

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ
Проректор по учебной работе,
качеству образования – первый
проректор
Хагуров Т.А.
подпись



«30» мая 2025

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ **Б1.О.13«Основы программирования»**

Направление подготовки 02.03.03 Математическое обеспечение и
администрирование информационных систем

Направленность (профиль) Технологии разработки программных систем

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Основы программирования» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем.

Программу составил(и):

Харченко Анна Владимировна, доцент, канд. пед наук

И.О. Фамилия, должность, ученая степень, ученое звание



подпись

Рабочая программа дисциплины «Основы программирования» утверждена на заседании кафедры информационных технологий протокол №15 от «14» мая 2025г.

Заведующий кафедрой (разработчика)

В. В. Подколзин



подпись

Рабочая программа обсуждена на заседании кафедры информационных технологий протокол №15 от «14» мая 2025г.

Заведующий кафедрой (выпускающей)

В. В. Подколзин



подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол №4 от «23» мая 2025 г.

Председатель УМК факультета

А. В. Коваленко



подпись

Рецензенты:

Бегларян М. Е., Проректор по учебной работе, Краснодарский кооперативный институт (филиал) АНО ВО Центросоюза РФ «Российский университет кооперации»

Рубцов Сергей Евгеньевич, кандидат физико-математических наук, доцент кафедры математического моделирования ФГБОУ ВО «КубГУ»

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Формирование у студентов базовых знаний в области современного программирования, включая основные понятия, принципы разработки алгоритмов, их реализацию на языке программирования, а также методы тестирования и отладки программного кода.

Воспитательной целью дисциплины является развитие у студентов научного и творческого мышления в процессе освоения технологий, методов и инструментов разработки программного обеспечения. Особое внимание уделяется изучению принципов взаимодействия между приложениями и работе в современных средах разработки.

Содержание дисциплины базируется на актуальных научных и практических знаниях, включая: технологии разработки программного обеспечения; парадигмы и современные методики программирования.

Научной основой дисциплины является теоретико-прагматический подход в обучении.

1.2 Задачи дисциплины

- изучение базовых конструкций и концепций языков программирования;
- освоение принципов организации типов и структур данных;
- анализ и применение базовых алгоритмов обработки информации;
- изучение методов оценки сложности алгоритмов;
- развитие навыков самостоятельного анализа и решения задач с использованием классических алгоритмических подходов.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Основы программирования» относится к «Обязательная часть» Блока 1 «Дисциплины (модули)» учебного плана.

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:

ОПК-2 Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности

ИД-3.ОПК-2 Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности

Знать Возможности существующей программно-технической архитектуры
Возможности современных и перспективных средств разработки программных продуктов, технических средств
Методологии разработки программного обеспечения и технологии программирования
Языки программирования и работы с базами данных
Инструменты и методы проектирования и дизайна ИС
Цели и задачи проводимых исследований и разработок
Методы и средства планирования и организации исследований и разработок
Методы проведения экспериментов и наблюдений, обобщения и обработки информации

- Уметь** *Вырабатывать варианты реализации требований*
Использовать существующие типовые решения и шаблоны проектирования программного обеспечения
Верифицировать структуру программного кода
- Владеть** *Разработка, изменение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения*
Разработка структуры программного кода ИС
Верификация структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС
Сбор, обработка, анализ и обобщение результатов экспериментов и исследований в соответствующей области знаний
Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач
- ИД-4.ОПК-2** ***Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения***
- Знать** *Возможности современных и перспективных средств разработки программных продуктов, технических средств*
Методологии разработки программного обеспечения и технологии программирования
Языки программирования и работы с базами данных
Инструменты и методы верификации структуры программного кода
Методы проведения экспериментов и наблюдений, обобщения и обработки информации
- Уметь** *Использовать существующие типовые решения и шаблоны проектирования программного обеспечения*
Верифицировать структуру программного кода
Применять методы анализа научно-технической информации
- Владеть** *Разработка, изменение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения*
Разработка структуры программного кода ИС
Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 6 зач. ед. (216 часов), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)					
		1					
Контактная работа, в том числе:	124,5	124,5					

Аудиторные занятия (всего):	118	118					
Занятия лекционного типа	50	50					
Лабораторные занятия	68	68					
Занятия семинарского типа (семинары, практические занятия)							
Иная контактная работа:	6,5	6,5					
Контроль самостоятельной работы (КСР)	6	6					
Промежуточная аттестация (ИКР)	0,5	0,5					
Самостоятельная работа, в том числе:	56	56					
Проработка учебного (теоретического) материала	25	25					
Выполнение индивидуальных заданий	25	25					
Подготовка к текущему контролю	6	6					
Контроль:	35,5	35,5					
Подготовка к экзамену	35,5	35,5					
Общая трудоемкость	час.	216	216				
	в том числе контактная работа	124,5	124,5				
	зач. ед	6	6				

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 1 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Языки программирования. Основы алгоритмизации	8	4		1	3
2.	Лексические основы языка	6	2		1	3
3.	Скалярные типы. Выражения. Операции	9	4		2	3
4.	Ввод/вывод	6	2		2	2
5.	Условный оператор	7	2		2	3
6.	Операторы цикла	18	4		10	4
7.	Статические массивы	26	8		14	4
8.	Указатели	14	4		4	6
9.	Динамические массивы	18	4		8	6
10.	Функции, прототипы функций	30	10		12	8
11.	Структуры	14	2		6	6
12.	Алгоритмы сортировки	18	4		6	8
ИТОГО по разделам дисциплины		174	50		68	56
Контроль самостоятельной работы (КСР)		6				
Промежуточная аттестация (ИКР)		0,5				
Подготовка к текущему контролю		35,5				
Общая трудоемкость по дисциплине		216				

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	2	3	4
1.	Языки программирования. Основы алгоритмизации	История развития языков программирования. Виды программирования: структурное, модульное, функциональное, процедурное, логическое. Принципы различных методов программирования. Понятия: алгоритм, программа. Свойства алгоритма. 1 Методы разработки и способы представления алгоритмов. Элементарные базовые управляющие структуры: последовательность, ветвление, циклы Трансляторы: интерпретаторы и компиляторы	К
2.	Лексические основы языка	Алфавит языка, лексемы, идентификаторы, служебные слова.	К
3.	Скалярные типы. Выражения. Операции	Типы данных: целый, действительный, логический, символьный. Понятия: константа и переменная. Способы описания и правила записи констант и переменных. Понятия: операнд, операция, выражение, приоритет операций. Правила записи выражений. Стандартные математические функции. Выражения и операции: арифметические и логические. Правила записи выражений и операций.	К
4.	Ввод/вывод	Общие сведения о библиотеке потокового ввода/вывода. Стандартные потоки для базовых типов. Особенности вывода данных различных типов.	К
5.	Условный оператор	Операторы языка: операторы условия и выбора. Логические операции и выражения. Краткая форма записи условия. Условный оператор IF (формат оператора, выполнение оператора). Оператор Switch (формат оператора, выполнение оператора). Особенности применения оператора выбора..	К
6.	Операторы цикла	Операторы языка: операторы цикла. Цикл с предусловием While, цикл с постусловием (Do while): форматы и правила записи. Ошибки при программировании циклов Различия в операторах передачи управления. Простые и вложенные циклы Операторы языка: операторы передачи управления. Оператор Goto, оператор Break, оператор Continue, оператор Return (форматы и правила записи). Использование циклических операторов For, While и Do/While	К
7.	Статические массивы	Массивы: описание и размерность. Многомерные массивы. Массивы, описание и обработка массивов C++.	К
8.	Указатели и ссылки	Указатели и адреса объектов, связи массивов и указателей. Инициализация указателей. Операции с указателями (операции резадресации, арифметические операции). Ссылки.	К
9.	Динамические массивы	Массивы указателей, динамические массивы.	К
10.	Функции, прототипы функций	Определение, описание и вызов функций. Функции с переменным количеством параметров. Рекурсивные функции. Перегрузка функций. Шаблоны функций. Аргументы функции main. Параметры функций: локальные и глобальные. Функции стандартной библиотеки. Функции ввода/вывода. Открытие потока, ввод/вывод в поток, закрытие потока, примеры работы с потоками. Функции работы со строками и символами.	К

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
11.	Структуры	Объявление и определение структур. Доступ к членам структур. Инициализация структур. Вложенные структуры. Массивы структур. Структуры и функции. Битовые поля. Объединения	К
12.	Алгоритмы сортировки	Сортировка массива. Оценка алгоритмов сортировки. Пузырьковая сортировка, сортировка методом Шелла, быстрая сортировка. Методы поиска элементов в одномерных и многомерных массивах	К

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.2 Занятия семинарского типа

Не предусмотрены

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Языки программирования. Основы алгоритмизации. Лексические основы языка	Создание консольного и пустого приложения в среде Microsoft Visual Studio. Структура простой программы	РЗ
2.	Скалярные типы. Выражения. Операции	Переменные. Выражения. Операции Оператор присваивания	РЗ
3.	Ввод/вывод	Ввод / вывод данных различных типов	РЗ
4.	Условный оператор	Условный оператор. Оператор выбора. Условная операция	РЗ
5.	Операторы цикла	Оператор цикла while. Обработка цифр в числе	РЗ
6.	Операторы цикла	Оператор цикла for. Поиск делителей числа.	РЗ
7.	Операторы цикла	Оператор цикла while. Обработка последовательности с маркером окончания	РЗ
8.	Операторы цикла	Оператор цикла for. Обработка последовательности их N элементов. Вычисление суммы ряда	РЗ
9.	Операторы цикла	Контрольная работа по теме «Операторы цикла»	РЗ
10.	Статические массивы	Одномерные массивы. Основные операции	РЗ
11.	Статические массивы	Одномерные массивы. Многократная обработка элементов массива	РЗ
12.	Статические массивы	Одномерные массивы. Обработка нескольких массивов	РЗ
13.	Статические массивы	Двумерные массивы. Обработка диагоналей	РЗ
14.	Статические массивы	Двумерные массивы. Формирование одномерного массива на основе двумерного	РЗ
15.	Статические массивы	Двумерные массивы. Обработка нескольких массивов	РЗ
16.	Статические массивы	Контрольная работа по теме «Операторы цикла»	РЗ
17.	Указатели и ссылки	Понятие указателя. Инициализация указателей. Операции с указателями	РЗ
18.	Указатели и ссылки	Понятие ссылки. Правила работы со ссылками	РЗ
19.	Динамические массивы	Одномерные динамические массивы	РЗ
20.	Динамические массивы	Двумерные динамические массивы	РЗ
21.	Динамические массивы	Обработка динамических массивов	РЗ
22.	Динамические массивы	Контрольная работа по теме «Динамические массивы»	РЗ
23.	Функции, прототипы функций	Функции. Параметры функций	РЗ
24.	Функции, прототипы функций	Передача параметров в функции	РЗ
25.	Функции, прототипы функций	Простейшие рекурсивные функции	РЗ
26.	Функции, прототипы функций	Рекурсивные функции	РЗ

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
27.	Функции, прототипы функций	Передача массивов в функции	РЗ
28.	Функции, прототипы функций	Контрольная работа по теме «Функции, прототипы функций»	РЗ
29.	Структуры	Объявление и определение структур. Доступ к членам структур. Инициализация структур	РЗ
30.	Структуры	Массивы структур	РЗ
31.	Структуры	Контрольная работа по теме «Структуры»	РЗ
32.	Алгоритмы сортировки	Сортировка пузырьком, сортировка выбором	РЗ
33.	Алгоритмы сортировки	Сортировка Шелла, быстрая сортировка	РЗ
34.	Алгоритмы сортировки	Контрольная работа по теме «Алгоритмы сортировки»	РЗ

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.4 Примерная тематика курсовых работ (проектов)

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	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019

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

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

- Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

- Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

- Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

- Информационно-коммуникационные технологии (ИКТ) – расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

- Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

- Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

- Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

- Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

- Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

- Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

- Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеперечисленных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

- работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

- проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

– анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

– развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	32
Итого			32

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

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

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

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.

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

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Основы программирования».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме тестовых заданий, разноуровневых заданий, и **промежуточной аттестации** в форме вопросов и заданий к зачету и 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

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Языки программирования. Основы алгоритмизации	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные вопросы 1-5	Вопрос на экзамене 1-9
2	Лексические основы языка	И ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные вопросы 6-7	Вопрос на экзамене 10-12
3	Скалярные типы. Выражения. Операции	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные вопросы 8-10, 14-17 Контрольные задания 1-2 Задания для самостоятельного решения 1-4	Вопрос на экзамене 13-28
4	Ввод/вывод	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные вопросы 11-13	Вопрос на экзамене 29-31
5	Условный оператор	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные задания 3-5 Задания для самостоятельного решения 5-12	Вопрос на экзамене 32-33

6	Операторы цикла	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные задания 6-10 Задания для самостоятельного решения 13-22	Вопрос на экзамене 34-38
7	Статические массивы	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные задания 11-15 Задания для самостоятельного решения 23-32	Вопрос на экзамене 39-42
8	Указатели и ссылки	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные задания 16-17 Задания для самостоятельного решения 33-37	Вопрос на экзамене 43-48
9	Динамические массивы	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные задания 18-23 Задания для самостоятельного решения 38-55	Вопрос на экзамене 49-50
10	Функции, прототипы функций	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные задания 24-27 Задания для самостоятельного решения 56-65	Вопрос на экзамене 51-64
11	Структуры	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные задания 28-29 Задания для самостоятельного решения 66-72	Вопрос на экзамене 65-72
12	Алгоритмы сортировки	ИД-3.ОПК-2 ИД-4.ОПК-2	Контрольные задания 30-31	Вопрос на экзамене 73-78

Показатели, критерии и шкала оценки сформированных компетенций

Соответствие **пороговому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **удовлетворительно /зачтено**):

ОПК-2 Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности

ИД-3.ОПК-2 Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности

Знать Основы синтаксиса и лексики языка C++.
 Скалярные типы данных, выражения и операции в C++.
 Принципы ввода/вывода данных.
 Условные операторы (if-else, switch) и операторы циклов (for, while, do-while).
 Работу со статическими и динамическими массивами, указателями и ссылками.
 Функции, их прототипы и передачу параметров.
 Структуры данных и их применение.

Уметь	Основные алгоритмы сортировки и их реализацию на C++.
	Разрабатывать алгоритмы и переводить их в программный код на C++.
Владеть	Использовать условные операторы и циклы для управления потоком выполнения программы.
	Работать с массивами (статическими и динамическими), указателями и ссылками.
ИД-4.ОПК-2	Организовывать код с помощью функций и структур.
	Применять базовые алгоритмы сортировки.
Знать	Навыками написания структурированного кода на C++.
	Методами отладки и оптимизации программ.
Уметь	Приемами работы с динамической памятью.
	Основами модульного программирования с использованием функций.
ИД-4.ОПК-2 Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения	
Знать	Современные среды разработки (IDE) и компиляторы для C++.
	Методы отладки и тестирования кода.
Уметь	Принципы оценки эффективности алгоритмов.
	Применять инструменты отладки (например, gdb, встроенные дебаггеры IDE).
Владеть	Проверять корректность работы программ с помощью тестовых случаев.
	Анализировать производительность кода (время выполнения, использование памяти).
Знать	Навыками тестирования и верификации программ на C++.
	Методами оптимизации кода для повышения эффективности.
Уметь	Приемами документирования и рефакторинга кода.

Соответствие **базовому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **хорошо /зачтено**):

ОПК-2 **Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности**

ИД-3.ОПК-2 **Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности**

Знать Основы синтаксиса и лексики языка C++ на уровне, позволяющем писать читаемый и эффективный код.

Скалярные типы данных, выражения и операции в C++ с пониманием их влияния на производительность.

Принципы ввода/вывода данных и способы их оптимизации.

Условные операторы (if-else, switch) и операторы циклов (for, while, do-while) с умением выбирать оптимальные конструкции для конкретных задач.

Работу со статическими и динамическими массивами, указателями и ссылками с учетом управления памятью и минимизации ошибок.

Функции, их прототипы и передачу параметров с пониманием областей видимости и времени жизни переменных.

Структуры данных и их применение в реальных задачах.

Основные алгоритмы сортировки и их реализацию на C++ с анализом временной и пространственной сложности.

Уметь	Разрабатывать алгоритмы и переводить их в программный код на C++ с соблюдением принципов структурного программирования. Использовать условные операторы и циклы для управления потоком выполнения программы с минимизацией вложенности и оптимизацией условий. Работать с массивами (статическими и динамическими), указателями и ссылками без утечек памяти и ошибок доступа. Организовывать код с помощью функций и структур с соблюдением принципов модульности и повторного использования. Применять базовые алгоритмы сортировки с обоснованием выбора конкретного алгоритма для задачи.
Владеть	Навыками написания структурированного, читаемого и эффективного кода на C++ в соответствии с best practices. Методами отладки, тестирования и оптимизации программ с использованием профилирования. Приемами работы с динамической памятью и предотвращения утечек. Основами модульного программирования с использованием функций и разделения кода на логические блоки.
ИД-4.ОПК-2	Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения
Знать	Навыками написания структурированного, читаемого и эффективного кода на C++ в соответствии с современными практиками Методами отладки, тестирования и оптимизации программ с использованием профилирования. Приемами работы с динамической памятью и предотвращения утечек. Основами модульного программирования с использованием функций и разделения кода на логические блоки.
Уметь	Применять инструменты отладки (например, gdb, встроенные дебаггеры IDE) для анализа сложных ошибок. Проверять корректность работы программ с помощью тестовых случаев и автоматизированного тестирования. Анализировать производительность кода (время выполнения, использование памяти) и вносить оптимизации.
Владеть	Навыками тестирования, верификации и валидации программ на C++ в соответствии с промышленными стандартами. Методами рефакторинга и оптимизации кода для повышения эффективности и поддерживаемости. Приемами документирования кода и создания технической документации.

Соответствие **продвинутому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **отлично /зачтено**):

ОПК-2	Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
ИД-3.ОПК-2	Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности
Знать	Основа синтаксиса и лексики языка C++ на профессиональном уровне, позволяющем писать высокоэффективный и надежный код

	Скалярные типы данных, выражения и операции в C++ с глубоким пониманием их влияния на производительность и возможностей оптимизации Принципы ввода/вывода данных и современные способы их оптимизации для различных сценариев использования Условные операторы и операторы циклов с мастерским владением техниками оптимизации и выбора наилучших конструкций Работу со статическими и динамическими массивами, указателями и ссылками с профессиональным управлением памятью и предотвращением ошибок Функции, их прототипы и передачу параметров с полным пониманием всех аспектов их работы Структуры данных и их эффективное применение в сложных практических задачах Основные алгоритмы сортировки и их реализацию на C++ с профессиональным анализом сложности и оптимизацией
Уметь	Разрабатывать оптимальные алгоритмы и профессионально реализовывать их в коде на C++ Грамотно использовать условные операторы и циклы для создания высокопроизводительного кода Профессионально работать с массивами, указателями и ссылками, гарантируя безопасность и эффективность Организовывать код с применением продвинутых техник модульности и повторного использования Осознанно выбирать и применять алгоритмы сортировки, адаптируя их под конкретные задачи
Владеть	Навыками написания профессионального, высокооптимизированного кода на C++ Продвинутыми методами отладки, тестирования и оптимизации программ Грамотным владением техниками работы с динамической памятью Профессиональными приемами модульного программирования и организации кода
ИД-4.ОПК-2	Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения
Знать	Современные методы и инструменты оценки качества кода на профессиональном уровне Продвинутые техники тестирования и верификации программного обеспечения Методы комплексного анализа производительности и оптимизации
Уметь	Профессионально применять инструменты отладки для решения сложных задач Проводить всестороннее тестирование программного обеспечения Выполнять глубокий анализ производительности и находить узкие места
Владеть	Профессиональными навыками тестирования и верификации кода Продвинутыми методами оптимизации и рефакторинга Техниками создания качественной технической документации

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

Типовые контрольные задания:

1. Даны катеты прямоугольного треугольника a и b . Найти его гипотенузу c и периметр P .
2. Найти расстояние между двумя точками с заданными координатами (x_1, y_1) , (x_2, y_2) на плоскости.
3. Пользователь вводит с клавиатуры координаты точки (x, y) . Необходимо написать программу, которая проверяет, какой четверти принадлежит точка.
4. Пользователь вводит с клавиатуры 3 числа от -10 до 10 . Написать программу, которая считает сумму только нечетных положительных чисел.
5. Пользователь вводит с клавиатуры 2 числа. Написать программу, которая проверяет, является ли первое число квадратом второго, и выводит соответствующее сообщение.
6. Дано целое число. Найти сумму его четных цифр.
7. Дана последовательность целых чисел, оканчивающаяся числом 0. Найти сумму квадратов двузначных отрицательных чисел. Если таких чисел нет, сообщить об этом.
8. Дана последовательность целых чисел, оканчивающаяся числом 0. Найти наименьшее четное число.
9. Для заданного натурального числа N найти сумму делителей, заканчивающихся заданной цифрой K .
10. Натуральное число N называется совершенным, если оно равно сумме всех своих делителей, исключая само это число (например: $6 = 1 + 2 + 3$; $28 = 1 + 2 + 4 + 7 + 14$). Найти все совершенные числа отрезка $[A, B]$.
11. Дан одномерный массив, состоящий из N ($0 < N < 100$) целочисленных элементов. Определить количество четных элементов, индексы которых не кратны 3.
12. Дан одномерный массив, состоящий из N ($0 < N < 100$) целочисленных элементов. Вывести значения, которые не превосходят среднего арифметического.
13. Дан одномерный массив, состоящий из N ($0 < N < 100$) целочисленных элементов. Определить, есть ли в нем два совершенных элемента.
14. Дана квадратная матрица целых чисел размером $N \times N$ ($3 < N < 100$). Найти произведение ненулевых элементов, лежащих выше главной диагонали матрицы.
15. Дана матрица вещественных чисел. Найти разность между максимальным и минимальным значениями элементов массива.
16. Используя вместо самой переменной указатель на нее, написать программу в соответствии с заданием. Вычислить высоту треугольника, зная две стороны треугольника и угол между ними.
17. Дан массив целых чисел. Вывести значения, которые не превосходят среднего арифметического.
18. Используя вместо самой переменной указатель на нее, написать программу в соответствии с заданием. Ввести сторону квадрата a . Вычислить площадь вписанной окружности. Дан массив целых чисел. Все нечетные элементы удвоить, а четные уменьшить вдвое.
19. Дан массив целых чисел. Определить количество четных элементов, индексы которых не кратны 3.
20. Дан двумерный массив вещественных чисел. Найти минимальный по абсолютной величине элемент матрицы.

21. Дан двумерный массив вещественных чисел. Найти сумму произведений элементов строки и столбца матрицы, на пересечении которых находится наибольший по абсолютной величине элемент матрицы.
22. Дан двумерный массив вещественных чисел. Найти сумму элементов столбца, в котором расположен максимальный элемент массива.
23. Дана квадратная матрица целых чисел. Найти отношение сумм элементов, лежащих выше и ниже главной диагонали матрицы.
24. Написать функцию, которая определяет, содержит ли последовательность целых чисел N хотя бы три отрицательных числа. Возвратить 1 – если содержит, 0 – в противном случае.
25. Дан двумерный массив целых чисел. С использованием функции каждой его строке найти первый нечетный элемент (принять, что нечетные элементы есть в каждой строке).
26. Дан двумерный массив целых чисел. С использованием функции для каждой строки выяснить, имеются ли в ней положительные элементы.
27. Написать функцию, которая определяет и возвращает количество совершенных чисел в последовательности натуральных чисел N .
28. Разработать структуру Abiturient: Фамилия, Имя, Отчество, Адрес, Оценки. Создать массив структур не менее чем из трёх элементов. Вывести на экран: список абитуриентов, имеющих неудовлетворительные оценки; список абитуриентов, сумма баллов у которых не меньше заданной; выбрать N абитуриентов, имеющих самую высокую сумму баллов, и список абитуриентов, имеющих проходной балл.
29. Разработать структуру Aeroflot: Пункт назначения, Номер рейса, Тип самолета, Время вылета, Дни недели. Создать массив структур не менее чем из трёх элементов. Вывести на экран: - список рейсов для заданного пункта назначения; список рейсов для заданного дня недели; список рейсов для заданного дня недели, время вылета для которых больше заданного.
30. Опишите алгоритм сортировки, дающий время работы $O(n^2)$.
31. Опишите алгоритм быстрой сортировки

Типовые задачи для самостоятельного решения:

1. Дан номер некоторого года (целое положительное число). Определить соответствующий ему номер столетия, учитывая, что, например, началом XX столетия был 1901 год.
2. Даны катет прямоугольного треугольника a и гипотенуза c . Найти другой катет b и периметр P .
3. Даны координаты двух противоположных вершин прямоугольника: (x_1, y_1) , (x_2, y_2) . Стороны прямоугольника параллельны осям координат. Найти периметр и площадь данного прямоугольника.
4. Даны три точки A, B, C на числовой оси. Точка C расположена между точками A и B . Найти произведение длин отрезков $|AC|$ и $|BC|$.
5. Пользователь вводит с клавиатуры 3 целых числа. Необходимо написать программу, которая выводит на экран только чётные числа.
6. Пользователь вводит с клавиатуры 3 целых числа. Необходимо написать программу, которая выводит на экран квадрат наибольшего из введенных чисел.
7. Пользователь вводит 2 целых числа. Необходимо написать программу, которая выводит на экран сумму этих чисел, если они оба положительные. В противном случае вывести на экран модуль суммы.
8. Пользователь вводит 2 целых числа. Необходимо написать программу, которая проверяет, делится ли первое число на второе без остатка и выводит сообщение об этом.

9. Пользователь вводит 2 целых числа. Необходимо написать программу, которая считает сумму квадратов введенных чисел и квадрат суммы и выводит наибольшую сумму на экран.

10. Пользователь вводит с клавиатуры два числа. Необходимо написать программу, которая проверяет, делится ли каждое число нацело на 2 и 3, и выводит сообщение об этом. Если же число не делится, выводит сообщение об этом и остаток от деления.

11. Пользователь вводит с клавиатуры число от 1 до 20. Если оно принадлежит интервалу $[1, 10]$, вывести на экран квадрат этого числа, если принадлежит интервалу $[11, 20]$, вывести на экран введенное число в кубе.

12. По длинам трех отрезков, введенных пользователем, определить возможность существования треугольника, составленного из этих отрезков. Если такой треугольник существует, то определить, является ли он разносторонним, равнобедренным или равносторонним.

13. Дано целое число. Найти количество его нечетных цифр.

14. Дано целое число. Верно ли, что в числе все цифры одинаковые?

15. Дано целое число. Найти количество заданной цифры в числе.

16. Дана последовательность целых чисел, оканчивающаяся числом 0. Найти произведение последних цифр положительных чисел.

17. Дана последовательность целых чисел, оканчивающаяся числом 0. Найти количество чисел, у которых последняя цифра больше предпоследней. Если таких чисел нет, сообщить об этом.

18. Дана последовательность целых чисел, оканчивающаяся числом 0. Найти сумму квадратов чисел, оканчивающихся на 2 или на 13.

19. Для заданного натурального числа N найти количество его нечетных делителей.

20. Является ли заданное натуральное число $N > 1$ простым?

21. Натуральное число N называется совершенным, если оно равно сумме всех своих делителей, исключая само это число (например: $6 = 1 + 2 + 3$; $28 = 1 + 2 + 4 + 7 + 14$). Определить, является ли заданное натуральное K совершенным?

22. Последовательность Фибоначчи (члены которой называются числами Фибоначчи) определяется рекуррентными соотношениями: $F_1 = F_2 = 1$; $F_n = F_{n-1} + F_{n-2}$ для $n \geq 3$. Найти сумму первых N чисел Фибоначчи.

23. Дан одномерный массив, состоящий из N ($0 < N < 100$) целочисленных элементов. Все элементы, кратные числу 17, заменить нулем.

24. Дан одномерный массив, состоящий из N ($0 < N < 100$) целочисленных элементов. Все нечетные элементы удвоить, а четные уменьшить вдвое.

25. Дан одномерный массив, состоящий из N ($0 < N < 100$) целочисленных элементов. Определить сумму положительных элементов с четными индексами.

26. Дан одномерный массив, состоящий из N ($0 < N < 100$) вещественных элементов. Заменить положительные числа, индексы которых четны, значением наибольшего по модулю числа.

27. Дан одномерный массив, состоящий из N ($0 < N < 100$) целочисленных элементов. Определить, есть ли в нем хотя бы одно простое число.

28. Дан одномерный массив, состоящий из N ($0 < N < 100$) целочисленных элементов. Определить, есть ли в нем хотя бы одно число Фибоначчи.

29. Дан одномерный массив, состоящий из N ($0 < N < 100$) вещественных элементов. Если в результате замены отрицательных элементов их квадратами элементы будут образовывать неубывающую последовательность, то получить сумму элементов исходной последовательности, в противном случае получить их произведение.

30. Дана квадратная матрица целых чисел размером $N \times N$ ($3 < N < 100$). Найти отношение сумм элементов, лежащих выше и ниже главной диагонали матрицы.

31. Дана матрица вещественных чисел. Найти сумму элементов, расположенных по периметру матрицы.
32. Дана матрица вещественных чисел. Найти произведение ненулевых элементов строки матрицы, на которой расположен максимальный элемент.
33. Используя вместо самой переменной указатель на нее, написать программу в соответствии с заданием. Вычислить площадь треугольника по стороне и высоте.
34. Используя вместо самой переменной указатель на нее, написать программу в соответствии с заданием. Вычислить площадь окружности по заданному радиусу.
35. Используя вместо самой переменной указатель на нее, написать программу в соответствии с заданием. Даны значения a и b , найти их среднее арифметическое, среднегеометрическое.
36. Используя вместо самой переменной указатель на нее, написать программу в соответствии с заданием. Задается длина окружности. Найти площадь круга, ограниченного этой окружностью.
37. Используя вместо самой переменной указатель на нее, написать программу в соответствии с заданием. Вычислить объем цилиндра, зная радиус основания и высоту.
38. Дан одномерный массив, состоящий из N целочисленных элементов. Вывести массив на экран в обратном порядке.
39. Одномерный массив, состоящий из N целочисленных элементов, заполнить случайными числами. Вычислить сумму нечетных элементов массива.
40. Дан одномерный массив, состоящий из N вещественных элементов. Вычислить среднеарифметическое положительных элементов массива.
41. Дан одномерный массив, состоящий из N вещественных элементов. Вывести отрицательные элементы на экран в обратном порядке.
42. Дан одномерный массив, состоящий из N целочисленных элементов. Найти максимальный отрицательный элемент.
43. Дан одномерный массив, состоящий из N целочисленных элементов. Найти минимальный положительный элемент.
44. Дан одномерный массив, состоящий из N целочисленных элементов. Вычислить сумму положительных элементов массива, кратных 3.
45. Дан массив целых чисел. Определить сумму положительных элементов с четными индексами.
46. Дан массив целых чисел. Определить сумму элементов больших заданного числа x , порядковые номера которых не кратны 5.
47. Дан двумерный массив вещественных чисел. Найти количество отрицательных элементов матрицы.
48. Дан двумерный массив вещественных чисел. Найти количество элементов матрицы, превышающих ее среднее арифметическое значение.
49. Дан двумерный массив вещественных чисел. Найти сумму элементов массива, для которых сумма индексов равна k , где k – введенное с клавиатуры значение. Проверить, что значение k позволяет найти решение.
50. Дан двумерный массив вещественных чисел. Найти сумму положительных элементов столбца, содержащего максимальный элемент массива.
51. Дан двумерный массив вещественных чисел. Найти разность между максимальным и минимальным значениями элементов массива.
52. Дан двумерный массив вещественных чисел. Найти сумму элементов, расположенных по периметру массива.
53. Дан двумерный массив вещественных чисел. Найти произведение ненулевых элементов строки массива, на которой расположен максимальный элемент.
54. Дан двумерный массив вещественных чисел. Найти разность между суммой значений элементов четных и нечетных строк.

55. Дана квадратная матрица целых чисел размером. Найти произведение ненулевых элементов, лежащих выше главной диагонали матрицы

56. Написать функцию, возвращающую процент нулевых чисел последовательности из N целых чисел.

57. Написать функцию, которая определяет, является ли последовательность натуральных чисел N строго возрастающей. Возвратить 1 – если является, 0 – в противном случае.

58. Написать функцию, которая определяет и возвращает разность между произведением нечётных чисел и наибольшим среди отрицательных чисел последовательности целых чисел N . Написать функцию, которая определяет, является ли последовательность строго убывающей. Возвратить 1 – если является, 0 – в противном случае. Использовать передачу массива по адресу.

59. Написать функцию, возвращающую i -й по порядку элемент последовательности Фибоначчи (0 1 1 2 3 5 8 ...). Использовать рекурсию.

60. Дан двумерный массив. С использованием функции найти строку с максимальной суммой элементов. Если таких строк несколько, должен быть найден номер самой нижней из них.

61. Дан двумерный массив. С использованием функции найти столбец с минимальной суммой элементов. Если таких столбцов несколько, должен быть найден номер самого левого из них.

62. Дан двумерный массив. С использованием функции проверить, является ли последовательность элементов некоторой строки массива упорядоченной по неубыванию. В случае отрицательного ответа в обеих задачах должны быть напечатаны координаты первого элемента, нарушающего указанную упорядоченность.

63. Дан двумерный массив целых чисел. С использованием функции найти минимальный номер строки, состоящей только из положительных элементов.

64. Дан двумерный массив целых чисел. С использованием функции в каждой его строке заменить любой минимальный элемент на максимальный.

65. Дан двумерный массив целых чисел. С использованием функции в каждой строке двумерного массива поменять местами первый нулевой элемент и последний отрицательный. Если таких элементов нет, то должно быть выведено соответствующее сообщение.

66. Разработать структуру Book: Автор, Название, Издательство, Год, Количество страниц. Создать массив структур не менее чем из трёх элементов. Вывести на экран: список книг заданного автора; список книг, выпущенных заданным издательством; список книг, выпущенных после заданного года.

67. Разработать структуру Worker: Фамилия и инициалы, Должность, Год поступления на работу, Зарплата. Создать массив структур не менее чем из трёх элементов. Вывести на экран: список работников, стаж работы которых на данном предприятии превышает заданное число лет; список работников, зарплата которых больше заданной; список работников, занимающих заданную должность.

68. Разработать структуру Train: Пункт назначения, Номер поезда, Время отправления, Число общих мест, Купейных, Плацкартных. Создать массив структур не менее чем из трёх элементов. Вывести на экран: список поездов, следующих до заданного пункта назначения; список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа; список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.

69. Разработать структуру Product: Наименование, Производитель, Цена, Срок хранения, Количество. Создать массив структур не менее чем из трёх элементов. Вывести на экран: список товаров для заданного наименования; список товаров для заданного наименования, цена которых не превышает указанной; список товаров, срок хранения которых больше заданного.

70. Разработать структуру Patient: Фамилия, Имя, Отчество, Адрес, Номер медицинской карты, Диагноз. Создать массив структур не менее чем из трёх элементов. Вывести на экран: список пациентов, имеющих данный диагноз; список пациентов, номер медицинской карты которых находится в заданном интервале.

71. Разработать структуру Bus: Фамилия и инициалы водителя, Номер автобуса, Номер маршрута, Марка, Год начала эксплуатации, Пробег. Создать массив структур не менее чем из трёх элементов. Вывести на экран: список автобусов для заданного номера маршрута; список автобусов, которые эксплуатируются больше 10 лет; список автобусов, пробег у которых больше 10 000 км.

72. Разработать структуру Customer: Фамилия, Имя, Отчество, Адрес, Телефон, Номер кредитной карточки. Создать массив структур не менее чем из трёх элементов. Вывести на экран: список покупателей в алфавитном порядке; список покупателей, номер кредитной карточки которых находится в заданном интервале.

Типовые контрольные вопросы

1. Основные принципы алгоритмизации и программирования.
2. Алгоритмы и программы.
3. Данные. Понятие типа данных.
4. Логические основы алгоритмизации.
5. Языки программирования: эволюция, классификация.
6. Идентификаторы. Ключевые слова. Символы.
7. Данные: переменные и константы.
8. Описание данных целого типа. Целые константы. Инициализация переменных целого типа. Модификатор unsigned.
9. Числа с плавающей точкой. Описание переменных с плавающей точкой
10. Модификаторы доступа. Модификатор const. Определение констант через #define.
11. Функции printf() и scanf(). Использование функции printf().
12. Модификаторы спецификации преобразования, используемые в функции printf().
13. Использование функции printf() для преобразования данных. Применение функции scanf().
14. Основные операции. Операция присваивания: =. Операция сложения: +. Операция вычитания: -. Операция изменения знака: -. Операция умножения: *. Операция деления: /.
15. Поразрядные операции. Поразрядное И (AND). Поразрядное ИЛИ (OR). Поразрядное исключающее ИЛИ (OR).
16. Поразрядный сдвиг влево и вправо. Операции отношения и логические операции. Условная операция (? :). Операция запятая (,). Порядок выполнения операций.
17. Выражения и операторы. Выражения. Операторы. Составные операторы (блоки).

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен/зачет)

Вопросы для подготовки к экзамену

1. Поколения языков программирования
2. Типы(парадигмы) языков программирования
3. Принципы структурного программирования

4. Понятие алгоритма, свойства алгоритма
5. Сущность трансляции. Компиляция и интерпретация
6. Синтаксис, семантика и прагматика
7. Форма Бэкуса-Наура (БНФ)
8. Синтаксические диаграммы
9. Типы ошибок
10. Состав языка программирования
11. Типы данных
12. Классификация типов данных в C++
13. Целочисленный (целый) тип данных
14. Вещественный тип данных
15. Символьный тип данных
16. Логический тип данных
17. Преобразования типов
18. Структура программы
19. Переменная в СИ++. Классы памяти
20. Операции и выражения
21. Унарные операции
22. Бинарные операции
23. Тернарная операция
24. Арифметические операции. Инкремент/декремент
25. Логические операции
26. Операции присваивания
27. Побитовые логические операции
28. Стандартные функции C++
29. Ввод и вывод данных средствами C++
30. Базовые конструкции структурного программирования
31. Составной оператор и блок
32. Условный оператор
33. Оператор выбора/Переключатель
34. Итерационные и арифметические циклы
35. Цикл с предусловием
36. Цикл с постусловием
37. Цикл с параметром
38. Операторы перехода
39. Одномерные массивы
40. Классы задач обработки массивов
41. Двумерные массивы. Диагонали
42. Двумерные массивы. Типы задач
43. Понятие указателя
44. Инициализация указателей
45. Операции с указателями
46. Понятие ссылки. Правила работы со ссылками
47. Динамические переменные
48. Указатели и массивы
49. Одномерные динамические массивы
50. Многомерные динамические массивы
51. Объявление и определение функций
52. Прототип функции
53. Локальные и глобальные переменные
54. Параметры функции
55. Передача массивов как параметров функции

56. Функции с начальными (умалчиваемыми) значениями параметров
57. Подставляемые (inline) функции
58. Функции с переменным числом параметров
59. Перегрузка функций
60. Шаблоны функций
61. Указатель на функцию
62. Передача имен функций в качестве параметров
63. Ссылки на функцию
64. Рекурсивные функции
65. Перечисляемые типы
66. Объявление и определение структур
67. Доступ к членам структур. Инициализация структур
68. Вложенные структуры
69. Массивы структур
70. Структуры и функции. Методы в структурах
71. Битовые поля
72. Объединения
73. Сортировка массива. Оценка алгоритмов сортировки
74. Сортировка методом "пузырька" (простого обмена)
75. Шейкерная сортировка
76. Сортировка методом простого выбора (простой перебор)
77. Сортировка методом простого включения (сдвиг-вставка, вставками)
78. Сортировка Хоара

4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Методические рекомендации, определяющие процедуры оценивания выполнения контрольных заданий:

Задание считается выполненным при выполнении следующих условий:

- предоставлен исходный код на C++ в среде Visual Studio
- продемонстрирована работоспособность программы
- студент понимает исходный код и отвечает на вопросы по его организации.

Методические рекомендации, определяющие процедуры оценивания на зачете:

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ». Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является зачет и экзамен. Студенты обязаны получить зачет в соответствии с расписанием и учебным планом ФОС промежуточной аттестации состоит из контрольных заданий и заданий для самостоятельной работы.

Зачет по дисциплине преследует цель оценить работу студента, получение теоретических и практических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Результат сдачи зачета заноситься преподавателем в экзаменационную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом тестовых заданий, контрольных заданий и заданий для самостоятельной работы.

Критерии оценки:

Оценка	
Не зачтено	Зачтено
• если студент правильно решил менее 70 % задач и/или не имеет представление как решать остальные задачи	• если студент правильно решил 70 % задач, имеет представление как решать остальные задачи

Методические рекомендации к сдаче экзамена

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Текущий контроль успеваемости студентов проводится в целях совершенствования и непрерывного контроля качества образовательного процесса, проверки усвоения учебного материала, активизации самостоятельной работы студентов, стимулирования их учебной работы, обеспечения эффективности образовательного процесса, предупреждения рисков отчисления студентов.

Текущий контроль знаний студентов осуществляется постоянно в течение всего семестра.

Виды текущего контроля: устный (письменный) опрос на занятиях; проверка выполнения домашних заданий; проведение контрольных работ; оценка активности студента на занятии.

При отсутствии зачетной книжки у студента экзаменатор не имеет права принимать у него зачет/экзамен. Такой студент считается не явившимся на зачет/экзамен. В исключительных случаях, на основании распоряжения декана преподаватель может допустить студента к зачету/экзамену при наличии документа, удостоверяющего личность.

В целях объективного оценивания знаний во время проведения зачетов и экзаменов не допускается наличие у студентов посторонних предметов и технических устройств.

Студенту, использующему в ходе экзамена неразрешенные источники и средства получения информации, выставляется неудовлетворительная оценка, и он удаляется из аудитории.

Во время экзамена студенты могут пользоваться утвержденной рабочей программой учебной дисциплины, которая должна быть в наличии на экзамене, а также с разрешения экзаменатора справочной литературой и другими пособиями.

Студенты, нарушающие правила поведения при проведении зачетов и экзаменов, могут быть незамедлительно удалены из аудитории, к ним могут быть применены меры дисциплинарного воздействия.

На зачете/экзамене могут присутствовать ректор, проректор по учебной работе, декан факультета, заведующий кафедрой, которая обеспечивает учебный процесс по данной

дисциплине. Присутствие на экзаменах и зачетах посторонних лиц без разрешения ректора или проректора по учебной работе не допускается.

После прослушивания лекции рекомендуется выполнить упражнения, приводимые в лекции для самостоятельной работы, а также выполнить на компьютере с использованием среды Visual Studio задачи, приводимые в лекции в качестве примеров.

При самостоятельной работе студентов необходимо изучить литературу, приведенную в перечнях, для осмысления вводимых понятий, анализа предложенных подходов и методов разработки параллельных программ. Разрабатывая решение новой задачи, студент должен уметь выбрать методы решения задачи с учетом целевой аппаратной платформы, проводить отладку и профилирование программы на языке C++.

В качестве систем программирования для решения задач и изучения методов и алгоритмов, приведенных в лекциях, рекомендуется использовать на практических занятиях и при самостоятельной работе стандартную реализацию языка C++ в связке со средой разработки Visual Studio.

Методические рекомендации, определяющие процедуры оценивания на экзамене:

Критерии оценивания и шкала оценки экзамена

Оценка	Критерии выставления оценки
«Отлично»	Дан развернутый ответ на поставленные вопросы. Материал изложен последовательно. Имеются логичные и аргументированные выводы. В течении семестра студент правильно решил 80 % задач, имеет представление как решать остальные задачи
«Хорошо»	Дан развернутый ответ на поставленные вопросы. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы. В течении семестра студент правильно решил 70 % задач, имеет представление как решать остальные задачи
«Удовлетворительно»	Ответ на вопрос не является полным. Материал изложен непоследовательно. Выводы не аргументированы. В течении семестра студент правильно решил 60 % задач, имеет представление как решать остальные задачи
«Неудовлетворительно»	Обучающийся не знает значительной части программного материала, допускает существенные ошибки. В течении семестра студент правильно решил менее 70 % задач и/или не имеет представление как решать остальные задачи

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 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 основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

5.1 Основная литература:

1. Конова, Е. А. Алгоритмы и программы. Язык C++ / Е. А. Конова, Г. А. Поллак. — 7-е изд., стер. — Санкт-Петербург : Лань, 2023. — 384 с. — ISBN 978-5-507-46070-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/297002> (дата обращения: 08.06.2025). — Режим доступа: для авториз. пользователей.
2. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2022. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492984> (дата обращения: 08.06.2025).
3. Кувшинов, Д. Р. Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2022. — 104 с. — (Высшее образование). — ISBN 978-5-534-07559-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/493460> (дата обращения: 08.06.2025).

Для освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья имеются издания в электронном виде в электронно-библиотечных системах «Лань» и «Юрайт».

5.2 Дополнительная литература:

1. Страуструп, Б. Язык программирования C++ для профессионалов : учебное пособие / Б. Страуструп. — 2-е изд. — Москва : ИНТУИТ, 2016. — 670 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100542> (дата обращения: 08.06.2025). — Режим доступа: для авториз. пользователей.
2. Павловская, Т. А. Программирование на языке C++ : учебное пособие / Т. А. Павловская. — 2-е изд. — Москва : ИНТУИТ, 2016. — 154 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100409> (дата обращения: 08.06.2025). — Режим доступа: для авториз. пользователей.
3. Керниган, Б. В. Язык программирования C : учебник / Б. В. Керниган, Д. М. Ричи. — 2-е изд. — Москва : ИНТУИТ, 2016. — 313 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100543> (дата обращения: 08.06.2025). — Режим доступа: для авториз. пользователей.

5.3 Периодические издания:

1. Базы данных компании «Ист Вью» <http://dlib.eastview.com>

2. Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

5.4. 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» <http://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. Scopus <http://www.scopus.com/>
2. ScienceDirect <https://www.sciencedirect.com/>
3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
7. Президентская библиотека им. Б.Н. Ельцина <https://www.prilib.ru/>
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
9. Springer Journals: <https://link.springer.com/>
10. Springer Journals Archive: <https://link.springer.com/>
11. Nature Journals: <https://www.nature.com/>
12. Springer Nature Protocols and Methods: <https://experiments.springernature.com/sources/springer-protocols>
13. Springer Materials: <http://materials.springer.com/>
14. Nano Database: <https://nano.nature.com/>
15. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
16. "Лекториум ТВ" <http://www.lektorium.tv/>
17. 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. КиберЛенинка <http://cyberleninka.ru/>;
2. Американская патентная база данных <http://www.uspto.gov/patft/>
3. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
4. Федеральный портал "Российское образование" <http://www.edu.ru/>;
5. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
6. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
7. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
8. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
9. Служба тематических толковых словарей <http://www.glossary.ru/>;
10. Словари и энциклопедии <http://dic.academic.ru/>;

11. Образовательный портал "Учеба" <http://www.ucheba.com/>;
12. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ

1. Электронный каталог Научной библиотеки КубГУ <http://megapro.kubsu.ru/MegaPro/Web>
2. Электронная библиотека трудов ученых КубГУ <http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>
3. Среда модульного динамического обучения <http://moodle.kubsu.ru>
4. База учебных планов, учебно-методических комплексов, публикаций и конференций <http://infoneeds.kubsu.ru/>
5. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru;>
6. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
7. 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)

.....

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. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю)

7.1 Перечень информационно-коммуникационных технологий

- Проверка домашних заданий и консультирование посредством электронной почты.
- Использование электронных презентаций при проведении лекционных занятий
- Система MOODLE
- Проверка домашних заданий и консультирование посредством ЭОИС КубГУ

а. Перечень лицензионного и свободно распространяемого программного обеспечения

OpenOffice
Компилятор C++
Google Chrome 97
GitHub Desktop 2.9
Visual Studio Code, 1.52+

8. Материально-техническое обеспечение по дисциплине (модулю)

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
---	-----------	--

1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
4.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.