащий описание процесса проектирования, обоснование принятых решений и описание получившегося документа. Практическое задание №2 Тема: Документирование процесса проектирования базы данных для информационной системы учёта продаж интернет-магазина. Цель задания: Овладеть навыками документирования процесса проектирования базы

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			данных с упором на учёт коммерческих данных. Задания: 1. Проведите анализ предметной области «учёт продаж интернет-магазина». Определите основные сущности (заказ, товар, покупатель, оплата, доставка и т.д.). 2. Разработайте концептуальную модель данных (ER-модель) и приведите её к третьей нормальной форме (3NF). 3. Создайте логическую модель базы данных (DDL-описание таблиц и связей). 4. Разработайте физическую модель базы данных (физическое размещение данных, индексы, ограничения целостности). 5. Подготовьте комплект технической документации, включающий: Постановку задачи и описание предметной области. Модель данных и описание структуры таблиц. Руководство по загрузке и выгрузке данных. Инструкцию по резервному копированию и восстановлению данных. Политику обеспечения безопасности данных. 6. Подготовьте отчет, содержащий описание процесса проектирования, обоснование принятых решений и описание полученного комплекта документации.
Тема 16. Организация и планирование работ	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Организация и планирование работ по разработке пользовательского интерфейса информационной системы. Цель задания: Ознакомиться с основами организации и планирования работ по проектированию пользовательских интерфейсов, приобретя навыки разработки дорожной карты и расписания проекта. Порядок выполнения: 1. Определите цели и задачи проекта по разработке пользовательского интерфейса информационной системы, укажите целевые группы пользователей. 2. Разделите общий процесс проектирования интерфейса на отдельные этапы и задачи (анализ требований, создание прототипов, тестирование, правки и внедрение).

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			3. Разработайте дорожную карту проекта, установив взаимосвязи между задачами и оценив их продолжительность. 4. Составьте расписание проекта с указанием дат начала и окончания каждой задачи, назначив ответственных исполнителей. 5. Оцените риски и ограничивающие факторы, которые могут повлиять на ход проекта, предложите меры по их предотвращению. 6. Подготовьте отчет, содержащий описание целей, задач, дорожной карты и расписания проекта, а также мероприятий по управлению рисками. Практическое задание №2 Тема: Организация и планирование работ по созданию API-интерфейса информационной системы. Цель задания: Научиться организовывать и планировать работы по проектированию программных интерфейсов (API), разрабатывая полноценную структуру задач и контролируя их исполнение. Порядок выполнения: 1. Определите цели и задачи проекта по созданию API-интерфейса информационной системы, определите целевое назначение API. 2. Разделите процесс проектирования API на отдельные этапы и задачи (определение ресурсов, проектирование методов, тестирование, интеграция с системой). 3. Разработайте дорожную карту проекта, расположив задачи последовательно и устанавливая зависимости между ними. 4. Составьте расписание проекта, задав сроки выполнения каждой задачи и назначив ответственных исполнителей. 5. Оцените возможные риски и ограничения, которые могут негативно сказаться на ходе проекта, предложите способы их минимизации. 6. Подготовьте отчет, содержащий описание целей, задач, дорожной карты и расписания проекта, а также предложений по контролю рисков.
Тема 17. Оценка	ПК-6 Способен применять	ИПК-6.2 Уметь применять методы и	Практическое задание №1 Тема: Оценка стоимости разработки базы данных для информационной системы

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
стоимости разработки программного продукта	методы и средства проектирования компьютерного программного обеспечения	средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	производственного предприятия. Цель задания: Научиться оценивать стоимость разработки базы данных с учётом требований заказчика и ресурсных затрат. Порядок выполнения: 1. Получите исходные данные: описание производственной деятельности предприятия, объём хранимой информации, планируемые масштабы использования базы данных. 2. Определите необходимые объёмы данных и интенсивность их обработки. 3. Проведите оценку трудозатрат на проектирование, разработку и тестирование базы данных. 4. Оцените материальные затраты на приобретение необходимого оборудования, лицензионного программного обеспечения и сопутствующих ресурсов. 5. Рассчитайте суммарную стоимость проекта, включив косвенные затраты (поддержка, администрирование, обучение персонала). 6. Разработайте финансовый план, предусматривающий возможную амортизацию оборудования и возможное изменение цен на услуги и материалы. 7. Подготовьте отчёт, содержащий оценку стоимости, обоснование выбора метода оценки и предложения по уменьшению стоимости проекта. Практическое задание №2 Тема: Оценка стоимости модернизации существующей базы данных интернет-магазина. Цель задания: Узнать, как рассчитать стоимость модернизации базы данных и учесть дополнительные затраты, возникающие в результате внедрения изменений. Порядок выполнения: 1. Получите исходные данные: характеристику действующей базы данных, описание планируемых изменений и прогнозируемый рост трафика. 2. Оцените трудозатраты на проектирование и разработку необходимых модификаций, включая проверку целостности данных и миграцию.

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			3. Оцените затраты на дополнительное оборудование и лицензию на новое программное обеспечение. 4. Рассчитайте стоимость переобучения персонала и консультационных услуг консультантов. 5. Учтите риски и непредвиденные обстоятельства, которые могут увеличить затраты (например, неполадки при миграции данных). 6. Проведите итоговый расчёт совокупной стоимости модернизации базы данных. 7. Подготовьте отчёт, содержащий финансовую смету, обоснование расчётов и предложения по минимизации рисков и экономии затрат.
Тема 18. Инструментальные средства разработки программ	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Практическое задание №1 Тема: Использование инструментальных средств разработки программ для проектирования и создания пользовательского интерфейса веб-приложения. Цель задания: Научиться пользоваться инструментами разработки программ (интегрированными средами разработки IDE, редакторами кода, системами контроля версий) для проектирования и реализации качественного пользовательского интерфейса. Порядок выполнения: 1. Выберите удобную среду разработки (Visual Studio Code, PyCharm, IntelliJ IDEA, WebStorm и т.п.) и подключите нужные плагины для удобной работы с языком разметки HTML/CSS и JavaScript. 2. Установите систему контроля версий (Git), создайте локальный репозиторий и заведите удалённый репозиторий на GitHub или Bitbucket. 3. Создайте простую одностраничную веб-страницу с оформлением стилей CSS и добавлением интерактивных элементов на JavaScript. 4. Загрузите созданный проект в удалённый репозиторий, продемонстрировав владение навыками работы с Git. 5. Проведите ревью своего кода коллегами или преподавателем, убедитесь в соблюдении стандартов и лучших практик. 6. Подготовьте отчёт, содержащий описание используемых инструментов, скриншоты рабочей среды, описание созданного проекта и рекомендации по улучшению. Практическое задание №2 Тема: Применение инструментальных средств разработки программ для проектирования и

Тема	Код и формулировка компетенции	Индикаторы достижения компетенции	Типовое задание для самостоятельной работы
			создания API-интерфейса информационной системы. Цель задания: Ознакомиться с инструментами разработки программ, такими как среды разработки (IDE), системы контроля версий и вспомогательные инструменты, для проектирования и реализации надежного API-интерфейса. Порядок выполнения: 1. Выберите подходящую среду разработки (PyCharm, Visual Studio Code, IntelliJ IDEA и т.п.) и подготовьте её для работы с выбранным языком программирования (Python, Java, Node.js и т.д.). 2. Установите и настройте систему контроля версий (Git), создадите локальный и удалённый репозитории для сохранения истории разработки. 3. Разработайте минимальный API-интерфейс (RESTful API), реализующий основные функции чтения, записи и обновления данных. 4. Проверьте работоспособность API с помощью инструментов вроде Postman или Insomnia. 5. Загрузите проект в удалённый репозиторий и убедитесь в правильности хранения истории изменений. 6. Подготовьте отчёт, содержащий описание выбранных инструментов, примеров работы с API и рекомендации по улучшению проекта.

Критерии оценивания самостоятельной работы:

Оценка	Критерии оценивания
отлично	выставляется если работа носит научно-исследовательский характер, проанализирован и сделан сравнительный анализ нескольких литературных источников, приведены примеры
хорошо	выставляется если проанализирован и сделан сравнительный анализ нескольких литературных источников, приведены примеры
удовлетворительно	выставляется если проведен сравнительный анализ научно-методической литературы, приведены примеры
неудовлетворительно	выставляется если работа прошла проверку на антиплагиат и соответствует требованиям оформления

4.5. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ ДЛЯ КУРСОВЫХ РАБОТ

Курсовая работа позволяет выявить степень владения базовыми знаниями, умениями и навыками, необходимыми для обучения, и определить уровень владения новым материалом.

Примерные индивидуальные задания (темы) для курсовых работ:

1. Проектирование облачной аналитической платформы для мониторинга KPI розничной сети с адаптивными дашбордами и прогнозированием спроса
2. Проектирование системы анализа клиентского поведения в реальном времени с проекцией данных на интерактивную карту магазина
3. Проектирование low-code платформы для бизнес-аналитиков по созданию кастомных отчётов без участия разработчиков
4. Проектирование системы когнитивной аналитики с естественно-языковым интерфейсом для формирования запросов к данным
5. Проектирование системы мониторинга операционных рисков финансовой организации с автоматической генерацией отчётов регулятору
6. Проектирование кросс-платформенного приложения для полевых исследований с офлайн-режимом синхронизации данных и визуализацией на карте
7. Проектирование системы рекомендаций для персонализации маркетинговых коммуникаций на основе сегментации клиентов
8. Проектирование системы оптимизации логистических маршрутов с учётом динамических факторов (пробки, погода, загрузка складов)
9. Проектирование системы раннего выявления мошенничества в онлайн-платежах с объяснимым ИИ для аналитика
10. Проектирование системы поддержки клинических решений для врачей первичного звена на основе анализа электронных медкарт
11. Проектирование системы прогнозирования оттока клиентов с автоматизированным формированием удерживающих предложений
12. Проектирование корпоративной шины данных (Data Fabric) для унификации доступа к разнородным источникам с единым семантическим слоем
13. Проектирование системы управления API (API Gateway) с функциями мониторинга, ограничения скорости и аналитики использования
14. Проектирование системы интеграции с партнёрами через стандартизированные API с поддержкой разных протоколов (REST, SOAP, EDI)
15. Проектирование событийно-ориентированной архитектуры (Event-Driven) для микросервисной системы с гарантией доставки и упорядочивания событий

16. Проектирование системы федеративной аналитики для анализа данных без их централизованного хранения (аналитика «на месте»)
17. Проектирование мобильного приложения для оперативного мониторинга бизнес-показателей руководителем с голосовым управлением
18. Проектирование веб-приложения для совместного анализа данных с возможностью комментирования визуализаций и совместной работы в реальном времени
19. Проектирование приложения для визуального анализа временных рядов с поддержкой прогнозирования и обнаружения аномалий «одним кликом»
20. Проектирование приложения для анализа социальных сетей с визуализацией сетевых графов и выявлением сообществ
21. Проектирование приложения для геопространственного анализа с наложением бизнес-данных на карту и возможностью пространственных запросов
22. Проектирование системы управления производственными активами (EAM) с прогнозированием отказов оборудования на основе данных с датчиков IoT
23. Проектирование системы управления цепочками поставок (SCM) с цифровым двойником логистической сети и симуляцией сценариев
24. Проектирование системы электронного документооборота с интеллектуальной маршрутизацией документов на основе их содержания
25. Проектирование системы управления обучением (LMS) с адаптивной траекторией обучения на основе анализа прогресса студентов
26. Проектирование системы управления взаимоотношениями с партнёрами (PRM) для дистрибьюторской сети с аналитикой эффективности партнёров
27. Проектирование конструктора аналитических пайплайнов с визуальным интерфейсом «перетащи-и-соедини» для построения цепочек обработки данных
28. Проектирование системы автоматизированной генерации отчётов с шаблонизатором и возможностью настройки триггеров запуска
29. Проектирование платформы для проведения A/B-тестов с автоматическим распределением трафика и статистическим анализом результатов
30. Проектирование системы управления метаданными (Data Catalog) с автоматической классификацией данных и поиском по семантике

Критерии оценивания курсовой работы:

Оценка	Критерии оценивания
отлично	Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Представлены результаты структурированного и логически последовательного обзора литературных и иных источников по теме исследования. Структура курсовой работы логически и методически выдержана. Верно определены исходные данные для расчетов. Все аналитические расчеты выполнены верно, корректно применены методы экономического анализа, не нарушена методика анализа предмета исследования. Все выводы и предложения убедительно аргументированы. При защите курсовой работы обучающийся правильно и уверенно отвечает на вопросы преподавателя, демонстрирует глубокое знание теоретического материала, способен аргументировать собственные утверждения и выводы

хорошо	Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Представлены результаты структурированного и логически последовательного обзора литературных и иных источников по теме исследования. Структура курсовой работы логически и методически выдержана. Верно определены исходные данные для расчетов. В расчетах допускаются незначительные (не искажающие общего итога оценки) погрешности/ошибки. Большинство выводов и предложений аргументировано, корректно применены методы экономического анализа, не нарушена методика анализа предмета исследования. Оформление курсовой работы и полученные результаты в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две несущественные ошибки в использовании терминов, в построенных диаграммах и схемах, в оформлении таблиц. Наличествует незначительное количество грамматических и/или стилистических ошибок. При защите курсовой работы обучающийся правильно и уверенно отвечает на большинство вопросов преподавателя, демонстрирует хорошее знание теоретического материала, но не всегда способен аргументировать собственные утверждения и выводы. При наводящих вопросах преподавателя исправляет ошибки в ответе
удовлетворительно	Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Результаты обзора литературных и иных источников представлены недостаточно полно, недостаточно логично и последовательно. Верно определены исходные данные для расчетов, но имеются грубые ошибки в расчетах. Аргументация выводов и предложений слабая или отсутствует. Экономические выводы носят констатирующий (описательный) характер. Имеются одно-два существенных отклонений от требований в оформлении курсовой работы. Полученные результаты в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две существенных ошибки в использовании терминов, в построенных диаграммах и схемах. Много грамматических и/или стилистических ошибок. При защите курсовой работы обучающийся допускает грубые ошибки при ответах на вопросы преподавателя, демонстрирует слабое знание теоретического материала, в большинстве случаев не способен уверенно аргументировать собственные утверждения и выводы
неудовлетворительно	Содержание курсовой работы не соответствует заданию, содержащемуся в методических указаниях, и плану. Неверно определены исходные данные для расчетов, неверно и не корректно применены методы экономического анализа. Экономические выводы содержат неверную экономическую оценку. Имеются более двух существенных отклонений от требований в оформлении курсовой работы. Большое количество существенных ошибок по сути работы, много грамматических и стилистических ошибок и др. Полученные результаты не отвечают требованиям, изложенным в методических указаниях. При защите курсовой работы обучающийся демонстрирует слабое понимание программного материала, студент не может защитить свои решения, допускает грубые фактические ошибки при ответах на поставленные вопросы или вовсе не отвечает на них. Курсовая работа не представлена преподавателю. Обучающийся не явился на защиту курсовой работы

5. МАТЕРИАЛЫ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Промежуточная аттестация проводится в форме экзамена.

5.1. Вопросы к экзамену:

1. Основные этапы процесса разработки информационной системы и их содержание. Жизненный цикл информационных систем.
2. Модели жизненного цикла.
3. Состав информационных систем.
4. Проектирование программного обеспечения при структурном подходе.
5. Проектирование программного обеспечения при объектном подходе.
6. Профили открытых информационных систем.
7. Управление проектом по созданию программного обеспечения.
8. Понятие бизнес-процесса. Реинжиниринг бизнес-процессов. Модель «AS-IS» и «AS-TO-BE».
9. Общие принципы проектирования информационных систем.
10. Унифицированный язык моделирования (UML): диаграммы прецедентов использования и диаграммы деятельности.
11. Унифицированный язык моделирования (UML): диаграммы пакетов и диаграммы компонентов.
12. Унифицированный язык моделирования (UML): диаграммы классов и диаграммы последовательностей.
13. Унифицированный язык моделирования (UML): диаграммы объектов и диаграммы конечных автоматов.
14. Математическое обеспечение процесса разработки информационной системы.
15. CASE-технологии в проектировании информационных систем.
16. Оценка трудоемкости и стоимости проекта по созданию информационной системы.
17. Структурное проектирование программного обеспечения.
18. Объектно-ориентированное проектирование программного обеспечения.
19. Методология быстрой разработки приложений.
20. Стандартизация разработки программных средств.

Критерии оценивания зачета с оценкой/экзамена:

Оценка	Критерии оценивания
зачтено	Обучающийся должен: уметь строить ответ в соответствии со структурой излагаемого вопроса; продемонстрировать прочное, достаточно полное усвоение знаний программного материала; продемонстрировать знание основных теоретических понятий; правильно формулировать определения; последовательно, грамотно и логически стройно изложить теоретический материал; продемонстрировать умения самостоятельной работы с литературой; уметь сделать достаточно обоснованные выводы по излагаемому материалу.
не зачтено	Обучающийся демонстрирует: незнание значительной части программного материала; не владение понятийным аппаратом дисциплины; существенные ошибки при изложении учебного материала; неумение строить ответ в соответствии со структурой излагаемого вопроса; неумение делать выводы по излагаемому материалу.

Критерии оценивания зачета с оценкой/экзамена:

Оценка	Критерии оценивания
отлично	Выставляется, если обучающимся правильно и полностью раскрыто содержание материала в пределах программы, чётко и правильно даны определения и раскрыто содержание понятий, точно использованы научные и технические термины, в ответе использованы ранее приобретённые теоретические знания, сделаны необходимые выводы и обобщения
хорошо	Выставляется, если обучающимся раскрыто основное содержание материала в пределах программы, даны определения и раскрыто содержание понятий, в ответе использованы ранее приобретённые теоретические знания, сделаны необходимые выводы и обобщения, но присутствуют незначительные нарушения в последовательности изложения, имеются одна-две неточности в содержании ответа
удовлетворительно	Выставляется, если обучающимся содержание учебного материала изложено фрагментарно, не всегда последовательно, не даны определения, не раскрыто содержание понятий, или они изложены с ошибками, допускаются ошибки и неточности в использовании научной терминологии, отсутствуют выводы и обобщения из предыдущего материала, или возможны ошибки в их изложении
неудовлетворительно	Выставляется, если обучающимся основное содержание учебного материала не раскрыто, не даются ответы на основные вопросы, допускаются грубые ошибки в определении понятий, в использовании терминологии, отсутствуют выводы и обобщения

6. МАТЕРИАЛЫ ДЛЯ ДИАГНОСТИЧЕСКОЙ РАБОТЫ

Задания для диагностической работы должны обеспечивать оценку полностью или частично сформированных компетенций. Каждое задание должно быть привязано к тому или иному индикатору сформированности компетенций.

При формировании заданий для диагностической работы необходимо использовать тестовые задания следующих типов:

Тип задания 1. Задания закрытого типа на установление соответствия.

Тип задания 2. Задания закрытого типа на установление последовательности.

Тип задания 3. Задания комбинированного типа, предполагающие выбор одного правильного ответа из предложенных

Тип задания 4. Задания комбинированного типа, предполагающие выбор нескольких ответов из предложенных

Тип задания 5. Задания открытого типа с развернутым ответом.

<table>
<tr><th>№ п/п</th><th>Тема занятия</th><th>Код компетенции</th><th>Индикатор</th><th>Тип задания</th><th colspan="2">Задание</th><th>Уровень освоения компетенции</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Вариант 1</td><td>Вариант 2</td><td></td></tr>
<tr><td></td><td>Тема 1. Проектирование информационных систем</td><td>ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения</td><td>ИПК-6.1 Знать методы и средства проектирования баз данных</td><td>1</td><td>1. Установите соответствие между этапами проектирования базы данных и их основными результатами:
<table><tr><td>Этап проектирования</td><td>Результат этапа</td></tr><tr><td>1. Инфологическое (концептуальное) проектирование</td><td>А. Нормализованная логическая структура таблиц с указанием первичных и внешних ключей</td></tr><tr><td>2. Даталогическое (логическое) проектирование</td><td>Б. Диаграмма «сущность-связь» (ER-диаграмма) без привязки к СУБД</td></tr><tr><td>3. Физическое проектирование</td><td>В. Оптимизированные индексы, выбор типов данных и параметров хранения</td></tr></table></td><td>1. Установите соответствие между типом связи в ER-модели и способом её реализации в реляционной базе данных:
<table><tr><td>Тип связи</td><td>Реализация в реляционной БД</td></tr><tr><td>1. Один ко многим (1:N)</td><td>А. Создание отдельной связующей таблицы с двумя внешними ключами</td></tr><tr><td>2. Многие ко многим (M:N)</td><td>Б. Добавление первичного ключа одной сущности как внешнего ключа в таблицу другой сущности</td></tr><tr><td>3. Один к одному (1:1)</td><td>В. Объединение атрибутов обеих сущностей в одну таблицу либо размещение внешнего ключа с уникальным ограничением (UNIQUE)</td></tr></table>Правильное соответствие:
1 → Б</td><td>Базовый (1-3 минуты)</td></tr>
</table>

			ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов		с учётом СУБД Правильное соответствие: 1 → Б 2 → А 3 → В 2. Название: Разработка структуры базы данных информационной системы предприятия торговли бытовой техникой. Цель: Применение методов проектирования баз данных и структур данных для хранения информации о товарах, поставщиках, заказчиках и продажах. Задания: 1. Анализ предметной области Составьте список сущностей, участвующих в процессе торговли бытовой техники. Определите атрибуты каждой сущности. 2. Проектирование ER-диаграммы Разработайте концептуальную модель данных (ER-модель). Представьте связи между сущностями (отношения). 3. Нормализация Приведите таблицы базы данных к третьей нормальной форме (3NF). 4. Разработка физической модели Создайте физическую структуру таблиц, включая типы полей, ключи и индексы. Предложите оптимальные стратегии индексирования для ускорения запросов. 5. Описание архитектуры БД	2 → А 3 → В 2. Название: Проектирование клиент-серверного приложения информационной системы учета рабочего времени сотрудников. Цель: Применить принципы проектирования программного обеспечения и разработки программных интерфейсов для автоматизации процесса учёта рабочего времени сотрудников. Задания: 1. Сбор требований Провести интервью с представителями отдела кадров и руководителями подразделений организации. Записать функциональные требования к приложению. 2. Выбор технологий Подберите инструменты и технологии для реализации проекта (например, Python/Django, Java/Spring Boot, PHP/Laravel, Node.js + Express.js и др.). Назовите преимущества выбранного решения. 3. Проектирование архитектурных компонентов Спроектируйте общую архитектуру приложения, включающую клиентскую	**Повышенный (3-5 минут)**

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Опишите основные компоненты будущей информационной системы, необходимые для реализации хранилища данных.</td>
<td>и серверную части.
Описать взаимодействия клиента и сервера.
4. Реализация основных функций
Реализовать регистрацию пользователей, аутентификацию и авторизацию.
Предоставить интерфейс для ввода рабочих часов сотрудниками.
Автоматизировать процесс формирования отчетов.
5. Документирование API-интерфейса
Подготовьте документацию по API-приложения с описанием всех эндпоинтов, передаваемых данных и возвращаемых результатов.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Тема 2. Жизненный цикл информационной системы</td>
<td>ПК-7 Способен осуществлять проектирование программных интерфейсов</td>
<td>И П К - 7 . 1 Знать методы и средства проектирования программных интерфейсов</td>
<td>1</td>
<td>3. Установите соответствие между этапами жизненного цикла информационной системы и задачами проектирования программных интерфейсов (API), характерными для каждого этапа:
<table>
<tr><td>Этап жизненного цикла ИС</td><td>Задача проектирования API</td></tr>
<tr><td>1. Анализ и проектирование</td><td>А. Настройка rate limiting, мониторинг доступности эндпоинтов и логирование запросов</td></tr>
<tr><td>2. Реализация и интеграция</td><td>Б. Определение контракта API: эндпоинты, форматы данных (JSON/CSV), методы аутентификации</td></tr>
<tr><td>3. Тестирован</td><td>В. Версионирование API (v1, v2) и</td></tr>
</table></td>
<td>3. Установите соответствие между моделью жизненного цикла ИС и подходом к проектированию и эволюции программных интерфейсов:
<table>
<tr><td>Модель жизненного цикла</td><td>Подход к проектированию и развитию API</td></tr>
<tr><td>1. Каскадная (водопадная) модель</td><td>А. Постепенное расширение функционала API через новые эндпоинты с сохранением старых версий</td></tr>
<tr><td>2. Итеративная модель</td><td>Б. Чёткое определение полного контракта API на этапе проектирования с минимальными изменениями в будущем</td></tr>
</table></td>
<td>Базовый (1-3 минуты)</td>
</tr>
</table>

					бизнеса. 2. Анализ требований: Составьте перечень функциональных и нефункциональных требований к системе. Используйте диаграмму Use Case для описания вариантов использования системы. 3. Проектирование системы: Выполните проектирование архитектуры системы, спроектируйте базу данных и определите структуру классов. Представьте UML-диаграммы (классовые, последовательностные, компонентные). 4. Разработка прототипа: Разработайте рабочий прототип системы, демонстрирующий ключевые функции (управление складскими остатками, оформление заказов). 5. Тестирование и верификация: Организуйте тестирование разработанного прототипа, используя Unit-тесты и интеграционные тесты. Оформите отчёт о результатах тестирования. 6. Эксплуатация и сопровождение: Спланируйте процессы эксплуатации и сопровождения системы. Разработайте рекомендации по внедрению и обучению персонала работе с системой.	Постройте детальный график выполнения проекта с использованием методик планирования (метод критического пути, сетевое моделирование). Выделите контрольные точки и риски. 3. Организация команды: Распределите роли участников проекта (аналитик, разработчик, тестировщик, администратор). Опишите критерии оценки качества работы каждого члена команды. 4. Управление качеством: Выберите метрики и показатели качества продукта (нагрузка сайта, скорость загрузки страниц, безопасность данных). Опишите стратегию улучшения показателей. 5. Обеспечение внедрения: Создайте подробный план запуска ресурса, включив этапы подготовки инфраструктуры, тестирования, обучения пользователей и технической поддержки. 6. Оценка эффективности: Оцените достигнутые результаты проекта, проанализировав удовлетворенность пользователей и выполнение поставленных задач. Оформите итоговый отчет о проекте.	
	Тема 3. Стандартизация разработ	ПК-6 Способен применять методы и	ИПК-6.1 Знать методы и средства проектирован	1	5. Установите соответствие между стандартами и методологиями проектирования баз данных и их основным назначением в контексте	5. Установите соответствие между правилами стандартизации при проектировании баз данных и их практическим применением в	**Базовый (1-3 минуты)**

	ки программных средств	средства проектирования компьютерного программного обеспечения	ия баз данных		разработки информационных систем для анализа данных: <table><tr><th>Стандарт / Методология</th><th>Основное назначение</th></tr><tr><td>1. ГОСТ 34.601-90</td><td>А. Определение трёхуровневой архитектуры БД (внешний, концептуальный, внутренний уровни)</td></tr><tr><td>2. Модель ANSI/SPARC</td><td>Б. Регламентация этапов жизненного цикла автоматизированных систем (включая проектирование БД)</td></tr><tr><td>3. Нотация IDEF1X</td><td>В. Формализованное описание семантики данных и метаданных для обеспечения их повторного использования в аналитических системах</td></tr><tr><td>4. ГОСТ Р ИСО/МЭК 11179</td><td>Г. Графическое моделирование структуры БД с чёткими правилами отображения сущностей, атрибутов и связей</td></tr></table>Правильное соответствие: 1 → Б 2 → А	аналитических информационных системах: <table><tr><th>Правило стандартизации</th><th>Практическое применение в аналитической ИС</th></tr><tr><td>1. Единые соглашения об именовании (naming conventions)</td><td>А. Применение префиксов dim_ для таблиц измерений и fact_ для таблиц фактов в схеме «звезда»</td></tr><tr><td>2. Нормализация до 3НФ</td><td>Б. Разделение таблицы «Клиенты» на отдельные таблицы «Клиенты», «Адреса», «Контакты» для устранения избыточности</td></tr><tr><td>3. Денормализация для OLAP-систем</td><td>В. Создание единого глоссария бизнес-терминов с привязкой к техническим атрибутам таблиц для согласования метрик между аналитиками и разработчиками</td></tr><tr><td>4. Стандарты документирования моделей данных</td><td>Г. Объединение таблиц измерений с таблицей фактов в единую плоскую структуру для ускорения агрегации данных</td></tr></table>Правильное соответствие: 1 → А	

					3 → Г 4 → В 6. Название: Проектирование реляционной базы данных туристической фирмы Цель: Отработать навыки стандартизации разработки программных средств, ориентируясь на проектирование баз данных, применяя стандарты нормализации и принципов объектно-реляционного проектирования. Задания: 1.Анализ предметной области Провести исследование предметной области туристических услуг и выделить главные объекты, события и процессы. Выявить ключевые элементы данных, необходимых для регистрации туров, клиентов, бронирования мест и оплаты услуг. 2. Создание концепции модели данных Спроектировать ER-диаграмму (Entity Relationship Diagram) и определить взаимосвязи между объектами. Использовать общепринятые обозначения стандартов для отображения связей ("один-к-одному", "один-ко-многим", "многие-ко-многим"). 3. Нормализация схемы базы данных Преобразовать схему данных в третью нормальную форму (3NF). Проанализировать и устранить	2 → Б 3 → Г 4 → В 6. Название: Создание корпоративной базы данных отдела закупок крупной торговой сети Цель: Формирование навыков стандартизированного подхода к созданию схем баз данных с соблюдением отраслевых стандартов и подходов к документированию проектов. Задания: 1. Формулировка требований Перечислить функциональные и нефункциональные требования к проекту, основанные на стандарте ISO/IEC 29148 (Software and systems engineering — Life cycle processes — Requirements engineering). Подробно описать желаемые характеристики и функциональность системы закупки товаров. 2. Моделирование данных Создать подробную ERD (entity relationship diagram) для представленной системы, обозначив сущности, атрибуты и отношения. Используя правила стандарта IDEF1X, представить систему понятным образом, учитывающим стандарты проектирования и интеграции с существующими системами компании. 3. Стандартизация документов	**Повышенный (3-5 минут)**

					избыточность и аномалии данных, возникающие при вставке, обновлении и удалении записей. 4. Физический дизайн базы данных Осуществить преобразование нормализованной логической модели в физическую модель данных, включая определение типов данных, первичных ключей, индексов и ограничений целостности. Реализовать использование триггеров и хранимых процедур для поддержания согласованности данных. 5. Оптимизация производительности Произвести оценку операций над базой данных и разработать рекомендации по улучшению производительности путем оптимизации запросов и индексации.	Составить техническую спецификацию базы данных согласно ГОСТ Р ИСО/МЭК 12207-2010 («Информационная технология. Процессы жизненного цикла программных средств»). Документально оформить выбранные алгоритмы обработки данных и условия транзакций, используемые при взаимодействии модулей. 4. Генерация документации Оформить полные SQL-скрипты для создания всех объектов базы данных, следуя стандартам написания комментариев и оформления документации SQL-запросов. Сформировать описание всех созданных объектов и подготовить инструкцию по развертыванию базы данных. 5. Автоматизация процессов миграции данных Разработать сценарии миграционных скриптов для переноса данных из существующих устаревших систем в новую базу данных, соблюдая совместимость данных и целостность всей информации.	
	Тема 4. Модели жизненного цикла	ПК-7 Способен осуществлять проектирование программных интерфейсов	И П К - 7 . 1 Знать методы и средства проектирования программных интерфейсов	2	7. Расположите этапы проектирования и внедрения RESTful API для аналитической информационной системы в соответствии с каскадной моделью жизненного цикла (ГОСТ 34.601-90): А. Разработка технического задания на API: определение эндпоинтов, форматов данных (JSON/CSV), требований к	7. Расположите фазы итерации в методологии Agile/Scrum при разработке версии 2.0 API для системы бизнес-аналитики, начиная с планирования и заканчивая ретроспективой: А. Демонстрация заинтересованным лицам новых эндпоинтов для получения аналитических дашбордов в реальном времени	**Базовый (1-3 минуты)**

					аутентификации Б. Написание программного кода эндпоинтов, реализация логики агрегации и фильтрации аналитических данных В. Составление отчётности по результатам приёмочных испытаний: проверка корректности возврата метрик, производительности запросов Г. Проведение модульного и интеграционного тестирования эндпоинтов с использованием инструментов (Postman, Swagger) Д. Анализ предметной области: выявление источников данных, бизнес-метрик и требований заинтересованных лиц к аналитическим отчётам Правильная последовательность: Д → А → Б → Г → В 8. Название: Проектирование RESTful API для системы отслеживания активности фитнес-клуба Цель: Развитие навыков проектирования программных интерфейсов в рамках каскадной модели жизненного цикла информационной системы. Ход выполнения задания: 1. Анализ требований Провести сбор требований и постановку задачи, основываясь на анализе потребностей клиентов и руководства фитнес-клуба. Фиксировать функциональную и	Б. Адаптация бэклога продукта: приоритизация задач по улучшению производительности API на основе метрик мониторинга В. Проведение спринт-планирования: формирование бэклога спринта с задачами «реализовать пагинацию», «добавить фильтрацию по времени» Г. Ежедневные стендапы команды: синхронизация прогресса по реализации эндпоинтов, выявление блокеров интеграции с источниками данных Д. Проведение ретроспективы: анализ эффективности процесса разработки API, выработка улучшений для следующей итерации Правильная последовательность: Б → В → Г → А → Д 8. Название: Проектирование мобильного приложения с графическим интерфейсом для образовательного учреждения Цель: Овладение навыками проектирования программных интерфейсов мобильных устройств в рамках гибких моделей жизненного цикла (Agile/Scrum). Ход выполнения задания: 1. Постановка задачи Определение ключевых требований к мобильному приложению, предназначенному для учащихся и	**Повышенный (3-5 минут)**

					нефункциональную составляющую требований к разрабатываемому API. 2. Проектирование API Использовать методологию проектирования RESTful API, определяя ресурсные маршруты и поддерживаемые HTTP-методы. Описать контракт API, включающий JSON-структуры запросов и ответов, коды ошибок и возможные исключения. 3. Документация Оформить полную документацию API с примерами запросов и примеров данных. Включить разделы по аутентификации, обработке ошибок и поддержке версий API. 4. Кодировка и тестирование Реализовать минимальный MVP (Minimum Viable Product) версии API на одном из современных фреймворков (Django Rest Framework, Flask-RESTful, Spring Boot и т.п.). Написать автотесты для проверки соответствия поведения API спецификации и требованиям. 5. Интеграция Интегрировать разработанный API с фронтенд-приложением фитнес-клуба, обеспечивая надежную передачу данных о посетителях, занятиях и тренировках. 6. Оценка результата Проверить работоспособность API и качество взаимодействий между клиентом и сервером. Оценить полноту покрытия требований и эффективность принятой модели	преподавателей университета. Постановка конкретных задач (просмотр расписания занятий, получение уведомлений, управление учебными материалами). 2. Разработка прототипа интерфейса Использование Sketch/Figma/Miro для создания wireframe и mockup графического интерфейса. Организация сессий Agile-группового мозгового штурма для уточнения деталей интерфейса. 3. Пользовательские истории Работа с командой разработчиков по формулированию User Stories и Backlog'а итераций (sprints). Назначение приоритетов задачам и оценке трудозатрат. 4. Реализация Программирование интерфейса мобильного приложения с использованием React Native или Flutter. Проведение ежедневных Scrum-встреч для контроля прогресса и устранения проблем. 5. Тестирование и улучшение Тщательное тестирование интерфейса с участием конечных пользователей. Сбор обратной связи и проведение ретроспектив для внесения улучшений. 6. Результат Презентация готового продукта команде заказчиков и участникам проекта. Итоговая оценка успешности проекта по критериям достижения заявленных целей и соблюдения сроков.	

					жизненного цикла.		
	Тема 5. Прототипы	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	2	9. Расположите этапы создания прототипа базы данных для аналитической системы в правильной последовательности от концептуального проектирования до верификации: А. Формирование таблиц измерений (измерения «Время», «Клиент», «Товар») и таблицы фактов с внешними ключами Б. Построение концептуальной модели «сущность-связь» с выделением бизнес-сущностей и их атрибутов В. Преобразование ER-модели в логическую реляционную схему с нормализацией до 3НФ Г. Денормализация логической схемы в модель «звезда» для оптимизации аналитических запросов Д. Верификация прототипа: выполнение типовых OLAP-запросов (агрегация, срезы, сводные отчёты) и оценка производительности Правильная последовательность: Б → В → А → Г → Д	9. Расположите действия при итеративном прототипировании витрины данных (data mart) для отдела маркетинга в порядке их выполнения в рамках спиральной модели жизненного цикла: А. Реализация ETL-скрипта для загрузки данных из источников в прототип витрины (таблицы фактов продаж и измерений) Б. Сбор обратной связи от маркетологов: анализ корректности метрик, полноты измерений, удобства формирования отчётов В. Определение границ прототипа: выбор ключевых метрик (выручка, конверсия) и измерений (период, канал привлечения, сегмент клиента) Г. Оценка рисков: проверка качества исходных данных, доступности источников, соответствия бизнес-требованиям Д. Визуализация результатов: подключение прототипа к инструменту (Power BI / Tableau) и демонстрация интерактивных дашбордов Правильная последовательность: В → Г → А → Д → Б	**Базовый (1-3 минуты)**
			ИПК-6.2 Уметь применять методы и		10. Название: Разработка низкоуровневого прототипа информационной системы управления проектами.	10. Название: Высокоуровневый прототип сервиса для онлайн-обучения иностранным языкам.	**Повышенный (3-5 минут)**

			средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов		Цель: Научиться создавать прототип информационной системы, способствующий быстрому пониманию функциональных возможностей программы перед началом полномасштабной разработки. Основные шаги: 1. Анализ предметной области Исследование базовых характеристик проектной деятельности (учёт ресурсов, сроки исполнения, мониторинг состояния задач). Формулирование списка минимальных функциональных требований к системе. 2. Подготовка технического задания Составление чёткого технического задания на разработку прототипа. Выбор подходящей технологической платформы (Python/Django, Ruby on Rails, Vue.js, Angular и т.д.). 3. Создание макета интерфейса Нарисовать low-fi wireframes экранов системы с основными элементами (таблицы задач, графики нагрузки, формы редактирования проектов). Инструментальная поддержка: Figma, Balsamiq Mockups, Adobe XD. 4. Низкоуровневая реализация Кодирование базовой логики приложения с минимальным набором контроллеров и шаблонов представлений. Простое хранение данных в памяти (без полноценной базы данных). 5. Демонстрация и тестирование	Цель: Развитие навыков высокоуровневой разработки прототипов для быстрой проверки гипотез относительно востребованности будущих продуктов. Основные шаги: 1. Исследование спроса Проведение исследования потенциальной аудитории (школьники, студенты, взрослые пользователи). Сбор информации о текущих продуктах-конкурентах (Duolingo, Lingualeo и т.д.). 2. Формулировка идеи Четкое изложение идеи будущего сервиса, описание уникального торгового предложения (USP). Планирование минимальной жизнеспособной версии продукта (MVP). 3. Высокий уровень детализации прототипа Детальное проектирование экрана главной страницы, профиля пользователя, курсов и уроков. Работа с инструментами высокого уровня детализации (Figma Prototype Mode, InVision Studio). 4. Интерфейсное прототипирование Связывание отдельных экранов в единый прототип, симулируя работу реального приложения. Настройка анимаций переходов и взаимодействия с элементами интерфейса. 5. Проведение тестов и фокус-группы	

					Демонстрация работоспособности интерфейса и логики основной функциональности. Корректировка недостатков, выявленных в ходе демонстрации. 6. Заключение Подготовка отчета о проделанной работе, содержащего выводы о целесообразности дальнейшего развития прототипа.	Тестирование прототипа среди потенциальных пользователей. Аналитика реакции пользователей на каждый элемент интерфейса и взаимодействия. 6. Отчет о проделанной работе Оформление отчета с выводами и рекомендациями по дальнейшему развитию продукта.	
	Тема 6. Анализ и моделирование функциональной области внедрения ИС	ПК-7 Способен осуществлять проектирование программных интерфейсов	И П К - 7 . 1 Знать методы и средства проектирования программных интерфейсов	2	11. Расположите этапы анализа функциональной области и проектирования API для внедрения системы бизнес-аналитики в порядке их выполнения: А. Декомпозиция бизнес-процессов «Продажи» и «Маркетинг» с выделением операций, генерирующих данные для анализа (оформление заказа, клик по баннеру, оплата) Б. Формирование требований к контракту API: определение эндпоинтов /api/sales/aggregate, /api/customers/segment, форматов ответа (JSON с метриками), параметров фильтрации В. Сбор требований от заинтересованных лиц: определение ключевых метрик (выручка, конверсия, LTV) и сценариев использования аналитических отчётов Г. Создание диаграммы потоков данных (DFD): отображение источников данных (CRM, веб-аналитика), процессов	11. Расположите действия при моделировании функциональной области «Управление клиентскими данными» для построения единого профиля клиента (Customer 360°) в правильной последовательности: А. Проектирование интерфейса агрегации данных: разработка эндпоинта /api/customer/{id}/profile с объединением данных из всех источников в единый JSON-ответ Б. Анализ источников данных: выявление систем-доноров (CRM, колл-центр, мобильное приложение) и их текущих методов экспорта данных (API, файлы, БД) В. Определение функциональных требований: составление списка атрибутов единого профиля (персональные данные, история покупок, поведенческие метрики, жалобы) Г. Моделирование бизнес-объекта «Клиент»: создание атрибутивной модели с классификацией атрибутов на	**Базовый (1-3 минуты)**

			ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов		агрегации и потребителей (дашборды, отчётность) Д. Моделирование интерфейсов взаимодействия с внешними системами: определение точек интеграции (API CRM, Google Analytics) и форматов обмена данными Правильная последовательность: В → А → Г → Д → Б 12. Название: Проектирование информационной системы управления логистическими операциями производственной компании. Цель: Научиться проводить функциональный анализ и моделирование конкретной предметной области, выявить базовые процессы и построить их модели. Ход выполнения задания: 1. Изучение предметной области Ознакомьтесь с процессом производства и доставки готовой продукции, исследовав роль поставщиков сырья, производственных цехов, склада готовой продукции и транспортных компаний. 2. Выделение ключевых процессов Определите границы системы и выделите основные процессы, подлежащие автоматизации: закупка материалов, производство, упаковка, доставка. 3. Диаграммы потоков данных (DFD) Постройте DFD первого уровня,	идентификаторы, измерения и факты Д. Спецификация требований к внешним интерфейсам: формирование технических заданий на адаптацию API источников (добавление полей, изменение форматов, обеспечение частоты обновления) Правильная последовательность: В → Б → Г → Д → А 12. Название: Моделирование информационной системы для клиники медицинского центра диагностики заболеваний. Цель: Овладеть навыком анализа и моделирования сложной функциональной области в сфере медицины, выявляя потребности пациентов и врачей. Ход выполнения задания: 1. Исследования и выявление процессов Проводится обследование текущего состояния клиники: регистрация пациента, запись на прием, назначение анализов, диагностика, лечение, выписка рецепта. 2. Сбор и систематизация требований Формулируются конкретные функциональные и нефункциональные требования к создаваемой информационной системе, такие как ведение электронной медицинской карты, интеграция с лабораториями, уведомление врача о готовности	**Повышенный (3-5 минут)**

					отражающего общие потоки данных внутри компании и внешние взаимодействия. Расширьте каждую подсистему до второго уровня детализации, показывая внутренние операции и обмен информацией. 4. Моделирование бизнес-процессов (BPMN) Построить диаграммы BPMN для каждого ключевого процесса (закупка, производство, транспортировка), фиксируя последовательности шагов, принимаемые решения и исполнителей. 5. Анализ полученной модели Проверьте правильность построенных моделей, убедитесь в отсутствии разрывов или противоречий в потоках данных и процессах. Докажите достаточность предложенной структуры для удовлетворения потребностей производственного предприятия. 6. Заключение Подготовьте отчет о проделанной работе, содержащий полученные модели и пояснения к ним.	результатов исследований. 3. Функциональный анализ Структурируется деятельность клиника в функциональные блоки, определяются подпроцессы и роли участников. 4. Построение UML-моделей Строятся use case-диаграммы, class-диаграммы и activity-диаграммы для наглядного представления общей картины функционирования клиники. 5. Применение нотации ARIS/EPC Для лучшего понимания внутренних процессов строится event-driven process chain (EPC)-диаграмма, отражающая последовательность и логику происходящих действий. 6. Финальная проверка и представление результатов Готовится отчет с моделью информационной системы, результатами анализа и рекомендациями по дальнейшей реализации системы.	
	Тема 7. Полная бизнес-модель компании	ПК-6 Способен применять методы и средства проектирования	ИПК-6.1 Знать методы и средства проектирования баз данных	3	13. При проектировании хранилища данных для полной бизнес-модели компании «Ритейл+», включающей операционные процессы (продажи, логистика, закупки) и стратегические метрики (рентабельность, доля рынка), архитектор БД должен определить	13. В полной бизнес-модели производственной компании выделены ключевые бизнес-объекты: «Заказ», «Партия», «Оборудование», «Сотрудник». При концептуальном проектировании БД возникает вопрос: как корректно отразить связь	**Повышенный (3-5 минут)**

		компьютерного программного обеспечения			структуру таблиц измерений. Какой подход к моделированию измерения «Продукт» обеспечит максимальную гибкость для кросс-функционального анализа (продажи по категориям, маржинальность по поставщикам, оборачиваемость по складам)? Варианты ответов: А. Создать единую плоскую таблицу dim_product со всеми атрибутами: product_id, name, category, supplier_id, warehouse_id, cost_price, selling_price. Б. Создать иерархическую структуру измерений: dim_product → dim_category → dim_product_group с внешними ключами, а атрибуты поставщика и склада вынести в отдельные измерения dim_supplier, dim_warehouse. В. Денормализовать модель, продублировав атрибуты категорий и поставщиков непосредственно в таблице фактов fact_sales для ускорения запросов. Г. Использовать документную модель (JSONB в PostgreSQL) для хранения всех атрибутов продукта в одном поле с индексацией по ключевым полям. Правильный ответ: Б Объяснение: Иерархическая структура измерений (Б) обеспечивает нормализацию справочников, исключает избыточность данных и поддерживает сквозную аналитику по разным бизнес-функциям. Для полной бизнес-модели критично разделение контекстов: категория	«Сотрудник участвует в производстве нескольких Партий, а одна Партия производится бригадой из нескольких Сотрудников». Какой тип связи и её реализация в реляционной модели соответствует бизнес-семантике и поддерживает аналитику трудозатрат? Варианты ответов: А. Связь «многие-ко-многим» (M:N) между employee и batch с созданием связующей таблицы employee_batch с атрибутами hours_worked, role, shift_date. Б. Связь «один-ко-многим» (1:N) с добавлением внешнего ключа employee_id в таблицу batch, предполагая ответственного за партию. В. Связь «один-к-одному» (1:1) через объединение атрибутов сотрудника и партии в единую таблицу production_unit. Г. Отказ от явной связи, хранение списка сотрудников в текстовом поле workers_list таблицы batch с последующим парсингом при анализе. Правильный ответ: А Объяснение: Связь M:N (А) точно отражает бизнес-семантику коллективного участия сотрудников в производстве партий. Связующая таблица employee_batch с атрибутами hours_worked, role позволяет анализировать трудозатраты на микроуровне (кто, сколько, в какой роли), что невозможно при упрощённых связях 1:N или 1:1 и критично для расчёта себестоимости в полной бизнес-модели.	

					относится к маркетингу, поставщик — к закупкам, склад — к логистике; их выделение в отдельные измерения позволяет строить гибкие срезы без дублирования бизнес-логики в таблице фактов. 14. Название: Проектирование реляционной базы данных для сервисной компании по обслуживанию компьютерной техники. Цель: Разработать полноценную реляционную модель базы данных, охватывающую всю необходимую информацию о клиентах, услугах, оборудовании и сотрудниках компании. Задания: 1. Анализ предметной области Провести исследование предметной области обслуживания компьютеров, проанализировать рабочие процессы компании и классифицировать основные виды предоставляемых услуг. Выделить ключевые объекты данных: клиенты, оборудование, сотрудники, заказы, услуги. 2. Концептуальное проектирование Спроектировать сущность-отношение (ER) модель данных, показывающую связи между объектами. Выбрать типы отношений (1:N, N:M, 1:1) и задать свойства сущностей. 3. Логическое проектирование Преобразовать ER-модель в	14. Название: Проектирование OLAP-куба для маркетингового анализа крупного продуктового ритейлера. Цель: Овладеть навыками проектирования многомерных баз данных, используемых для комплексного анализа больших объёмов данных. Задания: 1. Изучение особенностей предметной области Понимание специфики маркетинга продуктовых сетей, выделение важных факторов, влияющих на продажи (категория товара, регион, сезонность, скидки и акции). Идентифицировать измеряемые показатели (продажи, прибыль, остатки на складе, выручка) и измерения (регион, категория товара, временной интервал). 2. Определение мер и измерений Составить перечень важнейших мер (суммарные продажи, средняя цена единицы товара, доход) и измерений (время, товар, магазин, регион). Решить, какие меры являются ключевыми для принятия	

					реляционную модель данных. Нормализовать модель до 3-й нормальной формы (3NF), устранив повторяющиеся данные и зависимые поля. 4. Физическое проектирование Подобрать физические структуры таблиц, выбрать типы полей, установить первичные и вторичные ключи. Реализовать внешние ключи и ограничить значения, обеспечить консистентность данных с помощью индексов и триггеров. 5. Оптимизация и тестирование Оценить производительность выбранной структуры данных, составить запросы и протестировать быстродействие выборок и модификаций. Сделать вывод о качестве и пригодности предложенной модели для полного охвата деятельности компании.	управленческих решений руководством компании. 3. Проектирование OLAP-куба Разработать куб данных с иерархиями измерений и разрезами для проведения многомерного анализа. Продумать структуру куба с учётом задач оперативного анализа, отчетности и выявления закономерностей. 4. Использование витрины данных (Data Mart) Подготовить витрину данных, агрегированную по важным срезам, облегчающую быстрое извлечение необходимой информации для руководителей. Построить ETL-процедуры для регулярной синхронизации данных из оперативной базы данных в витрину. 5. Представление результатов Оформить отчёт о выполненных работах, включить диаграммы OLAP-куба, показать возможности многомерного анализа. Обосновать значимость предложенной структуры данных для повышения конкурентоспособности компании на рынке.	
	Тема 8. Спецификация функциональных требований к ИС	ПК-7 Способен осуществлять проектирование программных интерфейсов	ИПК-7.1 Знать методы и средства проектирования программных интерфейсов	3	15. При спецификации функциональных требований к системе бизнес-аналитики для онлайн-ритейла определено требование: «Аналитик должен иметь возможность получать агрегированные метрики продаж за произвольный период с разбивкой по регионам, категориям товаров и каналам продаж».	15. В спецификации функциональных требований к ИС указано: «Система должна предоставлять данные о клиентах для сегментации в режиме реального времени с задержкой не более 5 секунд». При проектировании интерфейса интеграции с системой сегментации возникает вопрос выбора	**Повышенный (3-5 минут)**

					Какой формат спецификации эндпоинта API наиболее полно и корректно отразит данное требование с учётом гибкости аналитических запросов? Варианты ответов: А. /api/sales/report — POST-запрос с телом в формате JSON, содержащим структуру фильтров (date_from, date_to, regions[], categories[], channels[]) и список запрашиваемых метрик (revenue, units_sold, avg_check). Б. /api/sales/2024/01/01/to/2024/12/31/by/region/category/channel — GET-запрос с жёстко закодированным периодом и измерениями в пути URL. В. /api/sales — GET-запрос без параметров, возвращающий полный дамп всех продаж за последний месяц в формате CSV для последующей обработки аналитиком вручную. Г. /api/report1, /api/report2, /api/report3 — набор фиксированных эндпоинтов для заранее определённых отчётов («продажи по регионам», «продажи по категориям» и т.д.). Объяснение: Вариант А обеспечивает декларативную спецификацию требований: структура JSON-тела точно отражает бизнес-логику аналитического запроса (произвольный период, множественные измерения, выбор метрик) без жёсткой привязки к конкретным значениям. Такой подход соответствует принципам проектирования аналитических API —	паттерна взаимодействия. Какой подход к проектированию API корректно отразит данное требование к временным характеристикам и обеспечит его верифицируемость? Варианты ответов: А. Синхронный REST API с заголовком X-Expected-Latency: 5000, где сервер обязан возвращать ответ в течение 5 секунд, иначе клиент получает статус 504 Gateway Timeout. Б. Асинхронный паттерн: эндпоинт /api/segmentation/jobs создаёт задачу с возвратом job_id, а клиент опрашивает статус через /api/segmentation/jobs/{id} до завершения обработки. В. Вебхуки (webhooks): система сегментации подписывается на события «новый клиент», «изменение поведения» и получает данные по push-механизму без запросов. Г. Потоковая передача (Server-Sent Events): эндпоинт /api/segmentation/stream устанавливает постоянное соединение и передаёт обновления клиентских данных в реальном времени. Правильный ответ: А Объяснение: Требование «режим реального времени с задержкой ≤5 сек» описывает синхронный сценарий взаимодействия, где аналитик ожидает немедленного результата для принятия решений. Вариант А транслирует бизнес-требование в верифицируемую	

					поддерживает динамические сценарии анализа, минимизирует количество эндпоинтов и позволяет валидировать сложные комбинации фильтров на уровне схемы запроса (например, через OpenAPI Specification). 16. Название: Разработка спецификаций функциональных требований для информационно-аналитической системы прогнозирования спроса на товары массового потребления. Цель: Освоить навыки составления спецификаций функциональных требований, грамотно выражающих ожидания пользователей и владельцев системы. Основные этапы: 1. Анализ предметной области Провести исследование и изучить рынок массовых потребительских товаров, выявить факторы, влияющие на спрос (ценообразование, сезонность, рекламные кампании). Сформулировать главную проблему, которую решает система (предсказание изменения спроса для оптимального заказа товаров). 2. Собеседование с заинтересованными	техническую спецификацию: SLA фиксируется в контракте API (заголовок или документация), а превышение лимита приводит к однозначному статусу ошибки, что позволяет автоматизировать тестирование производительности. Асинхронные паттерны (Б, В, Г) не соответствуют семантике «режима реального времени» для интерактивной аналитики, так как вводят неопределённость времени получения результата. 16. Название: Проектирование технических требований для информационной системы учета энергоэффективности предприятий малого и среднего бизнеса. Цель: Овладеть методом постановки и описания функциональных требований к комплексной автоматизированной системе сбора и анализа данных по энергопотреблению. Этапы выполнения: 1. Постановка проблемы Изучить сферу энергосбережения, выявить проблемы малых и средних предприятий, касающиеся неэффективного расходования энергии. Поставить задачу: создать инструмент для постоянного мониторинга энергопотребления, позволяющий оперативно снижать затраты.	

					сторонами Организовать встречу с менеджерами по снабжению, маркетологами и специалистами по данным. Получить от них информацию о требованиях к точности прогноза, скорости обновления данных и удобстве интерфейса. 3. Классификация требований Разделить требования на функциональные и нефункциональные группы. Функциональные требования включают алгоритм прогнозирования, ввод данных, настройки интерфейса, экспорт отчетов. Нефункциональные требования связаны с производительностью, надежностью, поддержкой разных платформ и масштабируемостью. 4. Спецификация требований Оформить техническое задание в виде документа, содержащее описание функциональных требований и приоритеты их реализации. Пример: Система должна автоматически загружать данные о продажах каждые сутки и выдавать прогнозы спроса на следующую неделю с точностью не менее 90%. 5. Контроль качества спецификации Выполнить проверку правильности и полноты написанных требований, убедиться, что каждая функция имеет четкий смысл и однозначную трактовку. Согласовать готовую спецификацию с	2. Интервью с пользователями Провести встречи с инженерами и экономистами предприятий, выяснить, какая информация необходима для эффективной работы с данными. Уточнить требуемый уровень детализации (час, день, месяц), типы оборудования, чьё потребление важно учитывать. 3. Структура функциональных требований Классифицировать требования на группы: учет показаний приборов, обработка аварийных ситуаций, автоматизация вычислений экономии. Например: Система должна обеспечивать возможность автоматического подключения новых датчиков без перезагрузки основного ПО. 4. Оформление требований Создать спецификацию требований, оформленную в табличной форме с указанием типа требования, важности, срока реализации и исполнителя. Утвердить спецификацию совместно с ответственными лицами предприятия-заказчика.	

					заказчиком и всеми участниками проекта.		
	Тема 9. Структурные и объектные модели предметной области	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	3	17. При анализе предметной области «Электронная коммерция» построены две модели: • Структурная модель (DFD): отражает потоки данных между процессами «Оформление заказа», «Оплата», «Доставка» с хранилищами «Корзина», «Заказы», «Платежи». • Объектная модель (диаграмма классов UML): выделяет бизнес-объекты Customer, Order, OrderItem, Product, Payment с атрибутами и связями. При проектировании операционной БД (OLTP) для обработки транзакций и хранилища данных (OLAP) для анализа продаж необходимо трансформировать эти модели в реляционные структуры. Какой подход к проектированию таблиц корректно учитывает различия между моделями и специфику аналитической нагрузки? Варианты ответов: А. Использовать единую нормализованную схему 3НФ для обеих систем, так как объектная модель уже обеспечивает структурную целостность данных. Б. Для OLTP реализовать нормализованную схему на основе объектной модели (таблицы orders, order_items, products), а для OLAP построить денормализованную модель «звезда» на основе структурной модели потоков данных (факты = события	17. В предметной области «Банковский риск-менеджмент» построена объектная модель с классами Client, Loan, Collateral, Guarantor и связями «один ко многим» между ними. При проектировании БД для анализа кредитного портфеля выявлено требование: «Рассчитывать динамику обеспечения по каждому кредиту с учётом истории изменений залогов (добавление, изъятие, переоценка)». Какой подход к проектированию структуры хранения обеспечит корректное отражение временной семантики из предметной области? Варианты ответов: А. Создать таблицу collateral с полями loan_id, asset_type, value и внешним ключом на loan, обновляя запись при изменении залога (UPDATE). Б. Добавить в таблицу collateral поля valid_from и valid_to для хранения периода действия каждой версии залога (модель Slowly Changing Dimension Type 2). В. Хранить историю изменений в отдельной таблице collateral_audit с полями old_value, new_value, changed_at, не используя её для аналитических расчётов. Г. Отказаться от реляционной модели и использовать графовую БД для отображения динамических связей «кредит — залог».	**Повышенный (3-5 минут)**

					потока: оформление, оплата, доставка). В. Игнорировать структурную модель и проектировать обе БД исключительно по объектной модели, так как она содержит полное описание бизнес-сущностей. Г. Применить документную модель NoSQL для обеих систем, чтобы сохранить иерархию объектов из UML-диаграммы без трансформации. Правильный ответ: Б Объяснение: OLTP-система оптимизирована под транзакции и требует нормализации (объектная модель → 3НФ) для обеспечения целостности операционных данных. OLAP-система оптимизирована под агрегацию, поэтому факты моделируются как события потока данных из структурной модели (оформление = факт продажи, оплата = факт поступления средств), а измерения берутся из объектной модели — это позволяет анализировать динамику бизнес-процессов, а не только статику сущностей.	Правильный ответ: Б Объяснение: Простое обновление записей (вариант А) уничтожает историю изменений, делая невозможным расчёт корректной динамики обеспечения на произвольную дату в прошлом — критически важного сценария для риск-анализа. Модель SCD Type 2 (Б) сохраняет все версии залога с периодами действия, что позволяет строить «срезы» кредитного портфеля на любую дату и точно рассчитывать метрики вроде «покрытие залогом на 01.01.2024», что напрямую отражает временную семантику предметной области в структуре БД.	
			ИПК-6.2 Уметь применять методы и средства проектирования компьютерно		18. Тема: Проектирование информационной системы с использованием объектных моделей предметной области. Цель задания: Разработать объектную модель предметной области автоматизированной информационной	18. Тема: Разработка структурной модели базы данных для предметной области "Управление проектами" Цель задания: Создать структуру реляционной базы данных для автоматизации процессов управления проектами в проектной организации.	**Повышенный (3-5 минут)**

			го программного обеспечения, структур данных, баз данных, программных интерфейсов		системы управления персоналом предприятия, включая описание классов объектов, атрибутов и связей между ними. Задания: 1. Опишите основные объекты, процессы и отношения, характерные для системы управления персоналом организации. 2. Разработайте классы и укажите их атрибуты, соответствующие следующим сущностям: Сотрудник Подразделение Должность Компетенция Контракт Навык Отчет сотрудника 3. Установите связи между классами согласно отношениям между объектами в реальной жизни. Определите типы ассоциаций (например, многие-к-многим, один-к-одному). 4. Создать диаграмму классов с указанием всех классов, атрибутов и связей. Используйте нотацию UML. 5. Написать пример реализации одного из классов (например, "Сотрудник") на выбранном вами языке программирования (Python, JavaScript, C#). Пример должен включать конструктор, свойства и метод отображения сведений о сотруднике.	Задания: 1. Анализ требований и определение сущности. Проведите анализ бизнес-процессов и определите ключевые сущности и их атрибуты для хранения информации о проектах, участниках проектов, ресурсах проекта, сроках исполнения этапов и статусах выполнения работ. 2. Создание ER-диаграммы. Нарисуйте сущность-отношение (ER) диаграмму с описанием сущностей, ключевых полей, взаимосвязей и типов данных каждого атрибута. 3. Нормализация схемы базы данных. Привести схему базы данных к третьей нормальной форме (3NF). 4. SQL-запросы. Составьте SQL-запросы для выполнения типичных операций над базой данных, таких как создание таблиц, вставка записей, выборка данных по критериям и обновление статусов проектных задач.	
	Тема 10.	ПК-7	И П К - 7 . 1	4	19. В фазе уточнения (Elaboration)	19. При применении Унифицированного	**Повышенны**

	Унифицированный процесс разработки	Способен осуществлять проектирование программных интерфейсов	Знать методы и средства проектирования программных интерфейсов		Унифицированного процесса разработки для системы бизнес-аналитики необходимо зафиксировать архитектурно значимые решения по проектированию API. Какие из перечисленных действий обязательны на этой фазе для обеспечения стабильности контракта интерфейсов при последующих итерациях? Варианты ответов (выберите 2–3 правильных): А. Разработка полного технического задания на все эндпоинты с указанием точных форматов ответа для каждого атрибута аналитической метрики. Б. Определение архитектурного стиля API (REST, GraphQL, gRPC) и принципов версионирования (в пути, заголовках, параметрах). В. Создание «скелета» критических эндпоинтов (например, /api/analytics/aggregate) с заглушками и валидацией параметров без реализации бизнес-логики. Г. Фиксация базового набора измерений и метрик, которые будут доступны через API (например: выручка, конверсия, период, регион), с описанием их семантики в глоссарии. Д. Написание нагрузочных тестов для всех эндпоинтов с эмуляцией 10 000 одновременных запросов от аналитиков. Правильные ответы: Б, В, Г Объяснение: Фаза уточнения направлена на снижение архитектурных рисков: выбор стиля и	процесса к проекту создания единой платформы аналитики для розничной сети выделены дисциплины «Требования» и «Анализ и проектирование». Какие практики проектирования программных интерфейсов корректно распределены между этими дисциплинами в рамках одной итерации фазы конструирования? Варианты ответов (выберите 2 правильных): А. Дисциплина «Требования»: описание пользовательских сценариев использования API («Аналитик формирует отчёт по продажам за квартал с разбивкой по категориям»); Дисциплина «Анализ и проектирование»: проектирование структуры запроса эндпоинта /api/sales/aggregate с параметрами period=Q1_2024, dimensions=["category"]. Б. Дисциплина «Требования»: выбор инструмента для документирования API (Swagger/OpenAPI); Дисциплина «Анализ и проектирование»: написание спецификации OpenAPI 3.0 для всех эндпоинтов. В. Дисциплина «Требования»: определение нефункционального требования «Время ответа эндпоинта агрегации ≤ 3 сек при объёме данных 100 млн записей»; Дисциплина «Анализ и проектирование»: выбор стратегии кэширования и индексирования для выполнения данного требования. Г. Дисциплина «Требования»:	**й (3-5 минут)**

					версионирования (Б) задаёт основу стабильности контракта, «скелет» эндпоинтов (В) позволяет проверить интегрируемость интерфейсов до реализации логики, а фиксация семантики метрик (Г) предотвращает разночтения между аналитиками и разработчиками. Полная спецификация форматов (А) и нагрузочное тестирование (Д) относятся к фазе конструирования, когда архитектура уже утверждена и ведётся детальная реализация.	проектирование схемы базы данных для хранения аналитических метрик; Дисциплина «Анализ и проектирование»: реализация SQL-запросов для эндпоинтов. Д. Дисциплина «Требования»: определение формата аутентификации (OAuth 2.0); Дисциплина «Анализ и проектирование»: реализация механизма выдачи токенов и мидлвара проверки прав доступа к эндпоинтам. Правильные ответы: А, В Объяснение: Дисциплина «Требования» фокусируется на что система должна делать (бизнес-сценарии, метрики SLA), а «Анализ и проектирование» — на как это реализовать технически (структура эндпоинтов, архитектурные паттерны). Вариант А корректно разделяет сценарий использования и техническую спецификацию запроса; вариант В — требование к производительности и техническое решение для его достижения. Варианты Б, Г, Д смешивают инструментальную/реализационную деятельность с определением требований, что нарушает принцип разделения ответственности дисциплин в UP.	
			И П К - 7 . 2 Уметь		20. Тема: Применение унифицированного процесса разработки (Unified Process) для проектирования	20. Тема: Реализация проекта автоматизации учета рабочего времени сотрудников с применением подходов	**Повышенный (3-5 минут)**

			применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов		корпоративной информационной системы управления запасами товаров. Цель задания: Применяя методологию Rational Unified Process (RUP), разработать требования и начальные этапы жизненного цикла проекта по созданию корпоративной информационной системы управления складскими остатками продукции. Задания: 1. Идентификация заинтересованных сторон и целей бизнеса. Определите основных пользователей будущей системы и сформулируйте цели проекта с точки зрения повышения эффективности логистического подразделения. 2. Сбор и анализ требований. Используя технику интервьюирования и анкетирования, соберите функциональные и нефункциональные требования будущих пользователей. Заполните шаблон Use Case для основного сценария взаимодействия пользователей с системой ("создание заказа"). 4. Моделирование вариантов использования (Use Cases). Постройте диаграмму вариантов использования (use case diagram) для описания функциональности системы, включив сценарии обработки заказов, инвентаризации запасов и выдачи товара клиентам.	унифицированного процесса. Цель задания: Используя методологии RUP и Scrum, спроектировать систему учета рабочего времени сотрудников компании с интеграцией в существующую ERP-систему предприятия. Задания: 1. Планирование итерационного подхода. Спланируйте проект, разделив его на несколько коротких итераций (sprints). Определите сроки каждой итерации и поставленные цели. 2. Исследование требований и разработка бэклога продукта. Совместно с заказчиком разработайте перечень функциональных и нефункциональных требований. Оформите их в виде элементов бэклога (Product Backlog Items). 3. Формулировка user stories. Формализуйте конкретные потребности пользователей системы в виде историй пользователей (User Stories), каждая из которых должна отвечать на вопросы: кто заказчик, какую функциональность ожидает увидеть и зачем эта функциональность необходима? 4. Выбор инструментов и технологий. Оцените доступные технологии и выберите наиболее подходящие инструменты для реализации проекта (например, фреймворки Python/Django, Node.js + MongoDB).	

					5. Архитектура решения. Разработайте архитектурную концепцию системы, используя инструмент анализа рисков (Risk Management Framework) и шаблоны проектирования (Design Patterns). Выберите подходящую технологию и инструменты для разработки веб-приложения. 6. Документирование результатов этапа. Подготовьте отчет по итогам проделанной работы, содержащий краткое описание всех выполненных шагов и полученных артефактов (документация по требованиям, use cases, архитектура).	5. Итерация разработки. Выполните первую итерацию, реализовав одну или две истории пользователей. Протестируйте полученные результаты и получите обратную связь от заказчика. 6. Отчёт по результатам первой итерации. Докажите успешность выбранного подхода и подтвердите правильность решений по развитию проекта, представив отчёт с результатами проведённой итерации.	
	Тема 11. Унифицированный язык моделирование (UML)	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	4	21. При проектировании хранилища данных для анализа клиентского поведения в розничной сети используется диаграмма классов UML для моделирования измерения «Клиент». Какие элементы диаграммы классов корректно отражают специфику аналитического измерения с поддержкой истории изменений атрибутов (например, смена адреса, обновление сегмента лояльности)? Варианты ответов (выберите 2 правильных): А. Ассоциация «один ко многим» между классами Customer и CustomerAddress с атрибутом is_current: boolean в CustomerAddress. Б. Композиция между Customer и CustomerVersion, где каждый экземпляр CustomerVersion содержит полный снимок атрибутов клиента на	21. При проектировании операционной БД и хранилища данных для системы электронной коммерции необходимо выбрать типы диаграмм UML для разных этапов проектирования. Какие соответствия между задачей проектирования и типом диаграммы UML являются методически корректными? Варианты ответов (выберите 3 правильных): А. Моделирование бизнес-сущностей «Заказ», «Товар», «Клиент» и их связей на этапе концептуального проектирования → Диаграмма классов. Б. Описание последовательности шагов преобразования данных из операционной БД в таблицу фактов при загрузке витрины → Диаграмма последовательностей. В. Отображение потоков данных между	**Высокий (5-10 минут)**

					определённую дату. В. Зависимость (dependency) от класса Address к классу Customer с пометкой «<<history>>» в стереотипе. Г. Атрибуты valid_from и valid_to непосредственно в классе Customer с ограничением {temporal} в заметке. Д. Обобщение (наследование) класса Customer от абстрактного класса SCD_Type2_Entity. Правильные ответы: А, Г Объяснение: Вариант А отражает паттерн хранения истории через выделение сущности-истории (CustomerAddress) с признаком актуальности — стандартный подход для атрибутов с частыми изменениями. Вариант Г моделирует медленно меняющееся измерение типа 2 (SCD Type 2) через временные метки непосредственно в классе — корректно для атрибутов с редкими изменениями. Вариант Б (композиция с полным снимком) избыточен и нарушает нормализацию; вариант В (зависимость) не передаёт семантику владения данными; вариант Д (наследование) некорректен, так как SCD Type 2 — это стратегия хранения, а не тип сущности.	источниками (CRM, веб-сайт) и процессами (очистка, агрегация) на этапе анализа требований → Диаграмма деятельности. Г. Моделирование жизненного цикла заказа («создан» → «оплачен» → «отгружен» → «доставлен») с триггерами переходов → Диаграмма состояний. Д. Спецификация контракта REST API для получения аналитических отчётов → Диаграмма компонентов. Е. Визуализация архитектуры системы: веб-сервер, приложение аналитики, СУБД, кэш → Диаграмма развёртывания. Правильные ответы: А, Г, Е Объяснение: Диаграмма классов (А) — основной инструмент концептуального моделирования сущностей и связей, напрямую транслируемый в структуру таблиц. Диаграмма состояний (Г) корректно отражает бизнес-логику жизненного цикла сущностей, влияющую на проектирование атрибутов статуса и триггеров. Диаграмма развёртывания (Е) необходима для проектирования физической архитектуры БД (мастер-реплика, шардинг). Диаграмма последовательностей (Б) фокусируется на взаимодействии объектов во времени, а не на преобразовании данных; для ETL-процессов уместнее диаграмма деятельности. Диаграмма деятельности	

					22. Тема: Проектирование базы данных информационной системы учёта заявок клиентов туристической фирмы с использованием языка UML. Цель задания: Научиться разрабатывать концептуальную модель данных средствами UML и преобразовывать её в физическую модель данных для последующей реализации базы данных. Задания: 1. Анализ предметной области. Описать основные понятия и отношения предметной области «Туризм»: клиент, заявка, тур, отель, экскурсия, рейс и др., определить главные характеристики каждого объекта. 2. Создание ER-модели. Разработать ERD (Entity Relationship Diagram) — диаграмму сущностей и отношений для заданной предметной области, обозначив сущности, атрибуты и связи между ними. 3. Трансформация в модель UML. Преобразовать ER-модель в UML-	(В) может дополнять анализ, но основной инструмент для потоков данных — DFD, а не UML. Диаграмма компонентов (Д) описывает модульную структуру ПО, а не контракт API — для этого используется спецификация OpenAPI. 22. Тема: Использование UML для проектирования информационной системы учета продаж недвижимости. Цель задания: Освоение методов построения UML-диаграмм и способов трансформации UML-моделей в реляционные структуры данных. Задания: 1. Постановка задачи и сбор требований. Описать процессы и объекты предметной области: продавец, покупатель, недвижимость, сделка купли-продажи, договор аренды и т.п.; выявить необходимые операции и запросы к данным. 2. Классовые диаграммы UML. Разработать классовые диаграммы (Class Diagrams) для отражения структуры данных системы учета сделок с недвижимостью. Включить необходимые классы, атрибуты и отношения между ними. 3. Последовательностные диаграммы (Sequence Diagrams). Создать последовательностные диаграммы для	

					классовые диаграммы (Class Diagrams): представить каждую сущность как класс, выделить ассоциации, наследования и композиции между классами. 4. Нормализация и оптимизация структуры БД. Нормализовать структуру данных до 3-й нормальной формы (3NF), привести обоснование преимуществ нормализованной структуры перед исходной моделью. 5. Реализация физической модели. Представить физические таблицы для СУБД MySQL на основании подготовленной модели UML-классов, составить SQL-код для создания таблиц и первичного наполнения тестовыми данными.	базовых сценариев взаимодействия пользователей с системой («оформление сделки», «поиск недвижимости»). 4. Переход к физическим структурам данных. Перевести UML-классы в таблицу реляционной базы данных (MySQL, PostgreSQL), описать поля таблиц, ключи и внешние связи. 5. Генерация SQL-скриптов. Подготовить SQL-команды для создания базы данных и представления основной структуры таблиц на основе полученной UML-модели.	
	Тема 12. Определение требований в рамках унифицированного процесса	ПК-7 Способен осуществлять проектирование программных интерфейсов	И П К - 7 . 1 Знать методы и средства проектирования программных интерфейсов	4	23. В дисциплине «Требования» Унифицированного процесса для системы бизнес-аналитики разработана карта прецедентов. Какие из перечисленных прецедентов корректно специфицируют функциональные требования к API с учётом специфики профиля «Анализ данных» и принципов трассируемости требований? Варианты ответов (выберите 2 правильных): А. «Аналитик получает агрегированные метрики продаж» с основным потоком: «Система возвращает сумму выручки, количество заказов и средний чек за указанный период, сгруппированные по выбранным измерениям (регион, категория)». Б. «Система обрабатывает 10 000	23. При определении требований к API системы аналитики в рамках итерации фазы уточнения (Elaboration) необходимо зафиксировать не только функциональные, но и архитектурно значимые нефункциональные требования. Какие из перечисленных требований следует включить в спецификацию на этом этапе, чтобы обеспечить стабильность контракта интерфейсов при масштабировании? Варианты ответов (выберите 3 правильных): А. «Эндпоинт агрегации должен поддерживать пагинацию результатов с параметрами page и page_size для обработки выборок свыше 10 000 записей». Б. «Все эндпоинты должны использовать	**Высокий (5-10 минут)**

					запросов в секунду» как отдельный прецедент с потоком событий. В. «Аналитик формирует когортный анализ удержания» с альтернативным потоком: «Если в запросе указан параметр cohort_type=date_of_first_purchase, система рассчитывает retention rate по неделям с момента первой покупки». Г. «Разработчик вызывает эндпоинт /api/v1/sales» как прецедент с актёром «Внешняя система». Д. «Система возвращает данные в формате JSON» как отдельный прецедент с постусловием «Ответ содержит поля data, metadata, pagination». Правильные ответы: А, В Объяснение: Прецеденты А и В корректны, так как описывают цели актёра-аналитика в бизнес-терминах (агрегация метрик, когортный анализ) с детализацией потоков, что обеспечивает трассируемость от бизнес-требования к контракту API. Вариант Б описывает нефункциональное требование (производительность), которое фиксируется в спецификации, а не в прецеденте; вариант Г ошибочно делает актёром разработчика вместо конечного пользователя; вариант Д специфицирует техническую деталь реализации, а не бизнес-цель, что нарушает абстракцию уровня прецедентов.	аутентификацию по токену OAuth 2.0 с сроком действия 1 час». В. «Система должна возвращать данные о продажах за 2024 год». Г. «При изменении схемы ответа эндпоинта должна поддерживаться обратная совместимость в течение 6 месяцев через механизм версионирования в заголовке API-Version». Д. «Цветовая схема дашборда должна соответствовать корпоративному стилю». Е. «Время ответа эндпоинта /api/analytics/aggregate при объёме данных до 100 млн записей не должно превышать 5 секунд для 95% запросов». Правильные ответы: А, Б, Г, Е (выбрать любые 3 из 4) Объяснение: Требования А (пагинация), Б (аутентификация), Г (версионирование) и Е (производительность) определяют контракт API и его масштабируемость — их изменение на поздних этапах потребует переделки клиентских приложений. Вариант В описывает конкретные данные, а не поведение интерфейса, и относится к наполнению системы; вариант Д касается UI/UX дашборда, а не программного интерфейса. Архитектурно значимые требования фиксируются на этапе уточнения для снижения рисков изменения контракта в фазе конструирования.	

					24. Тема: Проектирование API-интерфейса сервиса аналитики Big Data с использованием RESTful архитектуры. Цель задания: Получить опыт формулирования требований и проектирования API-сервисов в рамках унифицированного процесса (Rational Unified Process, RUP). Задания: 1. Определите целевую аудиторию и конечных пользователей API. Кто будут основными пользователями вашего API и каковы их потребности? 2. Опишите требуемые возможности API: Какие данные и операции необходимы пользователям (получение статистики по метрикам, фильтрация данных, агрегированные отчёты)? 3. Разработайте спецификацию RESTful API, учитывая лучшие практики проектирования API (REST-конвенции именования ресурсов, HTTP-методы CRUD, форматы передачи данных JSON/XML). 4. Спроектируйте API-ресурсы и маршруты (endpoints) для сервисов аналитики данных. Продемонстрируйте полное покрытие функций, указанных в пункте 2. 5. Оформите документацию API. Предоставьте описание формата	24. Тема: Проектирование GUI-интерфейса интерактивной панели визуализации данных (Dashboard) в рамках унифицированного процесса. Цель задания: Практиковать навыки проектирования графических интерфейсов приложений в процессе сбора и анализа требований. Задания: 1. Собрать требования к интерфейсной части приложения. Провести опрос целевой аудитории и зафиксировать ожидания относительно удобства пользования панелью Dashboard. 2. Проанализировать конкурентные продукты. Найти аналогичные интерфейсы и оценить их сильные стороны и недостатки. 3. Создать wireframes (макеты интерфейса). Использовать программы прототипирования (Figma, Adobe XD) для рисования схем расположения элементов интерфейса и кнопок навигации. 4. Предложить дизайн визуальных компонентов: палитру цветов, шрифты, иконки и элементы управления (таблицы, графики, фильтры). 5. Продемонстрировать взаимодействие с интерфейсом. Предусмотреть анимацию переходов между экранами и подсказки для улучшения юзабилити.	**Высокий (5-10 минут)**

					запросов и ответов, примерами данных и возможных ошибок. 6. Проверьте согласованность API с ключевыми требованиями (надежность, производительность, безопасность, масштабируемость).		
	Тема 13. Анализ структуры в рамках унифицированного процесса	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	5	25. Вопрос: В рамках какой фазы Унифицированного процесса выполняется анализ структуры предметной области для проектирования хранилища данных, и чем принципиально отличается этот анализ от структурного анализа операционной базы данных? Эталон ответа: Анализ структуры для хранилища данных выполняется в фазе уточнения (Elaboration), так как именно здесь закладывается архитектурная основа системы и снижаются ключевые риски, включая выбор модели данных. Для операционной БД (OLTP) анализ структуры ориентирован на нормализацию (3НФ), обеспечение целостности транзакций и минимизацию избыточности — артефакты: диаграмма классов с детализацией атрибутов и связей. Для хранилища данных (OLAP) анализ фокусируется на выделении измерений (устойчивые бизнес-контексты) и фактов (измеримые события), критерий качества — производительность агрегации и гибкость срезов. Артефакты:	25. Вопрос: Почему при анализе структуры в дисциплине «Анализ и проектирование» Унифицированного процесса для систем анализа данных недостаточно ограничиться только диаграммой классов, и какие дополнительные структурные артефакты необходимо разработать? Эталон ответа: Диаграмма классов недостаточна для аналитических систем, так как фокусируется на статической структуре сущностей и не отражает ключевые аспекты анализа: временную динамику, гранулярность фактов и иерархии измерений. Необходимо дополнить анализ тремя артефактами. Во-первых, матрица «измерения-метрики» — определяет, какие агрегаты допустимы для каждой комбинации измерений (например, сумма выручки по регионам допустима, а средняя дата заказа — нет). Во-вторых, спецификация гранулярности фактовой таблицы — фиксирует атомарное событие («продажа товара в заказе»), что критично для корректной агрегации. В-третьих, модель медленно меняющихся измерений (SCD) —	**Высокий (5-10 минут)**

					концептуальная модель «звезда/снежинка» с указанием гранулярности фактов. Принципиальное отличие: OLTP-структуры моделируют текущее состояние бизнес-процессов, OLAP-структуры — историю событий для анализа трендов. Это требует разных подходов к анализу: для OLTP — акцент на связи и ограничения; для OLAP — на семантику измерений и аддитивность метрик.	описывает стратегию хранения истории изменений атрибутов (например, смена сегмента клиента) через диаграмму состояний с указанием триггеров переходов. Эти артефакты обеспечивают проектирование структур, пригодных не только для хранения данных, но и для выполнения сложных аналитических запросов без искажения бизнес-семантики.	
			ИПК-6.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов		26. Тема: Анализ и проектирование структуры информационной системы учебного заведения с использованием метода UML (Unified Modeling Language) Цель задания: Изучить организацию структуры учебной среды вуза и спроектировать базу данных для эффективного управления образовательным процессом, а также создать программу на языке Python для заполнения учебных групп студентами. Порядок выполнения: 1. Проведите анализ структуры учебного заведения: факультеты, кафедры, преподаватели, учебные группы, дисциплины, расписание занятий, успеваемость учащихся. 2. Разработайте общую схему информационных потоков внутри	26. Тема: Проекты базы данных медицинской клиники с использованием унифицированных методов проектирования (UML) Цель задания: Исследовать структуру медицинского учреждения и разработать единую информационно-аналитическую систему управления медицинскими услугами с возможностью автоматического распределения пациентов по врачам и назначения лечения. Порядок выполнения: 1. Изучите организационную структуру медицинского учреждения: врачи, пациенты, отделения, процедуры диагностики и терапии, назначение лекарственных препаратов. 2. Проведите SWOT-анализ текущего состояния информационных систем в клинике и выявите проблемы и	**Высокий (5-10 минут)**

					образовательной среды и определите ключевые элементы данных. 3. Применяйте унифицированный язык моделирования (UML) для составления общей модели предметной области и диаграммы классов (class diagrams). 4. Сформулируйте требования к структуре базы данных, соответствующей предметной области учебного заведения. 5. Рассчитайте объем памяти, необходимой для хранения всей совокупности данных, оценивая среднее количество преподавателей, студентов, предметов и экзаменов. 6. Осуществите проектирование структуры базы данных (создавая таблицы и определяя связи между ними). 7. Создайте простой скрипт на языке Python, позволяющий добавлять новых студентов в учебный класс, получать список текущих учеников, распределять студентов по учебным дисциплинам и формировать расписание занятий. 8. По завершении напишите отчет, поясняющий ваши выводы и принятые проектные решения.	перспективы развития. 3. Создайте модель предметной области с помощью UML-диаграмм (включая диаграммы классов, диаграммы последовательности действий, компонентные диаграммы). 4. Разработайте техническое задание на разработку единой информационной системы управления пациентами и медицинским обслуживанием. 5. Определите структуру реляционной базы данных для регистрации пациентов, врачей, назначаемых процедур и медикаментов. 6. Сделайте расчет объема данных, хранящихся в каждом элементе структуры базы данных, исходя из средней численности персонала и количества приемов ежедневно. 7. Разработайте простой web-интерфейс для администраторов клиники, обеспечивающий регистрацию нового пациента, ведение медицинских карт и автоматический подбор врача на основе загруженности специалиста. 8. Проверьте работу системы путем тестирования отдельных модулей и оценки производительности.	
	Тема 14. Проектирование в рамках унифицированного процесса	ПК-7 Способен осуществлять проектирование программных интерфейсов	И П К - 7 . 1 Знать методы и средства проектирования программных интерфейсов	5	27. Почему проектирование контракта программного интерфейса (спецификация эндпоинтов, форматов данных, правил версионирования) должно быть завершено к концу фазы уточнения (Elaboration) Унифицированного процесса, а не отложено до фазы конструирования?	27. Как итеративная природа Унифицированного процесса влияет на подход к проектированию программных интерфейсов для аналитических систем? В чём состоит баланс между ранним утверждением контракта (для стабильности) и гибкостью при уточнении требований к метрикам и	**Высокий (5-10 минут)**

					Эталон ответа: Контракт API является архитектурно значимым решением, так как определяет точки интеграции между компонентами системы и внешними потребителями (дашборды, отчётные инструменты). Его изменение после начала массовой разработки клиентских приложений ведёт к каскадному переделыванию кода и нарушению обратной совместимости — критический риск для аналитических систем, где интерфейсы потребляются десятками пользователей-аналитиков. Фаза уточнения предназначена именно для снижения таких рисков: к её завершению должна быть подтверждена жизнеспособность архитектуры через прототипирование ключевых сценариев (например, агрегация данных за период). Для аналитики особенно важно зафиксировать семантику метрик и измерений в контракте на этом этапе, так как их изменение (например, переопределение «выручки» с НДС на без НДС) делает исторические данные несопоставимыми. Версионирование как механизм обратной совместимости также проектируется в фазе уточнения — это архитектурное решение, а не деталь реализации. Отсрочка проектирования контракта до фазы конструирования превращает архитектурный риск в операционную проблему, требующую дорогостоящей миграции клиентских приложений.	измерениям? Эталон ответа: Итеративность UP требует разделения контракта API на стабильные и эволюционирующие части. На ранних итерациях фазы уточнения фиксируется «каркас» интерфейса: структура запроса (параметры фильтрации по периоду, измерениям), формат ответа (вложенная структура {"data": [...], "metadata": {...}}), правила версионирования и аутентификации. Эти элементы обеспечивают стабильность интеграции клиентских приложений. Гибкость достигается через параметризацию содержимого: список доступных метрик и измерений выносится в отдельный эндпоинт /api/metadata/schema, а не жёстко кодируется в контракте основных эндпоинтов. При добавлении новой метрики (например, «коэффициент удержания») изменяется только содержимое справочника метаданных, а контракт эндпоинта агрегации /api/analytics/aggregate остаётся неизменным. Для критичных изменений схемы применяется стратегия параллельного развёртывания версий (v1 и v2 работают одновременно 6 месяцев). Такой подход соответствует духу итеративной разработки: бизнес-требования к метрикам уточняются в каждой итерации фазы конструирования, но технический контракт взаимодействия остаётся предсказуемым, что критично для аналитиков, строящих отчёты на	

			ИПК-7.2 Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов		28. Тема: Проектирование информационной системы CRM (Customer Relationship Management) с использованием принципов унифицированного процесса (RUP). Цель задания: Применение методики RUP для проектирования системы управления взаимоотношениями с клиентами для малого бизнеса. Порядок выполнения: 1. Проведите анализ потребностей клиента в управлении взаимодействием с потребителями (записи звонков, регистрация обращений, управление заказами, отслеживание лояльности клиентов). 2. Разработайте диаграмму вариантов использования (Use Case Diagram) и перечислите основные функции и сценарии поведения системы. 3. Создайте UML-диаграмму классов (Class Diagram), выделив основные сущности и отношения между ними. 4. Сформируйте модель состояний системы (State Machine Diagram), описывая поведение системы в зависимости от стадий взаимодействия с клиентом. 5. Определите требования к интерфейсу пользователя и проекту архитектуры,	основе стабильных интерфейсов. 28. Тема: Проектирование и реализация онлайн-магазина электронной коммерции в рамках унифицированного процесса. Цель задания: Изучение и применение методик RUP для проектирования инфраструктуры электронного магазина с управлением товарами, корзиной покупок и платежами. Порядок выполнения: 1. Определите круг заинтересованных лиц (клиенты, продавцы, администрация сайта) и изучите их нужды и пожелания относительно функционала магазина. 2. Постройте Use Case Диаграмму, подробно раскрывающую сценарий покупки товара, добавления в корзину, оформления заказа и оплаты. 3. Разработайте Class Diagram для идентификации классов и их взаимоотношений, отражающих сущность предметной области. 4. Разработайте Activity Diagram, показывающий последовательность действий при покупке товара и обработке заказа. 5. Выявите требования к архитектуре и инфраструктуре, включая серверное оборудование, программное обеспечение и хранилище данных.	**Высокий (5-10 минут)**

					включая модульную структуру и механизмы интеграции с внешним оборудованием (телефон, e-mail, мессенджеры). 6. Реализуйте прототип простейшего UI-интерфейса для ввода и вывода данных (на примере любого инструмента проектирования интерфейсов). 7. Настройте инфраструктуру развертывания приложения, спланировав шаги установки сервера и базы данных. 8. Проверьте корректность выполненного проектирования и напишите итоговый отчет, содержащий выводы и рекомендации по дальнейшей разработке.	6. Создайте макет страницы интерфейса пользователя, используя инструменты прототипирования дизайна (например, Figma, Sketch). 7. Определите базовые модули приложения и способы их взаимодействия, включая безопасность и обработку транзакций. 8. Проверить выполнение требований, провести тестирование интерфейса и архитектуры, написать подробный отчет о результатах исследования и проектирования.	
	Тема 15. Документирование процесса	ПК-6 Способен применять методы и средства проектирования компьютерного программного обеспечения	ИПК-6.1 Знать методы и средства проектирования баз данных	5	29. Почему в проектировании баз данных для аналитических систем (хранилищ данных, витрин) документирование бизнес-семантики метрик и измерений критически важно на этапе концептуального проектирования, и чем грозит отсутствие такого документа при эксплуатации системы? Эталон ответа: Документирование бизнес-семантики (глоссарий метрик и измерений) критично, так как аналитические системы потребляются пользователями из разных подразделений, имеющими разное понимание терминов. Например, метрика «выручка» может трактоваться как сумма заказов (без учёта возвратов), как сумма оплаченных заказов или как сумма отгруженных товаров. Без единой	29. Какие особенности документирования процесса проектирования баз данных обусловлены итеративной природой Унифицированного процесса, и почему «живая документация» предпочтительнее единовременно созданного технического задания для аналитических систем? Эталон ответа: Итеративная природа Унифицированного процесса требует, чтобы документация эволюционировала параллельно с моделью данных: на ранних итерациях фиксируется концептуальная модель с ключевыми измерениями и фактами, на последующих — уточняется гранулярность, добавляются атрибуты измерений, специфицируются правила	**Высокий (5-10 минут)**

					спецификации в документе «Бизнес-глоссарий» аналитики строят отчёты на основе разных интерпретаций, что приводит к противоречивым выводам и ошибкам в принятии решений. На этапе концептуального проектирования фиксация семантики позволяет корректно спроектировать структуру измерений и аддитивность фактов. При отсутствии документации при изменении источников данных (например, переход на новую платёжную систему) невозможно определить, изменилась ли бизнес-логика расчёта метрики или только техническая реализация. Это ведёт к «размыванию» доверия к данным, росту трудозатрат на поддержку и невозможности аудита исторических отчётов — ключевые риски для зрелости аналитической культуры организации.	обработки медленно меняющихся измерений (SCD). «Живая документация» (например, спецификация в Confluence или DataHub с версионированием) позволяет отслеживать историю изменений схемы и обоснование архитектурных решений. Для аналитических систем это критично: при добавлении нового канала продаж необходимо документировать не только техническое изменение (новое значение в dim_channel), но и бизнес-контекст («маркетплейс Яндекс.Маркет»), правила атрибуции продаж и влияние на расчёт метрик. Единовременное ТЗ становится устаревшим после первой итерации конструирования, тогда как живая документация служит источником истины для аналитиков, инженеров данных и бизнес-заказчиков. Она обеспечивает трассируемость от бизнес-требования к структуре таблицы и позволяет быстро оценить влияние изменений на существующие отчёты — ключевой фактор снижения рисков при развитии аналитической платформы.	
					30. Тема: Документирование процесса проектирования базы данных для информационной системы учета товарных остатков склада торговой сети. Цель задания: Закрепить навыки документирования процесса проектирования и разработки	30. Тема: Документирование процесса проектирования базы данных службы такси для крупного города. Цель задания: Приобрести навыки полного документирования процесса проектирования базы данных на примере	**Высокий (5-10 минут)**

					базы данных с целью оптимизации учета товарных остатков на складе крупной розничной сети. Задания: 1. Исследуйте структуру предметной области: проанализируйте поток данных, происходящих в учетной системе торговых точек, особенности работы склада и роль базы данных в этом процессе. 2. Разработайте логическую модель данных с использованием диаграмм Entity-Relationship (ERD), описывающих сущности, атрибуты и связи между ними. 3. Произведите нормализацию структуры данных до 3-й нормальной формы (3NF). 4. Разработайте физический проект базы данных, детализируя представление таблиц, индексов, ключей и ограничений целостности. 5. Создайте документацию, содержащую описание процесса проектирования, требования к базе данных, используемым инструментам и технологиям, логическую и физическую модели данных, инструкции по установке и эксплуатации. 6. Охарактеризуйте риски и угрозы для устойчивости и безопасности базы данных, разработайте меры по их минимизации. 7. Организуйте контроль качества документа: обеспечьте наличие рецензии	конкретной информационной системы — служб такси крупного российского города. Задания: 1. Подробно исследуйте специфику функционирования диспетчерской службы такси: изучение маршрутов, пассажиров, водителей, автомобилей, тарифов и платежей. 2. Разработайте концептуальную модель данных, использовав подход "сущность—атрибут—связь" (ER), детально отразив все значимые компоненты системы. 3. Приведите структуру данных к третьей нормальной форме (3NF). 4. Представьте физическое проектирование базы данных, описав структуру таблиц, индексы, первичные и внешние ключи, правила контроля целостности данных. 5. Оформите техническую документацию, охватывающую вводную часть, требования к системе, модели данных, порядок инсталляции и настройки базы данных, инструкцию администратора и оператора. 6. Оцените потенциальные риски и уязвимости системы, предусмотрите мероприятия по обеспечению надежности и безопасности данных. 7. Организуете процедуру проверки и утверждения документа: ознакомьтесь с комментариями коллеги, скорректируйте текст, зафиксируйте новую версию	

					коллег и руководителя проекта, внесите исправления, обновите версию документа.	документа.	
	Тема 16. Организация и планирование работ	ПК-7 Способен осуществлять проектирование программных интерфейсов	И П К - 7 . 1 Знать методы и средства проектирования программных интерфейсов	2	31. Расположите этапы планирования проектирования аналитического API в правильной последовательности в рамках итерации Унифицированного процесса: А. Спецификация контракта эндпоинта: параметры фильтрации, структура ответа, правила пагинации Б. Сбор бизнес-требований: определение метрик (выручка, конверсия) и измерений (период, регион, категория) В. Оценка трудозатрат на реализацию эндпоинта и интеграцию с источниками данных Г. Анализ источников данных: выявление систем-доноров и их текущих интерфейсов экспорта Д. Формирование бэклога спринта: включение задачи «Реализовать эндпоинт /api/analytics/aggregate» с критериями приёмки Правильная последовательность: Б → Г → А → В → Д 32. Тема: Планирование и организация работ по проектированию мобильного приложения для анализа спортивных показателей спортсменов. Цель задания: Организовать и спрогнозировать деятельность команды разработчиков,	31. Расположите действия при организации работ по внедрению версионирования API для системы бизнес-аналитики в порядке их выполнения: А. Разработка механизма параллельной поддержки версий v1 и v2 в течение 6 месяцев Б. Уведомление потребителей API (аналитиков, владельцев дашбордов) о предстоящих изменениях В. Фиксация стратегии версионирования в архитектурной документации (в пути URL: /api/v1/...) Г. Анализ воздействия: выявление всех клиентских приложений, использующих текущую версию API Д. Проведение пилотного тестирования новой версии с группой опытных аналитиков Правильная последовательность: В → Г → Б → Д → А 32. Тема: Организация и планирование работ по созданию веб-портала по изучению иностранного языка. Цель задания: Изучить принципы организации труда и планирования проекта по созданию образовательного портала,	**Базовый (1-3 минуты)**

					занимающихся созданием мобильного приложения для мониторинга и анализа спортивной активности пользователей. Последовательность выполнения: 1. Проведите исследование рынка аналогичных продуктов и определите требования к мобильному приложению. 2. Составьте план-график работ по этапам разработки приложения, распределяя роли среди членов команды (аналитик, дизайнер, фронтенд-разработчик, бекенд-разработчик, тестировщик). 3. Распределите обязанности и временные рамки выполнения каждого этапа, создав детализированный сетевой график работ. 4. Разработайте бюджет проекта, учитывающий затраты на персонал, закупку оборудования и лицензий, хостинг и продвижение приложения. 6. Зафиксируйте риски, влияющие на успех проекта, и разработайте стратегию их минимизации. 6. Сформируйте финальный отчет, включающий планы-графики, распределение обязанностей, оценку бюджета и описание стратегии риск-менеджмента.	предназначенного для изучения английского языка. Последовательность выполнения: 1. Проанализируйте рынок образовательных платформ и создайте техническое задание на разработку веб-сайта, включающее функциональность, интерфейс и требования к дизайну. 2. Разработайте сетевой график проекта, включающий подготовку технических документов, проектирование архитектуры, написание кода, тестирование и внедрение. 3. Назначьте исполнителей и руководителей команд, установите сроки выполнения задач и ожидаемые результаты. 4. Определите стоимость проекта, принимая во внимание оплату услуг подрядчиков, покупку лицензии на использование CMS-платформы и рекламные расходы. 5. Прогнозируйте вероятные препятствия и трудности, возникающих в ходе реализации проекта, разработайте контрмеры и планы резервных действий. 6. Подготовьте полный пакет документации, включающей план-график, бюджет, должностные инструкции и регламент действий при возникновении форс-мажорных обстоятельств.	
	Тема 17. Оценка стоимост	ПК-6 Способен применять	ИПК-6.1 Знать методы и средства	2	33. Расположите этапы оценки трудозатрат на проектирование витрины данных для аналитики продаж в	33. Расположите действия при экспертной оценке стоимости проектирования хранилища данных	**Базовый (1-3 минуты)**

	и разработки программного продукта	методы и средства проектирования компьютерного программного обеспечения	проектирования баз данных		правильной последовательности: А. Расчёт трудозатрат на проектирование измерений: нормализация справочников, реализация SCD Type 2 для клиентских сегментов Б. Определение функционального размера: подсчёт логических файлов данных (измерения «Клиент», «Товар», «Время») и групп транзакций (агрегация продаж) В. Корректировка оценки с учётом сложности источников: интеграция 5 разнородных систем (CRM, ERP, веб-аналитика) с разными форматами выгрузки Г. Формирование итоговой сметы: умножение функционального размера на норматив трудозатрат (20 чел.-часов на функциональную точку) с учётом корректирующих коэффициентов Д. Анализ требований: выделение бизнес-метрик (выручка, конверсия) и измерений (период, регион, категория) из спецификации заказчика Правильная последовательность: Д → Б → А → В → Г	методом «планирование покер» в порядке их выполнения: А. Коллективное обсуждение расхождений: архитектор БД поясняет сложность реализации истории изменений сегментов клиентов, аналитик — требования к гранулярности фактов Б. Индивидуальная оценка каждым экспертом: присвоение «очков сложности» (по шкале Фибоначчи) задаче «Проектирование схемы витрины маркетинговой аналитики» В. Формирование итоговой оценки: медиана индивидуальных оценок после достижения консенсуса умножается на норматив скорости команды (15 чел.-часов на 1 очко) Г. Декомпозиция задачи: выделение подзадач «концептуальное моделирование», «логическое проектирование измерений», «спецификация ETL-потоков» Д. Предоставление контекста: описание бизнес-требований (когортный анализ, атрибуция) и ограничений (срок 3 месяца, 2 источника данных) Правильная последовательность: Д → Г → Б → А → В	
			ИПК-6.2 Уметь применять методы и средства		34. Тема: Расчет стоимости разработки корпоративной информационной системы управления финансами. Цель задания:	34. Тема: Оценка стоимости разработки веб-сервиса для анализа больших данных (Big Data). Цель задания:	

			проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов		Оценить предварительную стоимость разработки специализированной информационной системы для бухгалтерского учета и финансового менеджмента небольшой коммерческой организации. Порядок выполнения: 1. Проведите анализ требований и особенностей предметной области финансовой отчетности. 2. Определите объемы предстоящих работ по каждому этапу жизненного цикла разработки программного обеспечения (проектирование, кодирование, тестирование, внедрение, сопровождение). 3. Составьте калькуляцию затрат, учитывая трудозатраты сотрудников разных категорий (аналитики, разработчики, дизайнеры, тестировщики), необходимое аппаратное и программное обеспечение, аренду помещений и прочие накладные расходы. 4. Примените один из методов стоимостной оценки разработки (метод COCOMO II, метод аналогий, Delphi-метод и т.д.) и рассчитайте приблизительные затраты на весь цикл разработки. 5. Выделите факторы риска, способные повлиять на увеличение затрат (непредвиденная сложность проекта, задержки в сдаче промежуточных этапов, изменение требований	Рассчитать ориентировочную стоимость создания веб-сервиса для предварительной обработки и анализа больших массивов данных. Порядок выполнения: 1. Исследование существующих аналогов на рынке, выявление основных характеристик и возможностей подобных решений. 2. Сбор требований к будущему сервису: доступность большого объёма хранимых данных, высокая скорость обработки запросов, поддержка масштабируемых вычислений, интеграция с облачными системами хранения. 3. Детальное разделение работ на стадии жизненного цикла разработки ПО: анализ требований, проектирование, разработка, тестирование, релиз и обслуживание. 4. Выбор подходящей методологии оценки стоимости (COCOMO II, Function Point Analysis, Parametric Cost Estimation и т.д.). Расчет трудозатрат и материальных затрат по каждому этапу разработки. 5. Анализ факторов риска и неопределенностей, которые могут увеличить общие затраты на проект. 6. Создание таблицы расчета общей стоимости проекта, оценка выгод и рисков запуска такого сервиса.	

					заказчиков и т.д.). 6. Разработайте сводную таблицу затрат и сделайте вывод о целесообразности реализации проекта.		
	Тема 18. Инструментальные средства разработки программ	ПК-7 Способен осуществлять проектирование программных интерфейсов	И П К - 7 . 1 Знать методы и средства проектирования программных интерфейсов	2	35. Расположите этапы применения инструментальных средств при проектировании аналитического API в правильной последовательности от спецификации до публикации документации: А. Генерация интерактивной документации (Swagger UI) из OpenAPI-спецификации для предоставления аналитикам Б. Валидация OpenAPI-спецификации в редакторе (Swagger Editor) на соответствие стандарту и семантическую корректность В. Прототипирование ответов эндпоинтов в Postman с использованием mock-сервера для демонстрации заказчику до реализации бэкенда Г. Создание черновика спецификации в формате OpenAPI YAML/JSON с описанием эндпоинтов, параметров и схем ответов Д. Экспорт коллекции запросов из Postman в формат OpenAPI для синхронизации прототипа со спецификацией Правильная последовательность: Г → Б → В → Д → А	35. Расположите действия при настройке CI/CD-конвейера для обеспечения качества программных интерфейсов в порядке их выполнения: А. Настройка автоматического тестирования контракта API в Postman/Newman при каждом пуше в ветку develop Б. Интеграция линтера Spectral в конвейер для проверки OpenAPI-спецификации на соответствие корпоративным стандартам именования эндпоинтов В. Развёртывание новой версии API на тестовом окружении после успешного прохождения всех проверок Г. Конфигурация Git-репозитория: размещение OpenAPI-спецификации в каталоге /spec и коллекции Postman в /tests Д. Добавление шага автоматической публикации обновлённой документации на портал (Redocly/ReadMe) после мержа в main Правильная последовательность: Г → Б → А → В → Д	**Базовый (1-3 минуты)**
			И П К - 7 . 2		36. Тема: Проектирование и разработка простого клиент-серверного приложения	36. Тема: Автоматизация процессов тестирования и деплоя с использованием	

			Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов		с использованием современных инструментов разработки. Цель задания: Освоить инструменты и техники разработки простых клиент-серверных приложений, используя языки программирования высокого уровня и специализированные библиотеки. Порядок выполнения: 1. Выберите подходящий стек технологий (например, Python+Flask+Django или NodeJS+Express+MongoDB) и определите среду разработки (IDE, редакторы кода, утилиты компиляции и сборки). 2. Поставьте цель разработки — создание простого клиент-серверного приложения для демонстрации списка книг с возможностями просмотра, редактирования и удаления записей. 3. Проектируйте приложение, следуя паттернам MVC/MVVM, используя встроенные инструменты выбранного языка программирования и среды разработки. 4. Реализуйте механизм аутентификации и авторизации пользователей, используя стандартные библиотеки или плагины выбранного стека технологий. 5. Интегрируйте в проект автоматизированные тесты, используя популярные фреймворки тестирования (pytest, Jest, Mocha и др.).	DevOps-инструментов. Цель задания: Познакомиться с инструментами непрерывной интеграции и доставки (CI/CD), применяемыми для ускорения и упрощения разработки программного обеспечения. Порядок выполнения: 1. Проведите анализ и сравнительный обзор популярных инструментов CI/CD (GitLab CI, Jenkins, Travis CI, CircleCI и др.) 2. Создайте простое веб-приложение на языке PHP/Laravel или Ruby on Rails с простым набором страниц и базовой функциональностью. 3. Настройте репозиторий проекта на GitHub/GitLab, настроив триггеры для автоматической сборки и тестирования проекта при коммите изменений. 4. Подготовьте Docker-образ приложения и настройте доставку образа в контейнеры Kubernetes или Docker Swarm. 5. Добавьте возможность автоматически запускать тесты (unit-тесты, end-to-end тесты) при сборке и отправке изменений. 6. Реализуйте систему уведомлений и оповещения о статусе сборки и тестов. 7. После завершения разработки продемонстрируйте работу конвейера сборки и доставки, а также протестированное приложение.	

					6. Разработайте простой пользовательский интерфейс с минимальным дизайном и проверенными элементами доступности. 7. Завершите проект демонстрацией работоспособности приложения и представлением всех используемых инструментов и библиотек.		

Критерии оценивания диагностической работы:

Оценка	Критерии оценивания
отлично	от 90 до 100 % правильно выполненных заданий
хорошо	от 70 до 89 % правильно выполненных заданий
удовлетворительно	от 50 до 69 % правильно выполненных заданий
неудовлетворительно	менее 50 % правильно выполненных заданий