

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 первым проректором РУТ (МИИТ)
Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Алгоритмы и структуры данных

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Вычислительные системы и сети

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 РУТ (МИИТ)
ID подписи: 5665
Подписал: заведующий кафедрой Нутович Вероника Евгеньевна
Дата: 22.06.2026

1. Общие сведения о дисциплине (модуле).

Дисциплина изучает фундаментальные алгоритмы и структуры данных как основу разработки компьютерных программ, пригодных для практического применения. Содержание ориентировано на студентов начальных семестров, осваивающих программирование на Java, и последовательно связывает теоретические модели данных, анализ вычислительной сложности, выбор алгоритмического подхода и реализацию корректных программных решений. В рамках дисциплины рассматриваются линейные структуры данных, деревья, кучи, хэш-таблицы, алгоритмы сортировки и поиска, графовые алгоритмы, рекурсия, перебор, жадные методы, динамическое программирование, комбинаторные и строковые алгоритмы. На лабораторных занятиях обучающиеся последовательно реализуют набор структур данных и алгоритмов на Java, проводят простую проверку корректности, оценивают временную и пространственную сложность и оформляют техническое описание принятых решений.

Целью освоения дисциплины является формирование способности разрабатывать алгоритмы и компьютерные программы на Java, пригодные для практического применения, на основе осознанного выбора структур данных, оценки вычислительной сложности, проверки корректности и документирования алгоритмических решений.

Для достижения поставленной цели в рамках дисциплины решается комплекс задач, направленных на формирование у обучающихся способности – объяснять свойства алгоритмов и абстрактных типов данных, оценивать временную и пространственную сложность программ, выбирать линейные, древовидные, хэшированные и графовые структуры данных под условия задачи, реализовывать базовые алгоритмы сортировки, поиска, обхода, кратчайших путей и обработки строк на Java, применять рекурсивные, жадные и динамические подходы к решению типовых задач, проверять корректность реализации и готовить техническую документацию.

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.

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

Знать:

- свойства алгоритма, абстрактного типа данных и компьютерной программы, пригодной для практического применения
- основы анализа временной и пространственной сложности алгоритмов
- асимптотические обозначения и правила сравнения роста функций
- устройство динамических массивов, стеков, очередей и деков
- устройство односвязных, двусвязных и циклических списков
- принципы выбора линейной структуры данных под характер операций
- терминологию деревьев, способы представления бинарных деревьев и базовые обходы
 - свойства бинарного дерева поиска и влияние формы дерева на эффективность операций
 - идеи балансировки деревьев поиска и назначение AVL-деревьев, красно-черных деревьев и B-деревьев
 - свойства бинарной кучи и назначение приоритетной очереди
 - принципы построения хэш-функций, коллизий и методов их разрешения
 - классификацию алгоритмов сортировки и условия выбора простых и эффективных сортировок
 - принципы линейного, бинарного и интерполяционного поиска
 - основные понятия графов и способы представления графов в памяти
 - принципы обхода графов в глубину и в ширину
 - идеи поиска кратчайших путей во взвешенных графах
 - рекурсию, метод разделения задачи, перебор с возвратом и условия завершения рекурсивного процесса
 - принцип жадного выбора и ограничения применимости жадных алгоритмов
 - принцип оптимальности и рекуррентные соотношения в динамическом программировании
 - комбинаторные объекты, строковые алгоритмы поиска подстроки и базовые идеи сжатия данных

Уметь:

- уметь оценивать временную и пространственную сложность алгоритма при помощи O-нотации в условиях программной реализации на Java
- уметь выбирать линейную структуру данных при помощи анализа операций доступа, вставки, удаления и обхода в условиях ограничений по времени и памяти
- уметь реализовывать стек, очередь и дек при помощи массивов и связанных узлов в условиях типовых задач обработки последовательностей

- уметь реализовывать связный список при помощи классов Java и ссылочной структуры узлов в условиях операций вставки, удаления, поиска и обхода

- уметь реализовывать бинарное дерево поиска при помощи рекурсивных и итеративных процедур в условиях операций поиска, вставки, удаления и обхода

- уметь применять идеи балансировки деревьев при помощи анализа высоты и вращений в условиях поддержания логарифмической сложности операций

- уметь реализовывать приоритетную очередь при помощи бинарной кучи в условиях частого извлечения элемента с наивысшим приоритетом

- уметь реализовывать хэш-таблицу при помощи хэш-функции и метода разрешения коллизий в условиях операций вставки, поиска и удаления

- уметь выбирать алгоритм сортировки при помощи анализа входных данных, устойчивости, памяти и асимптотической сложности в условиях обработки массивов Java

- уметь реализовывать алгоритмы поиска при помощи последовательного и бинарного подхода в условиях неотсортированных и отсортированных данных

- уметь представлять граф при помощи матрицы смежности и списков смежности в условиях разной плотности связей

- уметь реализовывать обходы графа и поиск кратчайшего пути при помощи DFS, BFS и алгоритма Дейкстры в условиях типовых задач на графах

- уметь решать типовые задачи при помощи рекурсии, жадного подхода и динамического программирования в условиях доказуемого ограничения перебора

- уметь реализовывать строковый поиск при помощи наивного алгоритма и алгоритма Кнута-Морриса-Пратта в условиях обработки текстовых данных

Владеть:

- навыком разработки Java-реализаций базовых структур данных с проверкой корректности операций

- навыком оценки асимптотической сложности программного решения

- навыком выбора структуры данных по профилю операций и ограничениям задачи

- навыком реализации сортировок, поиска, обходов деревьев и графов

- навыком применения рекурсии, жадных методов и динамического программирования к типовым алгоритмическим задачам

- навыком реализации алгоритмов обработки строк и комбинаторных переборов

- навыком подготовки набора проверочных примеров для алгоритмической программы

- навыком подготовки технической документации по структуре данных, алгоритму, сложности и ограничениям реализации

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 №4
Контактная работа при проведении учебных занятий (всего):	64	64
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	32	32

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 44 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	Алгоритм, абстрактный тип данных и вычислительная модель Рассматриваемые вопросы: - свойства алгоритма и требования к программному решению; - абстрактный тип данных и отличие интерфейса операций от реализации; - роль Java в реализации алгоритмов и структур данных.
2	Анализ сложности алгоритмов Рассматриваемые вопросы: - временная и пространственная сложность алгоритма; - асимптотические обозначения и сравнение роста функций; - анализ циклов, вложенных циклов и простых рекурсивных вызовов.
3	Массивы, динамические массивы и последовательный доступ Рассматриваемые вопросы: - массив как непрерывное представление последовательности; - динамическое расширение массива и амортизированная стоимость операций; - доступ по индексу, вставка, удаление и сдвиг элементов.
4	Стеки, очереди и деки Рассматриваемые вопросы: - стек, очередь и дек как абстрактные типы данных; - реализация на массиве и связанных узлах; - применение стеков и очередей в вычислении выражений, планировании и обработке последовательностей.
5	Связные списки Рассматриваемые вопросы: - односвязные, двусвязные и циклические списки; - операции вставки, удаления, поиска и обхода; - сравнение списков с массивами по памяти и сложности операций.
6	Сортировка простыми методами Рассматриваемые вопросы: - сортировка пузырьком, выбором и вставками; - устойчивость, число сравнений и число обменов; - области применимости простых сортировок и их ограничения.
7	Эффективные сортировки сравнением Рассматриваемые вопросы: - сортировка слиянием и разделение массива; - быстрая сортировка и выбор опорного элемента; - пирамидальная сортировка и связь с бинарной кучей.
8	Алгоритмы поиска в последовательностях Рассматриваемые вопросы: - линейный поиск в неупорядоченных данных; - бинарный поиск в отсортированном массиве; - интерполяционный поиск и требования к распределению значений.
9	Бинарные деревья и обходы Рассматриваемые вопросы: - корень, узел, лист, глубина, высота и поддереву; - прямой, симметричный, обратный обход и обход в ширину; - представление дерева узлами и связь обхода с порядком обработки данных.
10	Бинарное дерево поиска Рассматриваемые вопросы:

№ п/п	Тематика лекционных занятий / краткое содержание
	- инвариант бинарного дерева поиска; - операции поиска, вставки и удаления узла; - влияние высоты дерева на сложность операций.
11	Сбалансированные деревья поиска Рассматриваемые вопросы: - проблема вырождения бинарного дерева поиска; - идеи AVL-деревьев и красно-черных деревьев; - назначение B-деревьев при хранении больших объемов данных.
12	Бинарная куча и приоритетная очередь Рассматриваемые вопросы: - свойства min-heap и max-heap; - вставка, извлечение приоритетного элемента и построение кучи; - применение кучи в пирамидальной сортировке и очередях с приоритетом.
13	Хэш-таблицы Рассматриваемые вопросы: - хэш-функция, индексирование и коллизии; - метод цепочек и открытая адресация; - средняя и худшая сложность операций в хэш-таблице.
14	Графы и алгоритмы обхода Рассматриваемые вопросы: - вершины, ребра, ориентированные, неориентированные и взвешенные графы; - матрица смежности и списки смежности; - поиск в глубину и поиск в ширину.
15	Алгоритмы кратчайших путей и рекурсивные подходы Рассматриваемые вопросы: - алгоритм Дейкстры для графов с неотрицательными весами; - алгоритм Беллмана-Форда и учет отрицательных весов; - рекурсия, разделение задачи и перебор с возвратом.
16	Жадные, динамические, комбинаторные и строковые алгоритмы Рассматриваемые вопросы: - принцип жадного выбора и принцип оптимальности; - рекуррентные соотношения и таблицы динамического программирования; - генерация комбинаторных объектов, поиск подстроки и базовые идеи сжатия данных.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Оценка сложности алгоритмов на Java Студент подготавливает набор небольших Java-фрагментов с циклами, вложенными циклами и рекурсивными вызовами. Для каждого фрагмента фиксируется оценка времени и памяти. Результаты оформляются в виде таблицы с пояснением роста числа операций.
2	Реализация динамического массива Студент реализует динамический массив на Java с операциями доступа, добавления, вставки и удаления. В реализации проверяется расширение внутреннего массива и сохранение порядка элементов. Для операций подготавливаются проверочные вызовы и вывод результата.
3	Реализация стека и очереди Студент реализует стек и очередь на основе массива или связанных узлов. Для стека выполняется

№ п/п	Наименование лабораторных работ / краткое содержание
	проверка сбалансированности скобок, а для очереди моделируется обработка последовательности заявок. Результаты проверяются на обычных и граничных входных данных.
4	Реализация связного списка Студент создает односвязный и двусвязный список на Java. В коде реализуются вставка, удаление, поиск и обход элементов. Работа завершается сравнением поведения списка и массива на одинаковом наборе операций.
5	Реализация простых алгоритмов сортировки Студент реализует сортировку пузырьком, выбором и вставками. Для каждого алгоритма выполняется запуск на отсортированных, обратных и случайных данных. В отчете фиксируется число сравнений и перестановок для разных входов.
6	Реализация эффективных алгоритмов сортировки Студент реализует сортировку слиянием, быструю сортировку и пирамидальную сортировку. Для каждого алгоритма выполняется проверка корректности результата на одинаковых массивах. Дополнительно фиксируется потребность в дополнительной памяти.
7	Реализация алгоритмов поиска Студент реализует линейный и бинарный поиск. Для бинарного поиска подготавливаются отсортированные наборы данных и проверяются случаи наличия и отсутствия элемента. Результаты сравниваются по числу проверенных элементов.
8	Реализация рекурсивных алгоритмов Студент реализует рекурсивные функции для факториала, чисел Фибоначчи и обхода дерева. Для каждой функции задается условие останова и проверяются малые входные данные. При необходимости добавляется итеративный вариант для сравнения.
9	Реализация бинарного дерева поиска Студент реализует бинарное дерево поиска на Java. В программе выполняются вставка, поиск, удаление узла и три обхода дерева. Проверка проводится на наборах данных, создающих разные формы дерева.
10	Исследование балансировки деревьев поиска Студент реализует или программно моделирует малые и большие вращения в AVL-дереве. Для последовательностей вставок фиксируется изменение высоты дерева. Итогом становится проверка восстановления баланса после операций.
11	Реализация бинарной кучи и приоритетной очереди Студент реализует бинарную кучу с операциями вставки и извлечения приоритетного элемента. На основе кучи выполняется пирамидальная сортировка массива. Проверка включает восстановление свойства кучи после каждой операции.
12	Реализация хэш-таблицы Студент реализует хэш-таблицу с выбранной хэш-функцией и методом разрешения коллизий. В программе выполняются вставка, поиск и удаление пар ключ-значение. Для проверки создаются входные данные с намеренными коллизиями.
13	Представление графа и обходы Студент реализует представление графа матрицей смежности и списками смежности. Для одного набора вершин и ребер выполняются поиск в глубину и поиск в ширину. Результаты обходов выводятся как последовательности посещения вершин.
14	Поиск кратчайшего пути в графе Студент реализует алгоритм Дейкстры для взвешенного графа с неотрицательными весами. В программе вычисляются расстояния от исходной вершины до остальных вершин. Для проверки используется граф с несколькими альтернативными маршрутами.
15	Жадные алгоритмы и динамическое программирование Студент реализует жадное решение задачи выбора заявок и динамическое решение одной

№ п/п	Наименование лабораторных работ / краткое содержание
	классической задачи. В коде фиксируются промежуточные решения и итоговый ответ. Результаты сопоставляются с условиями применимости выбранного подхода.
16	Комбинаторные и строковые алгоритмы Студент реализует генерацию одного вида комбинаторных объектов и поиск подстроки в тексте. Для строковой задачи выполняется наивный алгоритм и алгоритм Кнута-Морриса-Пратта. Результаты сравниваются по числу сопоставлений символов.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендованной литературы.
2	Подготовка к лабораторным 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	Тюкачев, Н. А. С#. 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: 22.06.2026)
2	Кораблин, Ю. П. Структуры и алгоритмы обработки данных : учебно-методическое пособие / Ю. П. Кораблин, В. П. Сыромятников, Л. А. Скворцова. — Москва : РТУ МИРЭА, 2020. — 219 с. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/163860 (дата обращения: 22.06.2026)
3	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: 22.06.2026)
4	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: 22.06.2026)

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

ЭБС Лань – <https://e.lanbook.com/>.

Образовательная платформа Юрайт – <https://urait.ru/>.

Единый реестр российских программ для ЭВМ и баз данных – <https://reestr.digital.gov.ru/reestr/>.

Документация Java Platform, Standard Edition – <https://docs.oracle.com/en/java/javase/>.

Документация Maven – <https://maven.apache.org/guides/>.

Документация JUnit 5 – <https://junit.org/junit5/docs/current/user-guide/>.

Документация Git – <https://git-scm.com/doc>.

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

Операционные системы – Astra Linux, ALT Linux, РЕД ОС, Debian GNU/Linux.

Среда разработки и сборки – OpenJDK, IntelliJ IDEA Community Edition или Eclipse IDE, Maven или Gradle.

Проверка программ – JUnit 5, Java Microbenchmark Harness.

Сопровождение проекта – Git, Markdown.

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 занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Зачет в 4 семестре.

10. Оценочные материалы.

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

Авторы:

И.И. Бутов

Согласовано:

Заведующий кафедрой ВССиИБ

Б.В. Желенков

Заведующий кафедрой ЦТУТП

В.Е. Нутович

Председатель учебно-методической
комиссии

Н.А. Андриянова