


МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
Программирования и информационных технологий


_____ проф. Махортов С.Д.,
10.03.2021

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.О.29 Алгоритмы и структуры данных

1. Код и наименование направления подготовки/специальности:

10.03.01 Информационная безопасность

2. Профиль подготовки/специализация:

Безопасность компьютерных систем

3. Квалификация (степень) выпускника: Бакалавр

4. Форма обучения: Очная

5. Кафедра, отвечающая за реализацию дисциплины:

Программирования и информационных технологий

6. Составители программы:

ст. преподаватель каф. ПиИТ Соломатин Дмитрий Иванович

e-mail: solomatin@cs.vsu.ru

факультет: Компьютерных наук

кафедра: Программирования и информационных технологий

7. Рекомендована:

НМС ф-та компьютерных наук, протокол № 5 от 10.03.2021

8. Учебный год: 2021-2022

Семестр(ы): 2

9. Цели и задачи учебной дисциплины:

Изучение студентами классических структур данных (связные списки, различные виды деревьев, хеш-таблицы, графы) и алгоритмов, которые лежат в их основе или используют данные структуры, развитие базовых навыков проектирования и анализа алгоритмов, а также применения изученных алгоритмов и структур данных в решении практических задач.

10. Место учебной дисциплины в структуре ООП:

Учебная дисциплина относится к обязательной части блока Б1.

Для успешного освоения дисциплины необходимы знание одного из современных языков программирования и практические навыки составления программ.

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников):

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ОПК-3	Способен использовать необходимые математические методы для решения задач профессиональной деятельности	ОПК-3.28	знает основные понятия и методы математической логики и теории алгоритмов	Знать: - способы построения и применения логических выражений в реализации условных операторов и циклов; - технологии построения алгоритмов для решения практических задач.
		ОПК-3.30	знает основные методы решения задач профессиональной области с применением дискретных моделей	Знать: - комбинаторные алгоритмы для решения задач в области программирования.
		ОПК-3.31	умеет строить математические модели задач профессиональной области	Уметь: - строить математические модели для алгоритмов задач в области программирования.
		ОПК-3.33	владеет навыками самостоятельного решения комбинаторных задач	Владеть: - навыками разработки программ с использованием комбинаторных алгоритмов.
ОПК-7	Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности	ОПК-7.5	умеет разрабатывать и реализовывать на языке высокого уровня алгоритмы решения типовых профессиональных задач	Уметь: - разрабатывать и реализовывать на языке Java алгоритмы решения задач поиска, сортировки, работы со стеками и очередью, деревьями и графами.
		ОПК-7.6	владеет навыками разработки, документирования, тестирования и отладки программ	Иметь навыки: - документирования программного кода в виде комментариев; - тестирование и отладки программ.
		ОПК-7.7	знает базовые структуры данных	Знать: - базовые структуры данных языка Java; - способы представления данных в виде структур объектов и интерфейсов; - представление списков,

				деревьев, графов.
		ОПК-7.8	знает основные алгоритмы сортировки и поиска данных	Знать: - основные алгоритмы поиска и сортировки данных
		ОПК-7.9	знает основные комбинаторные и теоретико-графовые алгоритмы	Знать: - алгоритмы решений комбинаторных задач; - алгоритмы построения и поиска данных на деревьях и графах.
		ОПК-7.10	знает общие сведения о методах проектирования, документирования, разработки, тестирования и отладки программного обеспечения	Знать: - способы документирования программ в java с использованием комментариев и мета-данных; - технологии тестирования и отладки программ в средах разработки программ на языке java.
		ОПК-7.11	умеет применять известные методы программирования и возможности базового языка программирования для решения типовых профессиональных задач	Уметь: - применять стандартные классы языка java для разработки программ.
		ОПК-7.12	владеет навыками разработки алгоритмов решения типовых профессиональных задач	Иметь навыки: - разработки программ на языке java.
ОПК-1.3	Способен обеспечивать защиту информации при работе с базами данных, при передаче по компьютерным сетям	ОПК-1.3.4	знает правила математической логики при составлении запросов к реляционным моделям	Знать: - правила математической логики, для составления логических выражений в алгоритмах программ.
		ОПК-1.3.5	умеет оценивать сложность алгоритмов	Уметь: - оценивать вычислительную сложность алгоритмов.

12. Объем дисциплины в зачетных единицах/час. (в соответствии с уч. планом) – 4 / 144.

Форма промежуточной аттестации – Экзамен

13. Виды учебной работы

Вид учебной работы		Трудоемкость			
		Всего	По семестрам		
			2 сем.	–	–
Аудиторные занятия		68	68	–	–
в том числе:	лекции	34	34	–	–
	практические			–	–
	лабораторные	34	34	–	–
Самостоятельная работа		40	40	–	–
Форма промежуточной аттестации (зачет – 0 час. / экзамен – 36 час.)		36	36	–	–
Итого:		144	144	–	–

13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1. Лекции			
1.1	Введение в предмет	Цели и задачи изучения дисциплины, связь алгоритмов и структур данных	
1.2	Основы объектно-ориентированного программирования (инкапсуляция)	Принципы объектно ориентированного подхода; понятие класса и его экземпляров; различия между static-функциями и методами классов; классы как расширение концепции типа данных в виде объединения данных и методов их обработки, понятие состояния объекта, примеры; принципы инкапсуляции; понятие наследования и полиморфизма; класс Object и его методы	
1.3	Понятие сложности алгоритма	Понятие алгоритма, свойства алгоритма, задача оценки и сравнения эффективности алгоритмов; временная и пространственная сложность, асимптотические оценки сложности алгоритмов для различных случаев и их свойства; практические принципы оценки сложности алгоритмов в O-нотации, примеры алгоритмов с различной сложностью	
1.4	Алгоритмы сортировки: свойства, простые алгоритмы $O(n^2)$	Свойства алгоритмов сортировки: время (временная сложность), память (пространственная сложность), устойчивость и естественность поведения, внутренние и внешние сортировки; простые сортировки и их свойства: пузырьковая сортировка, сортировка выбором, сортировка вставками; реализация указанных сортировок в обобщенном (generic) виде с параметром в виде функции сравнения элементов; вызов встроенной функции сортировки	
1.5	«Быстрые» алгоритмы сортировки	Классические «быстрые» ($O(n \cdot \log(n))$) сортировки со сравнением элементов: сортировка слиянием (merge sort), быстрая сортировка (quick sort), пирамидальная сортировка (heap sort); сортировки «без сравнений»: сортировка подсчетом (counting sort), блочная сортировка (bucket sort)	
1.6	Классификация структур данных, обзор коллекций языка Java	Классификация структур данных по принципу обращения к элементам: записи (классы), массивы, списки, словари, а также стеки и очереди; обзор коллекций Java как реализаций указанных структур данных: List<T> (ArrayList, LinkedList), Map<K, V> (TreeMap, HashMap), Stack<T>, Queue<T>, практические примеры их применения	
1.7	Односвязные и двусвязные списки, стеки, очереди	Архитектура односвязных и двусвязных списков, основные операции; реализация указанных структур данных в виде классов с использованием инкапсуляции; реализация стека и очереди на основе односвязного списка	
1.8	Деревья, двоичные деревья поиска	Организация данных в виде дерева, терминология, обход и способы отображения деревьев, двоичные деревья; двоичные деревья поиска: идея и основные операции, реализация в виде классов, понятие сбалансированности дерева; реализация множества (Set<T>) и словаря (Map<T>) с помощью двоичного дерева поиска	
1.9	AVL-деревья	AVL-деревья: идея, операции вставки и удаления, балансировка (R-/L-, RL-/LR-повороты), оценка сложности операций	

1.10	2-3-, 2-3-4- и красно-черные деревья	2-3- и 2-3-4- деревья: идея и основные операции; красно-черные деревья как вариант представления 2-3-4-деревьев, основные операции для красно-черных деревьев и оценка сложности	
1.11	Хеширование (hash-таблицы)	Hash-таблицы как вариант структуры с потенциальным временем доступа по ключу $O(1)$, понятие hash-функции и ее качество, стратегии разрешения коллизий – открытая и закрытая адресации, реализация множества (Set<T>) и словаря (Map<T>) с помощью hash-таблиц, сравнение с деревьями поиска; комбинации hash-таблиц с деревьями поиска против «плохих» hash-функций	
1.12	Основы теории графов	Графы: основные понятия и определения; способы задания (хранения) графов (матрица смежности, списки смежности) и универсальный интерфейс представления графов для алгоритмов	
1.13	Обход графов, поиск в глубину, поиск в ширину	Реализация обхода графов (поиска) в глубину и ширину; отображение графов с помощью библиотеки Graphviz	
1.14	Оптимизационные алгоритмы на графах (кратчайшие пути, минимальное остовное дерево)	Поиск кратчайших путей в графе: алгоритм Дейкстры, алгоритм Беллмана-Форда, алгоритм Флойда-Уоршелла; поиск минимального остовного дерева: алгоритм Прима, алгоритм Краскала	
1.15	Примеры практических задач на графах	Эйлеров граф, Гамильтонов граф; примеры практических задач, которые сводятся к задачам на графах	
2. Практические занятия			
2.1	Основы объектно-ориентированного программирования (инкапсуляция)	Принципы объектно ориентированного подхода; понятие класса и его экземпляров; различия между static-функциями и методами классов; классы как расширение концепции типа данных в виде объединения данных и методов их обработки, понятие состояния объекта, примеры; принципы инкапсуляции; понятие наследования и полиморфизма; класс Object и его методы	
2.2	Понятие сложности алгоритма	Понятие алгоритма, свойства алгоритма, задача оценки и сравнения эффективности алгоритмов; временная и пространственная сложность, асимптотические оценки сложности алгоритмов для различных случаев и их свойства; практические принципы оценки сложности алгоритмов в O-нотации, примеры алгоритмов с различной сложностью	
2.3	Алгоритмы сортировки: свойства, простые алгоритмы $O(n^2)$	Свойства алгоритмов сортировки: время (временная сложность), память (пространственная сложность), устойчивость и естественность поведения, внутренние и внешние сортировки; простые сортировки и их свойства: пузырьковая сортировка, сортировка выбором, сортировка вставками; реализация указанных сортировок в обобщенном (generic) виде с параметром в виде функции сравнения элементов; вызов встроенной функции сортировки	
2.4	«Быстрые» алгоритмы сортировки	Классические «быстрые» ($O(n \cdot \log(n))$) сортировки со сравнением элементов: сортировка слиянием (merge sort), быстрая сортировка (quick sort), пирамидальная сортировка (heap sort); сортировки «без сравнений»: сортировка подсчетом (counting sort), блочная сортировка (bucket sort)	

2.5	Классификация структур данных, обзор коллекций языка Java	Классификация структур данных по принципу обращения к элементам: записи (классы), массивы, списки, словари, а также стеки и очереди; обзор коллекций Java как реализаций указанных структур данных: List<T> (ArrayList, LinkedList), Map<K, V> (TreeMap, HashMap), Stack<T>, Queue<T>, практические примеры их применения	
2.6	Односвязные и двусвязные списки, стеки, очереди	Архитектура односвязных и двусвязных списков, основные операции; реализация указанных структур данных в виде классов с использованием инкапсуляции; реализация стека и очереди на основе односвязного списка	
2.7	Деревья, двоичные деревья поиска	Организация данных в виде дерева, терминология, обход и способы отображения деревьев, двоичные деревья; двоичные деревья поиска: идея и основные операции, реализация в виде классов, понятие сбалансированности дерева; реализация множества (Set<T>) и словаря (Map<T>) с помощью двоичного дерева поиска	
2.8	AVL-деревья	AVL-деревья: идея, операции вставки и удаления, балансировка (R-/L-, RL-/LR-повороты), оценка сложности операций	
2.9	2-3-, 2-3-4- и красно-черные деревья	2-3- и 2-3-4- деревья: идея и основные операции; красно-черные деревья как вариант представления 2-3-4-деревьев, основные операции для красно-черных деревьев и оценка сложности	
2.10	Хеширование (hash-таблицы)	Hash-таблицы как вариант структуры с потенциальным временем доступа по ключу O(1), понятие hash-функции и ее качество, стратегии разрешения коллизий – открытая и закрытая адресации, реализация множества (Set<T>) и словаря (Map<T>) с помощью hash-таблиц, сравнение с деревьями поиска; комбинации hash-таблиц с деревьями поиска против «плохих» hash-функций	
2.11	Основы теории графов	Графы: основные понятия и определения; способы задания (хранения) графов (матрица смежности, списки смежности) и универсальный интерфейс представления графов для алгоритмов	
2.12	Обход графов, поиск в глубину, поиск в ширину	Реализация обхода графов (поиска) в глубину и ширину; отображение графов с помощью библиотеки Graphviz	
2.13	Оптимизационные алгоритмы на графах (кратчайшие пути, минимальное остовное дерево)	Поиск кратчайших путей в графе: алгоритм Дейкстры, алгоритм Беллмана-Форда, алгоритм Флойда-Уоршелла; поиск минимального остовного дерева: алгоритм Прима, алгоритм Краскала	
2.14	Примеры практических задач на графах	Эйлеров граф, Гамильтонов граф; примеры практических задач, которые сводятся к задачам на графах	
3. Лабораторные работы			
3.1	Основы объектно-ориентированного программирования (инкапсуляция)	Принципы объектно ориентированного подхода; понятие класса и его экземпляров; различия между static-функциями и методами классов; классы как расширение концепции типа данных в виде объединения данных и методов их обработки, понятие состояния объекта, примеры; принципы инкапсуляции; понятие наследования и полиморфизма; класс Object и его методы	

3.2	Понятие сложности алгоритма	Понятие алгоритма, свойства алгоритма, задача оценки и сравнения эффективности алгоритмов; временная и пространственная сложность, асимптотические оценки сложности алгоритмов для различных случаев и их свойства; практические принципы оценки сложности алгоритмов в O-нотации, примеры алгоритмов с различной сложностью	
3.3	Алгоритмы сортировки: свойства, простые алгоритмы $O(n^2)$	Свойства алгоритмов сортировки: время (временная сложность), память (пространственная сложность), устойчивость и естественность поведения, внутренние и внешние сортировки; простые сортировки и их свойства: пузырьковая сортировка, сортировка выбором, сортировка вставками; реализация указанных сортировок в обобщенном (generic) виде с параметром в виде функции сравнения элементов; вызов встроенной функции сортировки	
3.4	«Быстрые» алгоритмы сортировки	Классические «быстрые» ($O(n \cdot \log(n))$) сортировки со сравнением элементов: сортировка слиянием (merge sort), быстрая сортировка (quick sort), пирамидальная сортировка (heap sort); сортировки «без сравнений»: сортировка подсчетом (counting sort), блочная сортировка (bucket sort)	
3.5	Классификация структур данных, обзор коллекций языка Java	Классификация структур данных по принципу обращения к элементам: записи (классы), массивы, списки, словари, а также стеки и очереди; обзор коллекций Java как реализаций указанных структур данных: List<T> (ArrayList, LinkedList), Map<K, V> (TreeMap, HashMap), Stack<T>, Queue<T>, практические примеры их применения	
3.6	Односвязные и двусвязные списки, стеки, очереди	Архитектура односвязных и двусвязных списков, основные операции; реализация указанных структур данных в виде классов с использованием инкапсуляции; реализация стека и очереди на основе односвязного списка	
3.7	Деревья, двоичные деревья поиска	Организация данных в виде дерева, терминология, обход и способы отображения деревьев, двоичные деревья; двоичные деревья поиска: идея и основные операции, реализация в виде классов, понятие сбалансированности дерева; реализация множества (Set<T>) и словаря (Map<T>) с помощью двоичного дерева поиска	
3.8	AVL-деревья	AVL-деревья: идея, операции вставки и удаления, балансировка (R-/L-, RL-/LR-повороты), оценка сложности операций	
3.9	2-3-, 2-3-4- и красно-черные деревья	2-3- и 2-3-4- деревья: идея и основные операции; красно-черные деревья как вариант представления 2-3-4-деревьев, основные операции для красно-черных деревьев и оценка сложности	
3.10	Хеширование (hash-таблицы)	Hash-таблицы как вариант структуры с потенциальным временем доступа по ключу $O(1)$, понятие hash-функции и ее качество, стратегии разрешения коллизий – открытая и закрытая адресации, реализация множества (Set<T>) и словаря (Map<T>) с помощью hash-таблиц, сравнение с деревьями поиска; комбинации hash-таблиц с деревьями поиска против «плохих» hash-функций	
3.11	Основы теории графов	Графы: основные понятия и определения; способы задания (хранения) графов (матрица смежности, списки смежности) и универсальный интерфейс представления графов для алгоритмов	
3.12	Обход графов, поиск в глубину, поиск в ширину	Реализация обхода графов (поиска) в глубину и ширину; отображение графов с помощью библиотеки Graphviz	

3.13	Оптимизационные алгоритмы на графах (кратчайшие пути, минимальное остовное дерево)	Поиск кратчайших путей в графе: алгоритм Дейкстры, алгоритм Беллмана-Форда, алгоритм Флойда-Уоршелла; поиск минимального остовного дерева: алгоритм Прима, алгоритм Краскала	
3.14	Примеры практических задач на графах	Эйлеров граф, Гамильтонов граф; примеры практических задач, которые сводятся к задачам на графах	

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Введение в предмет	2	–	–	1	3
2	Основы объектно-ориентированного программирования (инкапсуляция)	2	–	3	4	9
3	Понятие сложности алгоритма	2	–	2	3	7
4	Алгоритмы сортировки: свойства, простые алгоритмы $O(n^2)$	2	–	2	1	5
5	«Быстрые» алгоритмы сортировки	3	–	2	3	8
6	Классификация структур данных, обзор коллекций языка Java	2	–	2	3	7
7	Односвязные и двусвязные списки, стеки, очереди	2	–	2	3	7
8	Деревья, двоичные деревья поиска	2	–	3	3	8
9	AVL-деревья	2	–	3	3	8
10	2-3-, 2-3-4- и красно-черные деревья	3	–	3	3	9
11	Хеширование (hash-таблицы)	2	–	2	2	6
12	Основы теории графов	2	–	2	1	5
13	Обход графов, поиск в глубину, поиск в ширину	2	–	2	3	7
14	Оптимизационные алгоритмы на графах (кратчайшие пути, минимальное остовное дерево)	3	–	3	4	10
15	Примеры практических задач на графах	3	–	3	3	9
	Итого:	34	–	34	40	108

14. Методические указания для обучающихся по освоению дисциплины

Рекомендуется работа с конспектами лекций, презентационным материалом, выполнение всех лабораторных и контрольных работ, заданий текущей аттестации. Учебные и методические материалы по дисциплине размещены на сетевом диске, доступным на любом компьютере в локальной сети ФКН.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1	Седжвик, Роберт. Фундаментальные алгоритмы на Java : Пер. с англ. / Р. Седжвик .— М. : DiaSoft, 2003. — Ч.1-4: Анализ. Структуры данных. Сортировка. Поиск .— 2003 .— 680 с. : ил. — Библиогр. в конце глав. - Предм. указ.: с. 680-687 .— Парал. тит. л. англ. — ISBN 966-7992-22-5.

2	Лафоре, Роберт. Структуры данных и алгоритмы в Java = Data structures @ algorithms in Java / Роберт Лафоре ; [пер. с англ. Е. Матвеева] .— 2-е изд. — Санкт-Петербург [и др.] : Питер, 2014 .— 701 с. : ил., табл. — (Классика computer science) .— Библиогр.: с.683-685 .— Алф. указ.: с.695-701 .— ISBN 985-5-496-00740-5.
3	Кормен, Томас. Алгоритмы : Построение и анализ : [Учебник] / Т. Кормен, Ч. Лейзерсон, Р. Ривест ; Пер. с англ. К. Белов и др.; Науч. ред. А. Шень .— М. : МЦНМО, 2002 .— 955 с. : ил. — (Классические учебники: computer science) .— ISBN 5-900916-37-5.
4	Ахо, Альфред В. Структуры данных и алгоритмы : [Учебное пособие] / Альфред В. Ахо, Джон Э. Хопкрофт, Джеффри Д. Ульман; Пер. с англ. и ред. А.А. Минько .— М. и др. : Вильямс, 2003 .— 382 с. : ил., табл. — Парал. тит. л. англ. — Библиогр.: с.369-374 .— Предм. указ.: с. 375-382 .— ISBN 5-8459-0122-7.

б) дополнительная литература:

№ п/п	Источник
5	Седжвик, Роберт. Фундаментальные алгоритмы на C++ / Р. Седжвик .— М. и др. : DiaSoft, 2002. — Ч.1-4: Анализ. Структуры данных. Сортировка. Поиск .— 2002 .— 687 с. : ил. — ISBN 5-93772-047-4.
6	Седжвик, Роберт. Фундаментальные алгоритмы на C++ / Р. Седжвик .— М. и др. : DiaSoft, 2002. — Ч.5: Алгоритмы на графах .— 2002 .— 484 с. : ил. — Парал. тит. л. англ. — ISBN 5-93772-054-7 .— ISBN 0-201-36118-3.
7	Седжвик, Роберт. Фундаментальные алгоритмы на C : Пер. с англ. / Р. Седжвик ; Принстонский ун-т .— 3-е изд. — СПб. и др. : DiaSoft, 2003. — Ч.1-4: Анализ. Структуры данных. Сортировка. Поиск .— 2003 .— 670 с. : ил. — Парал. тит. л. англ. — ISBN 5-93772-081-4.
8	Седжвик, Роберт. Фундаментальные алгоритмы на C : Пер. с англ. / Р. Седжвик ; Принстонский ун-т .— 3-е изд. — СПб., и др. : DiaSoft, 2003. — Ч.5: Алгоритмы на графах .— 2003 .— С.661-1127 : ил. — Парал. тит. л. англ. — ISBN 5-93772-082-2.

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет):

№ п/п	Источник
9	MAXimal::algo [Электронный ресурс] : — Режим доступа: http://e-maxx.ru/algo/
10	Алгоритмы: теория и практика. Методы [Электронный ресурс] : — Режим доступа: https://stepik.org/course/217/promo
11	Алгоритмы программирования и структуры данных [Электронный ресурс] : — Режим доступа: https://openedu.ru/course/ITMOUniversity/PADS/

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	Шилдт, Герберт. Искусство программирования на JAVA : пер. с англ. / Герберт Шилдт, Джеймс Холмс .— СПб. [и др.] : БХВ-Петербург, 2005 .— 331 с. : ил. — Парал. тит. л. англ. — Предм. указ. : с.330-331, 4000 экз.

17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости)

№ п/п	Наименование
1	OpenJDK - бесплатен
2	Среда разработки NetBeans или IntelliJ IDEA (академическая лицензия или версия Community) - бесплатны

18. Материально-техническое обеспечение дисциплины:

№ п/п	Наименование
1	Мультимедийная лекционная аудитория (корп. 1а, ауд. № 479 или другая подходящая): рабочее место преподавателя: ПК-Intel-i3, проектор, видеоконмутатор, микрофон, аудиосистема, специализированная мебель: доски меловые 2 шт., столы и стулья/лавки в количестве, достаточном для размещения потока студентов; выход в Интернет, доступ к фондам учебно-методической документации и электронным изданиям.
2	Компьютерный класс (корп. 1а, ауд. № 382-385 или другие подходящие): ПК-Intel-i3 16 шт., специализированная мебель: доска маркерная 1 шт., столы и стулья в количестве, достаточном для размещения академической группы (подгруппы) студентов; выход в Интернет, доступ к фондам учебно-методической документации и электронным изданиям.

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1	Введение в предмет	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
2	Основы объектно-ориентированного программирования (инкапсуляция)	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
3	Понятие сложности алгоритма	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
4	Алгоритмы сортировки: свойства, простые алгоритмы $O(n^2)$	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
5	«Быстрые» алгоритмы сортировки	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
6	Классификация структур данных, обзор коллекций языка Java	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)

7	Односвязные и двусвязные списки, стеки, очереди	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
8	Деревья, двоичные деревья поиска	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
9	AVL-деревья	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
10	2-3-, 2-3-4- и красно-черные деревья	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
11	Хеширование (hash-таблицы)	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
12	Основы теории графов	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
13	Обход графов, поиск в глубину, поиск в ширину	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
14	Оптимизационные алгоритмы на графах (кратчайшие пути, минимальное остовное дерево)	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
15	Примеры практических задач на графах	ОПК-3 ОПК-7 ОПК-1.3	ОПК-3.28, ОПК-3.30, ОПК-3.31, ОПК-3.33 ОПК-7.5, ОПК-7.6, ОПК-7.7, ОПК-7.8, ОПК-7.9, ОПК-7.10, ОПК-7.11, ОПК-7.12 ОПК-1.3.4, ОПК-1.3.5	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
Промежуточная аттестация форма контроля – экзамен				Перечень вопросов к экзамену из пункта 20.2

20. Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью контроля выполнения обязательных практических заданий. Перечень заданий:

№ п/п	Задание
1	Задача 1 – Инкапсуляция (≥ 30 индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)
2	Задача 2 – Связные списки (≥ 30 индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)
3	Задача 3 – Стеки и очереди (≥ 30 индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)
4	Задача 4 – Сортировки (≥ 30 индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)
5	Задача 5 – Деревья (≥ 30 индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)
6	Задача 6 – Словари (на основе деревьев или hash-функций, ≥ 30 индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)
7	Задача 7 – Графы 1 (≥ 30 индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)
8	Задача 8 – Графы 2 (интерактивное редактирование графа, ≥ 30 индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)

20.2 Промежуточная аттестация

Для оценивания результатов обучения на зачете используются следующие содержательные показатели (формулируется с учетом конкретных требований дисциплины):

1) знание теоретических основ учебного материала, основных определений, понятий и используемой терминологии;

2) умение проводить обоснование и представление основных теоретических и практических результатов (теорем, алгоритмов, методик) с использованием математических выкладок, блок-схем, структурных схем и стандартных описаний к ним;

3) умение связывать теорию с практикой, иллюстрировать ответ примерами, в том числе, собственными, умение выявлять и анализировать основные закономерности, полученные, в том числе, в ходе выполнения лабораторно-практических заданий;

4) умение обосновывать свои суждения и профессиональную позицию по излагаемому вопросу;

5) владение навыками программирования и экспериментирования в рамках выполняемых лабораторных заданий;

Различные комбинации перечисленных показателей определяют критерии оценивания результатов обучения (сформированности компетенций) на зачете:

- высокий (углубленный) уровень сформированности компетенций;
- повышенный (продвинутый) уровень сформированности компетенций;
- пороговый (базовый) уровень сформированности компетенций.

Для оценивания результатов обучения на зачете с оценкой используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Соотношение показателей, критериев и шкалы оценивания результатов обучения на экзамене представлено в следующей таблице.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Студент владеет основными понятиями учебной дисциплины, может пояснить большинство принципов на примерах; вовремя сдал все практические задания, которые выполнены на высоком уровне, без явных ошибок.	Повышенный уровень	Отлично

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Студент владеет основными понятиями учебной дисциплины, однако в ответах на некоторые вопросы допускает неточности; сдал все практические задания, однако к некоторым решениям студента у преподавателя есть замечания.	Базовый уровень	Хорошо
Студент знает основные определения из учебной дисциплины, однако пояснить многие понятия на примерах затрудняется; сдал большую часть практических заданий, однако продемонстрированные решения содержат существенные ошибки.	Пороговый уровень	Удовлетворительно
Студент путается в основных понятиях учебной дисциплины, не может привести примеры; не сдал большую часть практических заданий.	–	Неудовлетворительно

Перечень вопросов к экзамену (зачету):

№ п/п	Вопрос
1	Понятие сложности алгоритма
2	Практические приемы оценки сложности алгоритмов
3	Сортировка слиянием (merge sort): идея, реализация, оценка сложности
4	Быстрая сортировка (quick sort): идея, реализация, оценка сложности
5	Пирамидальная сортировка (heap sort): идея, реализация, оценка сложности
6	Сортировки «без сравнений»: сортировка подсчетом (counting sort), блочная сортировка (bucket sort)
7	Классификация структур данных, обзор классов-коллекций стандартной библиотеки языка Java
8	Принципы реализации структур данных в виде классов с применением инкапсуляции
9	Односвязные списки
10	Двусвязные списки
11	Реализация стека и очереди в виде связного списка
12	Очередь с приоритетом: идея, возможные варианты эффективной реализации
13	Деревья: терминология, обход и способы отображения деревьев, двоичные деревья
14	Двоичные деревья поиска: основные операции, понятие сбалансированности дерева
15	Реализация множества (Set) и словаря (Map) на основе двоичного дерева поиска
16	AVL-деревья: идея, операции вставки и удаления, балансировка (R-/L-, RL-/LR-повороты), оценка сложности операций
17	2-3- и 2-3-4- деревья: идея и основные операции, оценка сложности
18	Красно-черные деревья: идея и основные операции, оценка сложности
19	Hash-таблицы: понятие hash-функции, коллизии и стратегии их разрешения
20	Реализация множества (Set) и словаря (Map) на основе hash-таблицы
21	Графы: основные понятия и определения, способы задания (хранения) графов
22	Реализация обхода графов (поиска) в глубину и ширину
23	Отображение графов с помощью библиотеки Graphviz
24	Алгоритм Дейкстры
25	Алгоритм Беллмана-Форда
26	Алгоритм Флойда-Уоршелла
27	Алгоритм Прима
28	Алгоритм Краскала
29	Примеры практических задач, которые сводятся к задачам на графах