ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 10
СБОРКА ИНТЕГРИРОВАННОГО ПРОТОТИПА

Дисциплина: CSE842 «Capstone Project». **Образовательная программа:** 6B06102 «Computer Science», 4 курс, 8 семестр. **Связь с лекцией:** лекция 10 «Интеграция подсистем». **Результаты обучения:** РО4, РО5. **Объём:** 3 часа практики, 7 часов самостоятельной работы, включая СРСП. **Максимальный балл:** 5.

Цель занятия: соединить подсистемы прототипа, показать сквозной сценарий и зафиксировать результаты интеграции.

Задачи:

1. Выполнить интеграцию по плану недели 7 с записью в журнал.
2. Реализовать асинхронную связь подсистем варианта.
3. Написать интеграционные и сквозной тесты и включить их в CI.
4. Показать полный сквозной сценарий на версии из репозитория.

Программное обеспечение и инструменты: Git, CI, Docker и docker compose (рекомендуется), Playwright или Selenium (сквозной тест), имитация SMTP (например, MailHog).

1. Краткие теоретические сведения

Синхронные вызовы используют тайм-ауты и повторяют только идемпотентные операции; асинхронные — очередь или transactional outbox с гарантией «как минимум один раз» и идемпотентным получателем. Интеграционный тест проверяет взаимодействие через реальные интерфейсы, сквозной — сценарий пользователя, дымовой — запуск после развёртывания.

Шаг CI для сквозного теста:

```
- run: docker compose up -d
- run: python scripts/seed_demo.py
- run: python -m pytest tests/e2e -q
- run: docker compose down
```

Таблица 1 — Шаблон журнала интеграции

Дата	Версия	Интегрировано	Результат и проблемы
24.03	a91f2c0	ядро + БД	сценарий через тест-драйвер: успех
25.03	3c07d11	+ HTTP API	часовой пояс: исправлено (#31)
28.03	v0.10	+ клиент	полный сценарий в CI

2. Порядок выполнения работы

1. Интегрируйте подсистемы по шагам плана; после каждого шага прогоняйте сценарий и пишите журнал.
2. Реализуйте асинхронную связь варианта (таблица 2) с идемпотентной обработкой.

3. Напишите интеграционный тест для каждого стыка и сквозной тест основного сценария.
4. Подключите тесты к CI; добейтесь зелёной сборки main.
5. Поднимите систему одной командой (docker compose или скрипт).
6. Покажите полный сценарий; перечислите оставшиеся заглушки и ограничения.
7. Обновите диаграммы, контракты и ADR, если при интеграции что-то изменилось; отметьте тег v0.10.

3. Индивидуальные задания

Номер варианта совпадает с номером темы проекта, выбранной на практическом занятии № 1, и сохраняется до конца семестра. Команда (до 4 человек) выполняет задание своего варианта; распределение работы между участниками указывается в отчёте.

Таблица 2 — Индивидуальные задания: интеграция

№	Асинхронная связь	Заменяемая заглушка/имитация	Сбой для интеграционного теста
1	запись → очередь → SMS	SMS-шлюз	SMS-шлюз недоступен 10 мин
2	отметки → пакетная выгрузка отчёта	API расписания	потеря сети у студента во время сканирования
3	MQTT-публикация → подписчик сервиса приёма	физический датчик (эмулятор)	брокер перезапущен под нагрузкой
4	API → очередь → воркер песочницы	GitHub API	воркер падает во время проверки
5	заказ → событие → пересчёт рекомендаций	платёжный шлюз	рекомендательный сервис недоступен
6	ежедневная задача → напоминания	Open Library API	сканер отправляет повторный код
7	обращение → очередь сотрудника	Telegram Bot API	Telegram API недоступен
8	кадры → очередь → распознавание	видеопоток (записанные файлы)	одна камера перестала отвечать
9	офлайн-очередь → синхронизация	учётная система	сканер 30 мин без сети
10	позиции → WebSocket-рассылка	ресторанная касса	разрыв WebSocket у клиента
11	поток журналов → очередь → детектор	продуктивный сервер (воспроизведение журнала)	всплеск журнала ×10

№	Асинхронная связь	Заменяемая заглушка/имитация	Сбой для интеграционного теста
12	фото → очередь OCR → результат	облачный OCR	нет сети при съёмке чека
13	согласование → уведомления	NCALayer	сервис генерации PDF недоступен
14	платёж → webhook → подтверждение	платёжный шлюз	webhook оплаты приходит дважды
15	событие MQTT → движок правил → команда	физические реле	брокер недоступен 1 мин
16	кадры → распознавание → журнал	шлагбаум	камера отдаёт тёмные кадры
17	опрос счётчиков → очередь → запись	счётчики Modbus	счётчик не отвечает 1 ч
18	завершение попытки → очередь проверки	LMS Moodle	перезапуск сервера во время теста
19	ежесуточное обновление GTFS	поставщик GTFS	новый фид GTFS с ошибками
20	изменения календаря → webhook → обновление	Google Calendar API	Google API возвращает 429

4. Содержание отчёта

1. Журнал интеграции.
2. Интеграционные и сквозной тесты, отчёт CI.
3. Демонстрация (или видео) полного сценария на версии с тегом.
4. Перечень заглушек и ограничений; обновлённые документы.

5. Форма сдачи

Артефакты занятия размещаются в репозитории проекта (каталог docs/ или указанный в задании), в трекере задач создаются и закрываются соответствующие задачи. Преподавателю сдаётся ссылка на версию (коммит или тег) и краткий отчёт.

6. Критерии оценивания

Максимальный балл за работу — **5** (календарный план силлабуса). Баллы распределяются по трём критериям силлабуса.

Таблица 3 — Критерии оценивания текущего задания

Критерий	Доля	Что проверяется
Полнота и соответствие требованиям задания	40%	все пункты задания и индивидуального варианта выполнены
Техническая корректность	40%	решения обоснованы, расчёты и код верны, терминология точна
Проверяемость материалов	20%	артефакты в репозитории, ссылки, команды запуска, данные

Работа сдаётся через канал, указанный преподавателем; в командном проекте каждый участник указывает свой вклад. Использование генеративного ИИ раскрывается по правилам университета.

7. Контрольные вопросы

1. Какие способы связи подсистем вы используете и почему?
2. Как работает outbox и что такое идемпотентный получатель?
3. Какие проблемы обнаружены при интеграции?
4. Чем интеграционный тест отличается от сквозного?
5. Как система ведёт себя при сбое из таблицы 2?

Литература

1. NASA. Systems Engineering Handbook. NASA/SP-2016-6105 Rev2.
2. OpenAPI Initiative. OpenAPI Specification v3.1. — spec.openapis.org.
3. Fowler M. Continuous Integration. — martinfowler.com.