

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

Конструирование программного обеспечения

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	6 семестре

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 «Конструирование программного обеспечения».

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 «Конструирование программного обеспечения» 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-3, ПК-4.

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-3 – Владеет навыками разработки программного обеспечения: способен реализовывать алгоритмы, писать код на современных языках программирования, создавать веб- и desktop-приложения, работать с базами данных				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-3ид-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 знает метода АТАМ, не может назвать атрибуты качества	Знает основные атрибуты качества (производительность, безопасность), но не умеет формулировать сценарии по АТАМ	Формулирует 2–3 сценария атрибутов качества (стимул, источник, среда, ответ, мера), выявляет очевидные риски (например, узкое место БД)	Полноценно применяет АТАМ: формулирует 5+ сценариев, выявляет архитектурные риски и компромиссы (например, между производительностью и безопасностью), предлагает способы снижения рисков
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-3ид-4 – Пишет модульный, документированный и тестируемый код, соблюдая стандарты оформления и принципы SOLID, использует системы сборки (Make, Gradle, npm)	Не понимает понятие технического долга, не может его идентифицировать	Определяет 1–2 вида технического долга (например, дублирование кода), но не оценивает стоимость	Проводит инвентаризацию долга в небольшом проекте (3–5 элементов: устаревшие библиотеки, отсутствие тестов, «грязный» код), даёт грубую оценку времени (человеко-часы)	Полная инвентаризация архитектурного технического долга с классификацией по типам (код, архитектура, инфраструктура, документация), оценка стоимости исправления в деньгах/времени с учётом приоритетов
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 пишет тесты, не знает TestContainers	Пишет unit-тесты для 2–3 простых методов, покрытие <30%, не использует TestContainers	Пишет unit-тесты с покрытием 60–70%, использует TestContainers для интеграционных тестов (БД, Redis) на одном сценарии	Unit-тесты с покрытием $\geq 80\%$, интеграционные тесты с TestContainers для всех внешних зависимостей (БД, очереди, кэш), E2E-тесты для критических путей
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-4_{ид-2} – Использует инструменты отладки (GDB, IDE debugger), профилирования (perf, JProfiler) и нагрузочного тестирования (JMeter, k6) для выявления узких мест	Не умеет пользоваться JMeter и профилировщиками	Запускает JMeter с 10 потоками, получает базовые метрики (время отклика), не анализирует узкие места	Проводит нагрузочное тестирование с разной интенсивностью (ramp-up), выявляет точку насыщения, использует профилировщик (например, JProfiler) для поиска одного узкого места	Полное нагрузочное тестирование с моделированием реалистичной нагрузки, анализ процентилей, профилирование CPU и памяти, выявление и устранение узких мест (N+1 запрос, блокировки), предоставление отчёта с рекомендациями

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
1.	2	Какая техника тест-дизайна разбивает входные данные на группы, внутри которых поведение программы одинаково? 1) Граничные значения 2) Классы эквивалентности 3) Парное тестирование 4) Таблицы решений	ПК-4
2.	1,3	Какие значения являются граничными для функции, принимающей возраст от 0 до 120 лет? (Выберите все верные) 1) -1 2) 60 3) 0 4) 121	ПК-4
3.	3	Какой инструмент позволяет запускать реальные контейнеры с БД (PostgreSQL, MySQL) в модульных тестах без развёртывания внешней инфраструктуры? 1) JUnit 2) Mockito 3) Testcontainers 4) Selenium	ПК-4
4.	2	Какой тип тестирования измеряет время отклика системы при заданной нагрузке (например, 1000 одновременных пользователей)? 1) Модульное тестирование 2) Нагрузочное тестирование 3) Интеграционное тестирование 4) Приёмочное тестирование	ПК-4
5.	1	Какой инструмент профилирования позволяет анализировать использование CPU и памяти в работающем Java-приложении без изменения кода? 1) JProfiler / async profiler 2) JMeter 3) Selenium	ПК-4

		4) Postman	
6.	4	Какой «запах кода» описывает метод, который делает слишком много разных вещей и превышает 20-30 строк? 1) Duplicate code 2) Switch statement 3) Feature envy 4) Long method	ПК-3
7.	2	Какой рефакторинг рекомендуется при обнаружении повторяющегося кода? 1) Rename method 2) Extract method 3) Inline method 4) Move method	ПК-3
8.	1,4	Какие стратегии загрузки в JPA/Hibernate помогают решить проблему N+1 запроса? (Выберите все верные) 1) JOIN FETCH в JPQL 2) LAZY загрузка без изменений 3) Отключение кэша 4) @EntityGraph	ПК-3
9.	3	Какой алгоритм хеширования паролей рекомендуется использовать в современных приложениях? 1) MD5 2) SHA-1 3) BCrypt / Argon2 4) Base64	ПК-4
10.	2	Что такое JWT (JSON Web Token) в контексте безопасности? 1) Протокол аутентификации OAuth2 2) Компактный URL-безопасный токен, содержащий claims (утверждения) в формате JSON, подписанный ключом 3) Сессионная cookie 4) Маркер для CSRF-защиты	ПК-3
11.	1	Какой тип атаки заключается во внедрении вредоносного JavaScript-кода в веб-страницу через незранированный пользовательский ввод? 1) XSS (Cross-Site Scripting)	ПК-4

		2) SQL injection 3) CSRF 4) Path traversal	
12.	3	Как в Spring Security защититься от SQL-инъекций при работе с JPA? 1) Использовать конкатенацию строк в @Query 2) Отключить валидацию 3) Использовать параметризованные запросы (?1, :name) в @Query или методы производных от имени 4) Использовать nativeQuery = true	ПК-3
13.	2	Какой тип реактивного потока в Project Reactor представляет асинхронную последовательность из 0 или 1 элемента? 1) Flux 2) Mono 3) Observable 4) Single	ПК-3
14.	4	Что такое backpressure в реактивных системах? 1) Увеличение нагрузки на систему 2) Шифрование потока данных 3) Автоматическое масштабирование сервера 4) Механизм, позволяющий потребителю сигнализировать производителю о своей готовности обрабатывать данные, чтобы избежать переполнения	ПК-3
15.	1	Какой протокол альтернативен REST и использует бинарный формат Protobuf и HTTP/2? 1) gRPC 2) GraphQL 3) SOAP 4) WebSocket	ПК-3
16.	3	Какой подход к работе с GraphQL решает проблему N+1 запросов в резолверах? 1) Использование cache-control 2) Объединение запросов в один большой SQL 3) DataLoader (пакетная загрузка и кэширование) 4) Отказ от GraphQL	ПК-3

17.	2	Для каких сценариев наиболее подходит документная NoSQL база данных (MongoDB)? 1) Строгие транзакции ACID с join-запросами 2) Хранение иерархических/полуструктурированных данных с гибкой схемой и высокой горизонтальной масштабируемостью 3) Хранение графов связей (социальные сети) 4) Ключ-значение для сессий	ПК-3
18.	1,4	Какие инструменты статического анализа кода используются для проверки качества и безопасности? (Выберите все верные) 1) SonarQube 2) JMeter 3) Selenium 4) Checkstyle / PMD	ПК-4
19.	3	Что такое технический долг в разработке ПО? 1) Задолженность перед заказчиком 2) Отсутствие документации 3) Накопленные неоптимальные или временные решения, которые требуют дополнительных затрат на поддержку и рефакторинг 4) Ошибки в коде	ПК-4
20.	2	Как в Spring Boot настроить сбор метрик в Prometheus? 1) Добавить зависимость spring-boot-starter-web 2) Добавить micrometer-registry-prometheus и эндпоинт /actuator/prometheus 3) Установить Prometheus на ту же VM 4) Использовать JMX	ПК-3
21.	4	Какой инструмент используется для централизованного сбора логов из микросервисов? 1) Prometheus 2) Grafana 3) JMeter 4) ELK Stack (Elasticsearch, Logstash, Kibana)	ПК-4
22.	1	Какой уровень изоляции транзакций в SQL позволяет читать только зафиксированные данные, но не защищает от фантомного чтения? 1) Read Committed	ПК-3

		2) Repeatable Read 3) Serializable 4) Read Uncommitted	
23.	2	Какой метод CompletableFuture используется для объединения результатов двух асинхронных операций в одну? 1) thenApply 2) thenCombine 3) exceptionally 4) complete	ПК-3
24.	3	Какая аннотация в Spring Security используется для ограничения доступа к методу контроллера по роли? 1) @Secured 2) @RolesAllowed 3) @PreAuthorize("hasRole('ADMIN')") 4) @PostAuthorize	ПК-3
25.	2	Что такое CSRF (Cross-Site Request Forgery) и как защититься? 1) Внедрение SQL-кода – использовать параметризованные запросы 2) Подделка межсайтового запроса – использовать CSRF-токены и проверку Referer/Origin 3) Внедрение JavaScript – экранировать ввод 4) Переполнение буфера – использовать безопасные функции	ПК-4
26.	4	Какой инструмент используется для миграции схемы базы данных с версионированием и откатами? 1) Hibernate 2) JPA 3) JDBC 4) Liquibase / Flyway	ПК-3
27.	1,3	Какие из перечисленных практик относятся к TDD (Test-Driven Development)? (Выберите все верные) 1) Написание теста до реализации кода 2) Написание всего кода, затем тестов 3) Красный → зелёный → рефакторинг 4) Тестирование только после завершения спринта	ПК-4

28.	2	Какой тип атаки возможен при конкатенации пользовательского ввода в SQL-запрос? 1) XSS 2) SQL injection 3) CSRF 4) Buffer overflow	ПК-4
29.	3	Как в Spring WebFlux правильно выполнить блокирующую операцию (например, вызов JDBC) без блокировки event loop? 1) Вызвать её прямо в контроллере 2) Использовать @Async 3) Вынести в отдельный пул потоков (Schedulers.boundedElastic()) 4) Использовать Mono.just()	ПК-3
30.	2	Какой метод используется для отправки сообщения в gRPC-стриминге от сервера к клиенту? 1) responseObserver.onNext() 2) responseObserver.onNext() – верно, но вопрос: "какой метод в серверном стриминге?" Правильно: onNext. Уточним: 2) onNext	ПК-3
31.	4	Как в GraphQL определить тип данных с полями? 1) type Query { ... } 2) schema { ... } 3) interface { ... } 4) type User { id: ID! name: String }	ПК-3
32.	1	Что такое «покрытие кода» (code coverage) и для чего оно используется? 1) Процент строк или ветвей кода, выполненных во время тестов; помогает оценить полноту тестирования 2) Количество ошибок в коде 3) Скорость выполнения тестов 4) Количество тестов	ПК-4
33.	2	Как в GitHub Actions указать, что пайплайн должен запускаться при push в main? 1) on: [pull_request] 2) on: push: branches: [main] 3) on: workflow_dispatch 4) on: schedule	ПК-4

34.	3	Какой уровень сложности кода измеряет количество линейно независимых путей через программу (чем выше, тем труднее тестировать)? 1) Глубина наследования 2) Связность модулей 3) Цикломатическая сложность (Маккейба) 4) Количество строк	ПК-4
35.	2	Что такое «мутационное тестирование»? 1) Тестирование с изменением данных в БД 2) Внесение небольших изменений в код (мутантов) для проверки, убьёт ли их тест; оценивает качество тестов 3) Тестирование с мутацией объекта 4) Тестирование производительности	ПК-4
36.	1	Как в Reactor обработать ошибку в потоке и вернуть значение по умолчанию? 1) onErrorReturn 2) onErrorResume 3) doOnError 4) retry	ПК-3
37.	3	Какой инструмент используется для нагрузочного тестирования с графическим интерфейсом, поддержкой сценариев и отчётности? 1) JUnit 2) Postman 3) JMeter 4) Selenium	ПК-4
38.	4	Что такое «попарное тестирование» (pairwise testing)? 1) Тестирование в паре программистов 2) Тестирование всех комбинаций входных параметров 3) Тестирование только граничных значений 4) Техника комбинаторного тест-дизайна, которая проверяет все возможные пары значений параметров для сокращения числа тестов	ПК-4
39.	2	Как в Spring Boot включить поддержку асинхронного выполнения методов через @Async? 1) Добавить @EnableAsync в класс конфигурации и @Async на метод	ПК-3

		2) Добавить @EnableAsync и @Async – верно, также нужен TaskExecutor. Правильный ответ 2. 3) Только @Async 4) Использовать CompletableFuture без аннотаций	
40.	1	Какая команда Git используется для безопасного переноса коммитов из одной ветки в другую с сохранением истории? 1) git cherry-pick 2) git rebase 3) git merge 4) git revert	ПК-3
41.	3	Какой метод в Mockito используется для проверки, что метод был вызван определённое количество раз? 1) verify(mock, times(2)).method() 2) when(mock.method()).thenReturn(...) 3) verify(mock, times(2)).method() – это и есть. Выберем 1, но в ответах – 1. В вариантах: 1) verify(mock, times(2)) – верно.	ПК-4
42.	2	Что такое «чистая архитектура» (Clean Architecture) в контексте конструирования ПО? 1) Архитектура без зависимостей 2) Организация кода с чёткими слоями (entities, use cases, interface adapters, frameworks), где зависимости направлены внутрь 3) Использование только одного паттерна MVC 4) Отказ от тестирования	ПК-3
43.	1,4	Какие из перечисленных метрик качества ПО обычно отслеживаются в SonarQube? (Выберите все верные) 1) Покрытие тестами 2) Количество запусков приложения 3) Время отклика API 4) Цикломатическая сложность	ПК-4
44.	3	Как в JPA настроить каскадное сохранение связанных сущностей (например, при сохранении заказа сохраняются и его позиции)? 1) @JoinColumn 2) @OneToMany без cascade	ПК-3

		3) @OneToMany(cascade = CascadeType.ALL) 4) @ManyToOne	
45.	2	Какой тип асинхронного программирования позволяет выполнять код в фоне без блокировки потока, используя ключевые слова <code>async</code> и <code>await</code> в C#? (вопрос для кросс-языков, но в Java аналог – <code>CompletableFuture</code>). Для общности: правильный ответ – 2 (<code>CompletableFuture</code> в Java, <code>async/await</code> в C#/Python). Но в контексте дисциплины – Spring WebFlux + Reactor. Оставим вопрос про Java: Какой интерфейс в Project Reactor представляет поток из 0..N элементов? Ответ: Flux. Давайте заменим этот вопрос на более корректный: "Что такое Flux в Project Reactor?" Ответ: 3) Поток из 0..N элементов.	ПК-3

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 экзамена и включает выполнение практического задания и/или ответы на теоретические вопросы.

3. Критерии оценивания компетенций заданий открытого типа

Оценка «отлично» выставляется студенту, если являются верными результаты демонстрации знаний и умений по следующим направлениям:

– **ПК-3 (разработка ПО):** реализация алгоритмов на языках низкого и высокого уровня (C/C++, ассемблер) с учётом архитектуры ЭВМ (фон Нейман, гарвардская архитектура, конвейеризация, кэш-память, иерархия памяти); написание кода для микропроцессорных систем (работа с регистрами, система команд, режимы адресации, обработка прерываний и исключений); создание отладочных и тестовых приложений для встроенных систем (без непосредственной разработки веб- и desktop-приложений, но с уклоном в системное и низкоуровневое ПО).

– **ПК-4 (тестирование, отладка, качество ПО):** проведение тестирования и отладки на уровне ассемблера и C/C++ с использованием инструментов (gdb для ассемблера, objdump, QEMU, симуляторы/эмуляторы, JTAG-отладчики); применение методов обеспечения надёжности и безопасности (анализ критических участков кода, проверка работы с памятью, обработка прерываний, тайминги); оценка качества программного обеспечения для архитектур x86, ARM, RISC-V.

Представленный отчёт (программный код на ассемблере/C/C++, схемы, пояснения, результаты отладки) не содержит ошибок, решены все задачи лабораторных работ и индивидуальных заданий. Компетенции **ПК-3, ПК-4** освоены на **высоком уровне**.

Оценка «хорошо» выставляется студенту, если являются верными результаты демонстрации указанных знаний и умений, но отчётный документ содержит незначительные ошибки, не влияющие на общий результат (например, неточности в комментариях, неполное описание архитектурных особенностей, некритичные ошибки в ассемблерных вставках), либо не представлены отдельные второстепенные задачи, не влияющие на общий результат. Компетенции **ПК-3, ПК-4** освоены на **среднем уровне**.

Оценка «удовлетворительно» выставляется студенту, если в основном верными являются результаты демонстрации знаний и умений, но отчётный документ содержит ошибки, не позволяющие однозначно оценить качество разработки (например, некорректная работа с прерываниями, ошибки адресации, проблемы с конвейеризацией или кэшем); представлены не все данные, задачи решены не в полном объёме. Компетенции **ПК-3, ПК-4** освоены на **минимальном уровне**.

Оценка «неудовлетворительно» выставляется студенту, если являются неверными результаты демонстрации знаний об архитектуре ЭВМ и микропроцессорной технике, отчётный документ содержит много грубых ошибок (непонимание системы команд, неправильная работа с памятью и вводом-выводом, отсутствие отладки), не представлены ключевые данные (код, схемы, результаты тестирования), задачи не решены. Компетенции **ПК-3, ПК-4** не сформированы.

4. Критерии оценивания компетенций заданий закрытого типа

Оценка «отлично» выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенция ПК-3 освоена на высоком уровне.

Оценка «хорошо» выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция ПК-3 освоена на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенция ПК-3 освоена на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция ПК-3 не сформирована.

Оценка «отлично» выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенция ПК-4 освоена на высоком уровне.

Оценка «хорошо» выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция ПК-4 освоена на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенция ПК-4 освоена на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция ПК-4 не сформирована.