МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО Университет биотехнологий
Институт цифровых технологий
Кафедра информационных технологий и моделирования

ИНТЕГРАЦИЯ ИНФОРМАЦИОННЫХ РЕСУРСОВ С ДРУГИМИ СИСТЕМАМИ В СЕТИ ИНТЕРНЕТ

Методические указания для проведения лабораторно-практических занятий и самостоятельной работы студентов

Новосибирск 2026

УДК 004.738.5(075.32)
ББК 32.973.202я723
Ш 65

Кафедра информационных технологий и моделирования

Составитель: Шишина Л.Г.

Рецензент: Краснов С.Д., преподаватель кафедры информационных технологий и моделирования ФГБОУ ВО Университет биотехнологий

Интеграция информационных ресурсов с другими системами в сети Интернет: методические указания для проведения лабораторно-практических занятий и самостоятельной работы студентов / сост. Л. Г. Шишина. – Новосибирск: Университет биотехнологий, 2026.

Методические указания предназначены для студентов, обучающихся по специальности 09.02.09 Веб-разработка (дисциплина МДК.03.03 «Интеграция информационных ресурсов с другими системами в сети Интернет»). Содержат описание практических занятий, задания для самостоятельной работы, требования к оформлению отчётов и критерии оценки.

Методические указания разработаны в соответствии с рабочей программой дисциплины и фондом оценочных средств.

Содержание

Введение 4
1. Общие методические указания 7
1.1. Организация практических занятий 7
1.2. Правила работы на занятиях 7
1.3. Требования к программному коду 7
1.4. Требования к отчёту 8
1.5. Организация самостоятельной работы 8
1.6. Критерии оценки 8
2. Методические указания к практическим занятиям 9
Тема 1.1. Стандарты взаимодействия компонентов распределённых приложений 9
Тема 1.2. Архитектуры программных интерфейсов (API) 10
Тема 1.3. Форматы обмена данными 12
Тема 2.1. Интеграция функций внешних систем в алгоритм работы ресурса 13
Тема 2.2. Подключение информационных и сервисных систем 15
Тема 2.3. Модули расширения платформы для интеграции 16
Тема 2.4. Обработка ошибок и отказов при интеграции 17
Тема 3.1. Размещение информационного ресурса в сети Интернет 18
Тема 3.2. Проверка и мониторинг интеграционных решений 19
Тема 3.3. Документирование интеграционных решений 20
3. Методические указания к самостоятельной работе 22
4. Рекомендуемая литература и информационные ресурсы 23
Приложение А. Образец титульного листа отчёта 24

Введение

На изучение дисциплины МДК.03.03 «Интеграция информационных ресурсов с другими системами в сети Интернет» запланировано 112 ч, в том числе 108 ч аудиторных (38 ч – занятия лекционного типа, 44 ч – занятия семинарского типа (практические занятия), 24 ч – курсовое проектирование, 2 ч – консультации) и 4 ч (3,6 %) отведено для самостоятельной работы. Форма промежуточной аттестации – зачёт с оценкой и курсовой проект (3 семестр); к зачёту с оценкой допускаются студенты, защитившие курсовой проект. Практические занятия реализуются в форме практической подготовки в объёме 44 ч. Курсовое проектирование (24 ч, форма контроля – курсовой проект) выполняется в соответствии с рабочей программой и методическими указаниями по курсовому проектированию; настоящие указания охватывают практические занятия и самостоятельную работу.

Основная цель междисциплинарного курса – формирование умений интегрировать информационные ресурсы с внешними системами: работать с программными интерфейсами (API) и форматами обмена данными, подключать внешние сервисы, обрабатывать ошибки, размещать ресурс в сети Интернет, мониторить и документировать интеграции.

В процессе изучения дисциплины решаются следующие задачи:

- изучение стандартов взаимодействия компонентов распределённых приложений и архитектур программных интерфейсов;
- освоение форматов обмена данными (JSON, XML);
- освоение интеграции функций внешних систем в алгоритм работы ресурса и подключения сервисных систем;
- освоение модулей расширения платформы для интеграции;
- освоение обработки ошибок и отказов при интеграции;
- освоение размещения ресурса в сети Интернет, проверки, мониторинга и документирования интеграций.

В соответствии с требованиями ФГОС СПО дисциплина направлена на формирование следующих компетенций по специальности 09.02.09 Веб-разработка:

- выбирать способы решения задач профессиональной деятельности применительно к различным контекстам (ОК 01);
- использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности (ОК 02);
- осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста (ОК 05);
- пользоваться профессиональной документацией на государственном и иностранном языках (ОК 09);
- планировать коммуникации с заказчиком в рамках типовых регламентов организации с целью выбора платформы разработки информационного ресурса на визуальном и/или адаптированном (специальном) языках программирования (ПК 3.1);
- комбинировать и настраивать системы компонентов информационных ресурсов на визуальном и/или адаптированном (специальном) языках программирования (ПК 3.2);
- интегрировать информационные ресурсы с другими системами в информационно-телекоммуникационной сети «Интернет» (ПК 3.3).

Методические указания составлены в соответствии с рабочей программой дисциплины и фондом оценочных средств. Нумерация и содержание практических занятий соответствуют темам рабочей программы; контрольные вопросы и практические задания согласованы с фондом оценочных средств.

Распределение практических занятий и самостоятельной работы по темам дисциплины приведено в таблице 1.

Таблица 1 – Распределение часов по темам

Тема	**Практические занятия**	**ПЗ, ч**	**СР, ч**
1.1. Стандарты взаимодействия компонентов распределённых приложений	№ 1–2	5	0
1.2. Архитектуры программных интерфейсов (API)	№ 3–4	5	2
1.3. Форматы обмена данными	№ 5–6	4	0
2.1. Интеграция функций внешних систем в алгоритм работы ресурса	№ 7–8	5	0
2.2. Подключение информационных и сервисных систем	№ 9–10	5	0
2.3. Модули расширения платформы для интеграции	№ 11	3	0
2.4. Обработка ошибок и отказов при интеграции	№ 12	3	0
3.1. Размещение информационного ресурса в сети Интернет	№ 13–14	5	2
3.2. Проверка и мониторинг интеграционных решений	№ 15–16	5	0
3.3. Документирование интеграционных решений	№ 17–18	4	0
Итого	18 занятий	44	4

1. Общие методические указания

1.1. Организация практических занятий

Практические занятия проводятся в учебной компьютерной лаборатории. Используются операционная система ALT Linux, язык PHP 8 и менеджер зависимостей Composer, фреймворк Laravel, среда Node.js, инструмент проверки программных интерфейсов Hoppscotch, редактор кода VSCodium, система контроля версий Git, браузер Mozilla Firefox, LibreOffice. Внешние сервисы выбираются из открытых и тестовых (например, API Банка России, тестовые сервисы JSONPlaceholder и httpbin, картографические и почтовые песочницы).

Перед занятием необходимо изучить краткие теоретические сведения и документацию используемого сервиса. На занятии студент выполняет интеграцию в учебном проекте, проверяет её, оформляет отчёт и отвечает на контрольные вопросы.

1.2. Правила работы на занятиях

- соблюдать правила электробезопасности и санитарные нормы работы за персональным компьютером, делать перерывы для гимнастики для глаз;
- не устанавливать и не удалять программное обеспечение без разрешения преподавателя;
- сохранять результаты работы в личном каталоге и делать резервные копии;
- ключи доступа и токены хранить только в переменных окружения (файл .env, исключённый из репозитория), в код, отчёты и снимки экрана не помещать;
- соблюдать условия использования внешних сервисов и ограничения на частоту запросов; не создавать избыточную нагрузку;
- при тестировании платёжных и почтовых сервисов использовать только тестовые режимы и вымышленные данные;

– самостоятельно выполнять задания; использование чужих решений и готовых ответов без указания источника не допускается.

1.3. Требования к программному коду

– секреты и адреса сервисов задаются в конфигурации (.env), а не в исходном коде;
– для каждого запроса указываются время ожидания (timeout) и обработка ошибок; ответы проверяются (код статуса, формат, обязательные поля);
– код интеграции выделяется в отдельный класс или модуль (клиент сервиса), не смешивается с представлением;
– код форматируется по PSR-12, снабжается комментариями к нетривиальным решениям;
– интеграция покрывается тестами с имитацией ответов внешнего сервиса; изменения фиксируются коммитами Git.

1.4. Требования к отчёту

Отчёт по каждому занятию оформляется в электронном виде (LibreOffice Writer, PDF) и содержит: тему и цель, описание сервиса и схемы обмена, примеры запросов и ответов (с удалёнными секретами), листинги кода, результаты проверок, описание обработки ошибок, выводы. Титульный лист – по образцу приложения А.

1.5. Организация самостоятельной работы

Самостоятельная работа (4 ч) включает изучение теоретического материала и выполнение заданий по темам 1.2, 3.1 (раздел 3 настоящих указаний). Результаты проверяются преподавателем на очередном практическом занятии в форме собеседования.

1.6. Критерии оценки

Результаты выполнения практических занятий и самостоятельной работы учитываются при выставлении оценки по результатам текущего контроля (3 семестр). Для оценки используется традиционная четырёхбалльная система:

Оценка	**Критерии**
Отлично	Все задания выполнены полностью и верно, отчёт оформлен по требованиям. Ответ полный, логичный, обоснованный; студент свободно отвечает на дополнительные вопросы, приводит примеры.
Хорошо	Задания выполнены, есть незначительные неточности в решении или оформлении. Ответ по существу верный; допускаются несущественные ошибки при ответе на дополнительные вопросы.
Удовлетворительно	Выполнена основная часть заданий. Ответ неполный: знание основных положений, затруднения с пояснениями и примерами.
Неудовлетворительно	Задания не выполнены или выполнены неверно. Основной материал не усвоен, на дополнительные вопросы студент ответить не может.

2. Методические указания к практическим занятиям

Ниже приведены практические занятия по темам рабочей программы. Для каждого занятия указаны цель, краткие теоретические сведения, задания и контрольные вопросы.

Раздел 1. Основы интеграции информационных ресурсов

Тема 1.1. Стандарты взаимодействия компонентов распределённых приложений

Принципы построения распределённых приложений, синхронное и асинхронное взаимодействие, очереди сообщений, вызовы удалённых процедур, требования к надёжности и согласованности обмена. Объём практических занятий – 5 ч.

Практическое занятие № 1. Распределённые приложения: синхронное и асинхронное взаимодействие (2 ч)

Цель работы: изучить способы взаимодействия компонентов распределённых приложений.

Краткие теоретические сведения.

Синхронное взаимодействие (запрос – ответ по HTTP) блокирует отправителя до ответа; асинхронное (очереди сообщений, события, webhook) позволяет не ждать результата и повышает устойчивость. Вызовы удалённых процедур (RPC, gRPC, JSON-RPC) оформляют обращение к функции другой системы как локальный вызов.

Задания.

1. Составьте таблицу сравнения синхронной и асинхронной интеграции (задержка, надёжность, сложность, примеры).
2. Нарисуйте схему обмена между интернет-магазином и платёжным сервисом и определите, где нужен синхронный, а где асинхронный обмен.

3. Опишите принцип работы webhook на примере уведомления о платеже.

Контрольные вопросы.

1. Какие стандарты взаимодействия компонентов распределённых приложений вы знаете?
2. Чем синхронное взаимодействие отличается от асинхронного?
3. Что такое очередь сообщений?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Практическое занятие № 2. Надёжность и согласованность обмена (3 ч)

Цель работы: изучить требования к надёжности и согласованности обмена данными.

Краткие теоретические сведения.

Требования: гарантии доставки (не менее одного раза, ровно один раз), идемпотентность операций (повтор не приводит к дублированию), порядок сообщений, подтверждения (ack), контроль версий данных. Идемпотентность достигается уникальными идентификаторами запросов.

```
<?php
$key = $_SERVER['HTTP_IDEMPOTENCY_KEY'] ?? null;
if ($key && already_processed($key)) { echo
json_encode(get_result($key)); exit; }
```

Задания.

1. Опишите три сценария сбоя обмена (потеря ответа, повтор запроса, расхождение данных) и меры защиты.
2. Реализуйте на PHP или JavaScript идемпотентную обработку: повторный запрос с тем же идентификатором не создаёт дубликат.
3. Составьте требования к надёжности обмена для интеграции с платёжным сервисом.

Контрольные вопросы.

1. Что такое идемпотентность и зачем она нужна?

2. Какие требования предъявляются к надёжности обмена?
3. Как избежать дублирования заказа при повторе запроса?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Тема 1.2. Архитектуры программных интерфейсов (API)

Виды программных интерфейсов, принципы REST, ресурсы и методы, коды ответов, версионирование, документация API, аутентификация и авторизация по ключу и токену. Объём практических занятий – 5 ч.

Практическое занятие № 3. REST и ресурсы API. Проверка запросов в Hoppscotch (2 ч)

Цель работы: изучить принципы REST и научиться проверять запросы инструментом.

Краткие теоретические сведения.

REST представляет данные как ресурсы с адресами (URL) и действиями методами HTTP: GET (чтение), POST (создание), PUT/PATCH (изменение), DELETE (удаление). Ответ содержит код статуса, заголовки и тело (JSON). Hoppscotch позволяет составлять и отправлять запросы и сохранять коллекции.

Задания.

1. Выполните запрос к открытому API (например, курсы валют ЦБ РФ) в Hoppscotch и разберите структуру ответа в формате JSON.
2. Выполните запросы GET, POST, PUT, DELETE к тестовому сервису JSONPlaceholder и сравните коды ответов.
3. Сохраните коллекцию запросов и приложите её в отчёт.

Контрольные вопросы.

1. Что такое программный интерфейс (API) и для чего он нужен?
2. Каковы основные архитектурные стили API (например, REST)?
3. Что такое HTTP-методы и как они используются в API?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Практическое занятие № 4. Коды ответов, документация API, аутентификация: ключ и токен (3 ч)

Цель работы: освоить коды ответов, чтение документации API и передачу ключей и токенов.

Краткие теоретические сведения.

Коды: 2xx – успех, 3xx – перенаправление, 4xx – ошибка клиента (400, 401 нет аутентификации, 403 нет прав, 404, 429 слишком много запросов), 5xx – ошибка сервера. Документация (OpenAPI/Swagger) описывает ресурсы, параметры, примеры; версионирование (/v1/) сохраняет совместимость. API-ключ идентифицирует приложение, токен (Bearer, OAuth 2.0) подтверждает права и имеет срок действия; ключ передают в заголовке Authorization и хранят в переменных окружения, а не в коде и репозитории.

```
APP_RATES_KEY=change_me
# config/services.php: 'rates' => ['key' => env('APP_RATES_KEY')]
```

Задания.

1. Составьте таблицу кодов статуса, важных при интеграции, с причинами и реакцией клиента.
2. Найдите в документации выбранного сервиса описание трёх методов и выпишите параметры, форматы ответов, ограничения.
3. Выполните запрос с заголовком Authorization к тестовому сервису (httpbin /bearer); проверьте ответы при верном и неверном токене.
4. Настройте в учебном проекте хранение ключа в .env и чтение через конфигурацию; добавьте .env в .gitignore.

Контрольные вопросы.

1. Как устроен ответ API: коды статусов, тело ответа, заголовки?
2. Что такое ключ доступа (API-ключ) и токен авторизации; как хранятся и защищаются ключи?
3. Какие требования предъявляются к документации API?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Тема 1.3. Форматы обмена данными

JSON и XML: синтаксис и область применения, схемы описания данных, валидация сообщений, кодировки, разбор и формирование сообщений, преобразование форматов. Объём практических занятий – 4 ч.

Практическое занятие № 5. JSON и XML: синтаксис, разбор и формирование (2 ч)

Цель работы: освоить разбор и формирование сообщений в форматах JSON и XML.

Краткие теоретические сведения.

JSON – компактный формат пар «ключ – значение», массивов и вложенных объектов; XML – размеченный текст с тегами, атрибутами, пространствами имён, поддерживает схемы (XSD). В PHP: json_encode/json_decode, SimpleXML. Нужно задавать кодировку UTF-8 и обрабатывать ошибки разбора.

```
<?php
$data = json_decode($body, true, 512, JSON_THROW_ON_ERROR);
$xml = simplexml_load_string($xmlBody);
```

Задания.

1. Разберите ответ API в формате JSON и выведите нужные поля на страницу (PHP).
2. Сформируйте JSON и XML для одного и того же объекта «заказ» и сравните размер и читаемость.
3. Преобразуйте XML-ответ (например, курсы валют ЦБ РФ в XML) в массив и в JSON.

Контрольные вопросы.

1. Какие форматы обмена данными применяются при интеграции (JSON, XML)?
2. Чем JSON отличается от XML?

3. Как обработать ошибку разбора?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Практическое занятие № 6. Схемы данных и валидация сообщений (2 ч)

Цель работы: освоить описание структуры сообщений схемой и их валидацию.

Краткие теоретические сведения.

Схема (JSON Schema, XSD) задаёт обязательные поля, типы, форматы, ограничения; валидация отклоняет некорректные сообщения до обработки. Это защищает от ошибок и атак. Преобразование форматов выполняют отображением полей с проверкой типов.

Задания.

1. Опишите формат обмена данными между учебным ресурсом и внешним сервисом (поля, типы, пример) и составьте JSON Schema.
2. Реализуйте валидацию входящего сообщения по схеме; проверьте корректные и некорректные примеры.
3. Опишите правила преобразования полей между двумя форматами.

Контрольные вопросы.

1. Что такое схема данных и зачем нужна валидация сообщений?
2. Как проверить корректность данных, полученных из внешней системы?
3. Какие кодировки используются при обмене?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Раздел 2. Интеграция внешних систем

Тема 2.1. Интеграция функций внешних систем в алгоритм работы ресурса

Сценарии обмена данными с внешними системами, сопоставление полей, преобразование и валидация передаваемых данных, встраивание вызовов внешних сервисов в бизнес-логику. Объём практических занятий – 5 ч.

Практическое занятие № 7. Сценарии обмена и сопоставление полей (2 ч)

Цель работы: спроектировать обмен данными между ресурсом и внешней системой.

Краткие теоретические сведения.

Сценарий описывает направление обмена (получение, отправка, двусторонний), триггер (по действию пользователя, по расписанию, по событию), формат, частоту. Сопоставление полей (mapping) связывает поля систем с преобразованием типов и значений (единицы, форматы дат, справочники).

Задания.

1. Подготовьте схему обмена данными между двумя системами (учебный ресурс и сервис курсов валют или доставки).
2. Составьте таблицу сопоставления полей с правилами преобразования.
3. Определите точки валидации и обработки ошибок в сценарии.

Контрольные вопросы.

1. Как интегрируются функции внешних систем в алгоритм работы ресурса?
2. Что такое сопоставление полей?
3. Какие триггеры обмена бывают?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Практическое занятие № 8. Встраивание вызовов внешних сервисов в бизнес-логику (3 ч)

Цель работы: реализовать вызов внешнего сервиса в бизнес-логике учебного проекта.

Краткие теоретические сведения.

Вызов внешнего сервиса выделяют в отдельный класс (клиент) с настройками подключения, таймаутом и обработкой ошибок; бизнес-логика обращается к клиенту через интерфейс, что упрощает тестирование и замену сервиса. В Laravel используется HTTP-клиент (фасад Http).

```
$response = Http::timeout(5)->get(config('services.rates.url'));
if ($response->failed()) { /* обработка ошибки */ }
$usd = $response->json('Valute.USD.Value');
```

Задания.

1. Подключите к учебному ресурсу внешний сервис (курсы валют, погода) и выведите данные на страницу.
2. Выделите клиент сервиса в отдельный класс с таймаутом и проверкой ответа.
3. Напишите тест с имитацией ответа сервиса (Http::fake).

Контрольные вопросы.

1. Как интегрируются функции внешних систем в алгоритм работы ресурса?
2. Зачем выделять клиент сервиса в отдельный класс?
3. Как протестировать интеграцию без обращения к реальному сервису?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Тема 2.2. Подключение информационных и сервисных систем

Регистрация и получение доступа к сервису, параметры подключения, управление ключами, ограничения по числу запросов, подключение платёжных, картографических, почтовых сервисов и служб уведомлений. Объём практических занятий – 5 ч.

Практическое занятие № 9. Регистрация, ключи и ограничения сервиса (2 ч)

Цель работы: освоить подключение внешнего сервиса: регистрация, ключи, лимиты.

Краткие теоретические сведения.

Подключение включает регистрацию, получение ключа, настройку разрешённых адресов, изучение лимитов (число запросов в минуту/сутки) и условий использования. Кэширование ответов снижает число запросов; при превышении лимита сервис возвращает 429 – необходимо замедлиться.

Задания.

1. Зарегистрируйтесь в тестовой среде выбранного сервиса (карты, почта, уведомления) и получите тестовый ключ.
2. Настройте параметры подключения в конфигурации проекта; реализуйте кэширование ответа на заданное время.
3. Опишите ограничения сервиса и реакцию вашей интеграции на превышение лимита.

Контрольные вопросы.

1. Как подключаются информационные и сервисные системы к веб-ресурсу?
2. Какие ограничения по числу запросов бывают и как их учитывать?
3. Зачем кэшировать ответы?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Практическое занятие № 10. Типовые сервисы: карты, почта, платежи, уведомления (3 ч)

Цель работы: подключить типовые сервисы и протестировать их в тестовых режимах.

Краткие теоретические сведения.

Картографические сервисы выводят карту и маркеры; почтовые (SMTP, API-почтовые службы) отправляют письма (необходима настройка SPF/DKIM); платёжные проводят оплату через редирект или API, подтверждая платёж уведомлением (webhook); службы уведомлений отправляют сообщения в мессенджеры.

Задания.

1. Выведите на страницу карту с маркером (открытая библиотека карт) и проверьте на мобильном экране.
2. Настройте отправку письма из проекта через тестовый почтовый сервер (MailHog или аналог) и проверьте получение.
3. Опишите последовательность оплаты через платёжный шлюз в тестовом режиме и обработки уведомления о платеже.

Контрольные вопросы.

1. Какие типовые сервисы подключаются к ресурсам?
2. Как подтверждается платёж?
3. Что такое SPF и DKIM?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Тема 2.3. Модули расширения платформы для интеграции

Готовые модули обмена данными, их настройка, сопоставление справочников, расписание синхронизации, случаи, когда требуется собственная доработка. Объём практических занятий – 3 ч.

Практическое занятие № 11. Модули (пакеты) для интеграции и собственная доработка (3 ч)

Цель работы: освоить готовые модули обмена и случаи их доработки.

Краткие теоретические сведения.

Готовые модули (пакеты Composer, плагины платформ) реализуют обмен с типовыми системами; настраиваются сопоставлением справочников и расписанием синхронизации (планировщик задач). Если нужной функции нет, пакет расширяют через события или оборачивают собственным классом.

```
composer require guzzlehttp/guzzle
php artisan schedule:work
```

Задания.

1. Установите пакет интеграции через Composer, настройте его и проверьте обмен.
2. Настройте расписание синхронизации (планировщик) для периодической загрузки данных.
3. Определите ограничения пакета и опишите, какая часть требует собственной доработки.

Контрольные вопросы.

1. Какие модули расширения платформы используются для интеграции?
2. Как настраивается расписание синхронизации?
3. Когда готового модуля недостаточно?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Тема 2.4. Обработка ошибок и отказов при интеграции

Недоступность сервиса, превышение времени ожидания, некорректные данные, повторные попытки, очереди отложенных заданий, компенсирующие действия, журналирование и оповещение. Объём практических занятий – 3 ч.

Практическое занятие № 12. Ошибки и отказы: таймауты, повторные попытки, очереди (3 ч)

Цель работы: реализовать устойчивую обработку отказов внешнего сервиса.

Краткие теоретические сведения.

Отказы: недоступность сервиса, таймаут, некорректный ответ, лимит запросов. Стратегии: повторные попытки с нарастающей задержкой (exponential backoff), ограничение числа попыток, очередь отложенных заданий, предохранитель (circuit breaker), компенсирующие действия (отмена частично выполненных операций), резервное значение (кэш).

```
$response = Http::timeout(5)->retry(3, 500)->get($url);
```

Задания.

1. Реализуйте обработку ошибок при недоступности внешнего сервиса: сообщение пользователю и повторный запрос.
2. Настройте повторные попытки с задержкой и очередь отложенных заданий для отправки данных.
3. Организуйте журналирование обмена и оповещение ответственного о серии отказов.

Контрольные вопросы.

1. Как обрабатываются ошибки и отказы внешнего сервиса?
2. Что такое таймаут и повторная попытка?
3. Что такое очередь отложенных заданий?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Раздел 3. Размещение и сопровождение интеграций

Тема 3.1. Размещение информационного ресурса в сети Интернет

Подготовка к публикации и перенос на площадку, привязка домена, защищённое соединение, проверка доступности, проверка интеграций в среде эксплуатации. Объём практических занятий – 5 ч.

Практическое занятие № 13. Подготовка к публикации и перенос на площадку (2 ч)

Цель работы: подготовить ресурс к публикации и выполнить перенос.

Краткие теоретические сведения.

Подготовка: переключение в рабочий режим (отключение отладки), настройка конфигурации окружения, кэширование конфигурации, миграции базы, права на каталоги, резервная копия. Перенос – через Git, rsync или архив; секреты задаются на сервере.

```
php artisan config:cache
php artisan migrate --force
```

Задания.

1. Подготовьте учебный проект к публикации по чек-листу (режим, секреты, миграции, права, резервная копия).
2. Разместите учебный ресурс на тестовом сервере и проверьте доступность по адресу.
3. Проверьте работу ключей и конфигурации в рабочем окружении.

Контрольные вопросы.

1. Как размещается информационный ресурс в сети Интернет?
2. Что такое хостинг?
3. Какие настройки меняются при переходе в рабочий режим?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Практическое занятие № 14. Домен, DNS и защищённое соединение; проверка интеграций (3 ч)

Цель работы: привязать домен, настроить HTTPS и проверить интеграции в среде эксплуатации.

Краткие теоретические сведения.

Для публикации создают A/AAAA-записи домена (DNS), устанавливают сертификат (HTTPS), проверяют доступность и перенаправления. Интеграции проверяют «снаружи»: разрешённые адреса сервера у внешних сервисов, ключи рабочего режима, webhooks (внешний адрес должен быть доступен из Интернета).

Задания.

1. Настройте разрешение имени (локальный DNS или hosts) и HTTPS с самоподписанным сертификатом на тестовом сервере.
2. Проверьте работу интеграций в среде эксплуатации (запрос к сервису, получение webhook на тестовый адрес).
3. Составьте таблицу различий окружений (тестовое и рабочее) для ключей и адресов.

Контрольные вопросы.

1. Что такое доменное имя, DNS и хостинг?
2. Как проверяется работоспособность интеграции после публикации?
3. Зачем нужен HTTPS для webhook?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Тема 3.2. Проверка и мониторинг интеграционных решений

Проверка корректности обмена и сверка данных, контроль работоспособности, метрики и пороговые значения, оповещения об отказах, разбор инцидентов обмена. Объём практических занятий – 5 ч.

Практическое занятие № 15. Проверка и сверка данных между системами (2 ч)

Цель работы: научиться проверять корректность обмена и сверять данные.

Краткие теоретические сведения.

Проверка корректности: сравнение выборочных записей источника и приёмника, контроль количества записей и контрольных сумм, проверка форматов. Сверка (reconciliation) выявляет расхождения и причины (потеря, дубликат, задержка).

Задания.

1. Выполните обмен тестовыми данными между двумя системами и сверьте результаты (количество, суммы, выборочные записи).
2. Составьте акт сверки с найденными расхождениями и причинами.
3. Разработайте автоматическую проверку (скрипт) для периодической сверки.

Контрольные вопросы.

1. Как проверяется корректность обмена и сверка данных между системами?
2. Что такое сверка?
3. Какие причины расхождений возможны?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Практическое занятие № 16. Мониторинг интеграций: метрики, оповещения, инциденты (3 ч)

Цель работы: настроить мониторинг интеграции и разбор инцидентов.

Краткие теоретические сведения.

Метрики: доля успешных запросов, время ответа (среднее, 95-й процентиль), число ошибок по типам, глубина очереди, задержка

синхронизации. Пороговые значения вызывают оповещения; разбор инцидента – хронология, причина, меры.

Задания.

1. Настройте сбор метрик интеграции (логи, счётчики) и вывод сводки.
2. Задайте пороговые значения и настройте оповещение об отказах (письмо или запись в канал).
3. Разберите учебный инцидент: хронология, причина, действия.

Контрольные вопросы.

1. Что включает мониторинг интеграционных решений?
2. Какие показатели мониторинга важны для интеграции?
3. Как организовать поддержку интеграции после публикации?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Тема 3.3. Документирование интеграционных решений

Состав документации, описание интерфейсов и структур сообщений, регламент обмена и порядок восстановления, инструкция для администратора. Объём практических занятий – 4 ч.

Практическое занятие № 17. Документация интеграции: схемы, интерфейсы, сообщения (2 ч)

Цель работы: подготовить документацию интеграции.

Краткие теоретические сведения.

Документация включает схему обмена, описание интерфейсов (методы, параметры, форматы), структуры сообщений с примерами, правила преобразования, описание ошибок и реакций, требования к безопасности, контактные данные ответственных.

Задания.

1. Подготовьте документацию интеграции: схема обмена, описание API, структуры сообщений, обработка ошибок.

2. Опишите интерфейсы учебного проекта в формате OpenAPI (YAML).
3. Проверьте документацию: другой студент выполняет тестовый запрос по описанию.

Контрольные вопросы.

1. Как документируются интеграционные решения?
2. Что нужно документировать в интеграции?
3. Что такое OpenAPI?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

Практическое занятие № 18. Регламент обмена, восстановление и инструкция администратора (2 ч)

Цель работы: подготовить регламент обмена, порядок восстановления и инструкцию администратору.

Краткие теоретические сведения.

Регламент определяет периодичность и объёмы обмена, ответственных, действия при отказах и порядок повторной загрузки. Порядок восстановления описывает, как повторить обмен без дублирования. Инструкция администратора – запуск, остановка, проверка, обновление ключей, диагностика.

Задания.

1. Составьте регламент обмена и порядок восстановления при сбое.
2. Подготовьте инструкцию администратора по сопровождению интеграции (ключи, журналы, перезапуск, типовые ошибки).
3. Подготовьте итоговый комплект документации и защитите решение.

Контрольные вопросы.

1. Как обновляется интеграция при изменении API внешнего сервиса?
2. Что включает порядок восстановления?
3. Что должно быть в инструкции администратора?

Отчёт: описание сервиса и схемы обмена, запросы и ответы (без секретов), листинги, результаты проверок, ответы на контрольные вопросы.

3. Методические указания к самостоятельной работе

Объём самостоятельной работы – 4 ч. Задания выполняются в свободное от занятий время; результаты сдаются в электронном виде и проверяются на очередном практическом занятии.

Таблица 2 – Задания для самостоятельной работы

№	Тема	Задание	Время, ч
1	1.2. Документация API	Изучите документацию двух открытых API (например, курсы валют, геокодирование) и подготовьте сравнительную таблицу: методы, параметры, аутентификация, лимиты, форматы; составьте по три примера запросов и ответов.	2
2	3.1. Публикация	Подготовьте инструкцию по публикации учебного ресурса (чек-лист, переменные окружения, DNS, HTTPS, проверка интеграций) и выполните публикацию на тестовом сервере.	2

4. Рекомендуемая литература и информационные ресурсы

Основная литература

1. Сергеев, А. Н. Основы локальных компьютерных сетей: учебное пособие для СПО / А. Н. Сергеев. – 6-е изд., стер. – Санкт-Петербург: Лань, 2026. – 184 с. – ISBN 978-5-507-55422-5. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/527792
2. Компьютерные сети: учебник для СПО / Д. А. Бархатова, Д. Н. Буторин, А. А. Левин [и др.]. – Санкт-Петербург: Лань, 2025. – 304 с. – ISBN 978-5-507-51753-4. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/460619

Дополнительная литература

1. Акмаров, П. Б. Компьютерные сети. Лабораторный практикум: учебное пособие для СПО / П. Б. Акмаров. – 2-е изд., стер. – Санкт-Петербург: Лань, 2026. – 120 с. – ISBN 978-5-507-54543-8. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/509334
2. Золкин, А. Л. Организация, принципы построения и функционирования компьютерных сетей: учебник для СПО / А. Л. Золкин, В. Д. Мунистер. – 2-е изд., стер. – Санкт-Петербург: Лань, 2026. – 212 с. – ISBN 978-5-507-56689-1. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/518572

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Docs.CNTD.ru. – URL: https://docs.cntd.ru/
2. КиберЛенинка. – URL: https://cyberleninka.ru

Приложение А. Образец титульного листа отчёта

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО Университет биотехнологий
Институт цифровых технологий
Кафедра информационных технологий и моделирования

Отчёт по практическому занятию № …

по дисциплине «Интеграция информационных ресурсов с другими системами в сети Интернет»

Тема: …

Выполнил(а): студент(ка) … курса, группы …
ФИО

Проверил: должность,
ФИО

Новосибирск 202…