

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ястребов Олег Александрович

Должность: Ректор

Дата подписания: 25.05.2026 12:25:52

Уникальный программный ключ:

sa953a01204891083f939673078ef1a989dae18a

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский университет дружбы народов имени Патриса Лумумбы»

Факультет искусственного интеллекта

(наименование основного учебного подразделения (ОУП)-разработчика ОП ВО)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

(наименование дисциплины/модуля)

Рекомендована МССН для направлений подготовки:

**02.03.02 ФУНДАМЕНТАЛЬНАЯ ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ;**

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

(код и наименование направления подготовки/специальности)

Освоение дисциплины ведется в рамках реализации основной профессиональной образовательной программы высшего образования (ОП ВО):

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ: РАЗРАБОТКА И ОБУЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

(наименование (профиль/специализация) ОП ВО)

2026 г.

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина «Алгоритмы и структуры данных» входит в программу бакалавриата «Искусственный интеллект: разработка и обучение интеллектуальных систем» по направлениям подготовки 02.03.02 Фундаментальная информатика и информационные технологии и 09.03.03 Прикладная информатика, и изучается во 2 семестре 1 курса. Дисциплину реализует Кафедра прикладного искусственного интеллекта. Дисциплина состоит из 3 разделов и 26 тем и направлена на изучение фундаментальных алгоритмов и структур данных — сортировок, поиска, хеширования, деревьев, графовых алгоритмов, жадных стратегий и динамического программирования — с акцентом на анализ вычислительной сложности, обоснованный выбор структур данных для задач обработки и хранения данных в системах искусственного интеллекта, а также на проектирование эффективных программных решений.

Целью освоения дисциплины является формирование у студентов системного владения аппаратом анализа алгоритмов и структур данных для решения профессиональных задач в области ИИ: умения оценивать временную и пространственную сложность алгоритмов, обоснованно выбирать структуры данных для ML-пайплайнов и информационных систем с компонентами ИИ, проектировать эффективные программные решения с учётом требований к производительности, а также развитие навыков критического анализа и аргументированного выбора алгоритмических подходов.

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины «Алгоритмы и структуры данных» направлено на формирование у обучающихся следующих компетенций (части компетенций):

Таблица 2.1. Перечень компетенций, формируемых у обучающихся при освоении дисциплины (результаты освоения дисциплины)

Шифр	Компетенция	Индикаторы достижения компетенции (в рамках данной дисциплины)
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.3 Владеет навыками научного поиска и практической работы с информационными источниками; методами принятия решений;
ОПК-3	Способен разрабатывать алгоритмические и программные решения в области системного и прикладного программирования, математических и информационных моделей, баз данных, средств тестирования, пригодные для практического применения	ОПК-3.1 Знает основные алгоритмы и структуры данных, парадигмы программирования, принципы проектирования программных систем и баз данных; ОПК-3.3 Владеет навыками создания, тестирования и отладки алгоритмических и программных решений для систем ИИ, включая разработку пайплайнов обработки данных и обучения моделей;
ПК-2	Способен проектировать архитектуру информационных систем с компонентами ИИ, разрабатывать прототипы и базы данных таких систем	ПК-2.1. Проектирует архитектуру ИС с компонентами ИИ, выбирает архитектурные паттерны и технологический стек;
PL-1	Способен применять язык программирования Python для решения задач в области ИИ	PL-1.1. Разрабатывает и отлаживает прикладные решения разной сложности и для разного круга конечных пользователей с использованием языка программирования Python, тестирует, испытывает и оценивает качество таких решений;

3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Алгоритмы и структуры данных» относится к обязательной части блока 1 «Дисциплины (модули)» образовательной программы высшего образования.

В рамках образовательной программы высшего образования обучающиеся также осваивают другие дисциплины и/или практики, способствующие достижению запланированных результатов освоения дисциплины «Алгоритмы и структуры данных».

Таблица 3.1. Перечень компонентов ОП ВО, способствующих достижению запланированных результатов освоения дисциплины

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	Линейная алгебра; Дискретная математика; Математический анализ;	Теория вероятностей и математическая статистика; Статистические методы и первичный анализ данных; Онтология и графы знаний; Введение в базы данных; Hadoop, SPARK; Преддипломная практика;
ОПК-3	Способен разрабатывать алгоритмические и программные решения в области системного и прикладного программирования, математических и информационных моделей, баз данных, средств тестирования, пригодные для практического применения	Дискретная математика; История и теория программирования;	Программирование на языке C++; Введение в базы данных; Программирование на языке Python; Параллельное и распределенное программирование; Методы разработки решений на основе искусственного интеллекта (Git, Docker); Методы машинного обучения; Нейронные сети; Технологическая (проектно-технологическая) практика (производственная);
ПК-2	Способен проектировать архитектуру информационных систем с компонентами ИИ, разрабатывать прототипы и базы данных таких систем		Эксплуатационная практика (производственная); Преддипломная практика; Эксплуатационная практика (учебная); Технологическая (проектно-технологическая) практика (производственная); Программирование на языке C++; Параллельное и распределенное программирование; Методы разработки решений на основе искусственного интеллекта (Git, Docker); Hadoop, SPARK; Массово-параллельные вычисления в машинном обучении (GPU); MLOps и промышленная разработка систем искусственного интеллекта; Практическая подготовка на проектах отраслевых промышленных партнеров; Проектирование и разработка систем компьютерного зрения; Практикум по обработке естественного

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
			языка (NLP); Программирование на языке Python; Основы глубокого обучения; <i>Вайб-кодинг</i> **; Введение в базы данных; Онтология и графы знаний;
PL-1	Способен применять язык программирования Python для решения задач в области ИИ		Технологическая (проектно-технологическая) практика (производственная); Эксплуатационная практика (производственная); Программирование на языке Python; <i>Вайб-кодинг</i> **; Статистические методы и первичный анализ данных; Методы машинного обучения; Основы глубокого обучения; Параллельное и распределенное программирование; Hadoop, SPARK;

* - заполняется в соответствии с матрицей компетенций и СУП ОП ВО

** - элективные дисциплины /практики

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общая трудоемкость дисциплины «Алгоритмы и структуры данных» составляет «3» зачетные единицы.

Таблица 4.1. Виды учебной работы по периодам освоения образовательной программы высшего образования для очной формы обучения.

Вид учебной работы	ВСЕГО, ак.ч.		Семестр(-ы)
			2
<i>Контактная работа, ак.ч.</i>	51		51
Лекции (ЛК)	17		17
Лабораторные работы (ЛР)	0		0
Практически/семинарские занятия (СЗ)	34		34
<i>Самостоятельная работа обучающихся, ак.ч.</i>	57		57
<i>Контроль (экзамен/зачет с оценкой), ак.ч.</i>	0		0
Общая трудоемкость дисциплины	ак.ч.	108	108
	зач.ед.	3	3

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Таблица 5.1. Содержание дисциплины (модуля) по видам учебной работы

Номер раздела	Наименование раздела дисциплины	Наименование темы	Содержание темы	Вид учебной работы*	Формируемые индикаторы	
Раздел 1	Основы анализа алгоритмов. Сортировки и поиск	1.1	Введение в анализ алгоритмов. Асимптотическая сложность	Понятие алгоритма. Критерии оценки: корректность, время, память. Асимптотические обозначения: O , Ω , Θ . Классы сложности: $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(2^n)$, $O(n!)$. Сравнение функций роста. Анализ лучшего, среднего и худшего случая. Связь с эффективностью ML-пайплайнов: почему $O(n^2)$ для предобработки 10^6 записей неприемлемо	ЛК	ОПК-3.1, УК-1.3
		1.2	Элементарные сортировки и эффективные сортировки	Элементарные сортировки: вставками ($O(n^2)$), выбором ($O(n^2)$), пузырьком ($O(n^2)$). Эффективные сортировки: слиянием ($O(n \log n)$, стабильная), быстрая ($O(n \log n)$ среднее, $O(n^2)$ худшее). Нижняя граница сортировки сравнениями: $\Omega(n \log n)$. Сортировка подсчётом и поразрядная ($O(n+k)$). Выбор алгоритма сортировки для различных задач: сортировка датасета по признаку, top-K элементов	ЛК	ОПК-3.1, PL-1.1
		1.3	Поиск: линейный, бинарный, интерполяционный	Линейный поиск: $O(n)$. Бинарный поиск: $O(\log n)$, требование упорядоченности. Вариации: нижняя/верхняя граница (bisect_left, bisect_right). Интерполяционный поиск: $O(\log \log n)$ при равномерном распределении. Поиск в отсортированных данных: применение к поиску порога классификации (threshold tuning), подбор гиперпараметра на отсортированной сетке	ЛК	ОПК-3.1, PL-1.1, УК-1.3
		1.4	Практикум: анализ сложности алгоритмов	Определение асимптотической сложности для заданных фрагментов кода. Вложенные циклы, рекурсивные вызовы. Мастер-теорема для анализа рекурсий. Задачи: «какова сложность данного кода?», «какой алгоритм быстрее для данных размером n ?». Замеры времени на Python: %timeit	СЗ	ОПК-3.1, УК-1.3
		1.5	Практикум: реализация сортировок на Python	Реализация сортировки вставками, слиянием, быстрой сортировки на Python. Замер производительности на массивах разного размера (10^2 , 10^4 , 10^6). Сравнение с sorted() и numpy.sort(). Визуализация зависимости времени от размера входных данных (Matplotlib)	СЗ	PL-1.1, ОПК-3.1, ОПК-3.3
		1.6	Практикум: бинарный поиск и его вариации	Реализация бинарного поиска: итеративная и рекурсивная версии. Нижняя и верхняя граница (bisect). Задачи: поиск элемента, поиск первого вхождения, поиск в повернутом массиве. Применение: поиск оптимального порога классификации через бинарный поиск по метрике	СЗ	PL-1.1, ОПК-3.1
		1.7	Практикум: рекурсия и анализ рекурсивных алгоритмов	Рекурсия: базовый случай, рекурсивный вызов. Стек вызовов. Рекурсия vs итерация. Мемоизация (functools.lru_cache). Рекуррентные соотношения для анализа сложности: $T(n) = 2T(n/2) + O(n)$. Практика: реализация рекурсивных алгоритмов, анализ глубины рекурсии	СЗ	ОПК-3.1, PL-1.1, УК-1.3

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*	Формируемые индикаторы
		1.8	Практикум: алгоритмы для задач ИИ — top-K, медиана, квантили	Алгоритм быстрого выбора (QuickSelect): $O(n)$ среднее. Нахождение k-го порядкового элемента. Медиана и квантили. Применение: выбор top-K предсказаний модели, отсечение выбросов по квантилям, нахождение пороговых значений. Реализация на Python, сравнение с <code>numpy.partition()</code>	СЗ	PL-1.1, ОПК-3.3, ПК-2.1
		1.9	Контрольная работа: анализ сложности, сортировки, поиск	Самостоятельная работа: задачи на анализ сложности, выбор оптимального алгоритма, реализация алгоритма поиска/сортировки, обоснование выбора для прикладной задачи	СЗ	УК-1.3, ОПК-3.1, PL-1.1
Раздел 2	Структуры данных: от массивов до деревьев	2.1	Линейные структуры данных: стек, очередь, дек, связный список	Массив (array) vs динамический массив (list в Python). Связный список: односвязный, двусвязный. Стек: LIFO, операции push/pop, $O(1)$. Очередь: FIFO, операции enqueue/dequeue. Дек (deque): двусторонняя очередь. Приоритетная очередь (куча). Выбор структуры данных для задачи: стек для DFS, очередь для BFS, приоритетная очередь для жадных алгоритмов	ЛК	ОПК-3.1, PL-1.1, ПК-2.1
		2.2	Хеш-таблицы и множества	Хеш-функция: определение, свойства (равномерность, детерминированность). Хеш-таблица: принцип работы, разрешение коллизий (цепочки, открытая адресация). Сложность: $O(1)$ среднее для вставки/поиска/удаления. dict и set в Python: реализация через хеш-таблицу. Применение: подсчёт уникальных значений, словари признаков, кэширование результатов инференса	ЛК	ОПК-3.1, PL-1.1, ПК-2.1
		2.3	Деревья: бинарные, сбалансированные, кучи	Бинарное дерево: определение, обходы (in-order, pre-order, post-order, BFS). Бинарное дерево поиска (BST): вставка, поиск, удаление, $O(h)$. Проблема несбалансированности. Сбалансированные деревья: AVL, красно-чёрное (обзор). Куча (heap): определение, heapify, $O(\log n)$. Модуль heapq в Python. Деревья в ИИ: дерево решений, kd-tree для поиска ближайших соседей	ЛК	ОПК-3.1, PL-1.1, ПК-2.1
		2.4	Практикум: реализация стека, очереди, связного списка	Реализация стека и очереди на Python: через список и через collections.deque. Реализация односвязного списка с операциями: вставка, удаление, поиск. Задачи: проверка скобочной последовательности (стек), BFS на матрице (очередь), реверс связного списка	СЗ	PL-1.1, ОПК-3.1
		2.5	Практикум: хеш-таблицы и их применение	Реализация простейшей хеш-таблицы на Python с разрешением коллизий цепочками. Задачи: два числа с заданной суммой (two sum), подсчёт частот слов, группировка анаграмм. Применение: построение словаря признаков (feature vocabulary), кэширование предсказаний	СЗ	PL-1.1, ОПК-3.1, ОПК-3.3
		2.6	Практикум: бинарное дерево поиска и обходы	Реализация BST на Python: класс Node, вставка, поиск, удаление, обход. Визуализация дерева. Задачи: проверка BST-свойства, нахождение k-го минимального элемента, общий предок двух узлов. Анализ: когда BST деградирует до списка	СЗ	PL-1.1, ОПК-3.1, УК-1.3
		2.7	Практикум: куча и	Реализация кучи: heapify, push, pop. Использование heapq в Python. Задачи: k	СЗ	PL-1.1, ОПК-3.1,

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*	Формируемые индикаторы
			приоритетная очередь	ближайших элементов, объединение k отсортированных списков, медиана потока данных. Применение: top-K рекомендаций, планирование задач в ML-пайплайне с приоритетами		ПК-2.1
		2.8	Практикум: выбор структуры данных для системы ИИ	Кейс-задача: проектирование подсистемы хранения данных для рекомендательной системы. Требования: быстрый поиск по ID пользователя (хеш-таблица), ранжирование по рейтингу (куча), история действий (очередь/дек). Обоснование выбора структур данных. Оформление как фрагмент архитектурного решения	СЗ	ПК-2.1, УК-1.3, ОПК-3.3
		2.9	Практикум: бенчмаркинг структур данных Python	Сравнение производительности: list vs deque (вставка в начало), dict vs list (поиск), set vs list (проверка принадлежности), heapq vs sorted(). Замеры на данных разного размера. Визуализация результатов. Формулирование правил: «когда использовать что». Связь с оптимизацией ML-пайплайнов	СЗ	ОПК-3.3, PL-1.1, УК-1.3
Раздел 3	Графовые алгоритмы, жадные стратегии и динамическое программирование	3.1	Графы: представление и обходы	Граф: вершины, рёбра, ориентированные и неориентированные. Представление: матрица смежности, список смежности, список рёбер. Выбор представления для разных задач. Обходы: BFS ($O(V+E)$), DFS ($O(V+E)$). Компоненты связности. Топологическая сортировка. Применение: DAG вычислительного графа нейронной сети, граф зависимостей ML-пайплайна	ЛК	ОПК-3.1, PL-1.1, ПК-2.1
		3.2	Кратчайшие пути и минимальные остовные деревья	Алгоритм Дейкстры: $O((V+E) \log V)$ с кучей. Алгоритм Беллмана-Форда: $O(V \cdot E)$, отрицательные веса. Алгоритм Флойда-Уоршелла: $O(V^3)$, все пары. Минимальное остовное дерево: алгоритмы Краскала и Прима. Применение: кластеризация через MST, маршрутизация в логистике (кейс X5 Group), поиск похожих объектов через графы расстояний	ЛК	ОПК-3.1, PL-1.1, ПК-2.1
		3.3	Жадные алгоритмы и динамическое программирование	Жадные алгоритмы: принцип, условия оптимальности. Примеры: задача о рюкзаке (дробный), код Хаффмана, планирование задач. Динамическое программирование: принцип оптимальности Беллмана, мемоизация vs табуляция. Примеры: задача о рюкзаке (целочисленная), расстояние Левенштейна, наибольшая общая подпоследовательность. Связь с RL (уравнение Беллмана) и NLP (edit distance, beam search)	ЛК	ОПК-3.1, PL-1.1, УК-1.3
		3.4	Практикум: BFS и DFS на графах	Реализация BFS и DFS на Python (список смежности). Задачи: нахождение компонент связности, проверка двудольности, обнаружение цикла в ориентированном графе. Топологическая сортировка. Визуализация обхода с NetworkX. Применение: определение порядка выполнения этапов ML-пайплайна (DAG)	СЗ	PL-1.1, ОПК-3.1, ПК-2.1
		3.5	Практикум: кратчайшие пути — Дейкстра и Беллман-Форд	Реализация алгоритма Дейкстры на Python с использованием heapq. Реализация Беллмана-Форда. Задачи: кратчайший путь в взвешенном графе,	СЗ	PL-1.1, ОПК-3.1

Номер раздела	Наименование раздела дисциплины	Наименование темы	Содержание темы	Вид учебной работы*	Формируемые индикаторы
			обнаружение отрицательного цикла. Применение: поиск кратчайшего маршрута доставки (логистика), построение оптимального расписания		
		3.6 Практикум: жадные алгоритмы	Реализация жадных алгоритмов на Python. Задачи: задача о рюкзаке (дробный вариант), планирование интервалов, обмен монет. Код Хаффмана для сжатия данных. Обсуждение: когда жадный подход оптимален, а когда нет. Контрпримеры	СЗ	PL-1.1, ОПК-3.1, УК-1.3
		3.7 Практикум: динамическое программирование	Реализация ДП на Python. Задачи: числа Фибоначчи (мемоизация + табуляция), задача о рюкзаке (0/1), расстояние Левенштейна (edit distance), наибольшая общая подпоследовательность (LCS). Визуализация таблицы ДП. Применение: расстояние Левенштейна для оценки качества NLP (метрика WER/CER), beam search в генерации текста	СЗ	PL-1.1, ОПК-3.1, УК-1.3
		3.8 Итоговое занятие: выбор алгоритма и структуры данных для проекта ИИ	Комплексная кейс-задача: проектирование архитектуры подсистемы поиска похожих товаров для e-commerce (кейс X5 Group). Требования: быстрый поиск ближайших соседей, ранжирование результатов, кэширование. Выбор и обоснование алгоритмов и структур данных. Оформление как фрагмент архитектурного описания ИС. Обсуждение компромиссов: точность vs скорость, память vs время	СЗ	ПК-2.1, УК-1.3, ОПК-3.3, ОПК-3.1

* - заполняется только по **ОЧНОЙ** форме обучения: ЛК – лекции; ЛР – лабораторные работы; СЗ – практические/семинарские занятия.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Таблица 6.1. Материально-техническое обеспечение дисциплины

Тип аудитории	Оснащение аудитории	Специализированное учебное/лабораторное оборудование, ПО и материалы для освоения дисциплины (при необходимости)
Лекционная	Аудитория для проведения занятий лекционного типа, оснащенная комплектом специализированной мебели; доской (экраном) и техническими средствами мультимедиа презентаций.	
Семинарская	Аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная комплектом специализированной мебели и техническими средствами мультимедиа презентаций.	Персональные компьютеры, необходимое ПО
Для самостоятельной работы	Аудитория для самостоятельной работы обучающихся (может использоваться для проведения семинарских занятий и консультаций), оснащенная комплектом специализированной мебели и компьютерами с доступом в ЭИОС.	Персональные компьютеры, необходимое ПО

* - аудитория для самостоятельной работы обучающихся указывается **ОБЯЗАТЕЛЬНО!**

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература:

1. Григорьев, А. А. Методы и алгоритмы обработки данных: учебное пособие / А.А. Григорьев, Е.А. Исаев. — 2-е изд., перераб. и доп. — Москва: ИНФРА-М, 2024. — 383 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/1032305. - ISBN 978-5-16-015581-4. - Текст: электронный. - URL: <https://znanium.ru/catalog/product/2084190>

2. Белов, В. В. Алгоритмы и структуры данных: учебник / В.В. Белов, В.И. Чистякова. — Москва: КУРС: ИНФРА-М, 2023. — 240 с. — (Бакалавриат). - ISBN 978-5-906818-25-6. - Текст: электронный. - URL: <https://znanium.ru/catalog/product/2110058>

Дополнительная литература:

1. Шкодина, Т. А. Алгоритмы и структуры данных в Python: лабораторный практикум: учебное пособие: [16+] / Т. А. Шкодина; Ростовский государственный экономический университет (РИНХ), Кафедра информационных систем и прикладной информатики. – Ростов-на-Дону: Издательско-полиграфический комплекс РГЭУ (РИНХ), 2022. – 80 с.: ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=704757>

2. Павлов, Л. А. Структуры и алгоритмы обработки данных: учебник для вузов / Л. А. Павлов, Н. В. Первова. — 3-е изд., стер. — Санкт-Петербург: Лань, 2021. — 256 с. —

ISBN 978-5-8114-7259-8. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/156929>

Ресурсы информационно-телекоммуникационной сети «Интернет»:

1. ЭБС РУДН и сторонние ЭБС, к которым студенты университета имеют доступ на основании заключенных договоров

- Электронно-библиотечная система РУДН – ЭБС РУДН
<https://mega.rudn.ru/MegaPro/Web>

- ЭБС «Университетская библиотека онлайн» <http://www.biblioclub.ru>

- ЭБС «Юрайт» <http://www.biblio-online.ru>

- ЭБС «Консультант студента» www.studentlibrary.ru

- ЭБС «Знаниум» <https://znanium.ru/>

2. Базы данных и поисковые системы

- Sage <https://journals.sagepub.com/>

- Springer Nature Link <https://link.springer.com/>

- Wiley Journal Database <https://onlinelibrary.wiley.com/>

- Научометрическая база данных Lens.org <https://www.lens.org>

Учебно-методические материалы для самостоятельной работы обучающихся при освоении дисциплины/модуля:*

1. Курс лекций по дисциплине «Алгоритмы и структуры данных».

* - все учебно-методические материалы для самостоятельной работы обучающихся размещаются в соответствии с действующим порядком на странице дисциплины **в ТУИС!**