МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

Рабочая программа дисциплины (модуля),
как компонент образовательной программы
базового высшего образования
по направлению подготовки
09.03.01 Информатика и вычислительная техника,
утвержденной директором ФГАОУ ВО РУТ
(МИИТ) Париновым Д.В.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Алгоритмы и структуры данных

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Технологии разработки программного обеспечения

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде электронного документа выгружена из единой корпоративной информационной системы управления университетом и соответствует оригиналу

Простая электронная подпись, выданная ФГАОУ ВО РУТ (МИИТ)
ID подписи: 937226
Подписал: директор центра Проневич Ольга Борисовна
Дата: 12.11.2026

1. Общие сведения о дисциплине (модуле).

Дисциплина формирует фундаментальную инженерную способность разработчика бизнес-приложений выбирать алгоритм и структуру данных по профилю реальной нагрузки программы, а не по привычке, и доказывать этот выбор корректностью, аналитической оценкой сложности и воспроизводимым измерением. Рыночная востребованность такой подготовки определяется дефицитом ИТ-кадров с фундаментальным образованием при устойчивом спросе на Java-разработчиков в Enterprise-сегменте и цифровых платформах транспорта. Студент работает в роли младшего инженера-исследователя над сквозным кейсом «Расчётное ядро транспортного сервиса» на обезличенных материалах индустриального партнёра: он формализует задачу выбора расчётной даты из потока транспортных событий, строит заведомо корректную базовую реализацию, проектирует и кодирует собственные структуры данных, собирает из них оптимизированный вариант, измеряет оба решения на идентичных наборах и завершает семестр разработкой планировщика фоновых заданий в технологическом окне. Результатом является защищаемое портфолио – библиотека алгоритмов и структур данных на Java с проверками, таблицей сложности и инженерной рекомендацией. Все занятия проводятся в импортозамещённом контуре – Astra Linux, Liberica JDK, Maven, JUnit 5, JMH, GitVerse, Р7-Офис.

Целью освоения дисциплины является формирование способности формализовать вычислительную задачу, выбирать и реализовывать подходящие алгоритмы и структуры данных, проверять корректность решения и оценивать его эффективность аналитически и экспериментально.

Для достижения поставленной цели в рамках дисциплины решается комплекс задач, направленных на формирование у обучающихся способности: формализовать прикладную задачу в виде спецификации входов, операций, ограничений и инвариантов. Разрабатывать заведомо корректную базовую реализацию и набор контрольных примеров, покрывающих нормальные, граничные и ошибочные сценарии. Реализовывать линейные и нелинейные структуры данных на языке Java с подтверждением контрактов операций автоматизированными проверками. Оценивать временную и пространственную сложность решений и стоимость предварительной подготовки данных. Проводить воспроизводимое сравнительное измерение вариантов реализации и объяснять расхождение теории и практики. Компоновать изученные структуры в работающий

программный компонент и обосновывать инженерный выбор с явным указанием цены решения.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ОПК-6 - Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения;

ППК-Р1 - Способен разрабатывать и отлаживать программный код.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

- методы создания и документирования контрольных примеров и тестовых наборов данных;
- требования к структуре и форматам хранения тестовых наборов данных;
- правила, алгоритмы и технологии создания тестовых наборов данных;
- методы и средства проверки работоспособности компьютерного программного обеспечения;
- государственные стандарты испытания автоматизированных систем;
- руководящие документы по стандартизации требований к документам автоматизированных систем;
- алгоритмы решения типичных задач, области и способы их применения;
- нотации и программное обеспечение для графического отображения алгоритмов;
- методы и приемы алгоритмизации поставленных задач;
- основные классы методов программной реализации моделей и критерии выбора алгоритмов.

Уметь:

- разрабатывать и оформлять контрольные примеры для проверки работоспособности компьютерного программного обеспечения;
- готовить тестовые наборы данных в соответствии с выбранной методикой тестирования компьютерного программного обеспечения;
- применять методы и средства проверки работоспособности компьютерного программного обеспечения;

- интерпретировать диагностические данные проверки работоспособности компьютерного программного обеспечения;

- анализировать значения полученных характеристик компьютерного программного обеспечения;

- использовать методы и приемы формализации и алгоритмизации поставленных задач;

- применять алгоритмы решения типовых задач в соответствующих областях;

- проводить сравнительный анализ и обоснование выбора алгоритмов для программной реализации модели.

Владеть:

- методами адаптации методов и алгоритмов под специфику задачи;

- навыками оценки эффективности выбранных алгоритмов.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 3 з.е. (108 академических часа(ов).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов	
	Всего	Семестр №2
Контактная работа при проведении учебных занятий (всего):	80	80
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	48	48

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 28 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме

контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Предмет дисциплины и инженерная постановка вычислительной задачи Рассматриваемые вопросы: - место алгоритмов и структур данных в роли инженера-программиста и стоимость неэффективного решения; - формализация прикладной задачи: входы, выходы, ограничения, инварианты; - постановка сквозного кейса «выбор источника расчётной даты» по материалам индустриального партнёра.
2	Абстрактные типы данных и базовая реализация алгоритма Рассматриваемые вопросы: - абстрактный тип данных как контракт операций, отделение интерфейса от реализации; - заведомо корректная базовая реализация в роли проверочного оракула; - псевдокод и нотации графического отображения алгоритмов.
3	Асимптотический анализ сложности Рассматриваемые вопросы: - временная и пространственная сложность, O-нотация, худший и средний случаи; - амортизационный анализ на примере расширения динамического массива; - чтение и сравнение оценок сложности в документации стандартной библиотеки.
4	Массив и динамический массив Рассматриваемые вопросы: - устройство массива в памяти и стоимость операций доступа, вставки, удаления; - стратегия расширения динамического массива и амортизированная стоимость; - соответствие собственной реализации и ArrayList из стандартной библиотеки Java.
5	Связные списки Рассматриваемые вопросы: - односвязный и двусвязный списки, стоимость операций в сравнении с массивом; - типовые ошибки работы со ссылками и их диагностика; - критерии выбора между массивом и списком по профилю операций.
6	Стек, очередь и дек Рассматриваемые вопросы: - контракты LIFO и FIFO, реализация на массиве и на связном списке; - кольцевой буфер и блочные очереди как инженерный компромисс; - применение стека и очереди в обработке событий и отложенных вычислениях.
7	Хеш-таблицы Рассматриваемые вопросы: - хеш-функции и требования к ним, равномерность распределения; - разрешение коллизий цепочками и открытой адресацией, фактор загрузки, рехеширование; - устройство HashMap и контракт equals/hashCode в Java.
8	Деревья поиска Рассматриваемые вопросы:

№ п/п	Тематика лекционных занятий / краткое содержание
	- двоичное дерево поиска, инвариант упорядоченности, стоимость операций; - деградация дерева в список и идея самобалансировки (АВЛ, красно-чёрные деревья); - упорядоченные отображения TreeMap и их роль в задаче выбора расчётной даты.
9	Приоритетная очередь и двоичная куча Рассматриваемые вопросы: - инвариант двоичной кучи, просеивание вверх и вниз, построение кучи за линейное время; - пирамидальная сортировка; - PriorityQueue в Java и применение кучи в событийном планировании.
10	Алгоритмы сортировки Рассматриваемые вопросы: - квадратичные сортировки вставками и выбором, их ниша на малых и частично упорядоченных данных; - сортировка слиянием, быстрая и пирамидальная сортировки: устойчивость, память, худший случай; - нижняя оценка сортировки сравнениями и гибридные сортировки в стандартной библиотеке.
11	Алгоритмы поиска и стоимость подготовки данных Рассматриваемые вопросы: - последовательный и двоичный поиск, требования к данным; - компромисс «подготовка – запрос»: когда сортировка и индексация окупаются; - выбор способа поиска по соотношению числа запросов и обновлений.
12	Методика воспроизводимого вычислительного эксперимента Рассматриваемые вопросы: - постановка измерения: идентичные наборы данных, прогрев виртуальной машины, контроль шума; - инфраструктура микробенчмарков JMH и типовые ошибки микрозамеров; - интерпретация и документирование расхождения аналитической оценки и измерения.
13	Представление графов и обходы Рассматриваемые вопросы: - ориентированный граф как модель зависимостей фоновых обработок; - матрица и списки смежности, выбор представления по плотности графа; - обходы в глубину и в ширину на собственных стеке и очереди, их стоимость.
14	Ацикличность и топологическая сортировка Рассматриваемые вопросы: - обнаружение циклов в ориентированном графе; - алгоритм Кана и топологическая сортировка на основе обхода в глубину; - применение к проверке корректности графа зависимостей заданий.
15	Событийное планирование фоновых заданий Рассматриваемые вопросы: - модель технологического окна: длительности, приоритеты, сроки, конкурирующие ресурсы; - событийный планировщик на приоритетной очереди как композиция изученных структур; - сравнение простого и приоритетного вариантов планирования, анализ причин ожидания.
16	Инженерный выбор структур данных под профиль нагрузки Рассматриваемые вопросы: - типовые профили нагрузки бизнес-приложений: потоковая запись, частые запросы, пакетная обработка; - комплексные критерии выбора структуры: время операций, память, стоимость подготовки и обновления; - композиция структур данных в промышленном компоненте и границы применимости решения.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Базовая реализация выбора расчётной даты Студент реализует на Java алгоритм последовательного просмотра событий по спецификации первого практического занятия. Он кодирует правила приоритета версий, отмен и запрета расчёта без обязательных данных. Реализация регистрируется в системе управления версиями и принимает роль эталона для последующих проверок.
2	Разработка контрольных примеров Студент строит на JUnit 5 наборы нормальных, граничных и ошибочных сценариев для эталонной реализации. Он документирует структуру тестовых наборов и добивается прохождения всех проверок. Полученный набор становится общим для всех последующих вариантов решения задачи.
3	Реализация динамического массива Студент реализует динамический массив с выбранной стратегией расширения без использования готовых коллекций. Он подтверждает контракт операций контрактными проверками и измеряет фактическую стоимость вставки при росте массива. Результат сопоставляется с поведением ArrayList.
4	Реализация связного списка и сравнение хранилищ Студент реализует односвязный и двусвязный списки с проверкой инвариантов при вставке и удалении. Он подключает оба хранилища к эталонной реализации через общий интерфейс и прогоняет общий набор проверок. По итогам запуска фиксируется фактическая стоимость операций массива и списка.
5	Реализация стека, очереди и дека Студент реализует стек, очередь и дек на массиве и на списке по проекту третьего практического занятия. Каждая структура проверяется контрактными тестами на порядок извлечения и граничные состояния. Структуры включаются в общую библиотеку с историей изменений.
6	Кольцевой буфер приёма событий Студент реализует кольцевой буфер ограниченной ёмкости и подключает его к генератору потока транспортных событий. Он проверяет поведение при переполнении и опустошении и фиксирует выбранную реакцию буфера. Проводится сопоставление с очередью на списке по стоимости операций.
7	Реализация хеш-таблицы Студент реализует хеш-таблицу с разрешением коллизий цепочками и управлением фактором загрузки. Он проверяет равномерность распределения выбранной хеш-функции на данных задачи и наблюдает рост стоимости поиска при переполнении. Реализация сопоставляется с HashMap на идентичных операциях.
8	Реализация двоичного дерева поиска Студент реализует двоичное дерево поиска с операциями вставки, поиска и удаления и проверкой инварианта упорядоченности. Он демонстрирует деградацию дерева на упорядоченной вставке и объясняет причину. Результат сопоставляется с TreeMap по стоимости операций.
9	Реализация двоичной кучи Студент реализует двоичную кучу с просеиванием вверх и вниз и построением из массива. Он проверяет инвариант кучи после каждой операции контрактными тестами. Куча сопоставляется с PriorityQueue и готовится к использованию в планировщике.
10	Оптимизированный вариант выбора расчётной даты Студент собирает индексированный вариант задачи из собственных хеш-таблицы и упорядоченной структуры по проекту пятого практического занятия. Он сохраняет внешний контракт базовой реализации и прогоняет общий набор проверок на обоих вариантах. Проверка эквивалентности результатов двух вариантов фиксируется в журнале.
11	Реализация сортировок и двоичного поиска Студент реализует сортировки вставками, слиянием и быструю, а также двоичный поиск. Он

№ п/п	Наименование лабораторных работ / краткое содержание
	применяет сортировки для подготовки наборов данных разного масштаба и проверяет устойчивость и стоимость на частично упорядоченных данных. Реализации пополняют библиотеку и проходят общий набор проверок.
12	Воспроизводимое измерение вариантов Студент ставит измерение на JMH по программе испытаний шестого практического занятия: базовый и оптимизированный варианты сравниваются на идентичных наборах. Он фиксирует время запросов, стоимость подготовки и память. Расхождение аналитической оценки и измерения объясняется письменно в протоколе.
13	Граф зависимостей и топологический порядок Студент реализует представление графа зависимостей фоновых обработок списками смежности и обходы в глубину и в ширину на собственных структурах. Он реализует обнаружение циклов и топологическую сортировку алгоритмом Кана. Проверяются графы с циклами, с изолированными вершинами и полный учебный каталог обработок.
14	Событийный планировщик технологического окна Студент реализует планировщик фоновых заданий на собственной двоичной куче по проекту седьмого практического занятия. Он моделирует выполнение каталога обработок с длительностями, сроками и конкурирующими ресурсами. Строится расписание простого варианта и журнал событий моделирования.
15	Приоритетный вариант планировщика и сравнение Студент добавляет в планировщик учёт приоритетов и сроков заданий и повторяет моделирование на том же каталоге. Он сравнивает оба расписания по укладке в технологическое окно и времени ожидания заданий. Причины расхождений расписаний фиксируются в отчёте моделирования.
16	Сборка портфолио библиотеки расчётного ядра Студент приводит репозиторий к итоговому виду: спецификации, обе реализации каждой задачи, общий набор проверок, таблица сложности и протоколы измерений. Он выполняет полный прогон всех проверок и измерений с нуля и подтверждает воспроизводимость сборки. Портфолио передаётся на приёмку и защиту.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Формализация задачи выбора источника расчётной даты Студент анализирует обезличенное описание правил выбора даты из транспортных событий с приоритетами версий, отменами и запретом расчёта без обязательных данных. Он выделяет входы, операции, ограничения и инварианты, фиксирует критерии корректности результата. Итогом становится спецификация и псевдокод, оформленные в Р7-Офис, которые принимаются как техническое задание на первую лабораторную работу.
2	Проектирование линейных хранилищ данных Студент сопоставляет профиль операций задачи с контрактами массива и связного списка и выбирает внутреннее устройство хранилища событий. Он проектирует интерфейсы динамического массива и списка, определяет стратегию расширения и составляет матрицу ожидаемой стоимости операций. Решения заносятся в схему структуры с обоснованием выбора.
3	Проектирование очередей обработки событий Студент моделирует поток входящих транспортных событий и определяет, где требуется порядок LIFO, где FIFO, а где двусторонний доступ. Он выбирает внутреннюю реализацию для стека, очереди и дека между массивом и списком и обосновывает выбор стоимостью операций. Отдельно проектируется кольцевой буфер ограниченной ёмкости для приёма событий. Проектные решения становятся заданием на реализацию.
4	Проектирование индексов: хеш-таблица и дерево поиска Студент анализирует запросы к задаче выбора расчётной даты и определяет ключи индексирования.

№ п/п	Тематика практических занятий/краткое содержание
	Он выбирает хеш-функцию и способ разрешения коллизий, рассчитывает допустимый фактор загрузки, а также проектирует упорядоченную структуру для диапазонных запросов. Оба проекта сопоставляются в матрице выбора по стоимости вставки, поиска и удаления.
5	Проектирование приоритетной очереди и оптимизированного варианта Студент проектирует двоичную кучу для управления заданиями по приоритету и сроку. Затем он собирает из спроектированных ранее структур оптимизированный вариант задачи выбора расчётной даты с тем же внешним контрактом, что у базовой реализации. Он планирует проверку эквивалентности двух вариантов на общих наборах данных.
6	Планирование сравнительного эксперимента Студент определяет, какие сортировки пригодны для подготовки наборов данных разного масштаба, учитывая устойчивость, память и частичную упорядоченность. Он составляет план измерений: перечень сравниваемых вариантов, наборы данных, число повторов и фиксируемые показатели. План оформляется таблицей и принимается как программа испытаний.
7	Моделирование графа зависимостей фоновых обработок Студент переводит обезличенный каталог фоновых обработок в ориентированный граф зависимостей и выбирает его программное представление по плотности. Вручную он проверяет граф на циклы и строит допустимый порядок запуска, затем проектирует событийный планировщик технологического окна. Итогом является схема модуля с выбранными структурами данных.
8	Анализ расписания и подготовка инженерной рекомендации Студент разбирает смоделированные расписания простого и приоритетного вариантов планировщика, объясняет причины ожидания заданий и оценивает укладку в технологическое окно. Он сводит аналитические оценки и измерения всего семестра в итоговую таблицу и формулирует рекомендацию по выбору вариантов под разные режимы нагрузки. Рекомендация оформляется отчётом по ГОСТ в Р7-Офис и готовится к защите.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендованной литературы.
2	Подготовка к лабораторным и практическим работам.
3	Подготовка к промежуточной аттестации.
4	Подготовка к текущему контролю.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Вирт, Н. Алгоритмы и структуры данных. Новая версия для Оберона : учебное пособие / Н. Вирт. — Москва : ДМК Пресс, 2010. — 272 с. — ISBN 978-5-94074-584-6. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/1261 (дата обращения: 16.10.2026).
2	Павлов, Л. А. Структуры и алгоритмы обработки данных / Л. А. Павлов, Н. В. Первова. — 2-е изд.,	Лань : электронно-библиотечная система. — URL:

	стер. — Санкт-Петербург : Лань, 2022. — 256 с. — ISBN 978-5-507-44105-1. — Текст : электронный	https://e.lanbook.com/book/207563 (дата обращения: 16.10.2026).
3	Нидхем, М. Графовые алгоритмы : руководство / М. Нидхем, Э. Холдер ; перевод с английского В. С. Яценкова. — Москва : ДМК Пресс, 2020. — 258 с. — ISBN 978-5-97060-799-2. — Текст : электронный	ань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/140578 (дата обращения: 16.10.2026).
4	Окулов, С. М. Программирование в алгоритмах : учебное пособие / С. М. Окулов. — 7-е изд. — Москва : Лаборатория знаний, 2025. — 386 с. — ISBN 978-5-93208-521-9. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/507633 (дата обращения: 16.10.2026).
5	Тюкачев, Н. А. C#. Алгоритмы и структуры данных / Н. А. Тюкачев, В. Г. Хлебостроев. — 6-е изд., стер. — Санкт-Петербург : Лань, 2023. — 232 с. — ISBN 978-5-507-47248-2. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/346067 (дата обращения: 16.10.2026).
6	Хиценко, В. П. Структуры данных и алгоритмы : учебное пособие / В. П. Хиценко. — Новосибирск : НГТУ, 2016. — 64 с. — ISBN 978-5-7782-2958-7. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/118222 (дата обращения: 16.10.2026).
7	Пантелеев, Е. Р. Алгоритмы и структуры данных : учебное пособие / Е. Р. Пантелеев, А. Л. Алыкова. — Иваново : ИГЭУ, 2018. — 142 с. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/154576 (дата обращения: 16.10.2026).

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

ЭБС «Лань» – https://e.lanbook.com/ – доступ к основной и дополнительной литературе дисциплины.

Java SE API Documentation – https://docs.oracle.com/javase/8/docs/api/java/util/package-summary.html – контракты и гарантии сложности коллекций JDK (ArrayList, HashMap, TreeMap, PriorityQueue, ArrayDeque).

OpenJDK – https://openjdk.org/ – спецификации и сборки открытой платформы Java.

Apache Maven – https://maven.apache.org/guides/ – официальные руководства по воспроизводимой сборке проектов.

JUnit 5 User Guide – https://junit.org/junit5/docs/current/user-guide/ – официальное руководство по модульному тестированию.

JMH (Java Microbenchmark Harness) – https://github.com/openjdk/jmh – официальный репозиторий и руководство по микробенчмаркам.

Git – Documentation – https://git-scm.com/doc – официальная документация системы управления версиями.

GitVerse – https://gitverse.ru – российская платформа размещения и совместной разработки программного кода.

Р7-Офис – https://r7-office.ru – документация офисного пакета для оформления отчётности по ГОСТ.

Документация Astra Linux – https://wiki.astralinux.ru – официальная вики-база знаний отечественной операционной системы.

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Операционные системы: Astra Linux (реестр российского ПО) / ALT Linux / РЕД ОС.

Офисные пакеты: Р7-Офис / МойОфис (для подготовки спецификаций, протоколов измерений и отчётов по ГОСТ).

Платформа Java: Liberica JDK 21 LTS (BellSoft, реестр российского ПО) / OpenJDK.

Среда разработки: IntelliJ IDEA Community Edition / Eclipse IDE (оффлайн-установка, open source).

Система сборки: Apache Maven, Maven Surefire (автоматический прогон проверок).

Проверка и измерение: JUnit 5 (контрактные проверки), JMH (воспроизводимые микробенчмарки).

Управление версиями и хостинг кода: Git, платформа GitVerse / локальный Forgejo (open source).

Моделирование и схемы: PlantUML, Graphviz, draw.io desktop (open source, оффлайн-версии).

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

Учебные аудитории для проведения учебных занятий, оснащенные компьютерной техникой и наборами демонстрационного оборудования.

Для лабораторных и практических занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Зачет во 2 семестре.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

доцент кафедры «Цифровые технологии управления транспортными процессами» Института железнодорожного транспорта — И.И. Бутов

Согласовано:

Заведующий кафедрой ЦТУТП — В.Е. Нутович

Директор ЦС IT - программ ВИШ — О.Б. Проневич

Председатель учебно-методической комиссии — Н.А. Андриянова