

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет физико-технический

УТВЕРЖДАЮ



Проректор по учебной работе, качеству
образования – первый проректор
Хагуров Т.А.
«02» июня 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.15.01 ИНФОРМАТИКА И ТЕОРИЯ АЛГОРИТМОВ

Направление подготовки/
специальность 09.03.02 Информационные системы и технологии

Направленность (профиль) /
специализация Аналитические информационные системы

Форма обучения очная, заочная
(очная, очно-заочная, заочная)

Квалификация бакалавр

Краснодар 2026

Рабочая программа дисциплины «Информатика и теория алгоритмов» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.02 Информационные системы и технологии (Аналитические информационные системы), 11.03.02 Инфокоммуникационные технологии и системы связи (Оптические системы и сети связи)

Программу составил(и):

Куликова Н.Н.

доцент кафедры теоретической физики
и компьютерных технологий, к.б.н.



Рабочая программа дисциплины утверждена на заседании кафедры теоретической физики и компьютерных технологий

протокол № 10 от «21» апреля 2026 г.

Заведующий кафедрой

Лебедев К.А

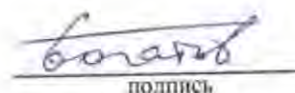

подпись

Утверждена на заседании учебно-методической комиссии факультета физико-технического

протокол № 11 «28» апреля 2026 г.

Председатель УМК факультета

Богатов Н.М.


подпись

Рецензенты:

Коваленко М.С., доцент кафедры физики и информационных систем, к.ф.-м.н.

Григорьян Л.Р., генеральный директор НПФ «Мезон», к.ф.-м.н.

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

формирование фундаментальных знаний в области информатики и теории алгоритмов, а также практических навыков разработки на языке C++ эффективных программных решений с использованием самостоятельно реализованных структур данных и алгоритмов для решения профессиональных задач.

1.2 Задачи дисциплины

- изучить синтаксис, семантику и основные конструкции языка C++.
- освоить базовые структуры данных.
- сформировать умения оценивать временную и пространственную сложность алгоритмов и выбирать оптимальные структуры данных для конкретной задачи.
- развить навыки работы в интегрированной среде разработки (IDE).
- получить общее представление об использовании объектно-ориентированного программирования при решении различных категорий задач.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Данная дисциплина входит в обязательную часть блока Б1 "Дисциплины (модули)" части учебного плана, формируемой участниками образовательных отношений.

Знания, полученные в рамках изучения данной дисциплины, у направления 09.03.02 Информационные системы и технологии являются фундаментальными для дальнейшего освоения учебного плана, в частности, являются базовыми для таких дисциплин, как «Учебная (ознакомительная) практика», «Основы программирования», «Программирование на Python и анализ данных», «ООП и паттерны проектирования», «Технологии программирования на C/C++». Для направления 11.03.02 Инфокоммуникационные технологии и системы связи знания, полученные в рамках изучения данной дисциплины, являются базовыми для таких дисциплин, как «Программирование на Python и анализ данных», «Цифровая обработка сигналов», курсового и дипломного проектирования.

1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций направление 09.03.02 Информационные системы и технологии:

Код и наименование индикатора*	Результаты обучения по дисциплине
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	
ОПК-2.1. Знать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Знать синтаксис и семантику языка C++, включая указатели, ссылки, динамическое выделение памяти, управляющие конструкции, функции, перегрузку, классы и наследование. Современные инструменты разработки и принципы написания качественного кода .
	Уметь писать программный код с учётом эффективности и надёжности
	Владеть навыками практического программирования на C++ для создания работающего программного кода.
ОПК-2.2. Уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Знать классификацию, свойства и области применения основных структур данных. Базовые алгоритмы. Понятия сложности алгоритмов (О-нотация) и критерии выбора оптимальной структуры данных для конкретной задачи
	Уметь анализировать входные данные и требования задачи для обоснованного выбора структуры данных и алгоритма

Код и наименование индикатора*	Результаты обучения по дисциплине
	Владеть навыками реализации динамических структур данных с помощью указателей и классов.
ОПК-2.3. Иметь навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	Знать принципы объектно-ориентированного программирования и их реализацию в C++
	Уметь создавать программы, пригодные для практического применения
	Владеть навыками разработки алгоритмов и программ на C++ с применением объектно-ориентированного подхода.
ОПК-3 Иметь навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	
ОПК-3.1. Знать принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Принципы и методы решения стандартных алгоритмических задач. Интегрированные среды разработки (IDE), системы контроля версий (Git), средства отладки и профилирования. Основные требования информационной безопасности при разработке на C++: защита от переполнения буфера, утечек памяти, некорректного ввода. Основы информационной и библиографической культуры при поиске научно-технической информации по алгоритмам и структурам данных.
	Использовать IDE, Git, отладчики для решения стандартных задач программирования на C++. Учитывать требования информационной безопасности при написании кода
	Владеть навыками работы в среде разработки C++. Навыками использования Git для управления версиями программ. Навыками безопасного программирования.
ОПК-3.2. Уметь решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Методы и алгоритмы решения типовых задач обработки данных. Основы применения ИКТ для автоматизации вычислений и отладки. Требования к защите информации при передаче и хранении данных в программах на C++
	Решать стандартные алгоритмические задачи, реализуя их на C++
	Владеть навыками практической реализации алгоритмов сортировки, поиска и структур данных на C++. Навыками использования отладочных средств IDE для выявления ошибок времени выполнения.
ОПК-3.3. Иметь навыки подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности	Основы научно-исследовательской работы в области программирования на C++. Правила обеспечения информационной безопасности при подготовке и распространении материалов: защита авторского кода, корректное цитирование, неразглашение критической информации.
	Составлять отчеты по лабораторным работам, включающие описание алгоритмов, листинги программ, результаты экспериментов и выводы.
	Навыками оформления технической документации к программам на C++.

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций направление 11.03.02 Инфокоммуникационные технологии и системы связи:

Код и наименование индикатора*	Результаты обучения по дисциплине
ОПК-3 Способен применять методы поиска, хранения, обработки, анализа и представления в требуемом формате информации из различных источников и баз данных, соблюдая при этом основные требования информационной безопасности	
ОПК-3.1. Знает основные закономерности и принципы передачи информации в инфокоммуникационных системах, основные виды сигналов, используемых в телекоммуникационных системах, особенности передачи различных сигналов по каналам и трактам телекоммуникационных систем	Знать способы представления информации в цифровом виде, типы данных и их размеры в C++.
	Уметь проектировать решение конкретной задачи: формулировать требования, выбирать структуры данных и алгоритмы, обосновывать выбор.
	Владеть навыками проектирования программ на C++ с использованием объектно-ориентированного подхода. Навыками выбора оптимальных алгоритмов и структур данных для заданных условий. Навыками оценки эффективности алгоритмов

Код и наименование индикатора*	Результаты обучения по дисциплине
ОПК-3.2. Способен решать задачи обработки данных с помощью средств вычислительной техники	Знать классификацию, свойства и области применения основных структур данных. Базовые алгоритмы. Понятия сложности алгоритмов (О-нотация) и критерии выбора оптимальной структуры данных для конкретной задачи
	Уметь анализировать входные данные и требования задачи для обоснованного выбора структуры данных и алгоритма
	Владеть навыками реализации динамических структур данных с помощью указателей и классов.
ОПК-3.3. Владеет методами и навыками обеспечения информационной безопасности	Знать принципы объектно-ориентированного программирования и их реализацию в C++
	Уметь создавать программы, пригодные для практического применения
	Владеть навыками разработки алгоритмов и программ на C++ с применением объектно-ориентированного подхода.
ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	
ОПК-4.1. Знает современные интерактивные программные комплексы и основные приемы обработки экспериментальных данных, в том числе с использованием стандартного программного обеспечения, пакетов программ общего и специального назначения	Знать методы и алгоритмы обработки числовых и текстовых данных на C++. Возможности средств вычислительной техники и их влияние на выбор алгоритма и структуры данных. Принципы организации ввода-вывода и хранения данных в файлах.
	Уметь решать задачи обработки данных с использованием языка C++ и средств вычислительной техники (выбор оптимального алгоритма по времени и памяти). Использовать стандартные возможности C++ для работы с массивами, строками, файлами. Проводить анализ эффективности программ.
	Владеть навыками реализации алгоритмов обработки данных на C++. Навыками оценки временной и пространственной сложности программ. Навыками работы с файловой системой для хранения и чтения данных..
ОПК-4.2. Использует возможности вычислительной техники и программного обеспечения для решения задач управления и алгоритмизации процессов обработки информации	Знать принципы безопасного программирования на C++: использование проверок границ, корректное освобождение памяти, обработка исключительных ситуаций
	Уметь применять методы обеспечения информационной безопасности при разработке программ на C++: проверять входные данные, избегать переполнения буфера, правильно управлять динамической памятью.
	Владеть навыками безопасного программирования на C++ (предотвращение утечек, переполнений)..
ОПК-4.3. Владеет методами компьютерного моделирования физических процессов при передаче информации, техникой инженерной и компьютерной графики	Знать методологию проектирования программного решения: анализ требований, выбор архитектуры, разработка алгоритмов. Критерии оптимальности: временная и пространственная сложность, простота реализации, надежность.
	Моделировать поведение алгоритмов на тестовых наборах данных с визуализацией промежуточных состояний.
	Владеть навыками проектирования программ на C++ с использованием объектно-ориентированного подхода (классы, инкапсуляция). Навыками выбора оптимальных алгоритмов и структур данных для заданных условий.
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	
ОПК-5.1. Проектирует решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений	Знать синтаксис и семантику языка C++, включая указатели, ссылки, динамическое выделение памяти, управляющие конструкции, функции, перегрузку, классы и наследование. Современные инструменты разработки и принципы написания качественного кода .
	Уметь писать программный код с учётом эффективности и надёжности
	Владеть навыками практического программирования на C++ для создания работающего программного кода.

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 4 зачетных единиц (144 часов), их распределение по видам работ представлено в таблице

Виды работ		Всего часов	Форма обучения			
			очная		очно-заочная	заочная
			1 семестр (часы)	X семестр (часы)	X семестр (часы)	X курс (часы)
Контактная работа, в том числе:		76,3	76,3			
Аудиторные занятия (всего):			68			8
занятия лекционного типа			34			4
лабораторные занятия			34			4
практические занятия			-			
семинарские занятия			-			
Иная контактная работа:			8,3			
Контроль самостоятельной работы (КСР)			8			
Промежуточная аттестация (ИКР)			0,3			
Самостоятельная работа, в том числе:			32			91
Самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т.д.)			32			91
Подготовка к текущему контролю						
Контроль:		35,7	35,7			9
Подготовка к экзамену		35,7	35,7			9
Общая трудоемкость	час.	144	144			144
	в том числе контактная работа		76,3			
	зач. ед	4	4			4

2.2 Содержание дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины. Разделы (темы) дисциплины, изучаемые в 1 семестре (очная форма обучения)

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1 семестр						
1.	Основы алгоритмизации и программирования	46	14	-	20	12
2.	Структуры данных и классические алгоритмы	26	10	-	6	10
3.	Объектно-ориентированное программирование	28	10	-	8	10
	ИТОГО по разделам дисциплины	100	34	-	34	32
	Контроль самостоятельной работы (КСР)	8				
	Промежуточная аттестация (ИКР)	0,3				
	Подготовка к текущему контролю	35,7				
	Общая трудоемкость по дисциплине	144				

Разделы (темы) дисциплины, изучаемые в 1 семестре (заочная форма обучения)

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1 семестр						
4.	Основы алгоритмизации и программирования	54	2	-	2	50
5.	Структуры данных и классические алгоритмы	52	1	-	1	50
6.	Объектно-ориентированное программирование	37	1	-	1	35
	ИТОГО по разделам дисциплины	135	4	-	4	135
	Контроль самостоятельной работы (КСР)					
	Промежуточная аттестация (ИКР)					
	Подготовка к текущему контролю	9				
	Общая трудоемкость по дисциплине	144				

Примечание: Л – лекции, ПЗ – практические занятия / семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1.	Основы алгоритмизации и программирования	Введение в информатику и алгоритмы. Свойства алгоритмов. Формы записей алгоритмов. Общие принципы построения алгоритмов. Анализ сложности алгоритмов. Язык C++ как надстройка над языком C. Структура программы. Операторы ввода– вывода. Базовые типы данных. Циклические конструкции. Массивы. Работа с указателями. Функции: способы организации и описание. Вызов функций, рекурсия. Строки. Сложные типы данных: структуры и объединения. Работа с файлами	ЛР, Т
2.	Структуры данных и классические алгоритмы	Базовые структуры данных. Алгоритмы сортировки, поиска. Линейные структуры данных: списки, стеки, очереди. Бинарные деревья и их обходы.	ЛР, Т
3.	Объектно-ориентированное программирование	Основы ООП в C++. Инкапсуляция, наследование и полиморфизм. Перегрузка операторов	ЛР, Т

2.3.2 Лабораторные работы

№	Наименование раздела (темы)	Тематика работ	Форма текущего контроля
1.	Основы алгоритмизации и программирования	Программирование линейных алгоритмов Использование условных операторов Программирование циклических алгоритмов Программирование с использованием одномерных массивов Указатели. Программирование с использованием динамических двумерных массивов Программирование с использованием строк Программирование с использованием структур Программирование с использованием функций Рекурсия Программирование с использованием файлов	ЛР
2.	Структуры данных и классические алгоритмы	Сортировка массива структур по ключевому полю с использованием алгоритмов вставками, слиянием и быстрой сортировки Реализация алгоритмов бинарного и интерполяционного поиска в массиве структур с предварительной сортировкой по ключу	ЛР
3.	Объектно-ориентированное программирование	Класс «Стек»: реализация динамического стека на основе связанного списка Класс «Очередь»: реализация циклической очереди и очереди на основе списка	ЛР

Защита лабораторной работы (ЛР), выполнение курсового проекта (КП), курсовой работы (КР), расчетно-графического задания (РГЗ), написание реферата (Р), эссе (Э), коллоквиум (К), тестирование (Т) и т.д.

При изучении дисциплины могут применяться электронное обучение, дистанционные образовательные технологии в соответствии с ФГОС ВО.

2.3.3 Примерная тематика курсовых работ (проектов)

Курсовые проекты не предусмотрены

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Подготовка к лекционным занятиям	1. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. - Москва : Юрайт, 2022. - 137 с. - URL: https://urait.ru/bcode/491215 (дата обращения: 26.08.2022). - Режим доступа: для авториз. пользователей. - ISBN 978-5-534-07834-3. - Текст : электронный. 2. Методические указания по организации аудиторной и самостоятельной работ, утвержденные кафедрой теоретической физики и компьютерных технологий, протокол № 9 от «14» марта 2017г 3. ОСМДО КубГУ «Информатика и теория алгоритмов». www.openedu.kubsu.ru
2	Подготовка к лабораторным работам	1. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. - Москва : Юрайт, 2022. - 137 с. - URL: https://urait.ru/bcode/491215 (дата обращения: 26.08.2022). - Режим доступа: для авториз. пользователей. - ISBN 978-5-534-07834-3. - Текст : электронный.

		2. Методические указания по организации аудиторной и самостоятельной работ, утвержденные кафедрой теоретической физики и компьютерных технологий, протокол № 9 от «14» марта 2017г 3. ОСМДО КубГУ «Информатика и теория алгоритмов». www.openedu.kubsu.ru
3	Подготовка к тестированию	1. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. - Москва : Юрайт, 2022. - 137 с. - URL: https://urait.ru/bcode/491215 (дата обращения: 26.08.2022). - Режим доступа: для авториз. пользователей. - ISBN 978-5-534-07834-3. - Текст : электронный. 2. Методические указания по организации аудиторной и самостоятельной работ, утвержденные кафедрой теоретической физики и компьютерных технологий, протокол № 9 от «14» марта 2017г 3. ОСМДО КубГУ «Информатика и теория алгоритмов». www.openedu.kubsu.ru

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии, применяемые при освоении дисциплины (модуля)

При изучении дисциплины предусмотрено использование комплекса образовательных технологий, обеспечивающих эффективное формирование у студентов как теоретических знаний, так и практических компетенций в области алгоритмизации и программирования на языке C++. К числу основных образовательных технологий относятся лекционные занятия и лабораторные работы, проблемное обучение, модульная технология, тестирование, а также различные формы самостоятельной работы студентов.

Лекционные занятия проводятся в двух форматах: классические лекции, предназначенные для изложения фундаментальных понятий теории алгоритмов, свойств алгоритмов и методов анализа сложности, они необходимы для пошагового объяснения сложных логических цепочек, а интерактивные лекции с мультимедийным сопровождением позволяют демонстрировать динамические структуры данных, визуализировать процессы сортировки и поиска, а также работу с динамической памятью, что способствует наглядному восприятию абстрактных концепций. При этом для ряда тем, особенно связанных с объектно-ориентированным программированием и перегрузкой операторов, активно используется проблемное обучение, когда лектор ставит перед

аудиторию проблемную задачу, например, разработку собственного контейнерного класса, и студенты совместно с преподавателем ищут оптимальные пути её решения. Модульная технология реализуется через разбиение всего курса на три логически завершённых модуля – «Основы алгоритмизации», «Структуры данных и классические алгоритмы» и «Объектно-ориентированное программирование», каждый из которых завершается контрольным мероприятием, что позволяет систематизировать процесс обучения и обеспечить поэтапный контроль усвоения материала.

Самостоятельная работа студентов охватывает изучение теоретического материала по учебникам и конспектам, подготовку к практическим и лабораторным занятиям, включающую написание кода, отладку и оформление отчётов, подготовку к тестированию и экзамену, а также выполнение дополнительных исследовательских заданий, таких как сравнительный анализ эффективности различных реализаций структур данных.

Компетентностный подход в рамках преподавания дисциплины реализуется в использовании интерактивных технологий и активных методов обучения в сочетании с внеаудиторной работой, к числу которых относятся проектные методики, мозговой штурм, разбор конкретных ситуаций, анализ педагогических задач и педагогический эксперимент. Проектные методики позволяют студентам разрабатывать собственные небольшие проекты, например, библиотеку контейнерных классов или программу для обработки структурированных данных из файлов, что способствует формированию навыков самостоятельной проектной деятельности. Мозговой штурм применяется при обсуждении возможных подходов к решению нетривиальных алгоритмических задач, например, при выборе оптимального алгоритма сортировки для больших объёмов данных, при этом студенты генерируют идеи и коллективно оценивают их эффективность.

Разбор конкретных ситуаций используется при анализе реальных ошибок из предыдущих потоков, когда студенты изучают некорректные реализации алгоритмов или структур данных, выявляют причины ошибок, таких как утечки памяти, переполнение буфера или некорректное использование указателей, и предлагают способы их исправления. Анализ педагогических задач направлен на выявление типичных затруднений, возникающих при изучении сложных разделов, и выработку методов их преодоления как для самого студента, так и для его потенциальной профессиональной деятельности.

При проведении лабораторных работ, каждому студенту выдается индивидуальное типовое задание. Студенты приступают к выполнению задания, взаимодействуя между собой. Преподаватель контролирует ход выполнения работы каждого студента, проверяет правильность выполнения лабораторных работ. Уточняя ход работы, если студенты что-то выполняют неправильно, преподаватель помогает им преодолеть сложные моменты и проверяет достоверность полученных результатов. Работа выполняется в интерактивном режиме, где студенты взаимодействуют друг с другом, обсуждают подходы и помогают в отладке, а преподаватель выступает в роли консультанта, контролируя ход выполнения, проверяя корректность кода, указывая на типичные ошибки и помогая преодолеть сложные моменты.

Особое внимание уделяется сочетанию аудиторной и внеаудиторной работы: для проведения лекций и демонстрации кода используется аудитория, оснащённая проектором и компьютером с установленной средой разработки, а при выполнении лабораторных работ студенты могут применять систему контроля версий Git для сохранения промежуточных результатов, что также является элементом современной инженерной практики. Кроме того, консультации с преподавателем проводятся как в очной, так и в дистанционной форме, через электронную почту или ОСМДО, для оперативного решения возникающих вопросов.

Таким образом, сочетание традиционных и инновационных образовательных технологий, активных методов обучения, модульной организации курса и различных форм самостоятельной и внеаудиторной работы обеспечивает комплексное формирование

профессиональных компетенций, предусмотренных рабочей программой дисциплины, и позволяет студентам не только освоить конкретные языковые конструкции C++, но и развить алгоритмическое мышление, навыки анализа сложности, командной работы и самостоятельного поиска информации.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Информатика и теория алгоритмов».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме защиты отчета по лабораторным работам и **промежуточной аттестации** в форме вопросов и заданий к экзамену.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации Направление 09.03.02 Информационные системы и технологии

№ п/п	Код и наименование индикатора	Результаты обучения	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
	ОПК-2.1. Знать современные информационные технологии и программные средства, в том числе отечественного производства при	Знать синтаксис и семантику языка C++, включая указатели, ссылки, динамическое выделение памяти, управляющие конструкции, функции, перегрузку, классы и наследование.	ЛР№1–ЛР№14	Вопрос на экзамене №1–32

	решении задач профессиональной деятельности	Современные инструменты разработки и принципы написания качественного кода.		
		Уметь писать программный код с учётом эффективности и надёжности	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
		Владеть навыками практического программирования на C++ для создания работающего программного кода.	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
	ОПК-2.2. Уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Знать классификацию, свойства и области применения основных структур данных. Базовые алгоритмы. Понятия сложности алгоритмов (О-нотация) и критерии выбора оптимальной структуры данных для конкретной задачи	ЛРН№1–ЛРН№14	Вопрос на экзамене №21–32
		Уметь анализировать входные данные и требования задачи для обоснованного выбора структуры данных и алгоритма	ЛРН№1–ЛРН№14	Вопрос на экзамене №21–32
		Владеть навыками реализации динамических структур данных с помощью указателей и классов.	ЛРН№1–ЛРН№14	Вопрос на экзамене №21–32
	ОПК-2.3. Иметь навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	Знать принципы объектно-ориентированного программирования и их реализацию в C++	ЛРН№1–ЛРН№14	Вопрос на экзамене №27–32
		Уметь создавать программы, пригодные для практического применения	ЛРН№1–ЛРН№14	Вопрос на экзамене №27–32
		Владеть навыками разработки алгоритмов и программ на C++ с применением объектно-ориентированного подхода.	ЛРН№1–ЛРН№14	Вопрос на экзамене №27–32
	ОПК-3.1. Знать принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Принципы и методы решения стандартных алгоритмических задач. Интегрированные среды разработки (IDE), системы контроля версий (Git), средства отладки и профилирования. Основные требования информационной безопасности при разработке на C++: защита от переполнения буфера, утечек памяти, некорректного ввода. Основы информационной и библиографической культуры при поиске научно-технической информации по алгоритмам и структурам данных.	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32

		Использовать IDE, Git, отладчики для решения стандартных задач программирования на C++. Учитывать требования информационной безопасности при написании кода	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
		Владеть навыками работы в среде разработки C++. Навыками использования Git для управления версиями программ. Навыками безопасного программирования.	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
	ОПК-3.2. Уметь решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Методы и алгоритмы решения типовых задач обработки данных. Основы применения ИКТ для автоматизации вычислений и отладки. Требования к защите информации при передаче и хранении данных в программах на C++	ЛРН№1–ЛРН№14	Вопрос на экзамене №21–32
		Решать стандартные алгоритмические задачи, реализуя их на C++	ЛРН№1–ЛРН№14	Вопрос на экзамене №21–32
		Владеть навыками практической реализации алгоритмов сортировки, поиска и структур данных на C++. Навыками использования отладочных средств IDE для выявления ошибок времени выполнения.	ЛРН№1–ЛРН№14	Вопрос на экзамене №21–32
	ОПК-3.3. Иметь навыки подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности	Основы научно-исследовательской работы в области программирования на C++. Правила обеспечения информационной безопасности при подготовке и распространении материалов: защита авторского кода, корректное цитирование, неразглашение критической информации.	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
		Составлять отчеты по лабораторным работам, включающие описание алгоритмов, листинги программ, результаты экспериментов и выводы.	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
		Навыками оформления технической документации к программам на C++.	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32

Направление 11.03.02 Инфокоммуникационные технологии и системы связи

№ п/п	Код и наименование индикатора	Результаты обучения	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
	ОПК-3.1. Знает основные закономерности и принципы передачи информации в	Знать способы представления информации в цифровом виде, типы	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32

	инфокоммуникационных системах, основные виды сигналов, используемых в телекоммуникационных системах, особенности передачи различных сигналов по каналам и трактам телекоммуникационных систем	данных и их размеры в C++.		
		Уметь проектировать решение конкретной задачи: формулировать требования, выбирать структуры данных и алгоритмы, обосновывать выбор.	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
		Владеть навыками проектирования программ на C++ с использованием объектно-ориентированного подхода. Навыками выбора оптимальных алгоритмов и структур данных для заданных условий. Навыками оценки эффективности алгоритмов	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
	ОПК-3.2. Способен решать задачи обработки данных с помощью средств вычислительной техники	Знать классификацию, свойства и области применения основных структур данных. Базовые алгоритмы. Понятия сложности алгоритмов (О-нотация) и критерии выбора оптимальной структуры данных для конкретной задачи	ЛРН№1–ЛРН№14	Вопрос на экзамене №21–32
		Уметь анализировать входные данные и требования задачи для обоснованного выбора структуры данных и алгоритма	ЛРН№1–ЛРН№14	Вопрос на экзамене №21–32
		Владеть навыками реализации динамических структур данных с помощью указателей и классов.	ЛРН№1–ЛРН№14	Вопрос на экзамене №21–32
	ОПК-3.3. Владеет методами и навыками обеспечения информационной безопасности	Знать принципы объектно-ориентированного программирования и их реализацию в C++	ЛРН№13–ЛРН№14	Вопрос на экзамене №27–32
		Уметь создавать программы, пригодные для практического применения	ЛРН№13–ЛРН№14	Вопрос на экзамене №27–32
		Владеть навыками разработки алгоритмов и программ на C++ с применением объектно-ориентированного подхода.	ЛРН№13–ЛРН№14	Вопрос на экзамене №27–32
	ОПК-4.1. Знает современные интерактивные программные комплексы и основные приемы обработки экспериментальных данных, в том числе с использованием	Знать методы и алгоритмы обработки числовых и текстовых данных на C++. Возможности средств вычислительной техники и их влияние на выбор алгоритма и	ЛРН№1–ЛРН№12	Вопрос на экзамене №1–26

	стандартного программного обеспечения, пакетов программ общего и специального назначения	структуры данных. Принципы организации ввода-вывода и хранения данных в файлах.		
		Уметь решать задачи обработки данных с использованием языка C++ и средств вычислительной техники (выбор оптимального алгоритма по времени и памяти). Использовать стандартные возможности C++ для работы с массивами, строками, файлами. Проводить анализ эффективности программ.	ЛРН№1–ЛРН№12	Вопрос на экзамене №1–26
		Владеть навыками реализации алгоритмов обработки данных на C++. Навыками оценки временной и пространственной сложности программ. Навыками работы с файловой системой для хранения и чтения данных..	ЛРН№1–ЛРН№12	Вопрос на экзамене №1–26
	ОПК-4.2. Использует возможности вычислительной техники и программного обеспечения для решения задач управления и алгоритмизации процессов обработки информации	Знать принципы безопасного программирования на C++: использование проверок границ, корректное освобождение памяти, обработка исключительных ситуаций	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
		Уметь применять методы обеспечения информационной безопасности при разработке программ на C++: проверять входные данные, избегать переполнения буфера, правильно управлять динамической памятью.	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
		Владеть навыками безопасного программирования на C++ (предотвращение утечек, переполнений)..	ЛРН№1–ЛРН№14	Вопрос на экзамене №1–32
	ОПК-4.3. Владеет методами компьютерного моделирования физических процессов при передаче информации, техникой инженерной и компьютерной графики	Знать методологию проектирования программного решения: анализ требований, выбор архитектуры, разработка алгоритмов. Критерии	ЛРН№11–ЛРН№14	Вопрос на экзамене №21–32

		оптимальности: временная и пространственная сложность, простота реализации, надежность.		
		Моделировать поведение алгоритмов на тестовых наборах данных с визуализацией промежуточных состояний.	ЛРН№11–ЛРН№14	Вопрос на экзамене №21–32
		Владеть навыками проектирования программ на С++ с использованием объектно- ориентированного подхода (классы, инкапсуляция). Навыками выбора оптимальных алгоритмов и структур данных для заданных условий.	ЛРН№11–ЛРН№14	Вопрос на экзамене №21–32
	ОПК-5.1. Проектирует решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений	Знать синтаксис и семантику языка С++, включая указатели, ссылки, динамическое выделение памяти, управляющие конструкции, функции, перегрузку, классы и наследование. Современные инструменты разработки и принципы написания качественного кода .	ЛРН№1–ЛРН№10	Вопрос на экзамене №1–20
		Уметь писать программный код с учётом эффективности и надёжности	ЛРН№1–ЛРН№10	Вопрос на экзамене №1–20
		Владеть навыками практического программирования на С++ для создания работающего программного кода.	ЛРН№1–ЛРН№10	Вопрос на экзамене №1–20

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы
Операторы цикла

Вывести на экран таблицу значений функции $Y(x)$ и ее разложения в ряд $S(x)$ для x , изменяющегося от a до b с шагом $h = (b - a)/10$, табл. 3.1.

Таблица 3.1.

№	a	b	$S(x)$	n	$Y(x)$
1	2	3	4	5	6
1	0.1	1	$x - \frac{x^3}{3!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$	160	$\sin x$
2	0.1	1	$1 + \frac{x^2}{2!} + \dots + \frac{x^{2n}}{(2n)!}$	100	$\frac{e^x + e^{-x}}{2}$

Использование условных операторов

При выполнении задания предусмотреть выбор вида функции $f(x)$: $\sin(x)$, x^2 или e^x . Предусмотреть вывод информации о выбранной ветви вычислений.

$$\begin{aligned}
 1. \quad a &= \begin{cases} (f(x)+y)^2 - \sqrt[3]{f(x)}, & xy > 0 \\ (f(x)+y)^2 + \sin(x), & xy < 0 \\ (f(x)+y)^2 + y^3, & xy = 0. \end{cases} & 2. \quad b &= \begin{cases} \ln(f(x)) + \sqrt[3]{f(x)}, & x/y > 0 \\ \ln|f(x)/y| \cdot (x+y)^3, & x/y < 0 \\ (f(x)^2 + y)^3, & \text{иначе.} \end{cases} \\
 3. \quad c &= \begin{cases} f(x)^2 + \sqrt[3]{y} + \sin(y), & x - y = 0 \\ (f(x) - y)^2 + \ln(x), & x - y > 0 \\ (y - f(x))^2 + \operatorname{tg}(y), & x - y < 0. \end{cases} & 4. \quad d &= \begin{cases} \sqrt[3]{f(x) - y} + \operatorname{tg}(f(x)), & x > y \\ (y - f(x))^3 + \cos(f(x)), & y < x \\ (y + f(x))^2 + x^3, & y = x. \end{cases} \\
 5. \quad e &= \begin{cases} y\sqrt{f(x)} + 3\sin(x), & x > y \\ x\sqrt{f(x)}, & x < y \\ \sqrt[3]{f(x)} + x^3/y, & \text{иначе.} \end{cases} & 6. \quad g &= \begin{cases} e^{f(x)+y}, & 0.5 < xy < 10 \\ \sqrt[3]{f(x)+y}, & 0.1 < xy < 0.5 \\ 2f(x)^2, & \text{иначе.} \end{cases} \\
 7. \quad s &= \begin{cases} e^{f(x)}, & 1 < xb < 10 \\ \sqrt[3]{f(x)+4y}, & 12 < xb < 40 \\ y \cdot f(x)^2, & \text{иначе.} \end{cases} & 8. \quad b &= \begin{cases} (f(x)^2 + y)^3, & x/y < 0 \\ \ln|f(x)/y| + x/y, & x/y > 0 \\ \sqrt[3]{\sin(y)}, & \text{иначе.} \end{cases} \\
 9. \quad l &= \begin{cases} 2f(x)^3 + 3y^2, & x > |y| \\ |f(x) - y|, & 3 < x < |y| \\ \sqrt[3]{f(x) - y}, & \text{иначе.} \end{cases} & 10. \quad k &= \begin{cases} \ln(|f(x)| + |y|), & |xy| > 10 \\ e^{f(x)+y}, & |xy| < 10 \\ \sqrt[3]{f(x)} + y, & |xy| = 10. \end{cases}
 \end{aligned}$$

Работа с файлами

В программе предусмотреть сохранение вводимых данных в файл и возможность чтения из ранее сохраненного файла. Результаты выводить на экран и в текстовый файл.

1. Список товаров, имеющихся на складе, включает в себя наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести список товаров, хранящихся больше месяца и стоимость которых превышает 1 000 000 р.

2. Для получения места в общежитии формируется список студентов, который включает ФИО студента, группу, средний балл, доход на члена семьи. Вывести информацию о студентах, у которых доход на члена семьи менее двух минимальных зарплат.

3. В справочной автовокзала хранится расписание движения автобусов. Для каждого рейса указаны его номер, пункт назначения, время отправления и прибытия. Вывести информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени.

Работа с многомерными массивами

В работе память для массива должна выделяться динамически. На экран выводить исходные данные и результат.

1. Задана матрица размером $N \times M$. Получить массив B , присвоив его k -му элементу значение 0, если все элементы k -го столбца матрицы нулевые, и значение 1 – в противном случае.

2. Задана матрица размером $N \times M$. Получить массив B , присвоив его k -му элементу значение 1, если элементы k -й строки матрицы упорядочены по убыванию, и значение 0 – в противном случае.

3. Задана матрица размером $N \times M$. Получить массив B , присвоив его k -му элементу значение 1, если k -я строка матрицы симметрична, и значение 0 – в противном случае.

4. Задана матрица размером $N \times M$. Определить количество «особых» элементов матрицы, считая элемент «особым», если он больше суммы остальных элементов своего столбца.

5. Задана матрица размером $N \times M$. Определить количество «особых» элементов матрицы, считая элемент «особым», если в строке слева от него находятся элементы, меньшие его, а справа – большие.

6. Задана символьная матрица размером $N \times M$. Определить количество различных элементов матрицы (т.е. повторяющиеся элементы считать один раз).

7. Дана матрица размером $N \times M$. Упорядочить ее строки по возрастанию их первых элементов.

8. Дана матрица размером $N \times M$. Упорядочить ее строки по возрастанию суммы их элементов.

9. Дана матрица размером $N \times M$. Упорядочить ее столбцы по возрастанию их наименьших элементов.

10. Определить, является ли заданная квадратная матрица n -го порядка симметричной относительно побочной диагонали.

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)

1. Алгоритм и его свойства. Формы представления алгоритмов.
2. Оценка вычислительной сложности алгоритмов. Асимптотические обозначения O , Ω , Θ .
3. Структура программы на C++. Этапы создания исполняемой программы (препроцессинг, компиляция, сборка).

4. Элементы языка C++: переменные, базовые типы данных (целые, вещественные, символьные, логические). Примеры.
5. Арифметические и логические операции. Приоритет операций. Примеры.
6. Директивы препроцессора C++: #include, #define, #undef, условная компиляция.
7. Математические функции. Библиотеки <math.h> и <cmath>. Примеры.
8. Поточный ввод/вывод в C++ (cin, cout). Примеры.
9. Управляющие структуры: условные операторы if, if-else, switch. Примеры.
10. Операторы цикла: for, while, do...while. Примеры. Организация вложенных циклов.
11. Программирование с использованием статических одномерных и многомерных массивов. Примеры.
12. Программирование с использованием динамических массивов. Операторы new и delete. Примеры.
13. Указатели и ссылки. Арифметика указателей. Передача параметров по указателю и по ссылке.
14. Модель памяти C/C++: стек, куча (динамическая память). Управление памятью.
15. Строки как массивы символов. Библиотека <cstring> (strlen, strcpy, strcmp, strcat). Примеры.
16. Сложные типы данных: структуры (struct). Вложенные структуры. Примеры.
17. Сложные типы данных: объединения (union). Перечисления (enum, enum class). Примеры.
18. Работа с файлами в C++ (fstream, ifstream, ofstream). Открытие, чтение, запись текстовых и бинарных файлов. Примеры.
19. Функции в C++. Определение, вызов, передача параметров (по значению, по ссылке, по указателю). Перегрузка функций. Параметры по умолчанию.
20. Рекурсия. Прямой и косвенный рекурсивный вызов. Базовый случай и рекурсивный шаг. Стек вызовов. Примеры.
21. Алгоритмы сортировки: пузырьковая, вставками, выбором (принципы, реализация, оценка сложности).
22. Алгоритмы сортировки: быстрая сортировка и сортировка слиянием. Сравнение эффективности.
23. Методы поиска: линейный (последовательный) поиск.
24. Методы поиска: бинарный поиск (рекурсивная и итеративная версии) на отсортированном массиве. Интерполяционный поиск.
25. Сортировка массива структур по ключевому полю. Примеры.
26. Поиск по ключу в массиве структур. Примеры.
27. Абстрактные типы данных. Классы и объекты в C++. Спецификаторы доступа (private, protected, public).
28. Конструкторы. Деструкторы. Статические члены класса. Дружественные функции и классы.
29. Правило трёх (конструктор копирования, оператор присваивания, деструктор) для классов, управляющих динамической памятью.
30. Наследование. Одиночное наследование. Базовый и производный классы. Порядок вызова конструкторов и деструкторов.
31. Полиморфизм. Виртуальные функции. Абстрактные классы и чистые виртуальные функции. Виртуальные деструкторы.
32. Перегрузка операторов (как член класса и как внешняя функция). Перегрузка унарных и бинарных операторов. Перегрузка операторов ввода/вывода.

В соответствии с «Положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся в Кубанском государственном университете и его филиалов» форма проведения экзамена устанавливается преподавателем и доводится до сведения студентов в начале семестра.

Экзамен является формой контроля усвоения студентом учебной программы по дисциплине или ее части, выполнения практических работ.

Результат сдачи экзамена по прослушанному курсу должны оцениваться как итог деятельности студента в семестре, а именно - по посещаемости лекций, результатам работы на практических занятиях, выполнения самостоятельной работы. При этом допускается на очной форме обучения пропуск не более 20% занятий, с обязательной отработкой пропущенных занятий. Студенты, у которых количество пропусков, превышает установленную норму, не выполнившие все виды работ и неудовлетворительно работавшие в течение семестра, проходят собеседование с преподавателем, который опрашивает студента на предмет выявления знания основных положений дисциплины.

Для получения положительной оценки по итогам семестра необходимо минимум выполнение следующих условий: выполнение и успешная защита всех лабораторных и практических занятий, а так же посещение 80% лекционных, лабораторных и практических занятий.

Экзамен оценивается, исходя из следующих критериев:

- Полнота ответа, последовательность и логика изложения;
- Правильность ответа, его соответствие рабочей программе учебной дисциплины;
- Действенность знаний, способность студента аргументировать свой ответ и приводить примеры;
- Осознанность излагаемого материала;
- Соответствие нормам культуры речи;
- Самостоятельность;
- Качество ответов на вопросы.

Критерии оценки:

Оценка	Критерии оценивания
отлично	ставится в том случае, когда студент обнаруживает систематическое и глубокое знание программного материала по дисциплине, умеет свободно ориентироваться в вопросе выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной программой. Ответ полный и правильный на основании изученного материала. Выдвинутые положения аргументированы и иллюстрированы примерами. Материал изложен в определенной логической последовательности, осознанно, литературным языком, с использованием современных научных терминов; ответ самостоятельный. Студент уверенно отвечает на дополнительные вопросы. Как правило, оценка "отлично" выставляется студентам, усвоившим взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.
хорошо	ставится в том случае, когда студент обнаруживает полное знание учебного материала, демонстрирует систематический характер знаний по дисциплине. Ответ полный и правильный, подтвержден примерами; но их обоснование не аргументировано, отсутствует собственная точка зрения. Также студент успешно выполняет предусмотренные в программе задания и усвоивший основную литературу, рекомендованную в программе. Материал изложен в определенной логической

	последовательности, при этом допущены 2-3 несущественные погрешности, исправленные по требованию экзаменатора. Студент испытывает незначительные трудности в ответах на дополнительные вопросы. Материал изложен осознанно, самостоятельно, с использованием современных научных терминов, литературным языком. Как правило, оценка "хорошо" выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.
удовлетворительно	ставится в том случае, когда студент обнаруживает знание основного программного материала по дисциплине, но допускает погрешности в ответе. Ответ недостаточно логически выстроен, самостоятелен. Основные понятия употреблены правильно, но обнаруживается недостаточное раскрытие теоретического материала. Выдвигаемые положения недостаточно аргументированы и не подтверждены примерами; ответ носит преимущественно описательный характер. Студент испытывает достаточные трудности в ответах на вопросы. Научная терминология используется недостаточно. Как правило, оценка "удовлетворительно" выставляется студентам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.
неудовлетворительно	выставляется студенту, обнаружившему пробелы в знаниях основного учебного материала по дисциплине. При ответе обнаружено непонимание студентом основного содержания теоретического материала или допущен ряд существенных ошибок, которые студент не может исправить при наводящих вопросах экзаменатора, затрудняется в ответах на вопросы. Студент подменил научное обоснование проблем рассуждением бытового плана. Ответ носит поверхностный характер; наблюдаются неточности в использовании научной терминологии. Также у студента имеются пробелы в знаниях основного учебно-программного материала, он допускает принципиальные ошибки в выполнении предусмотренных программой заданий. Как правило, оценка "неудовлетворительно" ставится студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании учёбы без дополнительных занятий по соответствующей дисциплине.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Перечень учебной литературы, информационных ресурсов и технологий

5.1. Учебная литература

1. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. - Москва : Юрайт, 2022. - 137 с. - URL: <https://urait.ru/bcode/491215>. - Режим доступа: для авториз. пользователей. - ISBN 978-5-534-07834-3. - Текст : электронный.

2. Конова, Е. А. Алгоритмы и программы. Язык С++ : учебное пособие для вузов / Е. А. Конова, Г. А. Поллак. — 6-е изд., стер. — Санкт-Петербург : Лань, 2021. — 384 с. — ISBN 978-5-8114-8487-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/176900> — Режим доступа: для авториз. пользователей.

3. Дейл, Н. Программирование на С++ [Электронный ресурс] / Н. Дейл, Ч. Уимз, М. Хедингтон; Пер. с англ. - Москва : ДМК Пресс, 2007. - 672 с.: ил. - (Серия «Учебник»). - ISBN 5-93700-008-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/407353> – Режим доступа: по подписке.

4. Огнева, М. В. Программирование на языке С++: практический курс : учебник для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — URL: <https://urait.ru/course/CDA0B311-FA79-41E8-BB3E-97ABF8B25877>

Для освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья имеются издания в электронном виде в электронно-библиотечных системах «Лань», «Юрайт», "Университетская библиотека ONLINE", "ZNANIUM.COM".

Дополнительная литература:

1. Седжвик, Роберт. Алгоритмы на С++ [Текст] : анализ, структуры данных, сортировка, поиск, алгоритмы на графах / Роберт Седжвик ; [при участии Кристофера Дж. Ван Вика] ; [пер. с англ. А. А. Моргунова]. — Москва [и др.] : Вильямс, 2016. — 1056 с. : ил., табл. ; 25 см. — ISBN 978-5-8459-2070-6.

2. Лафоре Р. Объектно-ориентированное программирование в С++. Классика Computer Science / Р. Лафоре ; пер. с англ. А. Кузнецов, М. Назаров, В. Шрага. — 4-е изд. — Санкт-Петербург : Питер, 2021. — 923 с.

3. Павловская, Татьяна Александровна. С/С++. Процедурное и объектно-ориентированное программирование : учебник для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" / Т. А. Павловская. - Санкт-Петербург [и др.] : Питер, 2019. - 495 с. - (Учебник для вузов) (Стандарт третьего поколения) (Для бакалавров и специалистов). - Библиогр.: с. 493-495. - ISBN 978-5-4461-0860-2 : 1820 р. - Текст : непосредственный.

4.

5.2. Периодическая литература

Указываются печатные периодические издания из «Перечня печатных периодических изданий, хранящихся в фонде Научной библиотеки КубГУ» <https://www.kubsu.ru/ru/node/15554>, и/или электронные периодические издания, с указанием адреса сайта электронной версии журнала, из баз данных, доступ к которым имеет КубГУ:

1. Базы данных компании «Ист Вью» <http://dlib.eastview.com>
2. Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

5.3. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» <https://urait.ru/>
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» www.biblioclub.ru
3. ЭБС «BOOK.ru» <https://www.book.ru>
4. ЭБС «ZNANIUM.COM» www.znanium.com
5. ЭБС «ЛАНЬ» <https://e.lanbook.com>

Профессиональные базы данных:

1. Web of Science (WoS) <http://webofscience.com/>
2. Scopus <http://www.scopus.com/>
3. ScienceDirect www.sciencedirect.com
4. Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. <http://www.machinelearning.ru>
5. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
6. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
7. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
8. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
9. Президентская библиотека им. Б.Н. Ельцина <https://www.prilib.ru/>
10. Электронная коллекция Оксфордского Российского Фонда <https://ebookcentral.proquest.com/lib/kubanstate/home.action>
11. Springer Journals <https://link.springer.com/>
12. Nature Journals <https://www.nature.com/siteindex/index.html>
13. Springer Nature Protocols and Methods <https://experiments.springernature.com/sources/springer-protocols>
14. Springer Materials <http://materials.springer.com/>
15. zbMath <https://zbmath.org/>
16. Nano Database <https://nano.nature.com/>
17. Springer eBooks: <https://link.springer.com/>
18. "Лекториум ТВ" <http://www.lektorium.tv/>
19. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

Информационные справочные системы:

1. Консультант Плюс - справочная правовая система (доступ по локальной сети с компьютеров библиотеки)

Ресурсы свободного доступа:

1. Справочник по C++ <https://cppreference.com/>
2. Полные тексты канадских диссертаций <http://www.nlc-bnc.ca/thesescanada/>
3. КиберЛенинка (<http://cyberleninka.ru/>);
4. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
5. Федеральный портал "Российское образование" <http://www.edu.ru/>;
6. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
7. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
8. Федеральный центр информационно-образовательных ресурсов (<http://fcior.edu.ru/>);
9. Образовательный портал "Учеба" <http://www.ucheba.com/>;
10. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--plai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ:

1. Среда модульного динамического обучения <http://moodle.kubsu.ru>
www.openedu.kubsu.ru
2. База учебных планов, учебно-методических комплексов, публикаций и конференций <http://mschool.kubsu.ru/>
3. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru>;
4. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
5. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" <http://icdau.kubsu.ru/>

6. Методические указания для обучающихся по освоению дисциплины (модуля)

Основными формами аудиторных занятий по дисциплине являются лекции, лабораторные работы и самостоятельная работа.

Успешное освоение дисциплины «Информатика и теория алгоритмов» требует систематической и целенаправленной самостоятельной работы. Дисциплина сочетает в себе фундаментальные основы алгоритмизации, глубокое понимание структур данных и методов их реализации, а также практические навыки программирования на языке C++ с ручным управлением памятью, поэтому для её освоения необходимо планировать время: регулярно выделять время для изучения теоретического материала, выполнения лабораторных работ и подготовки к контрольным мероприятиям; сочетать теорию и практику: изучение каждой новой темы должно сопровождаться практической реализацией на языке C++ в интегрированной среде разработки) с использованием указателей, динамической памяти; вести конспекты и сохранять код: фиксировать ключевые определения, блок-схемы алгоритмов, примеры реализации структур данных (списки, стеки, очереди, деревья) и алгоритмов сортировки и поиска, а также сохранять все версии кода в репозитории — это поможет при подготовке к зачёту и защите лабораторных работ; использовать электронные ресурсы: активно работать с литературой из списка, справочными материалами по языку C++, а также с открытыми онлайн-курсами и платформами для тренировки алгоритмического мышления (Codeforces, Stepik, cppreference.com); работать над сквозным проектом: все лабораторные работы объединены в единый сквозной проект, поэтому важно выполнять их последовательно, не пропуская этапов, чтобы к концу семестра иметь законченное работающее решение, например,

библиотеку контейнерных классов или программу для обработки структурированных данных.

Лекции следует проводить в лекционных аудиториях с использованием средств мультимедиа для демонстрации динамических структур данных, визуализации алгоритмов и работы с памятью. При подготовке отдельных вопросов лекций или лекций по определённым темам учебной программы рекомендуется активно привлекать студентов, реализуя такие виды интерактивных образовательных технологий, как «Студент в роли преподавателя» и «Работа в малых группах», что позволяет развивать навыки объяснения сложных концепций и командного обсуждения алгоритмических решений.

Лекции являются основной формой изложения теоретического материала. Для эффективного усвоения лекционного материала рекомендуется повторить базовые понятия из предыдущих лекций, так как материал дисциплины имеет логическую взаимосвязь: от основ алгоритмизации через структуры данных к объектно-ориентированному программированию. При необходимости изучить теоретические основы алгоритмов и структур данных по учебным пособиям, а также освежить в памяти принципы работы с указателями, динамической памятью и модели памяти C/C++.

Во время лекции необходимо внимательно слушать преподавателя, фиксировать основные определения, блок-схемы алгоритмов и примеры реализации структур данных на C++. Записывать примеры применения алгоритмов сортировки, поиска, работы со списками, стеками, очередями и деревьями в реальных задачах программирования. При возникновении вопросов задавать их преподавателю, не откладывая на потом, поскольку пропуск даже одного логического элемента может затруднить понимание последующих тем. Обращать особое внимание на связь лекционного материала с предстоящими лабораторными работами, чтобы понимать, какие алгоритмы и структуры данных предстоит реализовать самостоятельно на языке C++.

Лабораторные работы по дисциплине «Информатика и теория алгоритмов» следует проводить в аудиториях, оснащённых соответствующим мультимедийным оборудованием и персональными компьютерами с выходом в сеть Интернет. Лабораторные работы направлены на формирование практических навыков разработки алгоритмов и программ на языке C++ с использованием современных инструментов разработки и систем контроля версий. Все лабораторные работы выполняются в интегрированной среде разработки (IDE) на языке C++ без использования стандартной библиотеки шаблонов (STL), с акцентом на ручное управление памятью и реализацию структур данных «с нуля».

Общие рекомендации: перед выполнением лабораторной работы внимательно изучить её описание, цель и задачи, а также ознакомиться с теоретическим материалом, соответствующим теме работы (лекции, основная и дополнительная литература). Рекомендуется создать отдельный каталог для каждой лабораторной работы и вести репозиторий на GitHub/GitLab/GitVerse для сохранения всех версий кода и результатов.

В процессе выполнения лабораторной работы необходимо следовать пошаговой инструкции, фиксируя промежуточные результаты, и экспериментировать с параметрами реализуемых алгоритмов и структур данных, сравнивая их эффективность по времени выполнения и объёму используемой памяти. Результаты работы рекомендуется визуализировать с помощью графиков или таблиц, отражающих зависимость времени выполнения от объёма входных данных, а также комментировать код, чтобы было понятно назначение каждого блока и логика работы программы. Все версии кода и промежуточные результаты следует сохранять в репозитории (GitHub/GitLab/GitVerse) для последующего анализа и защиты.

После выполнения лабораторной работы необходимо оформить отчёт по установленной форме, включающий цель работы, описание реализованных методов и алгоритмов, выводы о проделанной работе и ответы на контрольные вопросы. Для защиты работы студент должен подготовить краткое представление проделанной работы, продемонстрировать код и полученные результаты, а также ответить на контрольные

вопросы преподавателя, демонстрируя понимание теоретических основ и практических аспектов реализации алгоритмов и структур данных на C++.

Структура дисциплины определяет следующие виды самостоятельной работы: самостоятельная работа студента (СРС) и контроль.

Самостоятельная работа студента является основным видом самостоятельной работы. Она проводится в целях закрепления знаний, полученных на всех видах учебных занятий, а также расширения и углубления знаний, т.е. активного приобретения студентами новых знаний.

СРС включает проработку и повторение лекционного материала. Для этого студенту рекомендуется прочитать текст лекции, пересказать его вслух, воспроизвести самостоятельно имеющиеся в тексте структурно-логические схемы, диаграммы, математические выкладки формул, доказательства теорем и т.п. Проработку лекционного материала следует проводить сначала последовательно, по каждому учебному вопросу, а затем повторно, по всему тексту лекции.

СРС также включает изучение материала по рекомендованным учебникам и учебным пособиям. Так как существует огромное количество учебной литературы, то для этого вида подготовки необходимо предварительное указание преподавателя. Преподаватель должен выступать здесь в роли опытного «путеводителя», определяя последовательность знакомства с литературными источниками и «глубину погружения» в каждый из них.

Одним из видов СРС является подготовка к лабораторным работам. Преподаватель накануне очередного занятия обозначает для студентов круг теоретического материала, необходимого для выполнения лабораторной работы. Студенты прорабатывают его. Затем, уже в аудитории, перед выполнением заданий, преподаватель производит контрольный опрос студентов. Это позволяет определить степень готовности группы по данной теме и скорректировать ход занятия.

Экзамен – вид итогового контроля усвоения содержания учебной дисциплины.

Для получения положительной оценки по экзамену студент сдаёт устный экзамен. На экзамене студент выбирает из разложенных (вопросы и задания скрыты) перед ним билет, который включает два вопроса, если не сданы лабораторные работы то плюс одно практическое задание. Студент, согласно «положения о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся в КубГУ» имеет право выбрать билет повторно, но со снижением полученной в последствии оценкой на один бал.

Сначала студенту дается возможность подготовиться, заготовив себе на чистом маркированном листе план и подсказки к ответу, записать решение задачи, в течение полутора часов после получения билета, при этом запрещено пользоваться студенту ни какими литературными, электронными и другими источниками информации, кроме собственных знаний. После подготовки, студент отвечает на вопросы по билету, а так же на дополнительные вопросы экзаменатора, показывает решенную задачу.

Если студент не сдал лабораторные работы, то после ответа на теоретические вопросы студенту даётся отдых не более двух часов, после которого он приступает к выполнению практической части задания по билету. На выполнение практической части задания студенту отводится два часа. По прошествии этих двух часов проверяется выполнение практического задания.

Решение об оценке принимается исходя из того, что студент должен был освоить теорию гораздо шире, нежели контролируют эти вопросы, а экзаменатор руководствуется «положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся в КубГУ».

Рекомендации по оцениванию лабораторных работ

В целях закрепления практического материала и углубления теоретических знаний по разделам дисциплины предполагается выполнение лабораторных работ, что позволяет

углубить процесс познания, раскрыть понимание прикладной значимости осваиваемой дисциплины. Комплект заданий репродуктивного уровня для выполнения на лабораторных занятиях, позволяющих оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, распознавание объектов изучения в рамках определенного раздела дисциплины. Лабораторные работы проводятся для получения навыков практического программирования на языке С и С++.

Консультации, выдача лабораторных заданий и прием результатов выполнения осуществляется только во время аудиторных занятий. Задания выполняются последовательно. Правильное выполнение некоторых заданий возможно только, если студент корректно выполнил предыдущие задания.

Поэтому приступать к следующему заданию студент может, только сдав преподавателю результат выполнения предыдущего. Результаты выполнения лабораторных работ демонстрируются преподавателю. Во время приема выполненной работы преподаватель вправе:

- 1) Требовать у студента демонстрации выполнения программного проекта, предусмотренной заданием.
- 2) Самостоятельно производить манипуляции с программным проектом и средой программирования, не изменяя программы, составленной студентом.
- 3) Требовать у студента пояснений, относящихся к исходному коду и способам реализации программы.

Задание считается выполненным и принимается преподавателем только в том случае, если реализован весь функционал, предусмотренный заданием.

Если какие-то функции, предусмотренные заданием, не работают, или работают неверно, то результат выполнения подлежит доработке. Студент должен работать над кодом программы максимально самостоятельно, использовать отладочные средства, предоставляемые изучаемой программной средой.

До конца семестра студент должен сдать результаты выполнения всех лабораторных работ, предусмотренных настоящей программой.

Критерии оценки лабораторных работ

Оценка	Критерии оценивания
10-12 баллов	Задание выполнено полностью, в представленном отчете обоснованно получено правильное выполненное задание.
8 балла	Задание выполнено полностью, но нет достаточного обоснования или при верном решении допущена незначительная ошибка, не влияющая на правильную последовательность рассуждений.
6 балла	Задания выполнены частично.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

7. Материально-техническое обеспечение по дисциплине (модулю)

По всем видам учебной деятельности в рамках дисциплины используются аудитории, кабинеты и лаборатории, оснащенные необходимым специализированным и лабораторным оборудованием.

При заполнении таблицы учитывать все виды занятий, предусмотренные учебным планом по данной дисциплине: лекции, занятия семинарского типа (практические занятия, лабораторные работы), а также курсовое проектирование, консультации, текущий контроль и промежуточную аттестацию.

Наименование специальных помещений	Оснащенность специальных помещений	Перечень лицензионного программного обеспечения
Учебные аудитории для проведения занятий лекционного типа: 201с, 300с	Мебель: учебная мебель Технические средства обучения: Лекционная аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) и соответствующим программным обеспечением (ПО) для воспроизведения файлов формата jpg и avi, достаточным количеством посадочных мест.	
Учебные аудитории для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации 207с, 217с, 213с, 133с, 142с	Мебель: учебная мебель Технические средства обучения: экран, проектор, компьютер	Аудитория, укомплектованная специализированными техническими средствами обучения – компьютерный класс, с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета
Учебные аудитории для проведения лабораторных работ. Лаборатория 207с, 217с, 213с, 133с, 142с	Лаборатория, укомплектованная специализированной мебелью и техническими средствами обучения	Аудитория, укомплектованная специализированными техническими средствами обучения – компьютерный класс, с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Наименование помещений для самостоятельной работы обучающихся	Оснащенность помещений для самостоятельной работы обучающихся	Перечень лицензионного программного обеспечения
Помещение для самостоятельной работы обучающихся (читальный зал Научной библиотеки)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры,	Доступ печатным и электронным Информационным ресурсам

	коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	
Помещение для самостоятельной работы обучающихся (ауд.208с)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	