Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)
Уральский технический институт связи и информатики (филиал) в г. Екатеринбурге
(УрТИСИ СибГУТИ)

УТВЕРЖДАЮ
Директор УрТИСИ СибГУТИ
_____ Минина Е.А.
«28» 11 2025 г.

ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

ПО ДИСЦИПЛИНЕ
Б1.В.13 Современные технологии программирования

Направление подготовки / специальность: **09.03.01 «Информатика и вычислительная техника»**

Направленность (профиль) /специализация: **Инженерия программного обеспечения и искусственного интеллекта**

Форма обучения: **очная**

Год набора: 2026

-

Разработчик (-и):
ст.преподаватель _____ / М.Ю. Казанцев /
подпись

к.т.н., доцент _____ / Т.А. Черных /
подпись

Оценочные средства обсуждены и утверждены на заседании информационных систем и технологий (ИСТ)
Протокол от 27.11.2025 г. № 3
Заведующий кафедрой _____ / Д.И. Бурумбаев /
подпись

Екатеринбург, 2025

Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)
Уральский технический институт связи и информатики (филиал) в г. Екатеринбурге
(УрТИСИ СибГУТИ)

УТВЕРЖДАЮ
Директор УрТИСИ СибГУТИ
_____Минина Е.А.
«____» _____2025 г.

ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

ПО ДИСЦИПЛИНЕ
Б1.В.13 Современные технологии программирования

Направление подготовки / специальность: **09.03.01 «Информатика и вычислительная техника»**

Направленность (профиль) /специализация: **Инженерия программного обеспечения и искусственного интеллекта**

Форма обучения: **очная**

Год набора: 2026

Разработчик (-и):
ст.преподаватель _____ / М.Ю. Казанцев /
подпись
к.т.н., доцент _____ / Т.А. Черных /
подпись

Оценочные средства обсуждены и утверждены на заседании информационных систем и технологий (ИСТ)
Протокол от 27.11.2025 г. № 3
Заведующий кафедрой _____ / Д.И. Бурумбаев /
подпись

Екатеринбург, 2025

1. Перечень компетенций и индикаторов их достижения

Процесс изучения дисциплины направлен на формирование следующих компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенций	Этап	Предшествующие этапы (с указанием дисциплин/практик)
ПК-1 Способен проектировать и разрабатывать программное обеспечение	ПК 1.1 Знает современные методы, средства и стандарты для проектирования и разработки программного обеспечения ПК 1.2 Умеет применять современные технологии для проектирования и разработки программного обеспечения ПК 1.3 Владеет навыками проектирования и разработки программного обеспечения	5	1 этап Б1.О.07 Программирование на языке Python Б1.О.17 Программирование на языке C# Б1.В.01 Web-технологии Б1.О.18 Программирование на языке C/C++ 2 этап Б1.О.11 Технологии баз данных Б1.В.15 Разработка интерактивных приложения Б1.В.22 Разработка на платформе JVM Б1.В.11 Разработка мобильных приложений Б2.О.02(П) Производственная технологическая практика 3 этап Б1.В.21 Методы и средства защиты баз данных Б1.В.06 Технологии разработки программного обеспечения Б1.В.19 Программирование на языке 1С Б2.В.01(П) Производственная эксплуатационная практика 4 этап Б1.О.23 Документирование программных решений Б1.В.18 Основы безопасной разработки

Форма итоговой аттестации по дисциплине – экзамен

2. Показатели, критерии и шкалы оценивания компетенций

2.1 Показателем оценивания компетенций на этапе их формирования при изучении дисциплины является уровень их освоения.

Индикатор освоения компетенции	**Показатель оценивания**	**Критерий оценивания**
ПК 1.1 Знает современные методы, средства и стандарты для проектирования и разработки программного обеспечения	Результаты теста/опроса по темам: парадигмы программирования (ООП/FP), принципы SOLID, паттерны проектирования (GoF — базовые), архитектурные подходы (слои, MVC, Clean Architecture — на уровне понимания), основы API и контрактов, тестирование (unit/integration), CI/CD (на уровне терминов), системы контроля версий (Git), код-стайл/линтин	не менее 70% верных ответов; корректно объясняет ключевые принципы (SOLID, назначение паттернов, разницу unit vs integration), знает назначение Git/CI, умеет привести примеры применения подходов
ПК 1.2 Умеет применять современные технологии для проектирования и разработки программного обеспечения	Выполнение практических заданий: проектирование структуры проекта, реализация модулей, работа с Git (commit/branch/merge), использование зависимостей (package manager), написание автотестов, настройка линтера/форматтера, сборка/запуск через стандартные инструменты, взаимодействие по API (если предусмотрено)	решение собирается и запускается; применены выбранные инструменты (Git, зависимости, линтер/форматтер); код соответствует ТЗ; тесты проходят (или представлены и воспроизводимы); допускаются незначительные недочеты, не влияющие на работоспособность
ПК 1.3 Владеет навыками проектирования и разработки программного обеспечения	Мини-проект в рамках практического задания: полный цикл разработки готового продукта; соблюдение практик качества (структура, читаемость, обработка ошибок, логирование по необходимости)	проект функционально завершен и стабильно работает; архитектура обоснована и соответствует масштабу; код поддерживаем (модули, именование, отсутствие дублирования); есть базовый набор тестов; есть README (запуск, зависимости, примеры использования); история Git отражает процесс разработки (осмысленные коммиты)

Шкала оценивания.

Экзамен

5-балльная шкала	**Критерии оценки**
«отлично»	На экзаменационные вопросы даны полные аргументированные ответы. Студент демонстрирует сформированность дисциплинарных компетенций на итоговом уровне,

	обнаруживает всестороннее, систематическое и глубокое знание учебного материала по тематике: современные подходы к проектированию ПО (модульность, слоистая архитектура, Clean Architecture/MVC на уровне понимания), принципы SOLID, базовые паттерны проектирования и области их применения, работа с зависимостями и сборкой (package manager), системы контроля версий (Git: ветвление, merge), основы тестирования (unit/integration, TDD на уровне подхода), инструменты качества кода (линтер/форматтер), основы CI/CD (понимание назначения), документация и стандарты оформления кода. Студент корректно решает практическую задачу (проектирование/фрагмент кода), объясняет решения и типовые компромиссы.
«хорошо»	На экзаменационные вопросы даны в целом полные ответы, но имеются незначительные неточности или замечания преподавателя (например, в формулировках SOLID, выборе паттерна, деталях Git-процессов или тестирования). Компетенции сформированы на среднем уровне: основные понятия и инструменты освоены, практическая задача выполняется, но возможны небольшие недочеты в архитектуре/читаемости/тестируемости решения.
«удовлетворительно»	Ответы даны со слабой аргументацией, студент отвечает неуверенно, требуется значительное количество наводящих вопросов. Компетенции сформированы на базовом уровне: знание материала фрагментарное, есть пробелы (паттерны, тестирование, Git, стандарты кода). При решении практической задачи допускаются существенные ошибки (неверная декомпозиция, отсутствие обработки ошибок, неуместные зависимости), но после подсказок студент способен частично исправить решение.
«неудовлетворительно»	Компетенции сформированы на уровне ниже порогового. Наблюдается недостаточность знаний по основным темам дисциплины, отсутствуют устойчивые навыки проектирования и разработки: студент не может объяснить базовые принципы (SOLID, тестирование, Git) и не способен выполнить практическую задачу даже с помощью наводящих вопросов. Дисциплинарные компетенции не сформированы.

3. Методические материалы, определяющие процедуры оценивания по дисциплине

3.1. В ходе реализации дисциплины используются следующие формы и методы текущего контроля

Тема и/или раздел	**Формы/методы текущего контроля успеваемости**
ПК-1 Способен проектировать и разрабатывать программное обеспечение	
Современные технологии программирования и быстрый запуск проекта	Самостоятельная работа, конспект лекции
Git и командная разработка на решениях, доступных в РФ	Самостоятельная работа, конспект лекции
Backend разработка, API first подход	Самостоятельная работа,

	конспект лекции
Аутентификация и авторизация в клиент серверных приложениях	Самостоятельная работа, конспект лекции
Frontend разработка на современном стеке	Самостоятельная работа, конспект лекции
Паттерны и архитектура прикладных приложений	Самостоятельная работа, конспект лекции
Работа с базами данных и миграции в проектах	Самостоятельная работа, конспект лекции
Интеграции и асинхронность в распределенных системах	Самостоятельная работа, конспект лекции
Тестирование как ускоритель разработки	Самостоятельная работа, конспект лекции
CI CD и автоматизация сборки в условиях РФ	Самостоятельная работа, конспект лекции
Контейнеризация и развертывание приложений	Самостоятельная работа, конспект лекции
Наблюдаемость производительность и эксплуатация	Самостоятельная работа, конспект лекции
Современные технологии программирования и быстрый запуск проекта	Самостоятельная работа, конспект лекции
Git и командная разработка на решениях, доступных в РФ	Самостоятельная работа, конспект лекции
Backend разработка, API first подход	Самостоятельная работа, конспект лекции
Аутентификация и авторизация в клиент серверных приложениях	Самостоятельная работа, конспект лекции
Frontend разработка на современном стеке	Самостоятельная работа, конспект лекции
Паттерны и архитектура прикладных приложений	Самостоятельная работа, конспект лекции
Работа с базами данных и миграции в проектах	Самостоятельная работа, конспект лекции
Интеграции и асинхронность в распределенных системах	Самостоятельная работа, конспект лекции
Тестирование как ускоритель разработки	Самостоятельная работа, конспект лекции
CI CD и автоматизация сборки в условиях РФ	Самостоятельная работа, конспект лекции
Контейнеризация и развертывание приложений	Самостоятельная работа, конспект лекции
Наблюдаемость производительность и эксплуатация	Самостоятельная работа, конспект лекции
Инициализация проекта по шаблону, настройка окружения, .env, структура репозитория и базовый запуск	Практическая работа
Настройка Git workflow, ветки, merge request и code review в GitLab CE self-host (или аналог в РФ)	Практическая работа
Разработка REST API для сущности, валидация входных данных и автогенерация OpenAPI Swagger	Практическая работа
Реализация аутентификации и авторизации, JWT или session, роли RBAC и защита эндпоинтов	Практическая работа

Создание SPA интерфейса, работа с формами и запросами к API, обработка ошибок на клиенте	Практическая работа
Рефакторинг под слоистую архитектуру, выделение сервисов и репозиториев, устранение технического долга	Практическая работа
Подключение PostgreSQL, проектирование схемы, миграции, транзакции и базовые индексы под запросы	Практическая работа
Интеграция с внешним сервисом через HTTP и webhooks, ретраи, идемпотентность и обработка ошибок	Практическая работа
Покрытие проекта тестами, unit и integration, тестирование API, фикстуры и мокирование	Практическая работа
Сборка CI pipeline, линтеры, тесты, сборка артефактов/образов, версия и тегирование релиза	Практическая работа
Контейнеризация приложения, docker compose, Nginx reverse proxy, деплой на VPS или российское облако	Практическая работа
Настройка логирования и метрик, базовый мониторинг и алерты, проверка требований к ПДн по 152-ФЗ для сценария проекта	Практическая работа

3.2. Типовые материалы текущего контроля успеваемости обучающихся

ПК-1 Способен проектировать и разрабатывать программное обеспечение.

Пример задания на практическое занятие

Цель: закрепить навыки проектирования и разработки прикладного сервиса с применением современных инструментов (Git, API-first, тестирование, CI/CD, контейнеризация).

Задание: Разработка мини-сервиса «Заметки (Notes API)» + простой SPA-клиент.

Задачи:

1. Инициализировать проект по шаблону, настроить окружение и .env, подготовить структуру репозитория.
2. Настроить Git workflow: ветки, merge request, code review.
3. Реализовать REST API для сущности (CRUD), валидацию входных данных и документацию OpenAPI/Swagger.
4. Подключить PostgreSQL, настроить миграции, добавить транзакцию для составной операции.
5. Реализовать аутентификацию и авторизацию (JWT или session), роли RBAC, защиту эндпоинтов.
6. Написать минимальный набор тестов (unit + integration) для ключевых сценариев.
7. Собрать CI pipeline: линтер + тесты + сборка артефакта/образа, тегирование версии.
8. Контейнеризировать проект (Docker Compose), добавить Nginx reverse proxy, подготовить инструкцию деплоя.
9. Отчет: ссылка на репозиторий, краткое описание архитектуры, как запустить локально/в контейнерах.

Типовые вопросы и задания к экзамену

1. Как выбрать технологический стек под задачу? Какие критерии учитывать (поддержка, команда, инфраструктура)?
2. Что такое Git workflow? Ветки, merge request, code review — зачем нужны.
3. Принципы API-first. Что такое контракт API и зачем нужна документация OpenAPI/Swagger?
4. Основы REST: ресурсы, методы, коды ответов, идемпотентность.

5. Валидация входных данных и обработка ошибок: зачем единый формат ошибок.

6. Аутентификация vs авторизация. JWT vs session. Что такое refresh token.

7. RBAC: как организовать роли и права доступа в приложении.

8. Базовые принципы слоистой архитектуры: контроллеры/сервисы/репозитории, зачем разделять.

9. Миграции БД: зачем, как версионировать схему.

10. Транзакции: когда нужны, что будет при ошибке в середине операции.

11. Unit, integration, e2e тесты: различия и где применять.

12. Моки/фикстуры: зачем и как использовать.

13. CI/CD: что входит в пайплайн (линтеры, тесты, сборка), что такое релизный тег/версия.

14. Docker и Docker Compose: зачем контейнеризация, что должно быть в compose для типового проекта.

15. Практическое задание: спроектировать REST API для сущности и описать структуру проекта (слои, зависимости, тесты и CI шаги).

3.3. Методические материалы проведения текущего контроля и промежуточной аттестации обучающихся

Перечень методических материалов для подготовки к текущему контролю и промежуточной аттестации:

1. Методические указания по выполнению практических занятий по дисциплине «Технологии командной разработки программного обеспечения». –URL: http://aup.uisi.ru/4629963/

2 Образовательная среда УрТИСИ СибГУТИ – URL: https://moodle.uisi.ru