

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Порохня Андрей Александрович  
Должность: И.о. директора института перспективной инженерии  
Дата подписания: 19.06.2026 16:00:03  
Уникальный программный ключ:  
990bea3aeec48c23bd8699e48288f8d1960c600

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
**Федеральное государственное автономное образовательное учреждение  
высшего образования**  
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**УТВЕРЖДАЮ**

канд. физ-мат. наук, доцент,  
Новикова Е.Н.

**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>Разработка и сопровождение программного обеспечения</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                                                                                                                                                                                   |
| <b>Компетенция: ПК-1</b> – Способен осуществлять управление программными проектами: планировать, контролировать сроки и ресурсы, применять системы контроля версий и инструменты управления задачами                                                                                                      |                                                                                  |                                                                                      |                                                                                                                                             |                                                                                                                                                                                                                      |
| Результаты обучения по дисциплине (модулю):<br><i>Индикатор:</i><br>ПК-1 <sub>ид-1</sub> ,<br>ПК-1 <sub>ид-2</sub> ,<br>ПК-1 <sub>ид-3</sub> ,<br>ПК-1 <sub>ид-4</sub> – 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                                                                          |
| <b>Компетенция: ПК-2</b> – Способен выполнять проектирование и разработку архитектуры программных систем среднего и крупного масштаба с применением современных паттернов и методов проектирования                                                                                                        |                                                                                  |                                                                                      |                                                                                                                                             |                                                                                                                                                                                                                      |
| Результаты обучения по дисциплине (модулю):<br><i>Индикатор:</i><br>ПК-2 <sub>ид-2</sub> – 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) |
| <b>Компетенция: ПК-3</b> – Владеет методами работы с данными и информационными системами: способен проектировать базы данных, выполнять анализ и обработку данных, интегрировать ПО с внешними системами                                                                                                  |                                                                                  |                                                                                      |                                                                                                                                             |                                                                                                                                                                                                                      |

|                                                                                                                                                                                                                                                                                                               |                                                                                         |                                                                                                                                         |                                                                                                                                                                                 |                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Результаты обучения по дисциплине (модулю):</p> <p><i>Индикатор:</i></p> <p>ПК-3<sub>ид-2</sub> – Создает клиент-серверные, веб- (frontend + backend) и desktop-приложения с графическим интерфейсом, применяя современные фреймворки (React, Spring, Django, WPF) и протоколы (REST, gRPC, WebSocket)</p> | <p>Не может реализовать работающий backend или frontend даже с базовым функционалом</p> | <p>Реализует мо-нолитный backend на Java/Python/Go (2–3 REST endpoint), frontend на React с простыми компонентами без асинхронности</p> | <p>Backend на про-мышленном языке с REST и WebSocket, интеграция с Kafka/RabbitMQ для одного события, frontend с React/Vue и Redux, кэширование через Redis (один ин-станс)</p> | <p>Полная распре-делённая си-стема: backend на Go/Java/C#/Python, frontend на React/Vue, асинхронное взаимодействие через Kafka/RabbitMQ, Redis Cluster для кэширова-ния, WebSocket для real-time уведомлений, gRPC между сервисами</p> |
| <p><b>Компетенция: ПК-4</b> – Способен проводить тестирование, отладку и оценивать качество программного обеспечения, применяя инструменты автоматизированного тестирования и методы обеспечения надежности и безопасности</p>                                                                                |                                                                                         |                                                                                                                                         |                                                                                                                                                                                 |                                                                                                                                                                                                                                         |
| <p>Результаты обучения по дисциплине (модулю):</p> <p><i>Индикатор:</i></p> <p>ПК-4<sub>ид-1</sub> – Применяет методы тест-дизайна (классы эквивалентности, граничные значения, pairwise) и пишет unit-, интеграци-онные и E2E-те-сты с использова-нием JUnit, pytest, Jest, Cypress, Selenium</p>            | <p>Не проводит те-стирования, не знает инстру-ментов</p>                                | <p>Проводит unit-тесты (JUnit/pytest) и одно нагрузоч-ное испытание (например, JMeter с 10 вир-туальными пользо-вателями)</p>           | <p>Пишет unit+, интеграцион-ные, E2E-тесты, проводит нагрузочное те-стирование (k6/JMeter) с от-чётом, выпол-няет SAST (SonarQube)</p>                                          | <p>Полный цикл: unit, интеграци-онные, E2E, нагрузочное те-стирование (RPS, латент-ность), chaos-тестирование (отказ узла, за-держки сети), SAST + D</p>                                                                                |
| <p><b>Компетенция: ПК-5</b> – Способен осуществлять сопровождение, настройку, развертывание и эксплуатацию программного обеспечения, в том числе в средах контейнеризации и облачных платформах</p>                                                                                                           |                                                                                         |                                                                                                                                         |                                                                                                                                                                                 |                                                                                                                                                                                                                                         |
| <p>Результаты обучения по дисциплине (модулю):</p> <p><i>Индикатор:</i></p> <p>ПК-5<sub>ид-1</sub> – Кон-тейнеризует при-ложения с помо-щью Docker (Dockerfile, multi-stage builds) и ор-кестрирует мно-гоконтейнерные</p>                                                                                    | <p>Не использует CI/CD, соби-рает и деплоит вручную</p>                                 | <p>Настраивает простой CI/CD (GitHub Actions) на сборку и запуск тестов без де-плоя</p>                                                 | <p>Настраивает CI/CD с контей-неризацией (Docker), де-плой на staging, автоматические тесты (unit + ин-теграционные)</p>                                                        | <p>Полный CI/CD пайплайн: SAST/DAST, сборка, тесты, деплой в K8s с НРА (горизон-тальное авто-масштабирование), настройка резервного ко-пирования, рол-лбэк</p>                                                                          |

|                                                                                                                                                                                                                                              |                                                       |                                                                                   |                                                                                                                                  |                                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| стеки через Docker Compose                                                                                                                                                                                                                   |                                                       |                                                                                   |                                                                                                                                  |                                                                                                                                                                                  |
| <b>Компетенция: ПК-6</b> – 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):<br><i>Индикатор:</i><br>ПК-6 <sub>ид-1</sub> – 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 |
| <b>Компетенция: ПК-7</b> – Способен применять современные информационные технологии, инструменты и платформы (включая отечественные), использовать DevOps-практики, облачные сервисы и средства автоматизации                                |                                                       |                                                                                   |                                                                                                                                  |                                                                                                                                                                                  |
| Результаты обучения по дисциплине (модулю):<br><i>Индикатор:</i><br>ПК-7 <sub>ид-1</sub> – Использует современные IDE, линтеры, формatters и инструменты статического анализа (SonarQube) для обеспечения качества кода                      | не 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)                     |
| <b>Компетенция: ПК-8</b> – 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):<br><i>Индикатор:</i><br>ПК-8 <sub>ид-3</sub> – 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)                                                                                                                                                                                       |                                                                        |                                                                                |                                                                                                                                                       |                                                                                                                                                                                                                    |
| <b>Компетенция: ПК-9</b> – Умеет оформлять техническую документацию, готовить презентации и отчеты, вести деловую коммуникацию на русском и иностранном языках в профессиональной среде                                                                                                                        |                                                                        |                                                                                |                                                                                                                                                       |                                                                                                                                                                                                                    |
| Результаты обучения по дисциплине (модулю):<br><i>Индикатор:</i><br>ПК-9 <sub>ид-1</sub> – 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)                                             |
| <b>Компетенция: ПК-10</b> – 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):<br><i>Индикатор:</i><br>ПК-10 <sub>ид-2</sub> – 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?<br>1) Waterfall<br>2) Scrum<br>3) Kanban<br>4) Lean                                   | ПК-1        |
| 2.            | 1,3              | Какие инструменты управления задачами и контроля версий используются в командной разработке? (Выберите все верные)<br>1) Git<br>2) Docker<br>3) Jira / Trello<br>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/*?<br>1) GitHub Flow<br>2) Trunk Based Development<br>3) GitLab Flow<br>4) Git Flow               | ПК-1        |
| 4.            | 2                | Какой архитектурный паттерн используется для защиты системы от каскадных отказов путём размыкания цепи при превышении порога ошибок?<br>1) Retry<br>2) Circuit Breaker<br>3) Bulkhead<br>4) Timeout                                 | ПК-2        |
| 5.            | 1,4              | Какие практики относятся к обеспечению отказоустойчивости распределённой системы? (Выберите все верные)<br>1) Retry с exponential backoff<br>2) Отсутствие мониторинга<br>3) Использование одного экземпляра сервиса без репликации | ПК-2        |

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

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

|     |     |                                                                                                                                                                                                                                                                                                                                                                                                                                          |      |
|-----|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 16. | 3   | <p>Какой паттерн интеграции ML-модели в высоконагруженную систему позволяет обрабатывать запросы пачками (batch inference) для повышения пропускной способности?</p> <ol style="list-style-type: none"> <li>1) Синхронный вызов модели на каждый запрос</li> <li>2) Предсказание на клиенте</li> <li>3) Асинхронный батчированный инференс с очередью (например, Kafka + ML-воркер)</li> <li>4) Хранение модели в базе данных</li> </ol> | ПК-8 |
| 17. | 2   | <p>Какая метрика из SRE (Site Reliability Engineering) определяет целевую долю успешных запросов за период?</p> <ol style="list-style-type: none"> <li>1) SLI (Service Level Indicator)</li> <li>2) SLO (Service Level Objective)</li> <li>3) SLA (Service Level Agreement)</li> <li>4) MTTR</li> </ol>                                                                                                                                  | ПК-7 |
| 18. | 1,4 | <p>Какие инструменты используются для E2E (сквозного) тестирования веб-приложений? (Выберите все верные)</p> <ol style="list-style-type: none"> <li>1) Cypress</li> <li>2) JUnit</li> <li>3) Mockito</li> <li>4) Selenium / Playwright</li> </ol>                                                                                                                                                                                        | ПК-4 |
| 19. | 3   | <p>Как в Docker Compose указать, что сервис backend зависит от БД и должен запускаться после неё?</p> <ol style="list-style-type: none"> <li>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</li> </ol>                 | ПК-5 |
| 20. | 2   | <p>Какой тип тестирования позволяет оценить производительность системы при ступенчатом увеличении нагрузки до критической точки?</p> <ol style="list-style-type: none"> <li>1) Smoke test</li> <li>2) Нагрузочное тестирование (load testing) с рамп-апом</li> <li>3) Chaos testing</li> <li>4) Интеграционное тестирование</li> </ol>                                                                                                   | ПК-4 |

|     |     |                                                                                                                                                                                                                                                                                                                  |       |
|-----|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 21. | 1   | Какой механизм аутентификации используется в микросервисах для передачи прав доступа в формате JWT после входа через Identity Provider?<br>1) OAuth2 / OpenID Connect<br>2) Basic Auth<br>3) API Key<br>4) mTLS (аутентификация сертификатами)                                                                   | ПК-10 |
| 22. | 4   | Какая команда Git создаёт новый коммит, который отменяет изменения указанного коммита, сохраняя историю?<br>1) git reset --hard<br>2) git checkout<br>3) git rebase<br>4) git revert                                                                                                                             | ПК-1  |
| 23. | 2   | Что такое «источник истины» (source of truth) для конфигурации приложений в Kubernetes?<br>1) Файлы в репозитории без контроля версий<br>2) Git репозиторий с манифестами (GitOps)<br>3) Ручное внесение изменений в кластер<br>4) База данных                                                                   | ПК-5  |
| 24. | 3   | Какой паттерн обеспечивает изоляцию сбоев путём ограничения количества параллельных вызовов к определённому компоненту?<br>1) Circuit Breaker<br>2) Retry<br>3) Bulkhead<br>4) Rate Limiter                                                                                                                      | ПК-2  |
| 25. | 1,4 | Какие практики относятся к обеспечению безопасности REST API? (Выберите все верные)<br>1) Использование HTTPS и HSTS<br>2) Возвращение подробных сообщений об ошибках для удобства отладки клиента<br>3) Отключение проверки входных данных для скорости<br>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)<br>2) OWASP ZAP<br>3) Trivy (сканирование образов)<br>4) Checkmarx                                                                                                                                                                                                                                         |      |
| 27. | 3 | Что такое «graceful shutdown» при завершении пода в Kubernetes?<br>1) Немедленное завершение всех соединений<br>2) Отправка SIGKILL сразу<br>3) Отправка SIGTERM, ожидание terminationGracePeriodSeconds, затем принудительное завершение; приложение не принимает новые запросы, завершает текущие<br>4) Бесконечное ожидание | ПК-5 |
| 28. | 1 | Как в Redis Cluster распределяются ключи по узлам?<br>1) По хэшу ключа (CRC16) по слотам (hash slots)<br>2) По порядку добавления<br>3) По размеру значения<br>4) Случайным образом                                                                                                                                            | ПК-6 |
| 29. | 4 | Какой инструмент используется для мониторинга отставания (lag) потребителей Kafka?<br>1) kcat<br>2) Prometheus<br>3) Grafana<br>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?<br>1) Модульное тестирование<br>2) Сквозное (E2E) тестирование<br>3) Нагрузочное тестирование<br>4) Компонентное тестирование                                                        | ПК-4 |
| 31. | 1 | Как в React Query (TanStack Query) выполнить повторный запрос данных при монтировании компонента?<br>1) useQuery с ключом автоматически выполняет запрос при монтировании<br>2) Нужно вызвать refetch в useEffect<br>3) Использовать useMutation                                                                               | ПК-3 |

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

|     |     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |      |
|-----|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|     |     | 1) Удалить все поды и создать новые<br>2) Использовать Blue-Green через смену сервиса<br>3) В Deployment указать стратегию rollingUpdate, которая постепенно заменяет старые поды новыми<br>4) Использовать StatefulSet                                                                                                                                                                                                                                                                                                                                                                        |      |
| 38. | 1   | Что такое «технический долг» и как его отслеживать в проекте?<br>1) Накопленные неоптимальные решения, требующие рефакторинга; отслеживать через задачи в Jira/Trello, технический бэклог<br>2) Затраты на облачные ресурсы<br>3) Долги по зарплате команды<br>4) Отсутствие лицензий                                                                                                                                                                                                                                                                                                          | ПК-1 |
| 39. | 2   | Какой паттерн обеспечивает единообразный доступ к данным в распределённой системе через кэш, расположенный перед БД?<br>1) CQRS<br>2) Cache-Aside (Lazy loading)<br>3) Event Sourcing<br>4) Saga                                                                                                                                                                                                                                                                                                                                                                                               | ПК-2 |
| 40. | 4   | Какой протокол чаще всего используется для синхронной высокопроизводительной коммуникации между микросервисами с поддержкой потоковой передачи?<br>1) SOAP<br>2) REST/JSON<br>3) GraphQL<br>4) gRPC (HTTP/2 + Protobuf)                                                                                                                                                                                                                                                                                                                                                                        | ПК-3 |
| 41. | 1,3 | Какие инструменты используются для визуализации дашбордов метрик (Prometheus + Grafana)? (Выберите все верные)<br>1) Grafana<br>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 | <p>Какой метод позволяет применять санитарную обработку пользовательского ввода для предотвращения XSS?</p> <ol style="list-style-type: none"> <li>1) Шифрование</li> <li>2) Экранирование (escape) HTML-тегов и использование CSP</li> <li>3) Хеширование</li> <li>4) Сжатие</li> </ol>                                                                                                                                                                              | ПК-10 |
| 43. | 3 | <p>Какая команда Docker запускает контейнер с пробросом порта хоста на порт контейнера?</p> <ol style="list-style-type: none"> <li>1) docker create -p 8080:80</li> <li>2) docker start -p 8080:80</li> <li>3) docker run -p 8080:80 myimage</li> <li>4) docker exec -p 8080:80</li> </ol>                                                                                                                                                                            | ПК-5  |
| 44. | 2 | <p>Какой тип тестирования проверяет, что API соответствует ожиданиям потребителя (контракту) без необходимости развёртывания обоих микросервисов?</p> <ol style="list-style-type: none"> <li>1) Нагрузочное тестирование</li> <li>2) Контрактное тестирование (Pact, Spring Cloud Contract)</li> <li>3) Интеграционное тестирование с реальными зависимостями</li> <li>4) Модульное тестирование</li> </ol>                                                           | ПК-4  |
| 45. | 1 | <p>Как в React Router v6 создать защищённый маршрут, доступный только авторизованным пользователям?</p> <ol style="list-style-type: none"> <li>1) Обернуть маршрут в компонент <code>&lt;PrivateRoute&gt;</code>, который проверяет наличие токена и редиректит на логин</li> <li>2) Использовать <code>useNavigate</code> вручную в каждом компоненте</li> <li>3) Отключить маршрутизацию</li> <li>4) Использовать <code>localStorage</code> без проверки</li> </ol> | ПК-3  |
| 46. | 4 | <p>Какой документ в архитектурном проекте содержит описание принятого решения, его контекст, альтернативы и последствия?</p> <ol style="list-style-type: none"> <li>1) Техническое задание</li> <li>2) План тестирования</li> <li>3) Руководство пользователя</li> <li>4) Architecture Decision Record (ADR)</li> </ol>                                                                                                                                               | ПК-9  |
| 47. | 2 | <p>Какой подход к версионированию API в микросервисах позволяет поддерживать несколько версий одновременно без изменения URL?</p>                                                                                                                                                                                                                                                                                                                                     | ПК-2  |

|     |   |                                                                                                                                                                                                                                                                                                        |       |
|-----|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
|     |   | 1) Версионирование через путь (/v1/resource)<br>2) Версионирование через заголовок Accept (Content-Type: application/vnd.myapp.v2+json)<br>3) Отказ от версионирования<br>4) Использование query параметра                                                                                             |       |
| 48. | 3 | Какой инструмент используется для статического анализа кода на предмет уязвимостей и качества в CI/CD?<br>1) JMeter<br>2) Postman<br>3) SonarQube<br>4) Selenium                                                                                                                                       | ПК-10 |
| 49. | 1 | Как в Kafka гарантировать exactly-once семантику при чтении и записи?<br>1) Использовать транзакционного producer и consumer с isolation.level=read_committed, идемпотентные producer'ы<br>2) Использовать at-least-once + ручная дедупликация<br>3) Использовать только at-most-once<br>4) Невозможно | ПК-6  |
| 50. | 2 | Какой принцип обеспечения надёжности означает, что система продолжает работу в штатном режиме при отказе одного или нескольких компонентов?<br>1) Scalability<br>2) Fault tolerance (отказоустойчивость)<br>3) Elasticity<br>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 «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция ПК-10 не сформирована.