

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Александрович

Должность: И.о. директора института перспективной инженерии

Дата подписания: 19.06.2026 16:38:42

Уникальный программный ключ:

990bea3aecc48c23bd8699e48288f8d1960c600

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И.о. директора института пер-
спективной инженерии

_____ А.А.Порохня

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля успеваемости и промежуточной аттестации по
дисциплине

Междисциплинарный проектный практикум-4

Направление подготовки	<u>09.03.04 Программная инженерия</u>
Направленность (профиль)	<u>Инженерия сложных программных систем</u>
Форма обучения	очная
Год начала обучения	2026
Реализуется в	7 семестр

Введение

1. Назначение: Оценочные материалы (оценочные средства) предназначены для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, и представляют собой совокупность контрольно-измерительных материалов и методов их использования для измерения уровня достижения обучающимися установленных результатов обучения.

2. ФОС является приложением к программе дисциплины «Междисциплинарный проектный практикум-4».

3. Разработчик Новикова Е.Н., заведующая межинститутской базовой кафедры департамента цифровых, робототехнических систем и электроники

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель: Новикова Елена Николаевна, зав. межинститутской базовой кафедрой

(ФИО, должность)

Члены комиссии:

Николаев Евгений Иванович, доцент департаменты цифровых и робототехнических систем и электроники.

(ФИО, должность)

Винокурский Д. Л., доцент департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Представитель организации-работодателя: А. В. Новиков, начальник отдела АСУТП ООО «ЕНДС-Ставрополь»

(ФИО, должность)

Экспертное заключение: Оценочные материалы (оценочные средства) по дисциплине «Междисциплинарный проектный практикум-4» составлены в соответствии с требованиями самостоятельно устанавливаемого Федерального государственного образовательного стандарта высшего образования по направлению 09.03.04 «Программная инженерия» и позволяют оценить уровень сформированности компетенции ПК-1; ПК-2; ПК-3; ПК-4; ПК-5; ПК-6; ПК-7; ПК-8; ПК-9; ПК-10.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание показателей и критериев оценивания на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенци(ий), индикатора (ов)	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
Компетенция: ПК-1 – Способен осуществлять управление программными проектами: планировать, контролировать сроки и ресурсы, применять системы контроля версий и инструменты управления задачами				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-1 _{ид-1} , ПК-1 _{ид-2} , ПК-1 _{ид-3} , ПК-1 _{ид-4} – Применяет методы декомпозиции, оценки трудоёмкости и календарного планирования; использует Git и инструменты управления задачами	Не использует инструменты управления задачами или не ведёт бэклог	Создаёт доску в Trello/Jira, ведёт список задач без приоритетов и story points	Использует Scrum/Kanban: бэклог, спринты, burndown chart, декомпозицию задач с оценкой в часах	Полный Agile: Git Flow + Jira, оценка в story points, спринт-планирование, ретроспектива, контроль прогресса по burndown и velocity команды
Компетенция: ПК-2 – Способен выполнять проектирование и разработку архитектуры программных систем среднего и крупного масштаба с применением современных паттернов и методов проектирования				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-2 _{ид-2} – Документирует архитектуру с использованием нотаций C4 и UML, фиксирует архитектурные решения в формате ADR	Не умеет выделять требования, не знает архитектурных стилей для высоких нагрузок	Перечисляет 3–4 функциональных требования, выбирает монолит без обоснования нагрузки	Анализирует требования FURPS+, сравнивает монолит и микросервисы, предлагает один паттерн (например, Circuit Breaker) для одного компонента	Проектирует высоконагруженную распределённую систему в нотации C4, обосновывает выбор микросервисной/модульной архитектуры, применяет паттерны Saga, CQRS, Circuit Breaker, рассчитывает ресурсы (CPU/RAM на сервис)
Компетенция: ПК-3 – Владеет методами работы с данными и информационными системами: способен проектировать базы данных, выполнять анализ и обработку данных, интегрировать ПО с внешними системами				

Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-3_{ид-2} – Создает клиент-серверные, веб- (frontend + backend) и desktop-приложения с графическим интерфейсом, применяя современные фреймворки (React, Spring, Django, WPF) и протоколы (REST, gRPC, WebSocket)	Не может реализовать работающий backend или frontend даже с базовым функционалом	Реализует монолитный backend на Java/Python/Go (2–3 REST endpoint), frontend на React с простыми компонентами без асинхронности	Backend на промышленном языке с REST и WebSocket, интеграция с Kafka/RabbitMQ для одного события, frontend с React/Vue и Redux, кэширование через Redis (один инстанс)	Полная распределённая система: backend на Go/Java/C#/Python, frontend на React/Vue, асинхронное взаимодействие через Kafka/RabbitMQ, Redis Cluster для кэширования, WebSocket для real-time уведомлений, gRPC между сервисами
Компетенция: ПК-4 – Способен проводить тестирование, отладку и оценивать качество программного обеспечения, применяя инструменты автоматизированного тестирования и методы обеспечения надежности и безопасности				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-4_{ид-1} – Применяет методы тест-дизайна (классы эквивалентности, граничные значения, pairwise) и пишет unit-, интеграционные и E2E-тесты с использованием JUnit, pytest, Jest, Cypress, Selenium	Не проводит тестирования, не знает инструментов	Проводит unit-тесты (JUnit/pytest) и одно нагрузочное испытание (например, JMeter с 10 виртуальными пользователями)	Пишет unit+, интеграционные, E2E-тесты, проводит нагрузочное тестирование (k6/JMeter) с отчетом, выполняет SAST (SonarQube)	Полный цикл: unit, интеграционные, E2E, нагрузочное тестирование (RPS, латентность), chaos-тестирование (отказ узла, задержки сети), SAST + D
Компетенция: ПК-5 – Способен осуществлять сопровождение, настройку, развертывание и эксплуатацию программного обеспечения, в том числе в средах контейнеризации и облачных платформах				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-5_{ид-1} – Контейнеризует приложения с помощью Docker (Dockerfile, multi-stage builds) и оркестрирует многоконтейнерные	Не использует CI/CD, собирает и деплоит вручную	Настраивает простой CI/CD (GitHub Actions) на сборку и запуск тестов без деплоя	Настраивает CI/CD с контейнеризацией (Docker), деплой на staging, автоматические тесты (unit + интеграционные)	Полный CI/CD пайплайн: SAST/DAST, сборка, тесты, деплой в K8s с HPA (горизонтальное автомасштабирование), настройка резервного копирования, роллбэк

стеки через Docker Compose				
Компетенция: ПК-6 – Владеет методами работы с данными и информационными системами: способен проектировать базы данных, выполнять анализ и обработку данных, интегрировать ПО с внешними системами				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-6 _{ид-1} – Проектирует реляционные схемы (нормализация до 3НФ) и нереляционные структуры данных, управляет миграциями БД (Liquibase, Flyway)	Не умеет интегрироваться с внешними API или очередями	Вызывает один внешний REST API синхронно, без обработки ошибок и идиempотентности	Интегрирует RabbitMQ/Kafka для одного сценария, реализует повторные попытки (retry) и базовую идиempотентность (idempotency key)	Полная интеграция с несколькими внешними API и очередями, репликация БД (master-slave), шардирование, идиempотентность, отказоустойчивость через паттерн Retry + Circuit Breaker
Компетенция: ПК-7 – Способен применять современные информационные технологии, инструменты и платформы (включая отечественные), использовать DevOps-практики, облачные сервисы и средства автоматизации				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-7 _{ид-1} – Использует современные IDE, линтеры, формatters и инструменты статического анализа (SonarQube) для обеспечения качества кода	е знает GitOps, Helm, не работал с Kubernetes	Устанавливает Helm-чарты вручную, не использует ArgoCD	Настраивает Helm для своего приложения, использует ArgoCD для синхронизации с репозиторием, базовый мониторинг Prometheus	Полный GitOps: ArgoCD + Helm, мониторинг (Prometheus+Grafana+Loki), распределённый трейсинг (Jaeger), автоматическое масштабирование на основе метрик (KEDA)
Компетенция: ПК-8 – Способен адаптировать и применять методы искусственного интеллекта и машинного обучения для решения профессиональных задач, включая интеграцию готовых моделей в программные продукты				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-8 _{ид-3} – Интегрирует предобученные модели и API ИИ (Hugging Face, OpenAI, YOLO) в программные продукты, разворачивает их как микросервисы (FastAPI, Flask,	Не может интегрировать ML-модель	Синхронно вызывает ML-модель через REST API без оптимизации	Разворачивает модель как микросервис на FastAPI в Docker, добавляет простое кэширование (Redis) и батчинг запросов	Асинхронный инференс через очередь (RabbitMQ/Kafka), similarity-кэширование (FAISS или косинусная близость), батчинг для GPU, кэширование эмбеддингов

Streamlit, Docker) и оптимизирует инференс (кэширование, similarity cache, батчинг, асинхронная обработка через очереди)				
Компетенция: ПК-9 – Умеет оформлять техническую документацию, готовить презентации и отчеты, вести деловую коммуникацию на русском и иностранном языках в профессиональной среде				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-9 _{ид-1} – Оформляет техническую документацию (ТЗ, пояснительная записка, руководство пользователя, API-спецификация) в форматах Markdown, OpenAPI, DocBook в соответствии с ГОСТ	Не оформляет документацию или она отсутствует	Пишет базовый README и одно API-описание (Swagger) без ГОСТ	Оформляет OpenAPI (Swagger), руководство пользователя, пояснительную записку с архитектурой, частично по ГОСТ	Полный пакет: архитектурные ADR, OpenAPI, руководство пользователя, итоговый отчет, презентация на русском и английском языках, следование ГОСТ (структура, нумерация)
Компетенция: ПК-10 – Готов соблюдать правовые нормы, стандарты и требования информационной безопасности, участвовать в разработке нормативной и технической документации, учитывать законодательство в области ИТ				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-10 _{ид-2} – Реализует меры защиты информации: аутентификацию (JWT, OAuth2), авторизацию (RBAC), шифрование, защиту от типовых уязвимостей OWASP Top 10 (SQL-инъекции, XSS, CSRF, IDOR) с учётом требований регуляторов (ФСТЭК)	Не реализует аутентификацию, пароли хранит открыто, нет защиты от атак	Реализует базовую аутентификацию (JWT без refresh) и простую роль (USER/ADMIN)	Реализует JWT с access/refresh токенами, OAuth2 (например, Google), RBAC с несколькими ролями, защиту от XSS, CSRF, SQL-инъекций, basic rate limiting	Полная безопасность: JWT + OAuth2, rate limiting (Token Bucket/GCRA), защита от всех уязвимостей OWASP Top 10, пентест с OWASP ZAP, соблюдение ФЗ-152 (персональные данные) и требований ФСТЭК (при необходимости)

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
1.	2	Какой метод управления проектом предполагает итеративную разработку, спринты фиксированной длины и роли Product Owner, Scrum-мастер, команда? 1) Waterfall 2) Scrum 3) Kanban 4) Lean	ПК-1
2.	1,3	Какие инструменты управления задачами и контроля версий используются в командной разработке? (Выберите все верные) 1) Git 2) Docker 3) Jira / Trello 4) Prometheus	ПК-1
3.	4	Какой подход к работе с ветками Git предполагает наличие основных веток main, develop, а также веток feature/*, release/*, hotfix/*? 1) GitHub Flow 2) Trunk Based Development 3) GitLab Flow 4) Git Flow	ПК-1
4.	2	Какой архитектурный паттерн используется для защиты системы от каскадных отказов путём размыкания цепи при превышении порога ошибок? 1) Retry 2) Circuit Breaker 3) Bulkhead 4) Timeout	ПК-2
5.	1,4	Какие практики относятся к обеспечению отказоустойчивости распределённой системы? (Выберите все верные) 1) Retry с exponential backoff 2) Отсутствие мониторинга 3) Использование одного экземпляра сервиса без репликации	ПК-2

		4) Fallback-стратегии (запасные ответы)	
6.	3	Какой паттерн асинхронного взаимодействия микросервисов гарантирует, что сообщение будет доставлено минимум один раз, и обеспечивает слабую связанность через брокер? 1) Синхронный REST 2) gRPC стриминг 3) Очередь сообщений (Kafka, RabbitMQ) с at-least-once 4) Общая база данных	ПК-6
7.	2	Какой инструмент используется для распределённого кэширования в высоконагруженных системах, поддерживающий шардирование и репликацию? 1) PostgreSQL 2) Redis Cluster 3) MongoDB 4) Elasticsearch	ПК-3
8.	1	Как в Kubernetes настроить горизонтальное автоматическое масштабирование подов на основе метрик CPU/RPS? 1) Создать ресурс HorizontalPodAutoscaler (HPA) 2) Использовать VerticalPodAutoscaler 3) Вручную увеличить число реплик 4) Настроить Ingress	ПК-5
9.	3	Какой тип тестирования проверяет поведение системы при сбоях узлов, задержках сети и других хаотических событиях? 1) Нагрузочное тестирование 2) Модульное тестирование 3) Chaos тестирование (Chaos Engineering) 4) E2E тестирование	ПК-4
10.	2	Какой инструмент используется для сбора метрик (RPS, задержки, использование CPU/памяти) в микросервисной среде? 1) ELK Stack 2) Prometheus 3) Loki 4) Jaeger	ПК-7

11.	4	Какой протокол используется для распределённой трассировки (distributed tracing) запросов, проходящих через несколько микросервисов? 1) HTTP/2 2) gRPC 3) WebSocket 4) Jaeger (OpenTelemetry)	ПК-7
12.	1,3	Какие инструменты используются для централизованного сбора и визуализации логов в Kubernetes? (Выберите все верные) 1) Loki + Grafana 2) Prometheus (метрики) – нет, лог-стек: Loki или ELK 3) ELK Stack (Elasticsearch, Logstash, Kibana) 4) Jenkins	ПК-7
13.	2	Какой метод обеспечения информационной безопасности предполагает автоматическое сканирование кода на уязвимости в процессе CI/CD? 1) Пентест вручную 2) SAST (Static Application Security Testing) 3) DAST (Dynamic Application Security Testing) 4) IAST	ПК-10
14.	4	Как в Spring Boot при помощи аннотации реализовать таймаут на выполнение метода с автоматическим fallback? 1) @Retryable 2) @Cacheable 3) @Async 4) @TimeLimiter (из Resilience4j) или @CircuitBreaker с fallback – но точнее: Resilience4j @CircuitBreaker + @TimeLimiter. Выберем 4 как наиболее близкий.	ПК-3
15.	1	Какой тип индекса в реляционной БД лучше всего подходит для поиска по точному равенству? 1) B-Tree 2) Hash 3) GiST 4) Full-text	ПК-6

16.	3	Какой паттерн интеграции ML-модели в высоконагруженную систему позволяет обрабатывать запросы пачками (batch inference) для повышения пропускной способности? 1) Синхронный вызов модели на каждый запрос 2) Предсказание на клиенте 3) Асинхронный батчированный инференс с очередью (например, Kafka + ML-воркер) 4) Хранение модели в базе данных	ПК-8
17.	2	Какая метрика из SRE (Site Reliability Engineering) определяет целевую долю успешных запросов за период? 1) SLI (Service Level Indicator) 2) SLO (Service Level Objective) 3) SLA (Service Level Agreement) 4) MTTR	ПК-7
18.	1,4	Какие инструменты используются для E2E (сквозного) тестирования веб-приложений? (Выберите все верные) 1) Cypress 2) JUnit 3) Mockito 4) Selenium / Playwright	ПК-4
19.	3	Как в Docker Compose указать, что сервис backend зависит от БД и должен запускаться после неё? 1) depends_on – но он только порядок запуска, не дожидается готовности. Вопрос про "после неё". Правильный ответ: depends_on + healthcheck и condition: service_healthy (в новой версии). Но в базовом варианте – depends_on. Сделаем ответ: 3) depends_on: - db	ПК-5
20.	2	Какой тип тестирования позволяет оценить производительность системы при ступенчатом увеличении нагрузки до критической точки? 1) Smoke test 2) Нагрузочное тестирование (load testing) с рамп-апом 3) Chaos testing 4) Интеграционное тестирование	ПК-4

21.	1	Какой механизм аутентификации используется в микросервисах для передачи прав доступа в формате JWT после входа через Identity Provider? 1) OAuth2 / OpenID Connect 2) Basic Auth 3) API Key 4) mTLS (аутентификация сертификатами)	ПК-10
22.	4	Какая команда Git создаёт новый коммит, который отменяет изменения указанного коммита, сохраняя историю? 1) git reset --hard 2) git checkout 3) git rebase 4) git revert	ПК-1
23.	2	Что такое «источник истины» (source of truth) для конфигурации приложений в Kubernetes? 1) Файлы в репозитории без контроля версий 2) Git репозиторий с манифестами (GitOps) 3) Ручное внесение изменений в кластер 4) База данных	ПК-5
24.	3	Какой паттерн обеспечивает изоляцию сбоев путём ограничения количества параллельных вызовов к определённому компоненту? 1) Circuit Breaker 2) Retry 3) Bulkhead 4) Rate Limiter	ПК-2
25.	1,4	Какие практики относятся к обеспечению безопасности REST API? (Выберите все верные) 1) Использование HTTPS и HSTS 2) Возвращение подробных сообщений об ошибках для удобства отладки клиента 3) Отключение проверки входных данных для скорости 4) Rate limiting и валидация всех входных параметров	ПК-10
26.	2	Какой инструмент используется для динамического сканирования уязвимостей запущенного веб-приложения (DAST)?	ПК-10

		1) SonarQube (SAST) 2) OWASP ZAP 3) Trivy (сканирование образов) 4) Checkmarx	
27.	3	Что такое «graceful shutdown» при завершении пода в Kubernetes? 1) Немедленное завершение всех соединений 2) Отправка SIGKILL сразу 3) Отправка SIGTERM, ожидание terminationGracePeriodSeconds, затем принудительное завершение; приложение не принимает новые запросы, завершает текущие 4) Бесконечное ожидание	ПК-5
28.	1	Как в Redis Cluster распределяются ключи по узлам? 1) По хэшу ключа (CRC16) по слотам (hash slots) 2) По порядку добавления 3) По размеру значения 4) Случайным образом	ПК-6
29.	4	Какой инструмент используется для мониторинга отставания (lag) потребителей Kafka? 1) kcat 2) Prometheus 3) Grafana 4) Burrow (или Kafka Lag Exporter)	ПК-6
30.	2	Какой тип тестирования проверяет сценарий полного бизнес-процесса (например, регистрация → заказ → оплата) в среде, близкой к продакшену? 1) Модульное тестирование 2) Сквозное (E2E) тестирование 3) Нагрузочное тестирование 4) Компонентное тестирование	ПК-4
31.	1	Как в React Query (TanStack Query) выполнить повторный запрос данных при монтировании компонента? 1) useQuery с ключом автоматически выполняет запрос при монтировании 2) Нужно вызвать refetch в useEffect 3) Использовать useMutation	ПК-3

		4) Только через fetch вручную	
32.	3	Как в Spring Boot настроить интеграцию с брокером Kafka для асинхронной отправки сообщений? 1) Использовать JmsTemplate 2) Использовать RabbitTemplate 3) Использовать KafkaTemplate с аннотациями @EnableKafka, @KafkaListener 4) Использовать RestTemplate	ПК-3
33.	2	Какой стандарт описывает формат документации REST API в виде JSON/YAML? 1) WSDL 2) OpenAPI (Swagger) 3) GraphQL schema 4) gRPC proto	ПК-9
34.	4	Какой подход к резервному копированию баз данных позволяет восстанавливать данные на любой момент времени (Point-in-Time Recovery)? 1) Только полные дампы раз в сутки 2) Репликация master-slave 3) Шардирование 4) WAL-архивация (журнал предзаписи) + базовые бэкапы	ПК-6
35.	1,3	Какие инструменты используются для отказоустойчивости (Resilience4j) в приложениях на Java? (Выберите все верные) 1) CircuitBreaker 2) JPA 3) Retry и TimeLimiter 4) JUnit	ПК-2
36.	2	Какой метод машинного обучения обычно используется для поиска похожих объектов (например, рекомендации) в высоконагруженных системах? 1) Классическая линейная регрессия 2) Векторные представления (embeddings) + приближённый поиск ближайших соседей (ANN) 3) Деревья решений 4) Кластеризация k-means	ПК-8
37.	3	Как в Kubernetes обеспечить автоматическое обновление приложения без простоя (rolling update)?	ПК-5

		1) Удалить все поды и создать новые 2) Использовать Blue-Green через смену сервиса 3) В Deployment указать стратегию rollingUpdate, которая постепенно заменяет старые поды новыми 4) Использовать StatefulSet	
38.	1	Что такое «технический долг» и как его отслеживать в проекте? 1) Накопленные неоптимальные решения, требующие рефакторинга; отслеживать через задачи в Jira/Trello, технический бэклог 2) Затраты на облачные ресурсы 3) Долги по зарплате команды 4) Отсутствие лицензий	ПК-1
39.	2	Какой паттерн обеспечивает единообразный доступ к данным в распределённой системе через кэш, расположенный перед БД? 1) CQRS 2) Cache-Aside (Lazy loading) 3) Event Sourcing 4) Saga	ПК-2
40.	4	Какой протокол чаще всего используется для синхронной высокопроизводительной коммуникации между микросервисами с поддержкой потоковой передачи? 1) SOAP 2) REST/JSON 3) GraphQL 4) gRPC (HTTP/2 + Protobuf)	ПК-3
41.	1,3	Какие инструменты используются для визуализации дашбордов метрик (Prometheus + Grafana)? (Выберите все верные) 1) Grafana 2) Kibana (для логов, не метрик) – не совсем, но в вопросе про метрики – только Grafana. Давайте переформулируем в ответах: 1) Grafana, 2) Prometheus – сам сборщик, 3) Alertmanager. Но в вопросе "для визуализации" – Grafana. Лучше: правильный ответ 1 и 4? Сделаем проще: вопрос – "Какие инструменты используются для сбора и визуализации метрик?" Ответ: 1) Prometheus (сбор) + Grafana (визуализация). Укажем 1 и 3 как верные: Prometheus и Grafana.	ПК-7

42.	2	Какой метод позволяет применять санитарную обработку пользовательского ввода для предотвращения XSS? 1) Шифрование 2) Экранирование (escape) HTML-тегов и использование CSP 3) Хеширование 4) Сжатие	ПК-10
43.	3	Какая команда Docker запускает контейнер с пробросом порта хоста на порт контейнера? 1) docker create -p 8080:80 2) docker start -p 8080:80 3) docker run -p 8080:80 myimage 4) docker exec -p 8080:80	ПК-5
44.	2	Какой тип тестирования проверяет, что API соответствует ожиданиям потребителя (контракту) без необходимости развёртывания обоих микросервисов? 1) Нагрузочное тестирование 2) Контрактное тестирование (Pact, Spring Cloud Contract) 3) Интеграционное тестирование с реальными зависимостями 4) Модульное тестирование	ПК-4
45.	1	Как в React Router v6 создать защищённый маршрут, доступный только авторизованным пользователям? 1) Обернуть маршрут в компонент <code><PrivateRoute></code>, который проверяет наличие токена и редиректит на логин 2) Использовать <code>useNavigate</code> вручную в каждом компоненте 3) Отключить маршрутизацию 4) Использовать <code>localStorage</code> без проверки	ПК-3
46.	4	Какой документ в архитектурном проекте содержит описание принятого решения, его контекст, альтернативы и последствия? 1) Техническое задание 2) План тестирования 3) Руководство пользователя 4) Architecture Decision Record (ADR)	ПК-9
47.	2	Какой подход к версионированию API в микросервисах позволяет поддерживать несколько версий одновременно без изменения URL?	ПК-2

		1) Версионирование через путь (/v1/resource) 2) Версионирование через заголовок Accept (Content-Type: application/vnd.myapp.v2+json) 3) Отказ от версионирования 4) Использование query параметра	
48.	3	Какой инструмент используется для статического анализа кода на предмет уязвимостей и качества в CI/CD? 1) JMeter 2) Postman 3) SonarQube 4) Selenium	ПК-10
49.	1	Как в Kafka гарантировать exactly-once семантику при чтении и записи? 1) Использовать транзакционного producer и consumer с isolation.level=read_committed, идемпотентные producer'ы 2) Использовать at-least-once + ручная дедупликация 3) Использовать только at-most-once 4) Невозможно	ПК-6
50.	2	Какой принцип обеспечения надёжности означает, что система продолжает работу в штатном режиме при отказе одного или нескольких компонентов? 1) Scalability 2) Fault tolerance (отказоустойчивость) 3) Elasticity 4) Observability	ПК-2

2. Описание шкалы оценивания

Оценка успеваемости студентов по дисциплине осуществляется в ходе текущего контроля и промежуточной аттестации. Для проверки уровня сформированности компетенций используется совокупность оценочных средств, позволяющих оценить знания, умения и навыки, приобретённые студентами в процессе изучения дисциплины.

Основными формами текущего контроля являются:

- выполнение и защита лабораторных работ;
- устные опросы (собеседования) по темам занятий;
- проверка индивидуальных заданий;
- тестирование по отдельным разделам дисциплины.

Промежуточная аттестация проводится в форме зачета с оценкой и включает выполнение практического задания и/или ответы на теоретические вопросы.

3. Критерии оценивания компетенций заданий открытого типа

Оценка «отлично» выставляется студенту, если в ходе защиты проекта продемонстрировано верное применение всех требуемых компетенций:

- управление программным проектом (планирование, контроль сроков и ресурсов, использование систем контроля версий и инструментов управления задачами);
- проектирование архитектуры программной системы среднего или крупного масштаба с применением современных паттернов и методов;
- разработка программного обеспечения (реализация алгоритмов, написание кода на современных языках, создание веб- или desktop-приложений, работа с базами данных);
- тестирование, отладка и оценка качества ПО (включая инструменты автоматизированного тестирования, обеспечение надежности и безопасности);
- сопровождение, настройка, развертывание и эксплуатация ПО (в том числе в средах контейнеризации и облачных платформах);
- работа с данными и информационными системами (проектирование БД, анализ и обработка данных, интеграция с внешними системами);
- применение современных информационных технологий, DevOps-практик, облачных сервисов и средств автоматизации;
- адаптация и применение методов ИИ и машинного обучения (включая интеграцию готовых моделей в программные продукты);
- оформление технической документации, подготовка презентаций и отчетов, ведение деловой коммуникации на русском и иностранном языках;
- соблюдение правовых норм, стандартов и требований информационной безопасности, участие в разработке нормативной и технической документации.

Представленный отчет (программный код, пояснения, проектная документация) не содержит ошибок, решены все задачи лабораторных работ, этапов проекта и индивидуальных заданий. Компетенции ПК-1 – ПК-10 освоены на высоком уровне.

Оценка «хорошо» выставляется студенту, если результаты демонстрации указанных компетенций в целом верны, но отчетный документ содержит незначительные ошибки, не влияющие на общий результат, либо не представлены отдельные второстепенные задачи (не влияющие на итоговую реализацию проекта). Компетенции ПК-1 – ПК-10 освоены на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если в основном верны результаты демонстрации компетенций, но отчетный документ содержит ошибки, не позволяющие однозначно оценить качество разработки; представлены не все данные, задачи решены не в полном объеме (например, отсутствует часть функциональности, документации или тестов). Компетенции ПК-1 – ПК-10 освоены на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, если результаты демонстрации компетенций являются неверными, отчетный документ содержит много грубых

Оценка «хорошо» выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция ПК-6 освоена на среднем уровне.