

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
БУРЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ДОРЖИ БАНАЗАРОВА
ИНСТИТУТ МАТЕМАТИКИ, ФИЗИКИ И КОМПЬЮТЕРНЫХ НАУК
КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И ИНФОРМАТИКИ

Утверждена на заседании
Ученого совета ИМФКН
«__»_____ 202__ г.
Протокол № __

Рабочая программа дисциплины
Технологии и платформы разработки информационных систем

Направление подготовки/ специальность
09.04.02 – Информационные системы и технологии

Профиль подготовки /специализация

Квалификация (степень) выпускника
магистр

Форма обучения
очная

Улан-Удэ

2025

Пояснительная записка

Цели освоения дисциплины

Цель дисциплины состоит в формировании у обучающихся профессиональных компетенций в области проектирования, разработки и сопровождения информационных систем с использованием современных технологических платформ и инструментариев.

Место дисциплины в структуре образовательной программы

Учебная дисциплина «Технологии и платформы разработки информационных систем» относится к обязательной части формируемая участниками образовательных отношений. Изучение учебной дисциплины "Технологии и платформы разработки информационных систем" базируется на знаниях, умениях и навыках, полученных обучающимися при изучении дисциплин уровня бакалавриата.

Планируемые результаты обучения по дисциплине и индикаторы достижения компетенций.

ПК-2. Способен организовывать и обеспечивать подтверждение уровня качества исполнения процессов в проектах в области ИТ любого уровня сложности

ПК.М-2.2 Проводит документирование и управление документированием собранных данных для формализации требований заказчика в проектах в области ИТ любого уровня сложности

ПК.М-2.3 Владеет методами управления требованиями в проектах в области ИТ

ПК-3. Способен планировать коммуникации в проектах любого уровня сложности в области ИТ

ПК.М-3.2. Понимает современные стандарты информационного взаимодействия систем

В результате освоения дисциплины студент должен:

Знать:

- Принципы построения многоуровневых архитектур информационных систем.
- Характеристики популярных платформ и сред разработки (JavaScript/TypeScript, Python, .NET Core, Ruby on Rails, PHP/Laravel).
- Стандартные методы и инструменты для интеграции баз данных и серверных приложений.
- Базовые принципы безопасности и защиты информационных систем.
- Процессы развертывания и масштабирования информационных систем.

Уметь:

- Осуществлять выбор оптимальной технологической платформы для конкретной задачи.
- Применять современные языки программирования и фреймворки для реализации функциональных требований.
- Работать с системами управления версиями (Git/GitHub, Bitbucket).
- Производить интеграцию информационных систем с внешними сервисами и API.
- Диагностировать и устранять технические неисправности и сбои в работе информационных систем.

Владеть:

- Навыками работы с интегрированными средами разработки (IDE): Visual Studio Code, PyCharm, IntelliJ IDEA.
- Техниками автоматизации сборки и тестирования (CI/CD).
- Способностью документировать архитектуру и функциональность разработанных решений.
- Методиками рефакторинга и оптимизации производительности информационных систем.
- Опытном работ с облачными платформами и контейнерами Docker/Kubernetes.

Планируемые результаты освоения образовательной программы:**Объем дисциплины в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем и на самостоятельную работу обучающихся**

Общая трудоемкость дисциплины составляет 3 зачетные единицы, 108 часа.

№	Название разделов дисциплины	Лекция	Лабораторная работа	Самостоятельная работа
Семестр 3		22	22	64
1	Теоретические основы информационных технологий	10	10	10
2	Платформы и инструментальные средства разработки информационных систем	12	12	54

Тематическое планирование курса**Темы****Теоретические основы информационных технологий**

Семестр 3

Архитектура информационных систем

Лекция. 2(0) ч. Типы архитектур информационных систем

Лабораторная работа. 2(0) ч. Постановка требований к архитектуре

Самостоятельная работа. 2(0) ч. Изучение примеров клиент-серверных и микросервисных архитектур

Программные платформы и среды разработки

Лекция. 2(0) ч. Средства разработки и платформы

Лабораторная работа. 2(0) ч. Установка и настройка инструментов разработки

Самостоятельная работа. 2(0) ч. Освоение среды разработки Visual Studio Code

Языки программирования для разработки ИС

Лекция. 2(0) ч. Обзор ключевых языков программирования

Лабораторная работа. 2(0) ч. Введение в синтаксис и базовые конструкции языка JavaScript

Самостоятельная работа. 2(0) ч. Практическая реализация простых скриптов на JS

Клиентская сторона web-разработки

Лекция. 2(0) ч. HTML/CSS/JS для фронтенд-разработки

Лабораторная работа. 2(0) ч. Создание статичного сайта с элементами оформления

Самостоятельная работа. 2(0) ч. Реализация динамического интерфейса с использованием JavaScript

Серверная сторона web-приложений

Лекция. 2(0) ч. Основы серверной стороны на примере Node.js

Лабораторная работа. 2(0) ч. Настройка сервера Express.js

Самостоятельная работа. 2(0) ч. Разработка RESTful API на платформе Node.js

Платформы и инструментальные средства разработки информационных систем

Семестр 3

Web-фреймворки и шаблонизаторы

Лекция. 2(0) ч. Популярные web-фреймворки и их применение

Лабораторная работа. 2(0) ч. Использование Django/Python для backend-разработки

Самостоятельная работа. 2(0) ч. Разработка простого приложения на Django

Backend-технологии и базы данных

Лекция. 2(0) ч. QL и NoSQL базы данных

Лабораторная работа. 2(0) ч. Проектирование и создание базы данных MySQL

Самостоятельная работа. 2(0) ч. Импорт и экспорт данных между различными типами хранилищ

Безопасность и авторизация в информационных системах

Лекция. 2(0) ч. Механизмы аутентификации и авторизации пользователей

Лабораторная работа. 2(0) ч. Добавление механизма OAuth2 в приложение

Самостоятельная работа. 2(0) ч. Настройка ролей и прав доступа в приложении

Облачные сервисы и контейнеризация

Лекция. 2(0) ч. Amazon AWS, Microsoft Azure, Kubernetes

Лабораторная работа. 2(0) ч. Запуск приложения в облаке на виртуальной машине

Самостоятельная работа. 2(0) ч. Развёртывание приложения в контейнере Docker

Автоматизация и DevOps-подходы

Лекция. 2(0) ч. CI/CD пайплайны и автоматизация релизов

Лабораторная работа. 2(0) ч. Настройка непрерывной интеграции с GitLab CI C

Самостоятельная работа. 2(0) ч. Автоматизация деплоя и мониторинга приложений

Итоговая тема

Лекция. 2(0) ч. Современное состояние индустрии информационных технологий

Лабораторная работа. 2(0) ч. Проектирование полноценного web-приложения

Самостоятельная работа. 44(0) ч. Самостоятельное изучение расширенных материалов и углубленное освоение одной выбранной платформы или инструмента

БРС

Семестр	Контрольные точки	Баллы
3	Текущий контроль в разделе «Теоретические основы информационных технологий»	
	Подготовка доклада и презентации	10
	Лабораторная работа	20
3	Текущий контроль в разделе «Платформы и инструментальные средства разработки информационных систем»	
	Лабораторная работа	20
	проект	10
3	Зачет	
	Зачет	40

Итого за семестр 3: 100

Учебно-методическое и информационное обеспечение учебного процесса

Образовательные технологии (в том числе на занятиях, проводимых в интерактивных формах).

Традиционные лекции: Предназначены для передачи основных понятий, концепций и принципов управления проектами. Используются мультимедийные средства (проектор, компьютер), материалы курса размещаются в электронной образовательной среде университета.

Лабораторные задания: Включают работу в группах и индивидуально. Студентам предлагается реализовать различные элементы управления проектами, такие как

составление планов, анализ рисков, моделирование процессов и выполнение контрольных мероприятий.

Интерактивные формы: Интеграция элементов геймификации и деловых игр помогает развивать лидерские качества, умение работать в команде и принимать обоснованные управленческие решения.

Деловые игры: Например, игра по разработке стратегии управления крупным ИТ-проектом с применением метода мозгового штурма.

Кейс-метод: Решение реальных бизнес-кейсов из области ИТ-индустрии способствует развитию аналитических способностей и применению полученных знаний на практике.

Учебно-методические материалы, в том числе методические указания для обучающихся по освоению дисциплины

В распоряжении преподавателей и обучающихся имеется основное необходимое материально-техническое оборудование, Интернет-ресурсы, доступ к полнотекстовым электронным базам, книжный фонд библиотеки Бурятского государственного университета имени Доржи Банзарова

Оценочные средства

По данной дисциплине разработаны оценочные средства, критерии их оценивания, а также методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

- [ФОС_Технологии и платформы разоаботки ИС.docx](#)

Список литературы

Перечень основной и дополнительной литературы, необходимой для освоения дисциплины.

Основная

1. [Базы данных. Лабораторный практикум](#): учебное пособие для вузов/Мамедли Р. Э.. —Санкт-Петербург: Лань, 2023. —152 с.
Режим доступа: <https://e.lanbook.com/book/319400>
2. [Базы данных: проектирование](#): Учебник для вузов/Стружкин Н. П., Годин В. В.. — Москва: Юрайт, 2022. —477 с.
Режим доступа: <https://urait.ru/bcode/489099>
3. [Архитектура ЭВМ и систем в 2 ч. Часть 1](#): Учебное пособие для вузов/Новожилов О. П.. —Москва: Юрайт, 2022. —276 с.
Режим доступа: <https://urait.ru/bcode/494314>

Дополнительная

1. Яргер Р. Д. MySQL и mSQL.Базы данных для небольших предприятий и Интернета/Р. Д. Яргер, Д. Риз, Т. Кинг. —СПб.: Символ, 2001. —557 с.
2. [MySQL](#)/Ларри Ульман. —Москва: ДМК Пресс, 2008. —351 с.
Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=1241
3. [АРХИТЕКТУРА ИНФОРМАЦИОННЫХ СИСТЕМ](#): Учебное пособие/Рыбальченко М.В.. —М.: Издательство Юрайт, 2016. —91 с.
Режим доступа: <http://www.biblio-online.ru/book/1E3097D3-2594-4FFA-A033-3A7FF7C31065>

Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины

Федеральный портал. Российское образование. <http://www.edu.ru/>

Информационно-коммуникационные технологии в образовании. <http://www.ict.edu.ru/>

Российский портал открытого образования. <http://www.openet.edu.ru/>

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Портал электронного обучения БГУ e.bsu.ru

Личный кабинет преподавателя или студента БГУ <https://my.bsu.ru/>

Электронные библиотечные системы: Руконт, издательство «Лань», Консультант студента

Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. Сетевой компьютерный класс, оснащенный современной техникой
2. Офисный программный пакет (например, MicrosoftOffice 2003 или более поздних версий).
3. Web-браузер Mozilla Firefox или Google Chrome
4. Экран для проектора

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО «БУРЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ДОРЖИ БАНЗАРОВА»

**Фонд оценочных средств по дисциплине
Технологии и платформы разработки информационных систем**

Направление подготовки/ специальность
09.04.02 – Информационные системы и технологии

Профиль подготовки /специализация

Квалификация (степень) выпускника
магистр

Форма обучения
очная

Улан-Удэ
2025

Паспорт фонда оценочных средств

	Контролируемые разделы, темы, модули ¹	Наименование компетенции	Оценочные средства		
			Этапы формирования	Вид	Количество
1	Теоретические основы информационных технологий	ПК-2.2 ПК-2.3	3 семестр	Подготовка доклада и презентации Лабораторная работа	1 5
2	Платформы и инструментальные средства разработки информационных систем	ПК-2.3 ПК-3.2	3 семестр	Лабораторная работа Проект	6 1
3	Экзамен	ПК-2.2 ПК-2.3 ПК-3.2	3 семестр	Экзамен	1 вопрос в билете
Всего:		3			

Примерные вопросы для зачета

1. Понятие информационной системы. Классификация информационных систем.
2. Жизненный цикл информационной системы. Основные этапы разработки ИС.
3. Современные технологии программирования и их применение в создании информационных систем.
4. Модели разработки программного обеспечения: каскадная, спиральная, инкрементальная.
5. Методология Agile и её отличие от традиционных подходов к разработке ИС.
6. Архитектура клиент-серверных приложений и трёхуровневая архитектура.
7. Принципы объектно-ориентированного программирования и их реализация в современных языках программирования.
8. Что такое API и как оно используется в интеграции информационных систем?
9. Мобильные приложения и технологии их разработки (Android, iOS, Flutter, React Native).
10. Основные принципы работы баз данных. SQL и NoSQL решения.
11. Web-технологии: HTML, CSS, JavaScript, фреймворки (React, Angular, Vue.js).
12. RESTful-сервисы и принципы их проектирования.
13. Платформы разработки информационных систем: ASP.NET Core, Django, Spring Boot.
14. Средства интеграции информационных систем: ESB, SOA, микросервисная архитектура.
15. Контейнеризация и виртуализация: Docker, Kubernetes.
16. Тестирование информационных систем: виды тестов, автоматизированное тестирование.
17. Информационная безопасность в системах разработки. Механизмы аутентификации и авторизации.
18. Тенденции и перспективы развития платформ разработки информационных систем.
19. Метрики качества программного обеспечения и их измерение.

¹ Наименования разделов, тем, модулей соответствуют рабочей программе дисциплины

20. Практические аспекты миграции информационных систем на новые технологические платформы.

Критерии оценивания:

- степень полноты, точности, самостоятельности ответа;
- качество изложения программного материала при ответе на основной и дополнительные вопросы преподавателя;
- способность увязывать теорию с практикой;
- использование в ответе материала разнообразных источников.

Оценивание ответа при собеседовании – максимальный балл - 40:

Баллы для учета в рейтинге (оценка ответа на зачете)	Степень удовлетворения критериям
34-40 баллов «отлично»	Дано исчерпывающее и точное раскрытие содержания вопроса. Приведён полный перечень терминов, понятий и фактов, имеющих прямое отношение к рассматриваемой теме. Продемонстрировано глубокое понимание взаимосвязей между элементами изучаемого материала. Применяются практические примеры, демонстрирующие осмысленное владение материалом.
27-33 баллов «хорошо»	Представлены главные аспекты и термины, раскрывающие суть вопроса. Неточности или пробелы в ответе имеют второстепенное значение и не влияют существенно на общую картину представления материала. Материал структурирован логично, последовательны переходы от одного элемента ответа к другому.
20-26 баллов «удовлетворительно»	Дан общий обзор темы, раскрыта ключевая идея вопроса, но существенные детали пропущены или недостаточно освещены. Информация представлена фрагментарно, отсутствует полная логика изложения материала. Присутствуют ошибки в терминологии, определениях или содержании ответа, не оказывающие серьёзного негативного воздействия на общее восприятие.
19 баллов и меньше «неудовлетворительно»	Тема раскрыта поверхностно или частично, важнейшие аспекты не затронуты. Имеются многочисленные фактические ошибки, искажено содержание понятия или явления. Сложности в формировании полного ответа, нарушена логика рассуждений, демонстрация слабых знаний теории.

Шкала перевода в баллы:

Оценка	Общий балл
34-40	5 зачтено
27-33	4 зачтено
20-26	3 зачтено
Менее 20 баллов	2 не зачтено

Примерные темы докладов для оценивания в разделе «Теоретические основы информационных технологий»

1. История возникновения и развития информационных технологий.
2. Основные компоненты компьютерной системы и их функции.

3. Современные операционные системы: сравнительный анализ Windows, macOS, Linux.
4. Архитектуры вычислительных сетей: Ethernet, Wi-Fi, Bluetooth.
5. Алгоритмы сортировки и их эффективность.
6. Принципы функционирования файловых систем и форматы файлов.
7. Интернет вещей (IoT): возможности и ограничения.
8. Искусственный интеллект и машинное обучение: основные направления и применение.
9. Языки программирования высокого уровня: Python, Java, C++.
10. Современные тенденции в кибербезопасности и защита персональных данных.
11. Виртуальные машины и контейнеризация: преимущества и недостатки.
12. Базы данных: реляционные vs. нереляционные (NoSQL).
13. Программирование мобильных приложений: Android и iOS.
14. Графические интерфейсы пользователя: эволюция и современные тенденции.
15. Роботы и автоматизация процессов: будущее профессий.
16. Электронная коммерция и цифровые платежи: опыт российских онлайн-платформ.
17. Открытый код и свободное программное обеспечение: плюсы и минусы.
18. Эффект больших данных и аналитики в бизнесе.
19. Семантический web и технология Linked Data.
20. Будущие перспективы информационных технологий и их влияние на общество.

Критерии оценивания:

- В докладе показано владение специальными терминами.
- Доклад сопровождается демонстрационным материалом, например презентацией
- Автор хорошо владеет тематикой и отвечает на дополнительные вопросы.

Оценивание доклада – максимальный балл - 20:

Баллы для учета в рейтинге (оценка ответа)	Степень удовлетворения критериям
18-20 баллов «отлично»	Доклад соответствует критериям оценивания и сдан во время
14-17 баллов «хорошо»	Доклад соответствует критериям оценивания, но отсутствует презентация, сдан во время
9-13 балла «удовлетворительно»	Доклад соответствует критериям оценивания, но отсутствует презентация и сдан с опозданием
8 баллов и менее «неудовлетворительно»	Студент не приступал к выполнению данного задания.

Примерные темы для индивидуальных проектов для оценивания в разделе «Платформы и инструментальные средства разработки информационных систем»

1. **Проектирование мобильного приложения на платформе Android с использованием Kotlin.** *Цель:* Разработать простое мобильное приложение с использованием языка Kotlin и платформы Android Studio.
2. **Разработка веб-приложения с применением Node.js и Express.** *Цель:* Создать функциональное веб-приложение с серверной частью на Node.js и фронтэндом на HTML/CSS/JavaScript.
3. **Интеграция облачных сервисов Google Cloud Platform для хранения и обработки данных.** *Цель:* Настроить инфраструктуру для загрузки и обработки данных с использованием сервиса Google Cloud Storage и BigQuery.
4. **Настройка CI/CD конвейера с использованием GitLab CI.** *Цель:* Организовать непрерывную интеграцию и доставку (CI/CD) с использованием системы контроля версий Git и возможностей GitLab CI.

5. **Разработка чат-бота на платформе Telegram с использованием Python и библиотеки PyTelegramBotAPI.** *Цель:* Написать простого бота, выполняющего полезные функции, такие как информирование пользователей о погоде или курсах валют.
6. **Создание десктопного приложения на языке Python с графическим интерфейсом на Tkinter.** *Цель:* Реализовать несложное настольное приложение с GUI-интерфейсом на библиотеке Tkinter.
7. **Построение REST API на языке Go с использованием Gorilla Mux.** *Цель:* Разработайте простую API платформу с поддержкой HTTP-запросов GET, POST, PUT и DELETE.
8. **Внедрение системы управления версиями Git и репозитория на GitHub/GitLab.** *Цель:* Научитесь управлять версиями вашего проекта с использованием Git и разместите его на площадке GitHub или GitLab.
9. **Создания простой игры на Unity с использованием C#.** *Цель:* Освоить среду разработки Unity и язык программирования C#, создав простейшую игру с минимальным функционалом.
10. **Автоматизация деплоя с использованием Ansible.** *Цель:* Узнать основы работы с инструментом Ansible и развернуть собственное приложение на удалённом сервере автоматически.
11. **Облачная инфраструктура на AWS: настройка инфраструктуры как кода (IaC) с Terraform.** *Цель:* Познакомьтесь с инфраструктурой Amazon Web Services и настройте облачную инфраструктуру с помощью декларативного инструмента Terraform.
12. **Безопасность веб-приложений: аудит и защита от XSS и CSRF атак.** *Цель:* Провести аудит собственного веб-приложения на уязвимости, внедрить защиту от Cross-Site Scripting (XSS) и Cross-Site Request Forgery (CSRF).
13. **Работа с системой управления конфигурацией Chef/Puppet/SaltStack.** *Цель:* Ознакомиться с принципами настройки серверов и автоматической конфигурации оборудования с помощью выбранного инструмента (Chef, Puppet или SaltStack).
14. **Модификация и расширение существующей open-source системы (пример: WordPress).** *Цель:* Изменить существующий открытый продукт (например, CMS WordPress), добавив новый функционал или улучшив интерфейс.
15. **Контейнеризация приложения с использованием Docker и оркестрация контейнеров с Docker Compose.** *Цель:* Изучите технологию контейнеризации, запустите приложение в контейнерах Docker и научитесь управлять ими с помощью Docker Compose.
16. **Фронтенд-разработка современного SPA-приложения с использованием Vue.js/React.js/Angular.** *Цель:* Создать одностраничное приложение (SPA) с одним из популярных фронтенд-фреймворков.
17. **Создание REST API для мобильных приложений с использованием Flask/Django Rest Framework.** *Цель:* Разработать полноценный backend API для поддержки мобильной версии приложения с использованием Python и популярного фреймворка Flask или Django Rest Framework.
18. **Подготовка и публикация пакета Python на PyPi.** *Цель:* Подготовьте собственный пакет Python и опубликуйте его на официальном ресурсе PyPi.
19. **Сборка собственного дистрибутива Linux с использованием Arch Linux.** *Цель:* Понять устройство операционной системы Linux изнутри, собрать собственную версию дистрибутива на основе Arch Linux.
20. **Проектирование и разработка чат-бота на платформе Discord с использованием discord.py.** *Цель:* Научиться создавать своего первого бота для мессенджера Discord, умеющего взаимодействовать с пользователями.

Критерии оценивания:

1. Корректность выполнения технического задания

Оценивает, насколько точно студент выполнил технические требования и постановку задачи проекта.

- Полное соответствие проекту: максимальный балл.

- Незначительные отступления от техзадания: снижение балла.
- Неправильная реализация ключевых компонентов: значительное уменьшение оценки.

2. Глубина освоения инструментальных средств и технологий

Определяет, насколько хорошо студент понял и использовал инструментальные средства и технологии, указанные в техническом задании.

- Использование заявленных технологий в полном объеме: высокий балл.
- Частичное использование или упрощенная реализация: частичное снижение.
- Игнорирование важных технологических аспектов: сильное снижение оценки.

3. Архитектурная продуманность и масштабируемость решения

Оценивает архитектурную организацию проекта, структуру кода, документацию и модульность.

- Ясная и правильная организация проекта: высокий балл.
- Невнятная структура, дублирование кода, плохая документация: значительное снижение оценки.

4. Исполнение обязательных функциональных требований

Показывает, насколько полно реализованы функциональные требования проекта.

- Полноценная реализация всех указанных функций: максимальный балл.
- Некоторые функции работают некорректно или отсутствуют: снижаются баллы пропорционально объему неисполнения.

5. Демонстрация работы проекта и презентация

Рассматривает проведение публичной демонстрации проекта и его публичную защиту.

- Профессиональная и уверенная презентация, качественная демонстрация проекта: максимальная оценка.
- Неготовность презентовать проект, плохо проведенная демонстрация: заметное снижение оценки.

Оценивание ответа – максимальный балл - 10:

Баллы для учета в рейтинге (оценка ответа)	Степень удовлетворения критериям
9-10 баллов «отлично»	Проект выполнен на высоком уровне, техническое задание исполнено детально и аккуратно, студент продемонстрировал отличное владение инструментами и технологиями.
7-8 баллов «хорошо»	Хорошая реализация, возможно незначительное несоответствие техзаданию, приемлемая архитектура и функциональность.
5-6 баллов «удовлетворительно»	Выявлены серьезные недочеты в архитектуре, некоторая функциональность отсутствует или работает нестабильно.
4 балла и меньше «неудовлетворительно»	Большое количество технических ошибок, неполная реализация, несоблюдение поставленных задач.

Критерии оценивания Лабораторных работ:

1. Теоретическая подготовка студента

Оценивается знание теоретического материала, соответствующего тематике лабораторной работы. Студент должен продемонстрировать понимание основополагающих принципов и методик, применяемых в работе.

2. Правильность выполнения заданий

Проверяется точность исполнения всех пунктов задания, включая расчеты, эксперименты и оформление отчета. Если задание предусматривает расчетные данные — важно проверить правильность вычислений и обоснованность полученных выводов.

3. Качество оформления отчёта

Отчет должен соответствовать установленным требованиям по структуре и оформлению (шрифт, заголовки, нумерация страниц). Оформление должно быть

аккуратным и четким, иллюстративный материал должен сопровождаться пояснениями и выводами

Оценивание выполнения Лабораторной работы – максимальный балл - 4:

Баллы для учета в рейтинге	Степень удовлетворения критериям
4 балла «отлично»	Правильность и самостоятельность выполнения лабораторной работы. Понимание теоретического материала по теме. На все ответы даны правильные ответы и задание сдано в срок.
3 балла «хорошо»	Допущена ошибка, но не более чем в 1-ом ответе. Или работа сдана с опозданием.
2 балла «удовлетворительно»	В целом все задания лабораторной работы выполнены, но с одной грубой ошибкой. При наводящих вопросах преподавателя дает правильные ответы. Работа сдана с опозданием.
1 балл и меньше «неудовлетворительно»	Работа выполнена не полностью, нет правильных ответов; допущены две и более грубые ошибки, которые не исправляются по требованию педагога. Студент не может ответить на уточняющие и дополнительные вопросы преподавателя.