

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Костровец Лариса Борисовна
Должность: директор
Дата подписания: 18.05.2026 10:07:04
Уникальный программный ключ:
6882606104c36dbde41c4ab93a65382136a292d6

Приложение 4
к образовательной программе

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.02 Проектирование сервисно-ориентированных систем
(индекс, наименование дисциплины в соответствии с учебным планом)

09.04.03 Прикладная информатика
(код, наименование направления подготовки/специальности)

Корпоративные информационные системы
(наименование образовательной программы)

очная
(форма обучения)

Год набора 2026

Донецк

Автор(ы)-составитель(и) РПД:

Брадул Сергей Владимирович, канд. экон. наук, доцент кафедры информационных технологий

Заведующий кафедрой:

Брадул Наталья Валерьевна, канд. физ.-мат. наук, доцент, заведующий кафедрой информационных технологий

Рабочая программа дисциплины Б1.В.02 Проектирование сервисно-ориентированных систем одобрена на заседании кафедры информационных технологий факультета государственной службы и управления Донецкого филиала РАНХиГС.

Протокол № 7 от «05» марта 2026 г.

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы
2. Объем и место дисциплины в структуре образовательной программы
3. Содержание и структура дисциплины
4. Типы оценочных материалов, показатели и критерии их оценивания
5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам
6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине
7. Методические материалы по освоению дисциплины
8. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет»
9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения

образовательной программы

Дисциплина (наименование) обеспечивает формирование у обучающихся следующих универсальных, общепрофессиональных и профессиональных компетенций*:

ОТФ/ТФ и реквизиты ИС (при наличии)**	Код компетенции **	Наименование Компетенции **	Код индикатора достижения компетенций **	Наименование индикатора достижения компетенций **	Образовательный результат **
-	ПК-1	Способен проектировать и разрабатывать дизайн ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	ПК-1.1	Разрабатывает структуру программного кода ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	ПК-1.1. 3-1 Знает: языки программирования и работы с базами данных ПК-1.1. У-1 Умеет: кодировать на языках программирования в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС
			ПК-1.2	Верифицирует структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	ПК-1.2. 3-1 Знает: инструменты и методы проектирования и дизайна ИС ПК-1.2. У-2 Умеет: анализировать и структурировать входные данные в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС
			ПК-1.3	Разрабатывает пользовательские интерфейсы ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	ПК-1.3. У-1 Умеет: кодировать на языках программирования в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС

			ПК-1.4	Верифицирует пользовательские интерфейсы ИС относительно требований заказчика к ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	ПК-1.4. У-1 Умеет: кодировать на языках программирования в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС
			ПК-1.5	Устраняет обнаруженные несоответствия в программном коде и в дизайне ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	ПК-1.5. 3-4 Знает: источники информации, необходимой для профессиональной деятельности при выполнении работ и управлении работами по созданию (модификации) и сопровождению ИС ПК-1.5. У-2 Умеет: анализировать и структурировать входные данные в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС

* Дисциплина может формировать компетенцию полностью или частично.

** Должно соответствовать Приложению 1 к образовательной программе

2. Объем и место дисциплины в структуре образовательной программы

Общий объем дисциплины:

4,00 з.е., 144 ак.час

Контактная работа обучающихся с преподавателем по видам учебных занятий:
47 ак. час на контактную работу с преподавателем, из них 18 ак. час на лекционные занятия, 18 ак.час на практические занятия. 79 ак. час на самостоятельную работу обучающихся.

Б1.В.02 Проектирование сервисно-ориентированных систем реализуется во 2-м семестре 1-го курса.

3. Содержание и структура дисциплины

3.1. Структура дисциплины

Очная форма обучения

№ п/п	Наименование тем и (или) разделов	Объем дисциплины, ак.час												Форма текущего контроля успеваемости, промежуточной аттестации	
		ВСЕГО	Контактная работа обучающихся с преподавателем по видам учебных занятий							Самостоятельная работа					
			Период теоретического обучения				Период промежуточной аттестации (сессия)			СРкр	СРэк	СР			
			Занятия лекционного типа		Занятия семинарского типа		ИК	КСР	КЭ				Каттэк		Контроль
			Л	ВЛ	ЛР	ПЗ									
Раздел 1. Современные подходы к интеграции информационных систем															
Тема 1	Виды (уровни) интеграции информационных систем	13	2	0	0	2	0	0	0	0	0	0	0	9	Устный опрос, Контрольное задание
Тема 2	Интеграция на уровнях пользователей,	14	2	0	0	2	0	0	0	0	0	0	0	10	Устный опрос, Контрольное задание

	интерфейсов, приложений и данных													
Тема 3	Сервисно-ориентированная архитектура	14	2	0	0	2	0	0	0	0	0	0	10	Устный опрос, Контрольное задание, КТ №1
Раздел 2. Современные стандарты интеграции														
Тема 4	Форматы представления данных при интеграции (XML, JSON и др.)	14	2	0	0	2	0	0	0	0	0	0	10	Устный опрос, Контрольное задание
Тема 5	Удаленный вызов процедур	14	2	0	0	2	0	0	0	0	0	0	10	Устный опрос, Контрольное задание
Тема 6	Тема 2.1 Simple Object Access Protocol протокол SOAP	14	2	0	0	2	0	0	0	0	0	0	10	Устный опрос, Контрольное задание, КТ №2

Раздел 3. Архитектурный стиль REST														
Тема 7	Основные принципы REST	16	3	0	0	3	0	0	0	0	0	0	10	Устный опрос, Контрольное задание
Тема 8	Модель зрелости сервисов REST	16	3	0	0	3	0	0	0	0	0	0	10	Устный опрос, Контрольное задание, КТ №3
Промежуточная аттестация		29	0	0	0	0	0	0	2	9	0	18	0	Экзамен
Итого		144	18	0	0	18	0	0	2	9	0	18	79	

Используемые сокращения:

Л – лекции - занятия, предусматривающие преимущественную передачу учебной информации обучающимся педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях,).

ВЛ – видео лекции.

ЛР – лабораторные работы.

ПЗ – практические занятия (за исключением лабораторных работ).

ИК – индивидуальные консультации.

КСР – контроль самостоятельной работы

КЭ – консультации перед экзаменом

Каттэк – контактная работа на аттестацию в период экзаменационных сессий

Контроль - контактная работа на аттестацию в период экзаменационных сессий для заочной формы обучения

СРкр – самостоятельная работа на подготовку курсовой работы/ курсового проекта.

СРэк – самостоятельная работа на подготовку к экзамену.

СР – самостоятельная работа в семестре на подготовку к учебным занятиям.

3.2. Содержание дисциплины

Раздел 1. Современные подходы к интеграции информационных систем

Тема 1. Виды (уровни) интеграции информационных систем. ПК-1.2, ПК-1.5.

Понятие интеграции информационных систем, классификация видов интеграции: физическая, логическая, семантическая. Уровни интеграции: уровень данных (общие базы данных, репликация, ETL), уровень приложений (корпоративные сервисные шины, очереди сообщений, API), уровень интерфейсов (порталы, единые точки входа), уровень пользователей (единая аутентификация, единое рабочее пространство). Сравнительный анализ уровней, критерии выбора подходящего уровня в зависимости от задач организации. Роль стандартизации протоколов и форматов при интеграции. Инструменты интеграции: EAI, ESB, message brokers. Студент учится анализировать входные требования к интеграции, структурировать их по уровням, выявлять несоответствия между потребностями бизнеса и выбранным уровнем интеграции.

Тема 2. Интеграция на уровнях пользователей, интерфейсов, приложений и данных. ПК-1.2, ПК-1.5.

Детальное рассмотрение каждого уровня интеграции. Интеграция на уровне данных: общие схемы баз данных, синхронизация, репликация, федеративные запросы, проблемы согласованности и версионности данных. Интеграция на уровне приложений: использование промежуточного слоя (middleware), ESB, очередей сообщений (JMS, RabbitMQ), publish/subscribe модели. Интеграция на уровне интерфейсов: корпоративные порталы, mashup-приложения, единая система виджетов. Интеграция на уровне пользователей: единый вход (SSO), единый профиль, управление доступом. Примеры реализации каждого уровня. Преимущества и недостатки. Студент формирует умение анализировать и структурировать входные данные для выбора конкретных технологий интеграции, а также выявлять и устранять несоответствия между требованиями к интеграции и возможностями выбранного уровня.

Тема 3. Сервисно-ориентированная архитектура. ПК-1.2

Сервисно-ориентированная архитектура (SOA) как подход к реализации информационных систем, основанный на создании и использовании служб (сервисов). Основные свойства сервисов: многократность использования, формальные интерфейсы, независимость от платформы и языка реализации. Принципы SOA: слабая связанность, автономность сервисов, обнаружение сервисов, единая контрактная спецификация. Модели зрелости SOA: модель Sonic Software (5 уровней: эпизодическое применение, архитектурный уровень, бизнес-сервисы, измеряемые бизнес-сервисы, оптимизация) и модель SOE от IBM/BEA. Роль репозитория сервисов, политик безопасности, ESB. Стандарты SOA: WSDL, SOAP, UDDI, WS-BPEL. Студент осваивает инструменты и методы проектирования ИС на основе SOA, учится анализировать и структурировать архитектурные требования заказчика для выбора подходящего

уровня зрелости SOA.

Раздел 2. Современные стандарты интеграции

Тема 4. Форматы представления данных при интеграции (XML, JSON и др.). ПК-1.1, ПК-1.2, ПК-1.5.

Роль форматов данных при интеграции информационных систем. XML (eXtensible Markup Language) как подмножество SGML, цели создания XML: совместимость при передаче слабоструктурированных данных через Интернет. Структура XML-документа: XML-декларация (версия, кодировка, standalone), пролог, элементы (пустые и непустые), атрибуты (правила задания, типы), комментарии, текстовые данные (PCDATA, CDATA). Пространства имён XML: префиксы, URI, наследование, пространство имён по умолчанию, разрешение конфликтов имён. Обработка пробелов в XML, атрибут `xml:space`. Ссылки на символы и сущности: встроенные сущности (`lt`, `gt`, `amp`, `apos`, `quot`), пользовательские сущности. Сравнение XML с JSON: области применения, производительность, читаемость, поддержка типов данных. Другие форматы: YAML, Protocol Buffers, Avro. Инструменты валидации: DTD, XSD. Студент получает знания о языках программирования и работе с данными, учится анализировать и структурировать входные данные для выбора формата представления, выявлять несоответствия в структуре XML-документов относительно требований и схем. Практическое закрепление: индивидуальное задание по разработке XML-документа и DTD для данных астрономической лаборатории.

Тема 5. Удаленный вызов процедур. ПК-1.1, ПК-1.3.

Концепция удалённого вызова процедур (Remote Procedure Call, RPC) как механизма межпроцессного взаимодействия в распределённых системах. Принцип работы: клиент вызывает процедуру, которая выполняется на удалённом сервере, с прозрачной сериализацией параметров и результатов. Проблемы RPC: несоответствие локальной и удалённой семантики вызовов, обработка сетевых ошибок, задержки, различия в представлении данных (big-endian/little-endian). Классические реализации: Sun RPC, DCE/RPC, CORBA, Java RMI, XML-RPC. Роль XML в RPC: сериализация параметров вызова в XML-сообщения, передача по HTTP, десериализация ответа. Сравнение RPC с другими подходами: message-oriented middleware, REST. Современные эволюции RPC: gRPC (с использованием Protocol Buffers), Apache Thrift. Студент получает знания о языках программирования и механизмах вызова процедур, учится кодировать на языках программирования реализации удалённых вызовов.

Тема 6. Simple Object Access Protocol (SOAP). ПК-1.1, ПК-1.3, ПК-1.4.

SOAP как протокол обмена структурированными сообщениями в распределённых системах, основанный на XML. Структура SOAP-сообщения: конверт (envelope), заголовок (header — необязательный, содержит контекстную информацию, маршрутизацию, безопасность), тело (body — обязательное, содержит вызов метода и параметры или ответ/ошибку), блоки fault (обработка ошибок). Стили обмена сообщениями: RPC-style (удалённый вызов процедур), document-style (обмен документами). Привязки SOAP к транспортным протоколам: SOAP over HTTP, SOAP over SMTP, SOAP over JMS. Расширения

WS-*: WS-Security (безопасность), WS-ReliableMessaging (надёжная доставка), WS-AtomicTransaction (транзакции), WS-Addressing (адресация). WSDL (Web Services Description Language) как язык описания SOAP-сервисов: типы, сообщения, порты, привязки, сервисы. Сравнение SOAP с REST: случаи применения, сложность, безопасность, поддержка транзакций. Студент получает знания о языках программирования и форматах обмена, учится кодировать SOAP-сервисы и клиенты, а также верифицировать пользовательские интерфейсы (включая API) на соответствие требованиям заказчика.

Раздел 3. Архитектурный стиль REST

Тема 7. Основные принципы REST. ПК-1.2, ПК-1.5.

REST (Representational State Transfer) как архитектурный стиль распределённых систем, предложенный Роем Филдингом. Шесть основных ограничений (принципов) REST: клиент-серверная архитектура (разделение ответственности), отсутствие состояния (stateless — каждый запрос содержит всю необходимую информацию), кэширование (ответы могут быть помечены как кэшируемые), единообразие интерфейса (унифицированные методы HTTP, идентификация ресурсов через URI, самодостаточные сообщения, гипермедиа как движок состояния приложения), слоистая система (прокси, шлюзы, балансировщики), код по требованию (опционально). Ресурсы и их идентификация через URI. Методы HTTP: GET (безопасный, идемпотентный), POST (создание ресурса), PUT (полная замена ресурса, идемпотентный), PATCH (частичное обновление), DELETE (удаление, идемпотентный). Коды состояния HTTP: 2xx (успех), 3xx (перенаправление), 4xx (ошибка клиента), 5xx (ошибка сервера). Представления ресурсов (XML, JSON, HTML, изображения). HATEOAS (Hypermedia as the Engine of Application State) как ключевое свойство зрелого REST. Сравнение REST с SOAP и RPC. Студент осваивает инструменты и методы проектирования RESTful ИС, учится находить источники профессиональной информации для реализации REST-архитектур.

Тема 8. Модель зрелости сервисов REST. ПК-1.2, ПК-1.5.

Модель зрелости REST сервисов Ричардсона (Richardson Maturity Model) — четыре уровня зрелости RESTful API. Уровень 0: Swamp of POX (Plain Old XML) — один URI, один метод (обычно POST), всё общение через один endpoint, отсутствие использования возможностей HTTP. Уровень 1: Resources — множественные URI для разных ресурсов, но один метод (POST), сервис начинает использовать понятие ресурсов. Уровень 2: HTTP Verbs — полное использование методов HTTP (GET, POST, PUT, DELETE, PATCH) с правильной семантикой, использование кодов состояния HTTP, поддержка идемпотентности. Уровень 3: Hypermedia Controls (HATEOAS) — ответы содержат гиперссылки на возможные действия, сервер управляет состоянием приложения через гипермедиа, клиент не нуждается в предварительном знании всех URI. Критерии перехода между уровнями. Практические примеры API каждого уровня. Связь модели Ричардсона с принципами REST. Оценка зрелости существующих API. Влияние уровня зрелости на сопровождаемость, эволюционируемость и удобство интеграции. Студент учится анализировать и структурировать требования к REST-сервисам, оценивать уровень зрелости по модели Ричардсона, выявлять несоответствия принципам REST и предлагать пути их устранения.

4. Типы оценочных материалов, показатели и критерии оценивания

4.1. Оценочные материалы по дисциплине Б1.В.02 Проектирование сервисно-ориентированных систем входят в состав оценочных материалов по образовательной программе. Совокупность оценочных материалов по всем дисциплинам (модулям) образовательной программы составляет фонд оценочных средств (далее – ФОС). ФОС используется при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся с целью оценивания достижения обучающимися планируемых результатов обучения.

4.2. ФОС разработан как комплекс проверочных заданий различного типа и уровня сложности, включает критерии и шкалы оценивания, а также «ключи» правильных ответов. ФОС формируется как отдельный документ и хранится в электронном виде, доступ к ФОС предоставлен ограниченному кругу лиц.

4.3. Для самостоятельной работы обучающихся при подготовке к текущему контролю успеваемости и промежуточной аттестации в рабочих программах дисциплин размещены типовые проверочные задания, которые можно условно разделить на задания закрытого, комбинированного и открытого типов.

Задания закрытого типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных.

Задания комбинированного типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных и обосновать свой выбор.

Задания открытого типа – это задания, в которых на каждый вопрос должен быть предложен развернутый обоснованный ответ.

В зависимости от типа задания рекомендованы определенная последовательность выполнения и система оценивания выполнения заданий.

4.4. Типы заданий, сценарии выполнения, критерии оценивания

ТИП ЗАДАНИЯ	ИНСТРУКЦИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	КРИТЕРИИ ОЦЕНИВАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных	Прочитайте текст, выберите правильный ответ	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В). 	Ответ считается верным, если правильно указана цифра или буква
Задание закрытого типа на установление соответствия	Прочитайте текст и установите соответствие	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4). 	Ответ считается верным, если правильно указаны цифры или буквы
Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных	Прочитайте текст, выберите правильные ответы	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г). 	Ответ считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)

<p>Задание закрытого типа на установление последовательности</p>	<p>Прочитайте текст и установите последовательность</p>	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты ответа. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135). 	<p>Ответ считается верным, если правильно указана вся последовательность цифр</p>
<p>Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора</p>	<p>Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа</p>	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа. 5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования). 	<p>Ответ считается верным, если правильно указана цифра или буква и приведены корректные аргументы, используемые при выборе ответа</p>
<p>Задание открытого типа с развернутым ответом</p>	<p>Прочитайте текст и запишите развернутый обоснованный ответ</p>	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять суть вопроса. 2. Продумать логику и полноту ответа. 3. Записать ответ, используя четкие компактные формулировки. 4. В случае расчетной задачи, записать решение и ответ 	<p>Ответ считается верным:</p> <ol style="list-style-type: none"> 1. Отсутствие фактических ошибок. 2. Раскрытие объема используемых понятий (полнота ответа). 3. Обоснованность ответа (наличие аргументов). 4. Логическая последовательность излагаемого материала.

4.5. Общая шкала оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся с применением БРС

Итоговая балльная оценка	Традиционная система	Бинарная система	ECTS	
			Для традиционной системы	Для бинарной системы
95-100	Отлично	Зачтено	A	P/ Passed
85-94			B	P/ Passed
75-84	Хорошо		C	P/ Passed
65-74			D	P/ Passed
55-64	Удовлетворительно		E	P/ Passed
0-54	Неудовлетворительно	Не зачтено	F	F/Failed

Соотношение баллов за текущий контроль успеваемости и промежуточную аттестацию, а также повторную промежуточную аттестацию:

Максимальная сумма баллов за текущий контроль успеваемости	Максимальная сумма баллов за промежуточную аттестацию	Максимальная итоговая балльная оценка	Максимальная сумма баллов за повторную промежуточную аттестацию
100 баллов	100 баллов	100 баллов	100 баллов

5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам

5.1. В ходе реализации дисциплины Б1.В.02 Проектирование сервисно-ориентированных систем используются следующие формы текущего контроля успеваемости обучающихся (в том числе, задания к контрольным точкам):

устный опрос, контрольное задание.

Распределение баллов по видам учебной деятельности (БРС)

Раздел/Темы	Формы текущего контроля		КТ
	УО	КЗ	
Р-1. / Т-1	3	5	
Р-1. / Т-2	3	5	
Р-1. / Т-3	3	5	12
Р-2. / Т-4	3	5	
Р-2. / Т-5	3	5	
Р-2. / Т-6	3	5	12
Р-3. / Т-7	3	5	
Р-3. / Т-8	3	5	12
Итого: 100 б	24	40	36

УО – устный опрос;
 ТЗ – тестовое задание;
 КЗ – контрольные задания;
 Д – доклад;
 КТ – контрольные точки.

Критерии оценивания опроса:

Балы	Описание критерия
3	Обучающийся полно излагает материал (отвечает на вопрос), дает правильное определение основных понятий; обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; излагает материал последовательно и правильно с точки зрения норм литературного языка.
2	Обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.
1	Обучающийся обнаруживает знание и понимание основных положений данной темы, но излагает материал неполно и допускает неточности в определении понятий или формулировке правил; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
0	Обучающийся обнаруживает незнание вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал.

0* - в журнал академической группы не выставляется

Критерии оценивания контрольных заданий:

Балы	Описание критерия
5	Обучающимся задание выполнено без ошибок и в полном объеме.
3-4	Обучающимся допущены отдельные ошибки при выполнении задания
1-2	У обучающегося отсутствуют ответы на большинство вопросов задачи, задание не выполнено или выполнено не верно.

0* - в журнал академической группы не выставляется

5.2. Типовые оценочные материалы для текущего контроля успеваемости обучающихся (вне контрольных точек):

Контрольные вопросы для проведения опроса:

Раздел 1. Современные подходы к интеграции информационных систем

Тема 1. Виды (уровни) интеграции информационных систем. ПК-1.2, ПК-1.5.

1. Какие существуют основные виды интеграции информационных систем?
2. Чем физическая интеграция отличается от логической?
3. Что понимается под семантической интеграцией данных?
4. На каких четырех уровнях может осуществляться интеграция информационных систем?
5. Какие технологии используются для интеграции на уровне данных?
6. Что такое ETL и для каких задач интеграции он применяется?
7. Какие критерии выбора уровня интеграции являются наиболее важными?
8. Чем интеграция на уровне приложений отличается от интеграции на уровне интерфейсов?
9. Какова роль стандартизации протоколов при интеграции разнородных систем?
10. Какие проблемы возникают при интеграции на уровне данных в случае использования разных СУБД?

Тема 2. Интеграция на уровнях пользователей, интерфейсов, приложений и данных. ПК-1.2, ПК-1.5.

1. Какие задачи решает интеграция на уровне пользователей?
2. Что такое единая точка входа (Single Point of Entry) в контексте интеграции?
3. Какие технологии применяются для интеграции на уровне интерфейсов?
4. Чем корпоративный портал отличается от обычного веб-сайта?
5. Какую роль выполняет сервисная шина предприятия (ESB) при интеграции на уровне приложений?
6. В чем разница между синхронной и асинхронной интеграцией на уровне приложений?
7. Какие проблемы согласованности данных возникают при интеграции на уровне данных?
8. Что такое федеративные запросы к базам данных?
9. Как решается проблема версионности схем данных при интеграции?
10. Какие преимущества дает использование очередей сообщений (message queues) при интеграции приложений?

Тема 3. Сервисно-ориентированная архитектура. ПК-1.2.

1. Какие основные свойства должны быть у служб (сервисов) в SOA?
2. Чем SOA отличается от традиционного монолитного подхода к построению ИС?
3. Какие пять уровней включает модель зрелости SOA, предложенная Sonic Software?
4. Что характеризует первый уровень зрелости SOA (эпизодическое применение)?

5. Какие подсистемы появляются на четвертом уровне зрелости SOA?
6. Чем отличается третий уровень зрелости SOA от второго?
7. Что такое репозиторий сервисов и политик в SOA?
8. Какие две группы сервисов выделяются на третьем уровне зрелости SOA?
9. В чем заключается автоматическая реакция на события на пятом уровне зрелости SOA?
10. Для какой аудитории предназначена модель зрелости SOE, разработанная IBM и BEA Systems?

Раздел 2. Современные стандарты интеграции

Тема 4. Форматы представления данных при интеграции (XML, JSON и др.). ПК-1.1, ПК-1.2, ПК-1.5.

1. Для каких целей был создан язык XML?
2. Какие разделы обязательно присутствуют в структуре XML-документа?
3. Какие правила необходимо соблюдать при именовании XML-элементов?
4. Чем пустой элемент в XML отличается от непустого?
5. В каких тегах XML могут располагаться атрибуты, а в каких не могут?
6. Что такое пространство имен (namespace) в XML и для чего оно используется?
7. Какие символы запрещены внутри XML-комментариев?
8. Для чего используются разделы CDATA в XML-документах?
9. Какие пять встроенных сущностей определены в XML?
10. Чем DTD отличается от XML Schema (XSD) при описании структуры документа?

Тема 5. Удаленный вызов процедур. ПК-1.1, ПК-1.3.

1. Какова основная идея удаленного вызова процедур (RPC)?
2. Какие проблемы возникают при передаче параметров через сеть в RPC?
3. Чем локальный вызов процедуры отличается от удаленного с точки зрения семантики?
4. Какие известные реализации RPC существовали до появления XML-RPC?
5. Какую роль выполняет сериализация данных в RPC?
6. Какие недостатки присущи классическому RPC по сравнению с message-oriented middleware?
7. Чем современный gRPC отличается от традиционного XML-RPC?
8. Как в RPC обрабатываются сетевые ошибки и таймауты?
9. В чем проблема несовместимости порядка байтов (endianness) при RPC-вызовах?
10. Какие сценарии более подходят для RPC, а какие — для REST?

Тема 6. Simple Object Access Protocol (SOAP). ПК-1.1, ПК-1.3, ПК-1.4.

1. Из каких обязательных и необязательных частей состоит SOAP-

сообщение?

2. Какие транспортные протоколы могут использоваться для передачи SOAP-сообщений?
3. Чем RPC-style SOAP отличается от document-style?
4. Для чего используется блок Fault в SOAP-сообщении?
5. Какие задачи решают расширения WS-Security и WS-ReliableMessaging?
6. Какова роль WSDL при описании SOAP-сервисов?
7. Какие элементы обязательно присутствуют в WSDL-документе?
8. В каких случаях использование SOAP предпочтительнее REST?
9. Как в SOAP реализуется поддержка распределенных транзакций?
10. Какие недостатки SOAP ограничивают его применение в современных веб-сервисах?

Раздел 3. Архитектурный стиль REST

Тема 7. Основные принципы REST. ПК-1.2, ПК-1.5.

1. Какие шесть ограничений (принципов) определяют архитектурный стиль REST?
2. Что означает требование «отсутствие состояния» (stateless) в REST?
3. Как кэширование влияет на производительность REST-сервисов?
4. Что такое единообразие интерфейса в контексте REST?
5. Какие методы HTTP считаются идиempотентными, а какие нет?
6. Какую смысловую нагрузку несут коды состояния HTTP 2xx, 3xx, 4xx, 5xx?
7. Что означает аббревиатура HATEOAS и какое место она занимает в REST?
8. Как происходит идентификация ресурсов в REST-архитектуре?
9. Чем REST-сервис принципиально отличается от SOAP-сервиса?
10. Может ли REST-сервис передавать данные в формате XML, а не в JSON, и будет ли он соответствовать принципам REST?

Тема 8. Модель зрелости сервисов REST. ПК-1.2, ПК-1.5.

1. Сколько уровней включает модель зрелости REST Ричардсона?
2. Чем характеризуется нулевой уровень (Swamp of POX) в модели Ричардсона?
3. Какое ключевое отличие первого уровня зрелости REST от нулевого?
4. Какие возможности HTTP добавляются на втором уровне модели зрелости?
5. Что позволяет реализовать третий уровень зрелости (HATEOAS)?
6. Почему использование только одного метода POST является признаком низкой зрелости REST-сервиса?
7. Какой уровень зрелости считается минимально необходимым для того, чтобы сервис назывался RESTful?
8. Как влияет уровень зрелости на возможность эволюции API без поломки клиентов?

9. Может ли сервис использовать HATEOAS, но при этом неправильно применять методы HTTP — считается ли он соответствующим третьему уровню?

10. Какие практические выгоды даёт переход с первого уровня зрелости на второй для команды разработки и для клиентов API?

Контрольные задания:

Раздел 1. Современные подходы к интеграции информационных систем

Тема 1. Виды (уровни) интеграции информационных систем. ПК-1.2, ПК-1.5.

Задание 1. Крупное производственное предприятие на протяжении 15 лет автоматизировало свою деятельность поэтапно. В результате были внедрены три независимые информационные системы: складская система управления запасами на базе MS SQL Server, бухгалтерская система на Oracle DB, а также CRM-система для работы с клиентами на PostgreSQL. В настоящий момент руководство столкнулось с проблемой: данные в этих системах не синхронизированы, что приводит к ошибкам в планировании закупок и отгрузках. Необходимо организовать обмен данными между всеми тремя системами без их полной замены.

Задачи:

1. Определите, какой уровень интеграции (данных, приложений, интерфейсов или пользователей) наиболее подходит для описанной ситуации.
2. Обоснуйте выбор уровня интеграции, указав не менее трех аргументов, связанных с условиями задачи.
3. Перечислите возможные технологические решения для реализации выбранного уровня и кратко охарактеризуйте каждое.

Задание 2. Холдинг, объединяющий шесть дочерних компаний, принял решение о создании единого корпоративного портала. Каждая дочерняя компания имеет собственные веб-приложения для управления проектами, документооборотом и внутренними коммуникациями. Все приложения разработаны на разных технологических стеках (Java, .NET, Python) и имеют собственные системы аутентификации. Требуется обеспечить сотрудникам любой дочерней компании возможность работать с любым приложением через единый портал без повторного ввода логина и пароля.

Задачи:

1. Определите, какие два уровня интеграции задействованы в описанной задаче.
2. Опишите подробно, как будет организована интеграция на каждом из выбранных уровней.
3. Укажите потенциальные риски и проблемы при реализации такой интеграции, а также способы их минимизации.

Задание 3. При попытке интеграции двух информационных систем — старой учетной системы (Legacy system) и новой аналитической платформы — архитекторы приняли решение организовать промежуточную базу данных. В эту базу старая система будет периодически выгружать данные, а новая система — считывать их оттуда. Однако спустя несколько недель эксплуатации возникли проблемы: данные в аналитической платформе устаревают, возникают конфликты при одновременной записи, а также сложно отследить, какие данные уже были обработаны, а какие нет.

Задачи:

1. К какому уровню интеграции относится выбранное решение с промежуточной базой данных?
2. Приведите два преимущества и два недостатка такого подхода в контексте описанной проблемы.
3. Предложите альтернативный способ интеграции, который позволил бы устранить выявленные недостатки, и объясните, почему он будет эффективнее.

Тема 2. Интеграция на уровнях пользователей, интерфейсов, приложений и данных. ПК-1.2, ПК-1.5.

Задание 1. Финансовая компания, насчитывающая более 2000 сотрудников, эксплуатирует семь различных корпоративных приложений: систему управления проектами, HR-портал, систему электронного документооборота, финансовую систему, систему управления клиентскими отношениями, внутренний мессенджер и корпоративную почту. Каждое приложение имеет собственную базу учетных записей, и сотрудникам приходится запоминать и вводить до семи различных паролей ежедневно. Руководство приняло решение внедрить систему единого входа (SSO) для всех приложений. При этом приложения работают на разных операционных системах (Windows, Linux) и имеют различную архитектуру.

Задачи:

1. К какому уровню интеграции относится SSO? Обоснуйте свой ответ.
2. Опишите последовательность действий пользователя при первом входе в систему с использованием SSO и при последующих входах.
3. Перечислите не менее трех протоколов и стандартов, которые могут быть использованы для реализации SSO в описанной разнородной среде, и кратко поясните назначение каждого.

Задание 2. Крупная логистическая компания разрабатывает корпоративный портал, который должен стать единой точкой входа для всех сотрудников. Портал должен отображать в режиме реального времени информацию из трёх источников: ERP-системы (данные о заказах и отгрузках), CRM-системы (информация о клиентах и контрактах) и системы управления персоналом (данные о доступных сотрудниках и их графике работы). Кроме того, портал должен предоставлять виджеты для быстрого доступа к наиболее часто используемым функциям этих систем. Пользователи портала не должны знать, из

какой именно системы приходят те или иные данные.

Задачи:

1. Определите, на каких двух уровнях интеграции будет строиться описанное решение.
2. Для каждого уровня опишите конкретные технологии и подходы, которые могут быть применены.
3. Сформулируйте три технических требования к корпоративному portalу, вытекающих из условий задачи.

Задание 3. Проектируется интеграция между двумя информационными системами: системой приёма заказов (интернет-магазин) и системой учёта на складе. Руководство требует, чтобы при оформлении заказа в интернет-магазине данные о заказе мгновенно попадали в складскую систему, а информация о наличии товаров на складе обновлялась в интернет-магазине в реальном времени. При этом в ночное время допускается асинхронный режим обмена для выполнения массовых операций. Дополнительное требование: при недоступности складской системы интернет-магазин должен продолжать работу, но с пометкой о возможной задержке подтверждения заказа.

Задачи:

1. Какие два режима интеграции (синхронный и асинхронный) должны быть реализованы в описанной системе? Для каких операций какой режим следует использовать?
2. Какое архитектурное решение (например, очереди сообщений, ESB, прямой HTTP-вызов) следует применить, чтобы обеспечить оба режима работы?
3. Опишите алгоритм обработки заказа в момент недоступности складской системы.

Тема 3. Сервисно-ориентированная архитектура. ПК-1.2.

Задание 1. Банк планирует переход к сервисно-ориентированной архитектуре для своей системы интернет-банкинга. В настоящее время все функции (просмотр баланса, история операций, переводы между счетами, оплата услуг, управление картами, настройки уведомлений) реализованы в одном монолитном приложении. Любое изменение требует остановки системы и переразвёртывания всего приложения. Банк хочет разбить функциональность на независимые сервисы, которые можно будет разрабатывать, развёртывать и масштабировать отдельно друг от друга. При этом критически важные операции (например, переводы) должны иметь более высокий уровень надёжности и безопасности, чем второстепенные (например, изменение темы оформления интерфейса).

Задачи:

1. Какие свойства сервисов в SOA должны быть обеспечены, чтобы удовлетворить требованиям банка? Перечислите не менее четырёх свойств с

пояснениями.

2. Предложите, на какие сервисы логически можно разделить функциональность интернет-банкинга.

3. Как в SOA решается вопрос обеспечения различных уровней надёжности и безопасности для разных сервисов?

Задание 2. Крупная страховая компания прошла аудит своей ИТ-инфраструктуры. Эксперты определили, что система находится на втором уровне зрелости SOA по модели Sonic Software (архитектурный уровень). В компании есть задокументированная архитектура, определены стандарты, внедрены политики безопасности, но при этом бизнес-процессы часто требуют ручного вмешательства, отсутствует автоматический мониторинг эффективности, а интеграция с партнёрами осуществляется через обмен файлами по электронной почте. Руководство хочет перейти на следующий уровень зрелости.

Задачи:

1. Какой уровень зрелости является следующим за архитектурным по модели Sonic Software? Опишите его основные характеристики.

2. Какие новые подсистемы и технологии необходимо внедрить для перехода на этот уровень?

3. Что должно измениться в организации интеграции с партнёрами при переходе на третий уровень зрелости?

Задание 3. Разрабатывается сервисная шина предприятия (ESB) для государственной организации, которая взаимодействует с тремя внешними ведомствами. Каждое ведомство предоставляет свои веб-сервисы, но использует разные протоколы: одно — SOAP/HTTP, другое — REST/JSON, третье — устаревший формат на основе файлового обмена по FTP. Организация хочет, чтобы её внутренние приложения вызывали все внешние сервисы через единый интерфейс с использованием только протокола SOAP, не заботясь о различиях в протоколах внешних систем.

Задачи:

1. Какую функцию ESB следует использовать для решения описанной задачи?

2. Опишите архитектуру решения: какие компоненты должны входить в состав ESB и как они будут взаимодействовать.

3. Какие форматы данных и протоколы необходимо будет преобразовывать при прохождении запросов через ESB? Приведите конкретные примеры преобразований.

Раздел 2. Современные стандарты интеграции

Тема 4. Форматы представления данных при интеграции (XML, JSON и др.). ПК-1.1, ПК-1.2, ПК-1.5.

Задание 1. Астрономическая лаборатория ведёт наблюдения за планетами Солнечной системы. Данные наблюдений необходимо передавать между несколькими исследовательскими центрами. Требования к формату передачи: поддержка группировки планет по различным признакам (например, «планеты земной группы», «газовые гиганты», «карликовые планеты»), указание названий планет на двух языках (национальном и английском), а также возможность указывать физические характеристики (масса, радиус, плотность, период обращения вокруг оси, расстояние до Солнца) с произвольными единицами измерения. Некоторые характеристики могут отсутствовать для отдельных планет. Кроме того, важно сохранить возможность валидации структуры документа перед передачей.

Задачи:

1. Разработайте XML-документ для передачи данных о трёх планетах (на ваш выбор), распределённых по двум группам. Документ должен соответствовать всем перечисленным требованиям.
2. Разработайте DTD-схему для созданного XML-документа, обеспечивающую проверку структуры.
3. Проверьте созданный XML-документ на валидность относительно DTD и перечислите все элементы и атрибуты, которые были проверены.

Задание 2. Компания разрабатывает интеграционное решение для обмена данными между двумя системами. Первая система отправляет данные в формате XML с использованием вложенных элементов и атрибутов. Вторая система способна принимать только данные в формате JSON. При этом структуры XML и JSON должны быть логически эквивалентны. Пример исходного

XML: `<order id="123"><customer>Иванов</customer><items><item count="2">Товар А</item><item count="1">Товар Б</item></items></order>`

Требуется преобразовать XML в JSON и наоборот.

Задачи:

1. Преобразуйте приведённый XML в соответствующий JSON-документ.
2. Укажите, как в преобразованном JSON были обработаны атрибуты (id, count) и вложенные элементы.
3. Напишите общее правило преобразования произвольного XML в JSON для данного интеграционного сценария.

Задание 3. При разборе XML-документа парсер выдал ошибку: символы <, >, & в текстовом содержимом элемента не были экранированы. Например, в документе присутствует фрагмент:

`<description>Важно: цена < 1000 & количество > 0</description>`

Кроме того, разработчик хочет вставить внутрь XML-документа фрагмент JavaScript-кода, содержащий символы &, <, >, ", ', и не хочет экранировать их

поштучно. Также требуется определить сущность `&company`; для многократного использования названия организации во всём документе.

Задачи:

1. Исправьте приведённый ошибочный фрагмент XML, используя встроенные сущности XML.
2. Предложите способ вставки JavaScript-кода без ручного экранирования каждого спецсимвола. Продемонстрируйте на примере.
3. Опишите синтаксис объявления и использования пользовательской сущности `&company`; в DTD и в XML-документе.

Тема 5. Удаленный вызов процедур. ПК-1.1, ПК-1.3.

Задание 1. Необходимо реализовать удалённый вызов процедуры для вычисления математических функций на специализированном вычислительном сервере. Клиентское приложение на языке Python отправляет на сервер имя функции и список аргументов (например, `sqrt(25)`, `pow(2, 10)`, `log(100, 10)`). Сервер производит вычисления и возвращает результат. В качестве транспортного протокола используется HTTP. Требования: вызов должен быть блокирующим (синхронным), а формат сообщений — XML. При этом сервер может быть недоступен, и клиент должен корректно обрабатывать таймауты.

Задачи:

1. Опишите структуру XML-запроса и XML-ответа для вызова функции `pow(2, 10)`.
2. Напишите псевдокод клиентской функции, которая формирует XML-запрос, отправляет его на сервер по HTTP и обрабатывает возможные ошибки (сетевые ошибки, таймаут, некорректный ответ).
3. Какие проблемы могут возникнуть при передаче чисел с плавающей точкой через XML-RPC и как их можно решить?

Задание 2. Разрабатывается распределённая система управления заказами, в которой используются два типа RPC: синхронный вызов для проверки наличия товара на складе и асинхронный вызов для оформления доставки. При синхронном вызове клиент ожидает ответа, при асинхронном — отправляет запрос и продолжает работу, а результат обрабатывается позже при наступлении события. Асинхронный вызов оформления доставки может выполняться от 2 до 10 секунд. В случае синхронного вызова максимальное время ожидания — 1 секунда. Система должна корректно работать при временной недоступности сервера проверки наличия товаров.

Задачи:

1. Объясните, почему для проверки наличия товара выбран синхронный вызов, а для оформления доставки — асинхронный.
2. Как должен вести себя клиент при синхронном вызове, если сервер не ответил в течение 1 секунды?

3. Опишите механизм обработки асинхронного вызова: как клиент узнает о завершении операции оформления доставки?

Задание 3. При использовании XML-RPC в распределённой системе возникли проблемы с производительностью: объём передаваемых XML-сообщений слишком велик, сериализация и десериализация занимают до 30% времени выполнения запроса. Команда разработчиков рассматривает возможность перехода на gRPC с использованием Protocol Buffers. Тестовый XML-RPC-вызов возвращает следующий результат:

```
<response><status>ok</status><data><id>100500</id><value>42.15</value></data></response>
```

Требуется оценить потенциальный выигрыш от перехода.

Задачи:

1. Перепишите приведённый XML-ответ в формате Protocol Buffers (синтаксис .proto-файла и пример сериализованных данных в текстовом представлении).

2. Укажите не менее трёх преимуществ gRPC/Protobuf перед XML-RPC, основываясь на описанной проблеме производительности.

3. В каких сценариях XML-RPC может оставаться предпочтительнее gRPC, несмотря на недостатки?

Тема 6. Simple Object Access Protocol (SOAP). ПК-1.1, ПК-1.3, ПК-1.4.

Задание 1. Разрабатывается веб-сервис для банковской системы перевода средств между счетами. Требования заказчика: все сообщения должны быть в формате SOAP; каждый запрос должен содержать идентификатор транзакции, данные счёта отправителя и получателя, сумму перевода и валюту; каждый ответ должен содержать статус операции и уникальный номер подтверждения; в случае ошибки (недостаточно средств, неверный номер счёта, недоступность базы данных) должны возвращаться коды ошибок и текстовые описания. Сервис должен поддерживать транзакционность: либо перевод выполняется полностью, либо не выполняется вовсе.

Задачи:

1. Разработайте структуру SOAP-сообщения (тело) для запроса на перевод средств.

2. Разработайте структуру SOAP-сообщения для успешного ответа и для ответа с ошибкой (блок Fault).

3. Какие расширения WS-* (не менее двух) потребуются для обеспечения транзакционности и безопасности в данном сервисе? Объясните, зачем они нужны.

Задание 2. Компания-интегратор получила задание описать существующий SOAP-сервис на языке WSDL. Сервис называется CurrencyConverter и

предоставляет единственную операцию `getExchangeRate`, которая принимает три параметра: `fromCurrency` (строка, код валюты, например, USD), `toCurrency` (строка, код валюты), `date` (дата, для которой требуется курс). Сервис возвращает значение типа `decimal` — курс обмена. Транспорт — SOAP over HTTP, стиль сообщений — `document`. Все сообщения используют пространство имён `http://example.com/currency`.

Задачи:

1. Опишите структуру WSDL-документа, перечислив все обязательные секции (`types`, `message`, `portType`, `binding`, `service`) и указав их содержание.
2. Как в данном WSDL будет описана операция `getExchangeRate`?
3. Чем отличается RPC-style от document-style в SOAP и почему в данном случае выбран document-style?

Задание 3. Две системы обмениваются SOAP-сообщениями через публичную сеть Интернет. Заказчик обнаружил, что сообщения передаются в открытом виде (без шифрования), не имеют цифровой подписи, а также не гарантируется доставка при сбоях сети. Кроме того, злоумышленник может перехватить сообщение и воспроизвести его повторно (`replay attack`), что критично для финансовых операций. Разработчикам поручено усилить безопасность и надёжность обмена без замены протокола.

Задачи:

1. Какие расширения WS-* следует использовать для обеспечения конфиденциальности и целостности сообщений?
2. Какое расширение WS-* обеспечивает надёжную доставку (гарантию доставки, устранение дубликатов, сохранение порядка сообщений)?
3. Как предотвратить атаку повторного воспроизведения (`replay attack`) в SOAP-сообщениях?

Раздел 3. Архитектурный стиль REST

Тема 7. Основные принципы REST. ПК-1.2, ПК-1.5.

Задание 1. Проектируется RESTful API для интернет-магазина. Требуется реализовать следующие операции: получение списка всех товаров, получение информации о конкретном товаре по его идентификатору, добавление нового товара, обновление информации о товаре, частичное изменение цены товара, удаление товара. Также требуется операция поиска товаров по названию (с возможностью указания неполного названия). Все ответы должны содержать соответствующие коды состояния HTTP. Некоторые операции должны поддерживать кэширование. Проектировщики спорят, какой метод HTTP использовать для операции поиска.

Задачи:

1. Для каждой из перечисленных операций укажите соответствующий метод HTTP и URL-шаблон.

2. Какой метод HTTP следует использовать для поиска товаров по названию? Обоснуйте выбор.

3. Для каких операций уместно использовать кэширование и как это указать в ответе сервера?

Задание 2. Разрабатывается RESTful API для системы управления библиотекой. Клиент отправляет запрос на сервер:

GET /books/123. Сервер возвращает ответ: {"id": 123, "title": "Война и мир", "author": "Толстой"}.

Через некоторое время клиент отправляет DELETE /books/123, затем снова GET /books/123. Первый разработчик утверждает, что сервер должен вернуть статус 404 (Not Found), а второй — что должен вернуть 410 (Gone), так как ресурс существовал, но был удалён. Кроме того, клиент после неудачного запроса повторяет его автоматически, не понимая разницы между ошибками клиента и сервера.

Задачи:

1. Какой код состояния HTTP более корректен в описанной ситуации после удаления? Обоснуйте ответ, опираясь на семантику кодов HTTP.

2. В чём разница между кодами 4xx и 5xx? Приведите по два примера из каждой группы.

3. Какие действия должен предпринимать клиент при получении кода 429 (Too Many Requests)?

Задание 3. Компания разрабатывает API для социальной сети. Требуется спроектировать ресурс «пользователь» и ресурс «пост». Пользователь может создавать посты, просматривать посты других пользователей, редактировать и удалять только свои посты. Идентификатор пользователя — число, идентификатор поста — строка вида post_XXXXXX. Кроме того, требуется реализовать возможность получения всех постов конкретного пользователя. Разработчики не могут определиться, как лучше организовать URL для получения постов пользователя: /posts?author_id=123 или /users/123/posts. Первый вариант называют «более простым», второй — «более RESTful».

Задачи:

1. Какой из двух вариантов URL соответствует принципам REST в большей степени? Почему?

2. Спроектируйте полную систему URL для всех перечисленных операций.

3. Как в REST реализуется принцип HATEOAS? Приведите пример ответа сервера при получении информации о пользователе, содержащий гиперссылки на доступные действия.

Тема 8. Модель зрелости сервисов REST. ПК-1.2, ПК-1.5.

Задание 1. Аналитики провели аудит трёх различных API, используемых в компании. Первый API использует один endpoint `https://api.example.com/process` и метод POST для всех операций, параметры операции передаются в теле запроса в JSON. Второй API использует отдельные URL для разных ресурсов (например, `/users`, `/orders`), но для всех операций (чтение, создание, обновление, удаление) применяет только метод POST. Третий API использует URL для ресурсов и методы GET, POST, PUT, DELETE в соответствии с их семантикой, но ответы содержат только сами данные, без каких-либо ссылок на связанные ресурсы или возможные последующие действия.

Задачи:

1. Определите уровень зрелости по модели Ричардсона для каждого из трёх API.
2. Для API, находящегося на втором уровне, опишите, какие изменения необходимо внести, чтобы подняться на третий уровень.
3. Какой уровень зрелости считается минимально необходимым, чтобы API назывался RESTful? Ответ обоснуйте.

Задание 2. Проектируется API для интернет-магазина. Заказчик настаивает на использовании одного endpoint `https://shop.api/command` с методом POST для всех операций (создание заказа, получение статуса заказа, отмена заказа, добавление товара в корзину, получение содержимого корзины). В качестве аргумента заказчик приводит простоту реализации и то, что «так проще для клиентов». Разработчик API предлагает перейти на подход с использованием разных URL и методов HTTP, так как это повысит зрелость API и упростит его сопровождение в будущем.

Задачи:

1. Какому уровню зрелости по модели Ричардсона соответствует предложение заказчика?
2. Перечислите не менее четырёх проблем, которые возникнут при таком подходе в долгосрочной перспективе (сопровождение, тестирование, документация, кэширование и т.д.).
3. Предложите компромиссное решение, которое было бы проще для заказчика, но при этом поднимало API хотя бы на первый или второй уровень зрелости.

Задание 3. Компания разработала API для управления задачами, которое было оценено как соответствующее второму уровню модели Ричардсона. После внедрения клиенты начали жаловаться: при получении задачи по GET `/tasks/123` они не знают, какие дальнейшие действия возможны (можно ли её редактировать, удалить, изменить статус). Им приходится обращаться к документации и самостоятельно формировать следующие запросы. Если API меняется (например, добавляется новое действие «переназначить задачу»), клиенты продолжают работать по старым правилам и не используют новую возможность. Команда разработки API хочет перейти на третий уровень зрелости,

чтобы клиенты могли динамически узнавать о доступных действиях.

Задачи:

1. Какое ключевое свойство REST необходимо реализовать для перехода с второго уровня на третий?
2. Перепишите пример ответа сервера на GET /tasks/123 в формате JSON (или XML) так, чтобы он соответствовал третьему уровню зрелости. Включите в ответ как минимум ссылки на редактирование, удаление и изменение статуса задачи.
3. Как третий уровень зрелости решает проблему изменения API без поломки существующих клиентов?

5.3. Один или несколько тематических блоков дисциплины завершаются контрольной точкой (далее – КТ). Текущий контроль успеваемости по дисциплине предусматривает не менее 2 (двух) и не более 10 (десяти) КТ в течение периода освоения дисциплины.

Максимальное количество баллов за любой тип работ в рамках КТ составляет 100 (сто) баллов.

Распределение весовых коэффициентов по КТ в рамках текущего контроля успеваемости по дисциплине и формулы расчета:

Наименование контрольной точки	Максимальное количество баллов за работу в рамках КТ, которое может набрать студент	Коэффициент веса контрольной точки	Результат контрольной точки, участвующий в формировании итоговой балльной оценки по дисциплине (отражается в журнале БРС в СДО)
КТ 1	100	0,35	35
КТ 2	100	0,35	35
КТ 3	100	0,3	30
Итого:	x	1,0	100

Формула расчета результата контрольной точки:

Результат контрольной точки = Количество баллов за работу в рамках КТ x Коэффициент веса контрольной точки.

5.4. Формы текущего контроля успеваемости обучающихся в рамках КТ и типовые оценочные материалы:

КТ – 1.

Контрольные задания:

Вариант 1

Крупный ритейлер имеет три изолированные системы: систему управления складом (WMS), систему управления заказами (OMS) и систему управления

взаимоотношениями с клиентами (CRM). В пик сезона (новогодние праздники) нагрузка на WMS возрастает в 10 раз, что приводит к сбоям при прямой синхронной интеграции. Руководство требует обеспечить интеграцию без потери данных и без остановки работы систем, а также дать возможность подключения новых партнёрских сервисов в будущем.

Задание

1. Какой уровень интеграции (данных, приложений, интерфейсов, пользователей) целесообразно выбрать в данной ситуации? Обоснуйте ответ с учётом пиковых нагрузок.
2. Предложите архитектурное решение с использованием сервисной шины предприятия (ESB). Опишите, как ESB поможет справиться с асинхронностью и масштабированием.
3. Какие свойства сервисов в SOA (минимум три) должны быть реализованы, чтобы обеспечить возможность подключения новых партнёрских сервисов без изменения существующей логики?
4. Оцените уровень зрелости SOA по модели Sonic Software, если в компании имеется только эпизодический опыт внедрения сервисов, нет задокументированной архитектуры, но уже используются XML, WSDL, SOAP. Ответ поясните.

Вариант 2

Государственная организация объединяет информационные системы трёх ведомств. Каждое ведомство использует свою СУБД (Oracle, PostgreSQL, MS SQL Server). Необходимо создать единую точку доступа для граждан, где можно получить сводную информацию из всех трёх систем (статус заявки, задолженности, документы). При этом менять структуру и код существующих систем запрещено — разрешена только надстройка.

Задание

1. Какой уровень интеграции (данных, приложений, интерфейсов, пользователей) наиболее подходит для описанной задачи? Почему другие уровни не подходят?
2. Предложите технологию или архитектурный паттерн, позволяющий получить сводные данные из трёх разных СУБД без изменения исходных систем.
3. Какие требования к сервисам в SOA (минимум три) нарушаются, если каждая из трёх систем вызывает другую напрямую (point-to-point)?
4. Какой уровень зрелости SOA по модели Sonic Software следует достичь, чтобы обеспечить автоматический мониторинг эффективности бизнес-процессов? Какие подсистемы для этого необходимы?

Вариант 3

Производственный холдинг принял стратегию перехода к сервисно-ориентированной архитектуре. На первом этапе создан реестр сервисов, задокументирована архитектура, внедрены политики безопасности и однократная аутентификация (SSO). Однако бизнес-процессы по-прежнему требуют ручного вмешательства, а интеграция с двумя ключевыми поставщиками осуществляется через

обмен файлами по FTP. Холдинг хочет перейти на следующий уровень зрелости, чтобы автоматизировать взаимодействие с партнёрами.

Задание

1. На каком уровне зрелости SOA (по модели Sonic Software) находится холдинг в данный момент? Аргументируйте.
2. Какой уровень является следующим? Перечислите не менее трёх изменений, которые необходимо внедрить для перехода.
3. Как на новом уровне будут организованы сервисы? В чём разница между бизнес-сервисами и сервисами сотрудничества (collaboration services)?
4. Какой стандарт (из лекционного материала) используется для описания бизнес-процессов на третьем уровне зрелости? Для чего он применяется?

Вариант 4

Банк решил внедрить систему, которая автоматически реагирует на события: при падении баланса клиента ниже порогового значения отправляется уведомление, при подозрительной транзакции блокируется счёт, при одобрении кредита запускается процесс оформления. Система должна самостоятельно определять эти ситуации и принимать решения без участия человека. Банк хочет оценить, достигнут ли пятый уровень зрелости SOA.

Задание

1. Какие признаки (не менее трёх) характеризуют пятый уровень зрелости SOA по модели Sonic Software?
2. Какие подсистемы (не менее двух) должны быть внедрены для автоматического принятия решений на основе событий?
3. Чем полуавтоматический режим работы системы поддержки принятия решений отличается от автоматического на пятом уровне?
4. В чём отличие модели зрелости SOE (от IBM/BEA) от модели Sonic Software с точки зрения целей оценки? Для какой аудитории предназначена каждая модель?

КТ – 2.

Контрольные задания:

Вариант 1

Метеорологический центр передаёт данные наблюдений с трёх метеостанций в единый аналитический центр. Для каждой станции необходимо передавать: название станции (на русском и английском), координаты (широта, долгота), список измерений (температура, давление, влажность, скорость ветра) с указанием единиц измерения. Часть измерений может отсутствовать. Требуется разработать формат XML и DTD для валидации. Также необходимо обеспечить возможность удалённого вызова процедуры для получения данных с конкретной станции за последние 24 часа.

Задание

1. Разработайте XML-схему (DTD) для описания данных трёх метеостанций. Укажите все элементы и атрибуты.

2. Напишите фрагмент XML-документа для одной станции с двумя измерениями.

3. Опишите структуру XML-RPC запроса и ответа для удалённого вызова `getStationData(stationId, hours=24)`.

4. Почему в данном сценарии XML-RPC может быть предпочтительнее SOAP? Укажите две причины.

Вариант 2

Компания разрабатывает SOAP-сервис для конвертации валют. Сервис должен принимать запрос с тремя параметрами: сумма, валюта источника (например, USD), валюта назначения (например, EUR). Возвращать — сконвертированную сумму. Требования: все сообщения должны соответствовать стилю document, транспорт — HTTP, коды ошибок при неверной валюте или отсутствии курса должны быть стандартизированы. Также необходимо описать сервис на WSDL для публикации в реестре.

Задание

1. Разработайте структуру тела SOAP-запроса и SOAP-ответа (без учёта конверта) в XML.

2. Разработайте структуру SOAP-сообщения с ошибкой (Fault) для случая, когда указана неизвестная валюта.

3. Какие секции WSDL-документа являются обязательными? Кратко опишите назначение каждой.

4. Чем RPC-style SOAP отличается от document-style? Почему в данном задании выбран document-style?

Вариант 3

При разработке интеграционного решения разработчик столкнулся с ошибками парсинга XML. В текстовом содержимом элементов встречаются символы `<`, `>`, `&`, а также требуется вставить большой фрагмент HTML-кода внутри XML без экранирования каждого тега. Кроме того, в нескольких местах документа повторяется длинное название организации «Общество с ограниченной ответственностью "Ромашка и партнёры"», и разработчик хочет определить сокращение для его многократного использования.

Задание

1. Какие встроенные сущности XML следует использовать для экранирования символов `<`, `>`, `&`, `"`, `'`? Приведите пример замены для фрагмента `"a < b & b > c"`.

2. Какой механизм XML позволяет вставить блок с HTML-кодом без ручного экранирования? Покажите синтаксис на примере.

3. Как определить пользовательскую сущность в DTD и如何使用 её в XML-документе для сокращения длинного названия организации?

4. Какие ограничения накладываются на содержимое CDATA-раздела? Почему CDATA-разделы не могут быть вложенными?

Вариант 4

В распределённой системе управления заказами используется XML-RPC для вызова двух операций: `checkStock(productId)` — синхронно, с таймаутом 500 мс, и `createOrder(orderData)` — асинхронно, с возможностью длительного выполнения (до 30 секунд). При тестировании выяснилось, что синхронные вызовы часто падают по таймауту во время пиковой нагрузки, а асинхронные вызовы теряются при перезапуске сервера. Руководство требует повысить надёжность.

Задание

1. Почему для `checkStock` выбран синхронный вызов, а для `createOrder` — асинхронный? В чём разница семантики?
2. Какую архитектурную альтернативу можно предложить вместо синхронного XML-RPC для `checkStock`, чтобы избежать падений по таймауту?
3. Какой механизм (из современных технологий) позволяет гарантировать доставку асинхронных сообщений при перезапуске сервера? Опишите принцип.
4. Назовите два основных недостатка XML-RPC по сравнению с gRPC/Protobuf в контексте производительности сериализации.

КТ – 3.

Контрольные задания:

Вариант 1

Проектируется RESTful API для интернет-магазина электроники. Требуются следующие операции: получение списка всех товаров с возможностью пагинации и фильтрации по категории и цене; получение информации о конкретном товаре; добавление нового товара (только для администратора); обновление цены товара; удаление товара; получение всех отзывов на товар; добавление нового отзыва. Клиенты жалуются, что после изменения структуры API их приложения перестают работать, так как они жёстко зашивают URL.

Задание

1. Для каждой из перечисленных операций укажите подходящий метод HTTP и URL-шаблон.
2. Как реализовать пагинацию и фильтрацию в RESTful API? Приведите примеры URL с параметрами.
3. Какой принцип REST позволяет клиентам не жёстко зашивать URL? Опишите, как должен выглядеть ответ сервера при получении информации о товаре, чтобы клиент мог динамически узнать URL для получения отзывов.
4. Какой код состояния HTTP следует вернуть при попытке удалить уже удалённый товар (200, 204, 404, 410)? Обоснуйте.

Вариант 2

Аналитик проводит аудит трёх API. API-A использует один endpoint `/api` и только метод POST, параметры передаются в теле. API-B использует разные URL для разных ресурсов (`/users`, `/orders`), но все операции выполняет через POST. API-B использует GET для чтения, POST для создания, PUT/PATCH для обновления,

DELETE для удаления, но в ответах не содержит гиперссылок. Заказчик требует, чтобы API-B поднялся на следующий уровень зрелости по модели Ричардсона.

Задание

1. Определите уровень зрелости (0, 1, 2 или 3) для каждого из трёх API.
2. Какой минимальный уровень необходим, чтобы API назывался RESTful? Почему API-A и API-B не соответствуют REST?
3. Какие изменения необходимо внести в API-B для перехода на третий уровень зрелости? Приведите пример ответа сервера на GET /orders/123 с реализацией этих изменений.
4. В чём преимущество третьего уровня зрелости перед вторым с точки зрения эволюции API без поломки клиентов?

Вариант 3

Команда разработчиков спорит о дизайне API для блога. Один предлагает URL /posts?author_id=5 для получения постов конкретного автора. Второй настаивает на /authors/5/posts. Третий предлагает использовать /authors/5/posts и дополнительно /posts/123 для конкретного поста. Также обсуждается, как быть с операцией поиска постов по ключевому слову: использовать GET /posts?search=keyword или POST /posts/search с телом запроса.

Задание

1. Какой вариант URL для получения постов автора более соответствует принципам REST? Обоснуйте, используя понятие «ресурс» и «вложенность».
2. Какой метод HTTP следует использовать для поиска по ключевому слову? Аргументируйте с учётом принципов REST (идемпотентность, безопасность, кэширование).
3. Если поисковый запрос сложный (много параметров, длинная строка), какой метод предпочтительнее и почему?
4. Что такое HATEOAS? Приведите пример ответа сервера при получении информации о посте, содержащего гиперссылки на автора и комментарии.

Вариант 4

Компания разработала API для управления проектами, которое соответствует второму уровню зрелости Ричардсона. После внедрения клиенты столкнулись с проблемами: при получении проекта по GET /projects/42 они не знают, можно ли его редактировать, удалять или архивировать. При изменении правил (например, проект можно архивировать только после завершения всех задач) клиенты не получают этой информации динамически. Руководство требует перейти на третий уровень зрелости, но разработчики не понимают, как это реализовать на практике.

Задание

1. Какое ключевое свойство REST необходимо реализовать для перехода на третий уровень? Дайте его определение.
2. Перепишите гипотетический JSON-ответ на GET /projects/42 так, чтобы он соответствовал третьему уровню. Добавьте гиперссылки на действия: редактирование

(только если статус "active"), удаление (если статус "draft"), архивирование (если все задачи завершены).

3. Как клиент, получивший такой ответ, должен определить, можно ли выполнить архивирование, не обращаясь к отдельной документации?

4. Какие изменения в API потребуются при добавлении нового действия (например, «заморозить проект») и как третий уровень зрелости позволяет внедрить это без поломки существующих клиентов?

Критерии оценивания контрольных заданий:

Диапазон баллов	Описание критерия
85-100	Обучающимся задание выполнено без ошибок и в полном объеме.
65-84	Обучающимся в целом задание выполнено, имеются отдельные неточности или недостаточно полные ответы, не содержащие ошибок.
55-64	Обучающимся допущены отдельные ошибки при выполнении задания
0-54	У обучающегося отсутствуют ответы на большинство вопросов задачи, задание не выполнено или выполнено не верно.

5.5. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий.

Не требуется.

6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине

6.1. Промежуточная аттестация по дисциплине Б1.В.02 Проектирование сервисно-ориентированных систем проводится в форме экзамена во 2-м семестре 1-го курса.

Обучающийся получает билет с вариантами заданий, чистые маркированные листы бумаги для записей решения заданий, затем приступает к решению.

6.2. Типовые оценочные материалы промежуточной аттестации.

Типовые проверочные задания для самоподготовки обучающегося к промежуточной аттестации:

Вопросы к экзамену:

1. Какие существуют уровни интеграции информационных систем и чем они отличаются друг от друга?

2. В чём заключается разница между физической, логической и семантической интеграцией данных?

3. Какие задачи решает интеграция на уровне пользователей и какие технологии для этого применяются?
4. Какую роль выполняет сервисная шина предприятия (ESB) при интеграции на уровне приложений?
5. Какие преимущества и недостатки имеет интеграция на уровне данных с использованием промежуточной базы данных?
6. Чем синхронная интеграция отличается от асинхронной и в каких сценариях применяется каждый подход?
7. Какие основные свойства должны быть у служб (сервисов) в сервисно-ориентированной архитектуре?
8. Чем SOA отличается от традиционного монолитного подхода к построению информационных систем?
9. Какие пять уровней включает модель зрелости SOA, предложенная компанией Sonic Software?
10. Какие подсистемы появляются на четвёртом уровне зрелости SOA и какие задачи они решают?
11. В чём заключается различие между бизнес-сервисами и сервисами сотрудничества на третьем уровне зрелости SOA?
12. Для какой аудитории предназначена модель зрелости SOE (Maturity Model for Service Oriented Enterprises), разработанная IBM и BEA Systems?
13. Для каких целей был создан язык XML и какие проблемы он решает в отличие от HTML?
14. Из каких обязательных и необязательных частей состоит XML-декларация?
15. Какие правила необходимо соблюдать при именовании элементов в XML-документе?
16. Чем пустой элемент в XML отличается от непустого и как синтаксически оформляется пустой элемент?
17. В каких тегах XML могут располагаться атрибуты и какие правила синтаксиса атрибутов следует соблюдать?
18. Какие символы запрещено использовать внутри XML-комментариев и почему?
19. Для чего предназначены разделы CDATA в XML-документах и какие ограничения на них накладываются?
20. Какие пять встроенных сущностей определены в XML и для представления каких символов они используются?
21. Как в XML решается проблема конфликта имён элементов и что такое пространства имён?
22. В чём разница между DTD и XML Schema (XSD) при описании структуры XML-документа?
23. Какие элементы и атрибуты могут быть описаны с помощью инструкции ELEMENT в DTD?
24. Какие типы данных могут быть заданы для атрибутов с помощью инструкции ATTLIST в DTD?
25. Какова основная идея удалённого вызова процедур и какие проблемы он решает?

26. Какие проблемы возникают при передаче параметров через сеть в классическом RPC?
27. Чем современный gRPC на основе Protocol Buffers отличается от XML-RPC?
28. Из каких обязательных и необязательных частей состоит SOAP-сообщение?
29. Чем RPC-style SOAP отличается от document-style и в каких случаях выбирается каждый стиль?
30. Для чего используется блок Fault в SOAP-сообщении и какова его структура?
31. Какова роль WSDL при описании SOAP-сервисов и какие секции обязательно присутствуют в WSDL-документе?
32. Какие расширения WS-* используются для обеспечения безопасности и надёжной доставки SOAP-сообщений?
33. Какие шесть ограничений (принципов) определяют архитектурный стиль REST?
34. Что означает требование «отсутствие состояния» (stateless) в REST и к каким последствиям оно приводит?
35. Какие методы HTTP считаются безопасными, какие — идемпотентными, и в чём разница между этими понятиями?
36. Что означает аббревиатура HATEOAS и какое место это понятие занимает в REST-архитектуре?
37. Какие четыре уровня включает модель зрелости REST Ричардсона и чем характеризуется каждый уровень?
38. Какие признаки позволяют отличить REST-сервис первого уровня зрелости от сервиса второго уровня?
39. Какие преимущества даёт переход с третьего уровня зрелости на второй с точки зрения эволюции API?
40. Как на третьем уровне зрелости Ричардсона решается проблема изменения API без поломки существующих клиентов?

Пример практического задания:

Туристическое агентство «Глобус» планирует создать информационную систему для бронирования путёвок. Система должна объединять данные из трёх независимых источников: базы данных отеля (содержит название отеля, список номеров, цены, наличие свободных мест), базы данных авиакомпании (рейсы, даты, количество свободных билетов) и базы данных экскурсионного бюро (список экскурсий, расписание, стоимость). В настоящее время все три источника работают изолированно. Агентство хочет предоставить клиентам единый веб-интерфейс для подбора и бронирования путёвки, а также обеспечить возможность интеграции с новыми поставщиками (другие отели, авиакомпании) в будущем. Техническое задание требует реализовать сервисно-ориентированную архитектуру, использовать REST для взаимодействия с клиентским приложением и обеспечить проверку структуры передаваемых данных.

Задание

1. Предложите вариант разделения функциональности системы на сервисы в рамках SOA (не менее трёх сервисов). Для каждого сервиса укажите его назначение и

какие источники данных он использует.

2. Разработайте фрагмент XML-документа для передачи информации об отеле (название отеля, список из двух номеров с указанием типа номера, цены за ночь и наличия свободных мест). Разработайте DTD-схему для проверки структуры этого XML-документа.

3. Спроектируйте RESTful API для операции получения списка доступных путёвок. Укажите: метод HTTP, URL-шаблон, параметры запроса (город, дата заезда, дата выезда, количество гостей) и пример ответа сервера в формате JSON для двух найденных вариантов. Ответ должен соответствовать как минимум второму уровню модели зрелости Ричардсона.

4. Оцените, на каком уровне зрелости SOA по модели Sonic Software будет находиться система после внедрения предложенных сервисов, если архитектура задокументирована, определены политики безопасности, но бизнес-процессы всё ещё требуют ручного вмешательства и отсутствует автоматический мониторинг. Обоснуйте ответ.

6.3. Критерии и шкала оценивания на основе БРС

КРИТЕРИИ ОЦЕНИВАНИЯ	РЕЗУЛЬТАТ В БАЛЛАХ
Дан полный, в логической последовательности развернутый ответ на поставленный вопрос, где он продемонстрировал знания предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно, и исчерпывающе отвечает на дополнительные вопросы, приводит собственные примеры по проблематике поставленного вопроса, решил предложенные практические задания без ошибок	90-100
Дан развернутый ответ на поставленный вопрос, где обучающийся демонстрирует знания, приобретенные на лекционных и семинарских занятиях, а также полученные посредством изучения обязательных учебных материалов по курсу, дает аргументированные ответы, приводит примеры, в ответе присутствует свободное владение монологической речью, логичность и последовательность ответа. Однако допускается неточность в ответе. Решил предложенные практические задания с небольшими неточностями.	75-89
Дан ответ, свидетельствующий в основном о знании процессов изучаемой дисциплины, отличающийся недостаточной глубиной и полнотой раскрытия темы, знанием основных вопросов теории, слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры, недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа и решении практических заданий.	60-74
Дан ответ, который содержит ряд серьезных неточностей, обнаруживающий незнание процессов изучаемой предметной	1-59

области, отличающийся неглубоким раскрытием темы, незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов, неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Выводы поверхностны. Решение практических заданий не выполнено, т.е. обучающийся не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя.	
--	--

6.4. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий

Компьютер с операционной системой RedOS или Windows с устойчивым Интернет-соединением, программные продукты с открытой лицензией: Visual Studio Code с расширением XML, Python с библиотеками requests и lxml, Mozilla Firefox, Bruno, SoapUI.

7. Методические материалы по освоению дисциплины (модуля)

Получение углубленных знаний по изучаемой дисциплине достигается за счет дополнительных часов к аудиторной работе самостоятельной работы студентов. Выделяемые часы целесообразно использовать для знакомства с дополнительной научной литературой по проблематике дисциплины, анализа научных концепций и современных подходов к осмыслению рассматриваемых проблем. К самостоятельному виду работы студентов относится работа в библиотеках, в электронных поисковых системах и т.п. по сбору материалов, необходимых для проведения практических занятий или выполнения конкретных заданий преподавателя по изучаемым темам. Студенты могут установить диалог с преподавателем, получать консультации по выполнению заданий. В качестве оценочных средств на протяжении семестра используются практические задания.

Обучение по дисциплине Б1.В.02 Проектирование сервисно-ориентированных систем предполагает изучение курса на аудиторных занятиях (лекции, практические занятия) и самостоятельную работу студентов. Практические занятия дисциплины предполагают их проведение в различных формах с целью выявления полученных знаний, умений, навыков и компетенций с проведением контрольных мероприятий. С целью обеспечения успешного обучения студент должен готовиться к лекции, поскольку она является важнейшей формой организации учебного процесса, поскольку:

- знакомит с новым учебным материалом;
- разъясняет учебные элементы, трудные для понимания;
- систематизирует учебный материал;
- ориентирует в учебном процессе.

Работа обучающегося на лекции:

Слушание и запись лекций – сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную

умственную деятельность обучающегося. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим обучающимся.

Подготовка к практическим занятиям:

Подготовку к каждому практическому занятию каждый обучающийся должен начать с ознакомления с планом, который отражает содержание предложенной темы. Тщательное продумывание и изучение вопросов плана основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованную к данной теме. Если программой дисциплины предусмотрено выполнение практического задания, то его необходимо выполнить с учетом предложенной инструкции. Все новые понятия по изучаемой теме необходимо внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности обучающегося свободно ответить на теоретические вопросы практического занятия, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий и контрольных работ.

Структура практического занятия:

В зависимости от содержания и количества отведенного времени на изучение каждой темы может практическое занятие состоять из четырех-пяти частей:

1. Устный опрос.
2. Обсуждение теоретических вопросов, определенных программой дисциплины.
3. Выполнение практических заданий с последующим разбором полученных результатов или обсуждение практического задания, выполненного дома.
4. Подведение итогов занятия.

Работа с литературными источниками:

В процессе подготовки к практическим занятиям, обучающимся необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем, что позволяет обучающимся проявить свою индивидуальность в рамках выступления на занятиях, выявить широкий спектр мнений по изучаемой проблеме.

8. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет

8.1. Основная литература

1. Быстренина, И. Е. Проектирование информационных систем : учебное пособие / И. Е. Быстренина. — Москва : Дашков и К, 2026. — 118 с. — ISBN 978-5-394-06431-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/513989> (дата обращения: 17.05.2026). — Режим доступа: для авториз. пользователей.

2. Токмаков, Г. П. Основы XML-технологий : учебное пособие / Г. П. Токмаков. — Ульяновск : УЛГТУ, 2017. — 229 с. — ISBN 978-5-9795-1701-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/165047> (дата обращения: 17.05.2026). — Режим доступа: для авториз. пользователей.

8.2. Дополнительная литература

1. Сакулин, С. А. Основы интернет-технологий: HTML, CSS, JavaScript, XML : учебное пособие / С. А. Сакулин. — Москва : МГТУ им. Н.Э. Баумана, 2017. — 112 с. — ISBN 978-5-7038-4724-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/103525> (дата обращения: 17.05.2026). — Режим доступа: для авториз. пользователей.

2. Диков, А. В. Клиентские технологии веб-дизайна. HTML5 и CSS3 / А. В. Диков. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 188 с. — ISBN 978-5-507-46740-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/318443> (дата обращения: 17.05.2026). — Режим доступа: для авториз. пользователей.

3. Смоленцева, Т. Е. Проектирование и разработка WEB-приложений: Практикум : учебное пособие / Т. Е. Смоленцева. — Москва : РТУ МИРЭА, 2023. — 68 с. — ISBN 978-5-7339-1759-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/368954> (дата обращения: 17.05.2026). — Режим доступа: для авториз. пользователей.

8.3. Нормативные правовые документы и иная правовая информация

1. Конституция Российской Федерации. — Текст : электронный // Сайт Президента Российской Федерации. — URL: <http://www.kremlin.ru/acts/constitution>

8.4 Интернет-ресурсы

1. Информационно-правовой портал ГАРАНТ.РУ. — URL: <https://www.garant.ru/>
2. Научная электронная библиотека eLIBRARY.RU. — URL: <https://elibrary.ru/>
3. Научная электронная библиотека «КиберЛенинка». — URL: <https://cyberleninka.ru>

4. Электронно-библиотечная система «Лань». – URL: <http://e.lanbook.com>
5. Документация по Python – URL: [https:// docs.python.org/3/](https://docs.python.org/3/)

9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

Материально-техническое обеспечение дисциплины включает в себя:

- лекционные аудитории, оборудованные видеопроекторным оборудованием для презентаций, средствами звуковоспроизведения, экраном;
- помещения для проведения практических занятий, оборудованные учебной мебелью.

Дисциплина поддержана соответствующими программными продуктами с открытой лицензией: Visual Studio Code с расширением XML, Python с библиотеками requests и lxml, Mozilla Firefox, Bruno, SoapUI.

Вуз обеспечивает каждого обучающегося рабочим местом в компьютерном классе в соответствии с объемом изучаемых дисциплин, обеспечивает выход в сеть Интернет.

Помещения для самостоятельной работы обучающихся включают следующую оснащенность: столы аудиторные, стулья, доски аудиторные, компьютеры с подключением к локальной сети института (для компьютерных аудиторий) и Интернет. Для изучения учебной дисциплины используются автоматизированная библиотечная информационная система и электронные библиотечные системы.