

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Алексеевич

Должность: И.о. директора института перспективной инженерии

Дата подписания: 19.06.2026 16:38:42

Уникальный программный ключ:

990bea3aeec48c23bd8699e48288f8d1960c600

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И.о. директора института

А.А. Порохня

Ф.И.О.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Алгоритмы и структуры данных

Направление подготовки
Направленность (профиль)

09.03.04 Программная инженерия
Инженерия сложных программных
систем

Год начала обучения

2026

Форма обучения

очная

Реализуется в семестре

2

Разработано

Заведующая межинститутской базовой
кафедрой департамента цифровых,
робототехнических систем и электроники
(должность разработчика)

Новикова Елена Николаевна

Ф.И.О.

Ставрополь 2026 г.

1. Цель и задачи освоения дисциплины (модуля)

Цель дисциплины: формирование у студентов системы теоретических знаний и практических навыков в области алгоритмов и структур данных, необходимых для разработки эффективного программного обеспечения на языке C++.

Задачи освоения дисциплины:

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Алгоритмы и структуры данных» относится к обязательной части.

1. Изучить основы языка C++ и принципы ООП
2. Реализовать основные структуры данных и алгоритмы
3. Научиться анализировать сложность алгоритмов
4. Разработать итоговый проект с использованием изученных технологий
5. Сформировать портфолио в Git-репозитории

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ОПК-1. Способен применять естественнонаучные и инженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ИД-1 ОПК- 1 Демонстрирует знание фундаментальных математических понятий и методов, необходимых для анализа и моделирования профессиональных задач	Применяет фундаментальные математические понятия (функции, множества, последовательности, рекуррентные соотношения) для формального описания и анализа алгоритмов
	ИД-2 ОПК- 1 Применяет методы математического анализа и моделирования для решения профессиональных задач	Анализирует временную и пространственную сложность алгоритмов, используя O-нотацию, и определяет классы сложности ($O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(2^n)$).
	ИД-3 ОПК- 1 Использует теоретические и экспериментальные методы исследования для оценки эффективности решений	Проводит экспериментальные исследования алгоритмов, сравнивает эффективность различных решений и строит графики зависимости времени выполнения от объёма входных данных.
ОПК-6 Способен разрабатывать алгоритмы и	ИД-1 ОПК- 6 Разрабатывает алгоритмы решения профессиональных	Разрабатывает корректные и эффективные алгоритмы решения типовых задач (поиск, сортировка, обход графов и деревьев) с учётом ограничений по времени и памяти.

программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	задач с учётом ограничений по времени и памяти	
	ИД-2 ОПК- 6 Реализует алгоритмы на языках программирования с соблюдением принципов структурного и объектно-ориентированного программирования	Программирует алгоритмы на языке C++, соблюдая принципы структурного и объектно-ориентированного программирования, и обеспечивает читаемость и сопровождаемость кода.
	ИД-3 ОПК- 6 Применяет методы тестирования для проверки корректности работы программных продуктов	Тестирует программный код, применяя методы модульного тестирования и тестирования граничных случаев для проверки корректности работы программ
	ИД-4 ОПК- 6 Создаёт программы, пригодные для практического использования, с понятным интерфейсом и обработкой ошибок	Создаёт программы с понятным пользовательским интерфейсом (консольное меню), обеспечивает обработку ошибок и управляет ресурсами (память, файлы) с использованием исключений и принципов RAII.
ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	ИД-1 ОПК-7 Демонстрирует понимание фундаментальных концепций, принципов и теорий информатики	Применяет фундаментальные концепции информатики (абстракция, инкапсуляция, иерархия типов данных, рекурсия, итерация) при проектировании и реализации программ.
	ИД-2 ОПК-7 Применяет теоретические знания о структурах данных и алгоритмах при разработке программного обеспечения	Выбирает и реализует подходящие структуры данных (массивы, списки, деревья, графы, хеш-таблицы) и алгоритмы на основе теоретических знаний о их свойствах и эффективности
	ИД-3 ОПК-7 Использует принципы организации вычислительных процессов для решения практических задач	Учитывает принципы организации вычислительных процессов и особенности аппаратного обеспечения при выборе алгоритмов и структур данных для решения практических задач.
	ИД-4 ОПК-7 Обосновывает выбор	Обосновывает выбор конкретных структур данных, алгоритмов и технологических

	конкретных технологических решений, опираясь на теоретические положения информатики	решений, опираясь на теоретические положения о их свойствах, преимуществах и недостатках.
ПК-2Способен выполнять проектирование и разработку архитектуры программных систем среднего и крупного масштаба с применением современных паттернов и методов проектирования	ИД-1пк-2: Выделяет и анализирует функциональные и нефункциональные требования (FURPS+), выбирает архитектурный стиль (монолит, микросервисы, EDA, Clean Architecture) на основе сценариев качества	Анализирует функциональные требования к программной системе, выделяет основные сущности предметной области, выбирает архитектурный стиль (монолитная архитектура) на основе требований к производительности и масштабируемости при проектировании учебного проекта.
	ИД-2пк-2: Документирует архитектуру с использованием нотаций C4 и UML, фиксирует архитектурные решения в формате ADR (Architecture Decision Record)	Документирует архитектуру программной системы, строит диаграммы классов UML (диаграмма классов, диаграмма компонентов), фиксирует основные архитектурные решения в отчётной документации.
	ИД-3пк-2: Применяет порождающие, структурные и поведенческие паттерны GoF, а также архитектурные паттерны (CQRS, Event Sourcing, Saga, Circuit Breaker)	Применяет порождающие и структурные паттерны проектирования (Фабрика, Одиночка, Компоновщик, Итератор, Стратегия) при разработке иерархий классов и организации взаимодействия между компонентами системы
	ИД-4пк-2: Реализует многослойную архитектуру (Clean Architecture, Hexagonal, PCMEF), обеспечивая независимость от фреймворков через Dependency Inversion	Реализует многослойную архитектуру приложения, разделяя уровни представления (пользовательский интерфейс), бизнес-логики и хранения данных, обеспечивает слабую связанность модулей и независимость слоёв.
ПК-3 Владеет навыками разработки программного обеспечения: способен реализовывать алгоритмы, писать код на современных языках программирования, создавать веб- и	ИД-1пк-3: Разрабатывает алгоритмы и реализует их на языках C++, C#, Python, Java, JavaScript/TypeScript, Go с использованием эффективных структур данных и стандартных библиотек	Разрабатывает алгоритмы обработки данных и реализует их на языке C++ с использованием эффективных структур данных (массивы, списки, деревья, графы, хеш-таблицы) и стандартной библиотеки шаблонов (STL).
	ИД-2пк-3: Создаёт	Разрабатывает desktop-приложения консольного

desktop-приложения, работать с базами данных	клиент-серверные, веб- (frontend + backend) и desktop-приложения с графическим интерфейсом, применяя современные фреймворки (React, Spring, Django, WPF) и протоколы (REST, gRPC, WebSocket)	типа с иерархическим меню, создаёт удобный пользовательский интерфейс для взаимодействия с реализованными алгоритмами и структурами данных.
	ИД-3пк-3: Проектирует, мигрирует и оптимизирует запросы к реляционным и нереляционным БД с использованием SQL и ORM (Hibernate, Entity Framework, Django ORM)	Выполняет сериализацию данных, сохраняет и загружает состояние программы с использованием текстовых и бинарных файлов, реализует импорт и экспорт данных.
	ИД-4пк-3: Пишет модулярный, документированный и тестируемый код, соблюдая стандарты оформления и принципы SOLID, использует системы сборки (Make, Gradle, npm)	Пишет чистый, структурированный и документированный код с соблюдением стандартов оформления, применяет принципы SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) при проектировании классов, ведёт Git-репозиторий.

4. Объем учебной дисциплины и формы контроля *

Объем занятий: всего: 5 з.е. 180 ак.ч.	ОФО, в ак. часах
Контактная работа:	96
Лекции/из них практическая подготовка	48
Лабораторных работ/из них практическая подготовка	48
Практических занятий/из них практическая подготовка	
Самостоятельная работа	57
Формы контроля	
Экзамен	27
Зачет	
Зачет с оценкой	
Расчетно-графические работы	
Курсовые работы	
Контрольные работы	

* Дисциплина (модуль) предусматривает применение электронного обучения, дистанционных образовательных технологий

5. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием количества часов и видов занятий

№	Раздел (тема) дисциплины и краткое содержание	Формируемые компетенции, индикаторы	очная форма			
			Контактная работа обучающихся с преподавателем /из них в форме практической подготовки, часов			Самостоятельная работа, часов
			Лекции	Практические занятия	Лабораторные работы	
1	Тема 1: Поточковый ввод/вывод. Работа с текстовыми файлами Понятие потока. Стандартные потоки (cin, cout, cerr). Буферизация и форматирование вывода. Классы ifstream, ofstream, fstream. Режимы открытия файлов. Чтение и запись текстовых файлов. Методы getline(), get(), put(). Проверка состояния потока (good(), eof(), fail(), bad()).	ПК-1 ^{ид-2} ПК-6 ИД-2 ОПК-6 ИД-4 ПК-3 ИД-3	2		2	2
2	Тема 2: Бинарные файлы и прямое позиционирование Текстовые vs бинарные файлы. Бинарный режим (ios::binary). Методы read() и write(). Сериализация структур. Позиционирование: seekg(), seekp(), tellg(), tellp(). Прямой доступ к записям. Обновление данных на месте. Обработка ошибок при работе с бинарными файлами.	ОПК-1.2, ОПК-6.2, ОПК-6.4, ПК-3.3	2		2	2
3	Тема 3: Основы ООП. Инкапсуляция. Классы и структуры Понятие класса и объекта. Инкапсуляция. Спецификаторы доступа (public, private, protected). Структуры (struct) vs классы (class). Поля и методы класса. Методы-аксессоры (геттеры и сеттеры). Константные методы.	ОПК-6.2, ОПК-7.1, ПК-2.1, ПК-3.4	2		2	2

4	Тема 4: Конструкторы и деструктор. Правило трёх Конструкторы: по умолчанию, параметризованный, копирования. Список инициализации. Деструктор. Правило трёх (деструктор, конструктор копирования, оператор присваивания). Поверхностное vs глубокое копирование. Конструктор перемещения и правило пяти (C++11).	ОПК-6.2, ОПК-6.4, ПК-3.1, ПК-3.4	2		2	2
5	Тема 5: Перегрузка операторов. Дружественные функции Перегрузка арифметических операторов (+, -, *, /). Перегрузка операторов сравнения (==, !=, <, >). Перегрузка оператора индексации ([]). Перегрузка операторов ввода/вывода (<<, >>). Дружественные функции и классы (friend). Правила перегрузки. Префиксный и постфиксный инкремент/декремент.	ОПК-6.2, ПК-2.3, ПК-3.4	2		2	2
6	Тема 6: Композиция и агрегация классов. Статические члены Отношения между классами: ассоциация, агрегация, композиция. Жизненный цикл объектов при разных отношениях. Реализация композиции (поля-объекты, unique_ptr). Реализация агрегации (указатели, ссылки). Статические поля и методы. Подсчёт количества объектов. Паттерн «Одиночка» (Singleton).	ОПК-6.2, ОПК-7.1, ПК-2.1, ПК-2.4, ПК-3.4	2		2	2
7	Тема 7: Анализ сложности алгоритмов. О-нотация Понятие временной и пространственной сложности. О-нотация (Big O). Классы сложности: $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(2^n)$, $O(n!)$. Амортизированный анализ. Правила упрощения сложности (отбрасывание констант и младших членов). Сравнение эффективности алгоритмов. Экспериментальный анализ производительности.	ОПК-1.1, ОПК-1.2, ОПК-1.3, ОПК-7.3	2		2	2

8	Тема 8: Шаблоны классов. Обобщённое программирование. Шаблоны функций. Шаблоны классов. Параметры шаблонов (типы, значения). Синтаксис: <code>template <typename T></code>. Специализация шаблонов (полная и частичная). Классы функциональных объектов. Использование шаблонов для создания контейнеров.	ОПК-6.2, ПК-3.1, ПК-3.4	2		2	2
9	Тема 9: Динамический массив (аналог <code>vector</code>). Проблемы статических массивов. Динамическая память (<code>new[]/delete[]</code>). Управление ёмкостью (<code>capacity</code>) и размером (<code>size</code>). Стратегия удвоения ёмкости. Амортизированный анализ <code>push_back</code> ($O(1)$ амортизированно). Реализация методов: <code>push_back</code>, <code>pop_back</code>, <code>insert</code>, <code>erase</code>, <code>reserve</code>, <code>resize</code>, <code>clear</code>. Оператор доступа по индексу (<code>[]</code>). Конструктор копирования и оператор присваивания (глубокое копирование).	ОПК-1.2, ОПК-6.2, ОПК-6.4, ПК-3.1, ПК-3.4	2		2	2
10	Тема 10: Односвязный список. Двусвязный список. Понятие связного списка. Узел (данные + указатель). Односвязный список: структура узла, вставка в начало/конец/середину, удаление из начала/конца/середины, поиск, обход. Двусвязный список: указатели на следующий и предыдущий, обход в обратном направлении. Кольцевой список. Сравнение с динамическим массивом (доступ по индексу, вставка/удаление, память).	ОПК-6.1, ОПК-6.2, ОПК-7.2, ПК-3.1	2		2	2

11	Тема 11: Стек и очередь. Реализация на массиве и списке Стек (LIFO): принцип работы, операции push, pop, top. Очередь (FIFO): принцип работы, операции enqueue, dequeue, front. Реализация стека на динамическом массиве. Реализация стека на связном списке. Реализация очереди на кольцевом буфере. Реализация очереди на связном списке. Очередь на двух стеках. Анализ сложности операций. Применение: проверка скобок, обратная польская запись.	ОПК-6.1, ОПК-6.2, ОПК-7.2, ПК-3.	2		2	2
12	Тема 12: Дек и приоритетная очередь Дек (Double-Ended Queue): двусторонняя очередь, операции push_front, push_back, pop_front, pop_back. Реализация дека на основе двусвязного списка. Приоритетная очередь: понятие приоритета, извлечение максимального элемента. Двоичная куча (max-heap, min-heap): представление в виде массива. Операции push (heapifyUp), pop (heapifyDown), top. Построение кучи из массива (heapify). Анализ сложности $O(\log n)$ для push/pop. Применение: алгоритм Дейкстры, сортировка Heap Sort.	ОПК-6.1, ОПК-6.2, ОПК-7.2, ПК-3.1	2		2	2
13	Тема 13: Рекурсивные алгоритмы. Простые сортировки Прямая и косвенная рекурсия. Базовый случай и шаг рекурсии. Рекурсивный обход списка и дерева. Ханойские башни: рекурсивное решение, подсчёт ходов. Обход лабиринта (backtracking). Сравнение рекурсивных и итеративных реализаций (факториал, числа Фибоначчи). Простые сортировки: пузырьковая (Bubble Sort), сортировка вставками (Insertion Sort), сортировка выбором (Selection Sort). Анализ сложности $O(n^2)$. Устойчивость и адаптивность простых сортировок.	ОПК-1.2, ОПК-6.1, ОПК-6.2, ОПК-7.1, ПК-3.1	2		2	2

14	Тема 14: Быстрая сортировка. Сортировка слиянием. Пирамидальная сортировка Принцип «разделяй и властвуй» (Divide and Conquer). Быстрая сортировка (Quick Sort): выбор опорного элемента (pivot), процедура разбиения (partition), рекурсивная сортировка подмассивов. Анализ сложности: лучший $O(n \log n)$, худший $O(n^2)$. Способы выбора опорного элемента (медиана трёх). Сортировка слиянием (Merge Sort): рекурсивное разделение, процедура слияния (merge). Анализ сложности $O(n \log n)$, дополнительная память $O(n)$. Пирамидальная сортировка (Heap Sort): построение кучи, извлечение элементов. Сравнение эффективных сортировок (Quick Sort vs Merge Sort vs Heap Sort).	ОПК-1.2, ОПК-1.3, ОПК-6.1, ОПК-6.2, ПК-3.1	2		2	2
15	Тема 15: Двоичный поиск. Хеш-таблицы Линейный поиск ($O(n)$) vs бинарный поиск ($O(\log n)$). Итеративная и рекурсивная реализации бинарного поиска. Поиск границ (первое/последнее вхождение). Поиск в циклически сдвинутом массиве. Хеш-таблицы: понятие хеш-функции. Требования к хеш-функции (детерминизм, равномерность, скорость). Коллизии. Методы разрешения коллизий: метод цепочек (chaining), открытая адресация (линейное пробирование, квадратичное пробирование, двойное хеширование). Рехеширование (динамическое расширение). Анализ сложности: $O(1)$ в среднем, $O(n)$ в худшем.	ОПК-1.2, ОПК-6.1, ОПК-6.2, ОПК-7.2, ПК-3.1	2		2	2

16	Тема 16: Бинарное дерево. Обходы. BST Понятие бинарного дерева. Терминология: корень (root), лист (leaf), внутренний узел, высота дерева, глубина узла, уровень. Обходы: прямой (pre-order — корень → левое → правое), симметричный (in-order — левое → корень → правое), обратный (post-order — левое → правое → корень), уровневый (level-order — с использованием очереди). Вычисление характеристик: высота, количество узлов, количество листьев. Бинарное дерево поиска (BST): свойство упорядоченности. Операции: вставка, поиск, удаление (три случая: лист, один потомок, два потомка). Анализ сложности: $O(\log n)$ в среднем, $O(n)$ в худшем (вырожденное дерево).	ОПК-6.1, ОПК-6.2, ОПК-7.2, ПК-2.4, ПК-3.1	2		2	2
17	Тема 17: AVL-деревья. Префиксное дерево (Trie) Проблема вырождения BST. AVL-деревья (самобалансирующиеся): фактор баланса (высота левого поддерева - высота правого). Типы поворотов: правый (Right Rotation), левый (Left Rotation), правый-левый (Right-Left Rotation), левый-правый (Left-Right Rotation). Вставка с балансировкой. Сравнение с красно-чёрными деревьями. Префиксное дерево (Trie): структура узла (массив указателей на потомков, флаг конца слова). Операции: вставка, поиск, поиск по префиксу. Автодополнение (поиск всех слов с заданным префиксом). Сравнение с хеш-таблицей для строковых ключей.	ОПК-1.2, ОПК-6.1, ОПК-6.2, ОПК-7.2, ПК-3.1	2		2	2

18	Тема 18: Представление графов. DFS и BFS Основные понятия теории графов: вершины (vertices), рёбра (edges), степень вершины. Ориентированный vs неориентированный граф. Взвешенный vs невзвешенный граф. Путь, цикл, компонента связности. Способы представления: матрица смежности ($O(V^2)$ памяти), список смежности ($O(V+E)$ памяти). Обход в глубину (DFS — Depth-First Search): рекурсивная и итеративная (с использованием стека) реализации. Поиск компонент связности. Обход в ширину (BFS — Breadth-First Search): с использованием очереди. Поиск кратчайшего пути в невзвешенном графе. Сравнение DFS и BFS (области применения).	ОПК-6.1, ОПК-6.2, ОПК-7.2, ПК-3.1	2		2	2
19	Тема 19: Алгоритм Дейкстры. Алгоритм Флойда-Уоршелла Кратчайшие пути в графах. Алгоритм Дейкстры: жадный алгоритм для графов с неотрицательными весами. Релаксация рёбер. Очередь с приоритетами для оптимизации ($O((V+E) \log V)$). Восстановление пути. Алгоритм Флойда-Уоршелла: динамическое программирование для всех пар вершин ($O(V^3)$). Инициализация матрицы расстояний. Обнаружение отрицательных циклов. Сравнение алгоритмов (Дейкстра vs Флойд-Уоршелл). Области применения.	ОПК-1.2, ОПК-6.1, ОПК-6.2, ПК-3.1	2		2	2

20	Тема 20: Алгоритмы Прима и Краскала. Минимальное остовное дерево Понятие остовного дерева (Spanning Tree). Минимальное остовное дерево (MST — Minimum Spanning Tree). Свойства MST (разрез, цикл). Алгоритм Прима: жадный алгоритм, начинающий с одной вершины, на каждом шаге добавляющий минимальное ребро, соединяющее текущее дерево с новой вершиной. Очередь с приоритетами. Алгоритм Краскала: сортировка рёбер по весу, добавление рёбер, не создающих цикла. Система непересекающихся множеств (DSU): операции find (сжатие пути) и union (объединение по рангу). Сравнение алгоритмов Прима и Краскала (плотные vs разреженные графы).	ОПК-1.2, ОПК-6.1, ОПК-6.2, ПК-3.1	2		2	2
21	Тема 21: Динамическое программирование. Задача о рюкзаке. LCS Принципы ДП: оптимальная подструктура, перекрывающиеся подзадачи, мемоизация (top-down), табличный метод (bottom-up). Задача о рюкзаке (0/1): рекурсивное решение, рекурсия с мемоизацией, табличное решение (DP-таблица). Восстановление выбранных предметов. Наибольшая общая подпоследовательность (LCS — Longest Common Subsequence): постановка задачи, рекуррентное соотношение, заполнение таблицы LCS ($O(m \times n)$), восстановление последовательности. Применение LCS (сравнение файлов, биоинформатика). Сравнение с наибольшей общей подстрокой.	ОПК-1.1, ОПК-1.2, ОПК-6.1, ОПК-6.2, ПК-3.1	2		2	2

22	Тема 22: Алгоритм Кнута-Морриса-Пратта. STL (обзор) Наивный поиск подстроки ($O(n \times m)$). Префикс-функция: определение, вычисление за $O(m)$. Алгоритм Кнута-Морриса-Пратта (КМП — КМР): использование префикс-функции для эффективного сдвига при несовпадении. Поиск всех вхождений. Анализ сложности $O(n+m)$. Обзор STL (Standard Template Library): контейнеры (vector, list, deque, set, map, unordered_set, unordered_map), итераторы (begin, end, обратные итераторы), алгоритмы (find, sort, copy, count_if, for_each). Примеры использования.	ОПК-1.2, ОПК-6.1, ОПК-6.2, ПК-3.1, ПК-3.4	2		2	2
23	Тема 23: Наследование и полиморфизм. Проектирование иерархий классов Наследование: базовый и производный классы. Открытое (public), защищённое (protected) и закрытое (private) наследование. Отношение «is a». Конструкторы и деструкторы в производных классах. Замещение методов базового класса. Виртуальные методы (virtual): позднее связывание (динамический полиморфизм). Таблица виртуальных методов (vtbl). Абстрактные классы и чисто виртуальные методы ($= 0$). Виртуальный деструктор. Диаграммы классов UML: обозначения для классов, атрибутов, методов, отношений (наследование, агрегация, композиция, зависимость). Паттерны проектирования (Стратегия, Шаблонный метод).	ОПК-6.2, ОПК-7.1, ПК-2.1, ПК-2.2, ПК-2.3, ПК-2.4, ПК-3.4	2		2	7

24	Тема 24: Сериализация. Обработка исключений. Обзор курса Сериализация и десериализация объектов: преобразование состояния объекта в форму, пригодную для сохранения/передачи. Текстовая сериализация (CSV, собственный формат). Бинарная сериализация (запись структур целиком). Обработка исключительных ситуаций (exceptions): try, catch, throw. Стандартные классы исключений (std::exception, std::runtime_error, std::logic_error, std::out_of_range, std::invalid_argument). Создание собственных классов исключений. Исключения в конструкторах (единственный способ сообщить об ошибке создания). Исключения в деструкторах (не должны выбрасывать). RAII (Resource Acquisition Is Initialization). Обзор всего курса: основные структуры данных, алгоритмы, принципы ООП.	ОПК-6.2, ОПК-6.4, ПК-3.3, ПК-3.4				
	Экзамен	27				
	ИТОГО за 2 семестр		48		48	58
	ИТОГО		48		48	58

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств (ФОС) по дисциплине (модулю) базируется на перечне осваиваемых компетенций с указанием индикаторов. ФОС обеспечивает объективный контроль достижения запланированных результатов обучения. ФОС включает в себя:

- описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;
- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций (включаются в методические указания по тем видам работ, которые предусмотрены учебным планом и предусматривают оценку сформированности компетенций);
- типовые оценочные средства, необходимые для оценки знаний, умений и уровня сформированности компетенций.

ФОС является приложением к данной программе дисциплины.

7. Методические указания для обучающихся по освоению дисциплины

Приступая к работе, каждый студент должен принимать во внимание следующие положения.

Дисциплина построена по тематическому принципу, каждая тема представляет собой логически завершённый раздел.

Практические занятия проводятся с целью закрепления усвоенной информации, приобретения навыков ее применения при решении практических задач в соответствующей предметной области.

Самостоятельная работа студентов направлена на самостоятельное изучение дополнительного материала, подготовку к лабораторным занятиям, а также выполнения всех видов самостоятельной работы.

Для успешного освоения дисциплины, необходимо выполнить все виды самостоятельной работы, используя рекомендуемые источники информации.

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

8.1.1. Перечень основной литературы:

1. Павловская Т.А., Щупак Ю.А. «С/С++. Структурное и объектно-ориентированное программирование: Практикум». — СПб.: Питер, 2011. — 352 с. — (Серия «Учебное пособие»). ISBN 978-5-459-00613-1.
2. Лафоре Р. «Объектно-ориентированное программирование в С++». — 4-е изд. — СПб.: Питер, 2011. — 928 с. — (Классика Computer Science). ISBN 978-5-459-00659-9.
3. Страуструп Б. «Язык программирования С++». — Спец. изд. — М.: Бином; СПб.: Невский Диалект, 2008. — 1104 с. ISBN 978-5-7940-0145-7.
4. Седжвик Р. «Фундаментальные алгоритмы на С++. Анализ / Структуры данных / Сортировка / Поиск». — Пер. с англ. — Киев: ДияСофт, 2001. — 688 с. ISBN 966-7393-67-X.
5. Липпман С.Б., Лажоие Ж., Му Б.Э. «Язык программирования С++. Вводный курс». — 4-е изд. — М.: Вильямс, 2007. — 1120 с. ISBN 978-5-8459-1347-2.

8.1.2. Перечень дополнительной литературы:

1. Мейерс С. «Эффективное использование С++. 55 верных способов улучшить структуру и код ваших программ». — М.: ДМК Пресс, 2006. — 300 с. ISBN 5-93700-012-X.
2. Саттер Г. «Решение сложных задач на С++». — М.: Вильямс, 2002. — 400 с. ISBN 5-8459-0225-8.
3. Александреску А. «Современное проектирование на С++: Обобщенное программирование и прикладные шаблоны проектирования». — М.: Вильямс, 2008. — 336 с. ISBN 978-5-8459-1248-2.
4. Вандевурд Д., Джосаттис Н. «Шаблоны С++: справочник разработчика». — М.: Вильямс, 2003. — 544 с. ISBN 5-8459-0386-6.
5. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. «Приемы объектно-ориентированного проектирования. Паттерны проектирования». — СПб.: Питер, 2007. — 366 с. ISBN 978-5-469-01136-9.
6. Кнут Д. «Искусство программирования» (в 3 т.). — М.: Вильямс, 2010. ISBN 978-5-8459-1710-4.
7. Шилдт Г. «Искусство программирования на С++». — СПб.: BHV, 2005. — 496 с. ISBN 5-94157-316-4.
8. Эккель Б. «Философия С++. Введение в стандартный С++. 2-е изд.». — СПб.: Питер, 2004. — 608 с. ISBN 5-94723-645-6.

8.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

Методические указания по выполнению лабораторных работ по дисциплине «Алгоритмы и структуры данных» (электронный ресурс).

8.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Справочно-документационные ресурсы

1. [cppreference.com](https://ru.cppreference.com) (<https://ru.cppreference.com>) — справочник по языку C++ и стандартной библиотеке на русском языке. Содержит полное описание синтаксиса, стандартной библиотеки шаблонов (STL) и алгоритмов. Доступ свободный.
2. [cplusplus.com](https://www.cplusplus.com) (<https://www.cplusplus.com>) — справочник по C++ с примерами кода и пояснениями (на английском языке). Доступ свободный.
3. isocpp.org (<https://isocpp.org>) — официальный сайт комитета по стандартизации C++. Содержит новости, обучающие материалы и FAQ. Доступ свободный.

Образовательные и научные ресурсы

1. **GeekBrains** (<https://gb.ru>) — образовательная платформа с курсами по C++ и алгоритмам. Доступ к базовым материалам свободный, углублённые курсы по подписке.
2. **Stepik.org** (<https://stepik.org>) — образовательная платформа с открытыми курсами по C++, алгоритмам и структурам данных. Доступ свободный.
3. **Habr.com** (<https://habr.com>) — профильные статьи, обзоры и tutorиалы по C++, алгоритмам и структурам данных. Доступ свободный.
4. **GitHub** (<https://github.com>) — хостинг IT-проектов. Доступ свободный. По состоянию на 2026 год доступ на территории РФ сохраняется.

Научно-образовательные ресурсы

1. [MCCME.ru](https://www.mccme.ru) (<https://www.mccme.ru>) — сайт Московского центра непрерывного математического образования. Содержит лекции и материалы по алгоритмам. Доступ свободный.
2. [Neerc.ifmo.ru](https://neerc.ifmo.ru) (<https://neerc.ifmo.ru>) — сайт кафедры компьютерных технологий ИТМО. Содержит материалы по олимпиадному программированию и алгоритмам. Доступ свободный.
3. [CyberLeninka.ru](https://cyberleninka.ru) (<https://cyberleninka.ru>) — научная электронная библиотека. Содержит научные статьи по алгоритмам и структурам данных. Доступ свободный.

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

При чтении лекций используется компьютерная техника, демонстрации презентационных мультимедийных материалов. На семинарских и практических занятиях студенты представляют презентации, подготовленные ими в часы самостоятельной работы.

Информационные справочные системы:

Информационно-справочные и информационно-правовые системы, используемые при изучении дисциплины:

Программное обеспечение:

Средства разработки (интегрированные среды разработки — IDE)

1. **Microsoft Visual Studio Community** — Бесплатная лицензия для индивидуальных разработчиков, образовательных учреждений и проектов с открытым кодом. Доступно для скачивания с официального сайта Microsoft.
2. **JetBrains CLion** — Бесплатная лицензия для студентов и преподавателей

(оформляется по корпоративной почте @edu). Доступно через официальный сайт JetBrains.

3. **Qt Creator** — Бесплатная лицензия (LGPL/GPL). Доступно для скачивания с официального сайта или GitHub.
4. **Code::Blocks** — Бесплатная лицензия (GNU GPL). Доступно для скачивания с официального сайта.
5. **Eclipse CDT** — Бесплатная лицензия (EPL). Доступно для скачивания с официального сайта.

Компиляторы и инструменты сборки

1. **GCC (GNU Compiler Collection)** — Бесплатная лицензия (GNU GPL с исключением для библиотек времени выполнения). Доступно для установки из официальных репозиторий.
2. **MinGW-w64** — Бесплатная лицензия (Zope Public License, совместима с GPL). Доступно для скачивания с официального сайта.
3. **Clang** — Бесплатная лицензия (Apache 2.0). Доступно для скачивания с официального сайта.
4. **CMake** — Бесплатная лицензия (BSD 3-Clause). Доступно для скачивания с официального сайта или через PyPI.

Инструменты документации и отладки

1. **Doxygen** — Бесплатная лицензия (GNU GPL). Доступно для скачивания с официального сайта.
2. **Valgrind** — Бесплатная лицензия (GNU GPL). Доступно для установки из репозиторий.
3. **GDB (GNU Debugger)** — Бесплатная лицензия (GNU GPL). Поставляется в составе GCC.

Системы контроля версий

1. **Git** — Бесплатная лицензия (GNU GPL). Доступно для скачивания с официального сайта.
2. **GitHub Desktop** — Бесплатная лицензия. Доступно для скачивания с официального сайта.

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

Лекционные занятия	Учебная аудитория для проведения учебных занятий, оснащенная мультимедийным оборудованием и техническими средствами обучения.
Лабораторные работы	Учебная аудитория для проведения учебных занятий, оснащенная мультимедийным оборудованием и техническими средствами обучения.
Самостоятельная работа	Помещение для самостоятельной работы обучающихся оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и возможностью доступа к электронной информационно-образовательной среде университета

11. Особенности освоения дисциплины (модуля) лицами с ограниченными возможностями здоровья

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

Освоение дисциплины (модуля) обучающимися с ограниченными возможностями здоровья может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины (модуля) обучающимися с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения высшего образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины (модуля) обеспечивается:

1) для лиц с ограниченными возможностями здоровья по зрению:

- присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),
- письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,
- специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),
- индивидуальное равномерное освещение не менее 300 люкс,
- при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;

2) для лиц с ограниченными возможностями здоровья по слуху:

- присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),
- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;
- обеспечивается надлежащими звуковыми средствами воспроизведения информации;

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата (в том числе с тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;
- по желанию студента задания могут выполняться в устной форме.

12. Особенности реализации дисциплины с применением дистанционных образовательных технологий и электронного обучения

Согласно части 1 статьи 16 Федерального закона от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» под *электронным обучением* понимается организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников. Под *дистанционными образовательными технологиями* понимаются образовательные технологии, реализуемые в основном с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) взаимодействии обучающихся и педагогических работников.

Реализация дисциплины может быть осуществлена с применением дистанционных образовательных технологий и электронного обучения полностью или частично. Компоненты УМК дисциплины (рабочая программа дисциплины, оценочные и

методические материалы, формы аттестации), реализуемой с применением дистанционных образовательных технологий и электронного обучения, содержат указание на их использование.

При организации образовательной деятельности с применением дистанционных образовательных технологий и электронного обучения могут предусматриваться асинхронный и синхронный способы осуществления взаимодействия участников образовательных отношений посредством информационно-телекоммуникационной сети «Интернет».

При применении дистанционных образовательных технологий и электронного обучения в расписании по дисциплине указываются: способы осуществления взаимодействия участников образовательных отношений посредством информационно-телекоммуникационной сети «Интернет» (ВКС-видеоконференцсвязь, ЭТ – электронное тестирование); ссылки на электронную информационно-образовательную среду СКФУ, на образовательные платформы и ресурсы иных организаций, к которым предоставляется открытый доступ через информационно-телекоммуникационную сеть «Интернет»; для синхронного обучения - время проведения онлайн-занятий и преподаватели; для асинхронного обучения - авторы онлайн-курсов.

При организации промежуточной аттестации с применением дистанционных образовательных технологий и электронного обучения используются Методические рекомендации по применению технических средств, обеспечивающих объективность результатов при проведении промежуточной и государственной итоговой аттестации по образовательным программам высшего образования - программам бакалавриата, программам специалитета и программам магистратуры с применением дистанционных образовательных технологий (Письмо Минобрнауки России от 07.12.2020 г. № МН-19/1573-АН "О направлении методических рекомендаций").

Реализация дисциплины с применением электронного обучения и дистанционных образовательных технологий осуществляется с использованием электронной информационно-образовательной среды СКФУ, к которой обеспечен доступ обучающихся через информационно-телекоммуникационную сеть «Интернет», или с использованием ресурсов иных организаций, в том числе платформ, предоставляющих сервисы для проведения видеоконференций, онлайн-встреч и дистанционного обучения (МТС-Линк), а также с использованием возможностей социальных сетей для осуществления коммуникации обучающихся и преподавателей.

Учебно-методическое обеспечение дисциплины, реализуемой с применением электронного обучения и дистанционных образовательных технологий, включает представленные в электронном виде рабочую программу, учебно-методические пособия или курс лекций, методические указания к выполнению различных видов учебной деятельности обучающихся, предусмотренных дисциплиной, и прочие учебно-методические материалы, размещенные в информационно-образовательной среде СКФУ.