

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

«29» мая 2026г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.18 Основы программирования

Направление подготовки 09.03.04 Программная инженерия

Направленность (профиль) Искусственный интеллект и аналитика данных

Форма обучения очная

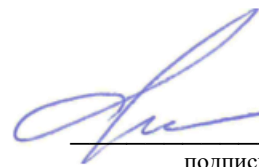
Квалификация бакалавр

Краснодар 2026

Рабочая программа дисциплины «Б1.О.18 Основы программирования» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.04 Программная инженерия.

Программу составил(и):

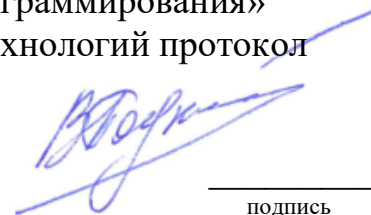
Н.Ю. Добровольская, доцент КИТ, к.п.н., доцент
И.О. Фамилия, должность, ученая степень, ученое звание


подпись

Рабочая программа дисциплины «Б1.О.18 Основы программирования» обсуждена на заседании кафедры информационных технологий протокол №20 от «12» мая 2026г.

Заведующий кафедрой (разработчика)

В. В. Подколзин


подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол №3 от «15» мая 2026 г.

Председатель УМК факультета

Н. Ю. Добровольская


подпись

Рецензенты:

Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг»,
e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, доктор экономических наук, кандидат технических наук, профессор кафедры компьютерных технологий и систем Федерального государственного бюджетного образовательного учреждения высшего образования «Кубанский государственный аграрный университет имени И.Т. Трубилина», e-mail: prof.lutsenko@gmail.com

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Формирование у студентов навыков программирования на C++ с акцентом на базовые конструкции языка, работу с памятью, структурами данных и основами объектно-ориентированного программирования (ООП), необходимых для дальнейшего изучения алгоритмов и машинного обучения.

1.2 Задачи дисциплины

Освоение синтаксиса C++: переменные, операторы, управляющие конструкции.
Изучение указателей, ссылок, динамического управления памятью.
Работа с массивами, строками, структурами данных (векторы, списки).
Основы ООП: классы, инкапсуляция, наследование, полиморфизм.
Разработка и отладка программ с использованием современных IDE.

1.3 Место дисциплины (модуля) в структуре образовательной программы

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

Пререквизиты:

Школьная информатика, Элементарная алгебра и основы логики.

Постреквизиты:

Б1.О.19 Алгоритмы и структуры данных (PL-4.1(Б)), Б1.О.24 Объектно-ориентированное программирование (PL-2.1(Б)), Б1.О.32 Параллельное и низкоуровневое программирование (PL-4.1(Б), PL-4.2(Б)), Б1.О.35 Обработка данных на Python (PL-1.4(Б), LLM-2.1(Б)), Б1.В.04 Математические модели нейронных сетей (DL-1.1(Б), DL-1.2(Б)), Б1.В.11 А/В-тестирование и Uplift-моделирование (MF-1.2(Б), ML-2.1(Б)), Б1.В.14 Введение в специальность (LLM-5.1(Б)), Б1.В.17 Технологии компьютерного зрения (PL-4.1(Б)), Б2.О.01.01(У) Технологическая (проектно-технологическая) практика (PL-4.1(Б), PL-4.2(Б)), Б1.О.27 Основы прикладной статистики (MF-1.2(С)), Б1.В.05 Методы ИИ в задачах классификации (MF-1.1(С), MF-2.1(С)), Б1.В.12 Машинное обучение (MF-3.1(С), ML-5.2(С)), Б1.В.13 Разработка ИИ агентов (LLM-4.1(С)), Б1.В.16 Нейросетевые технологии (DL-1.1(С)), Б1.В.18 Технологии создания и поддержки ПО (LC-4.1(С), LC-4.2(С)), Б1.В.20 Интеллектуальные методы оптимизации (MF-3.1(С), DL-1.1(С)), Б1.В.ДВ.01.01 Мультиагентные системы (LLM-4.1(П), LLM-4.2(П), LLM-5.4(П)), Б1.В.ДВ.01.02 Инженерия интеллектуальных агентов (LLM-4.1(П), LLM-4.2(П), LLM-5.4(П)), Б1.В.06 Современные методы фронтальных исследований ИИ (LLM-5.3(П), LLM-5.4(П)), Б1.В.08 Методы обучения с подкреплением (LLM-4.1(П)), Б3.01(Г) Подготовка к сдаче и сдача государственного экзамена (PL-4.1(П), PL-4.2(П)), Б3.02(Д) Выполнение и защита ВКР (PL-4.1(П)).

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: Data Analyst (Аналитик данных)

Задачи:

1. Статистический анализ, визуализация данных, предварительная обработка.
2. Создание прогнозных моделей
3. Построение аналитических моделей для поддержки бизнес-решений.

Роль 2: MLOps (Специалист по эксплуатации ИИ)

Задачи:

1. DevOps для ML.
2. Автоматизация, мониторинг ML-систем.
3. Операционное управление жизненным циклом ML-моделей.

Роль 3: AI PM (Менеджер проектов ИИ)

Задачи:

1. Управление ИИ-проектами от идеи до внедрения
2. Анализ бизнес-требований и постановка задач
3. Оценка эффективности и ROI ИИ-решений

1.5 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 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 ***Применяет современные информационные технологии и программные средства, в том числе отечественные, при определении требований и структуры программных продуктов и программных комплексов различного назначения***

Знать Знать современные отечественные и зарубежные информационные технологии и программные средства, используемые на этапах анализа, проектирования и разработки программных продуктов.

Уметь Уметь применять изученные технологии и средства для формализации требований, проектирования структуры и создания проектных решений программных комплексов

Владеть Владеть навыками выбора и использования современных (в том числе отечественных) инструментов для поддержки процессов определения требований и проектирования архитектуры ПО

ОПК-2.2 ***Применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности***

Знать Знать основные методологии, модели, этапы и инструменты жизненного цикла программного обеспечения, применяемые при проектировании, разработке и реализации программных продуктов.

Уметь Уметь выбирать и применять адекватные методы и инструменты для решения задач проектирования, разработки и интеграции программных модулей в рамках поставленной задачи из конкретной предметной области

Владеть Владеть навыками комплексного применения методов проектирования (например, структурного или объектно-ориентированного), современных инструментов разработки и практик реализации для создания работоспособного программного продукта, отвечающего исходным требованиям

ОПК-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОПК-3.1	<i>Способен осуществлять поиск и анализ информации на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности</i>
Знать	Основы информационного поиска: Принципы работы поисковых систем (Google, научные базы данных). Критерии оценки достоверности источников. Основные платформы для поиска технической документации (GitHub). Библиографическая культура: Правила цитирования и оформления ссылок. Виды источников (учебники, научные статьи, официальная документация). Информационная безопасность: Основные угрозы (вирусы, фишинг, утечка данных). Принципы безопасной работы в интернете
Уметь	Работа с информацией: Формулировать поисковые запросы для решения задач по программированию. Анализировать и сравнивать информацию из разных источников. Выделять ключевые идеи в технической документации. Использование ИКТ: Применять специализированные ресурсы для программистов (GitHub, онлайн-компиляторы). Работать с системами контроля версий (базовые команды Git). Безопасность: Создавать надежные пароли. Распознавать потенциально опасные ресурсы
Владеть	Навыками поиска: Эффективный поиск решений типовых задач на C++. Поиск и анализ примеров кода. Методами обработки информации: Конспектирование ключевых моментов. Сравнительный анализ разных подходов к решению задачи. Базовыми практиками безопасности: Безопасное хранение паролей. Проверка скачиваемых файлов
ОПК-3.2	<i>Способен участвовать в разработке и реализации программных продуктов на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности</i>
Знать	Основы прикладной математики: элементы линейной алгебры (матрицы, векторы), Основы математического анализа (производные, интегралы), Простейшие численные методы. Базовые концепции ПО: Основные парадигмы программирования (ООП, процедурное), Принципы работы алгоритмов (сложность, эффективность), Основные структуры данных (массивы, списки, деревья)
Уметь	Применять простейшие математические методы в программах. Реализовывать базовые алгоритмы на языке программирования. Анализировать простые технические задания. Работать с учебной литературой и документацией
Владеть	Навыками реализации математических формул в коде. Основными приемами отладки программ. Простейшими методами оптимизации кода

ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов
ОПК-6.1	<i>Разрабатывает алгоритмы для решения задач в области искусственного интеллекта и анализа данных, используя языки программирования</i>
Знать	Основные принципы разработки алгоритмов для задач искусственного интеллекта и анализа данных, фундаментальные структуры данных и методы их обработки, базовые алгоритмы машинного обучения и анализа данных, особенности реализации алгоритмов на различных языках программирования
Уметь	Формулировать задачи в терминах алгоритмических решений, выбирать подходящие структуры данных и алгоритмы для конкретных задач, реализовывать базовые алгоритмы искусственного интеллекта и анализа данных, оценивать эффективность и производительность разработанных алгоритмов.
Владеть	Навыками реализации алгоритмов на выбранном языке программирования, методами тестирования и отладки алгоритмических решений, приемами оптимизации алгоритмов по времени выполнения и используемой памяти, основами документирования алгоритмических решений.
ОПК-6.2	<i>Применяет принципы тестирования и отладки программного кода, обеспечивая его работоспособность и эффективность</i>
Знать	Основные принципы и методы тестирования программного обеспечения, виды и уровни тестирования, стратегии отладки кода, типовые ошибки программирования и способы их выявления, инструменты для тестирования и отладки, метрики оценки качества кода.
Уметь	Разрабатывать тестовые сценарии и наборы тестовых данных, применять различные методы тестирования, использовать инструменты отладки, анализировать результаты тестирования, локализовывать и исправлять ошибки в коде, оценивать эффективность программного кода.
Владеть	Навыками системного тестирования программных модулей, методами пошаговой отладки, техниками рефакторинга кода, приемами оптимизации производительности, способами документирования результатов тестирования.
PL-4	Способен применять языки программирования C/C++ для решения задач в области ИИ
PL-4.1	<i>Разрабатывает и отлаживает эффективные многопоточные решения на C++, тестирует, испытывает и оценивает качество таких решений</i> Знать: Понятие потока выполнения, способы создания и ожидания завершения потока посредством std::thread, назначение мьютекса для защиты разделяемых данных от одновременного доступа.

Уметь: Разрабатывать корректные программы с использованием изученных структур данных (массивы, матрицы, строки, структуры) без учёта проблем параллельного доступа к памяти.

Владеть: Базовыми навыками отладки последовательных программ с помощью точек останова и пошагового выполнения, обеспечивающими контроль линейного порядка исполнения операторов.

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:		55,8	55,8					
Курсовая работа								
Проработка учебного (теоретического) материала		25,8	25,8					
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)		30	30					
Реферат								
Подготовка к текущему контролю								
Контроль:		35,7	35,7					
Подготовка к экзамену		35,7	35,7					
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	СРС
1	2	3	4	5	6	7
1.	Введение в C++. Базовый синтаксис	32	10		12	10
2.	Управляющие конструкции, функции	34	10		14	10
3.	Указатели, ссылки, динамическая память.	32	10		12	10
4.	Массивы, матрицы	30	8		12	10
5.	Строки	20	4		8	8
6.	Структуры	17,8	6		8	5,8
7.	Введение в ИИ	6	2		2	2
ИТОГО по разделам дисциплины		173,8	50		68	55,8
Контроль самостоятельной работы (КСР)		6				
Промежуточная аттестация (ИКР)		0,5				
Подготовка к текущему контролю		35,7				
Общая трудоемкость по дисциплине		216				

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

Матрица формирования компетенций по разделам дисциплины

№ раздела	Наименование раздела (темы)	Формируемые индикаторы компетенций	Формы текущего контроля
1	Введение в C++. Базовый синтаксис	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1 (базовый)	Лаб. работа №1, контрольные задания
2	Управляющие конструкции, функции	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1 (базовый)	Лаб. работы №2, №3, контрольные задания
3	Указатели, ссылки, динамическая память	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1 (базовый)	Лаб. работы №4, №5, контрольные задания
4	Массивы, матрицы	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1 (базовый)	Лаб. работа №6, контрольные задания
5	Строки	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1 (базовый)	Лаб. работа №7, контрольные задания
6	Структуры	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1 (базовый)	Лаб. работа №8, контрольные задания
7	Введение в ИИ	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1 (базовый)	Лаб. работа №9, проектное задание

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.	Введение в C++. Базовый синтаксис	Основные типы данных, их размеры. Операторы ввода/вывода, форматирование. Область видимости переменных. Преобразование типов.	К
2.	Управляющие конструкции, функции	Условные операторы (if-else, switch-case). Циклы (for, while, do-while). Функции: передача параметров, возврат значений. Рекурсия. Перегрузка функций.	К
3.	Указатели, ссылки, динамическая память.	Указатели: арифметика, разыменование. Ссылки (отличие от указателей). Динамическая память (new, delete). Умные указатели (unique_ptr, shared_ptr). Способы передачи параметров в функции.	К, РЗ
4.	Массивы, матрицы	Одномерные и многомерные массивы. Контейнер vector: методы, преимущества. Алгоритмы обработки массивов.	К, РЗ
5.	Строки	Строки в стиле C (char[]) и C++ (std::string)	К, РЗ
6.	Структуры	Структуры (struct) и их использование.	К, РЗ
7.	Введение с ИИ	Использование библиотеки FastAI (упрощённый интерфейс для C++) для решения задачи классификации.	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.

2.3.2 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.	Введение в C++. Базовый синтаксис	ЛБ1 Основы программирования на C++	ЛР
2.	Управляющие конструкции, функции	ЛБ2 Условные операторы и циклы. ЛБ3 Функции и рекурсия	ЛР
3.	Указатели, ссылки, динамическая память.	ЛБ4 Работа с указателями и ссылками. ЛБ5 Умные указатели и передача параметров	ЛР
4.	Массивы, матрицы	ЛБ6 Одномерные и двумерные массивы.	ЛР
5.	Строки	ЛБ7 Строки и STL-контейнеры	ЛР
6.	Структуры	ЛБ 8 Обработка структур	ЛР
7.	Введение с ИИ	ЛБ 9 Введение в ИИ с помощью простой библиотеки на C++	ЛР

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 предусмотрено

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 вышеназванных 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»	18
Итого			18

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 «название дисциплины».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме заданий по темам, коллоквиума и контрольным работам, и **промежуточной аттестации** в форме вопросов и заданий **к экзамену и зачету**.

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	Введение в C++. Базовый синтаксис	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1	Типовые контрольные задания 1-7	Вопрос на экзамене 1-5
2	Управляющие конструкции, функции	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2 РЛ-4.1	Типовые контрольные задания 8-14	Вопрос на экзамене 6-10
3	Указатели, ссылки, динамическая память.	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-3.1, РЛ-3.2	Типовые контрольные задания 15-23	Вопрос на экзамене 11-15
4	Массивы, матрицы	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1	Типовые контрольные задания 24-28, 45-62	Вопрос на экзамене 16-17, 19
5	Строки	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2 РЛ-4.1	Типовые контрольные задания 29-35	Вопрос на экзамене 18
6	Структуры	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2 РЛ-4.1	Типовые контрольные задания 42-44	Вопрос на экзамене 20
7	Введение в ИИ	ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1	Типовые контрольные задания 63	Вопрос на экзамене 21

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 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 Применяет современные информационные технологии и программные средства, в том числе отечественные, при определении требований и структуры программных продуктов и программных комплексов различного назначения

Знать Знать назначение и основные возможности современных сред разработки (IDE), включая российские аналоги, для написания и отладки простых программ.

Уметь Уметь использовать базовые функции среды разработки (создание проекта, компиляция, отладка по шагам) для реализации простейших алгоритмов согласно формальному условию задачи (требованию).

Владеть Владеть на элементарном уровне навыками работы в интегрированной среде разработки для создания, запуска и отладки консольных приложений, решающих типовые вычислительные задачи.

ОПК-2.2 Применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности

Знать Знать базовые принципы структурного проектирования: последовательность выполнения, понятие алгоритма, элементарные структуры данных (переменные, массивы) и этапы создания простой программы (от идеи до исполняемого кода)

Уметь Уметь применять метод пошаговой детализации (сверху вниз) для разбиения простой задачи на последовательность действий и реализовывать её в виде линейной программы или программы с ветвлениями и циклами.

Владеть Владеть навыками реализации законченных консольных приложений, решающих типовые вычислительные задачи (расчёт формул, обработка массивов) на основе простейшего технического задания.

ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

ОПК-6.1 Разрабатывает алгоритмы для решения задач в области искусственного интеллекта и анализа данных, используя языки программирования

Знать Базовые понятия алгоритмов и структур данных, элементарные алгоритмы обработки информации, основные принципы составления программ на C++, простейшие методы оценки эффективности кода

Уметь Реализовывать линейные алгоритмы на C++, работать с базовыми структурами данных (массивы, строки), использовать стандартные конструкции языка для решения простых задач, находить и исправлять синтаксические ошибки.

Владеть Навыками написания простейших программ на C++, приемами ручной проверки кода, основами тестирования на элементарных примерах.

ОПК-6.2 *Применяет принципы тестирования и отладки программного кода, обеспечивая его работоспособность и эффективность*

Знать Базовые понятия тестирования программ, простейшие виды ошибок в коде, основные принципы отладки, элементарные приемы проверки работоспособности программ.

Уметь Проверять работу простых программ на корректность, использовать базовые средства вывода информации для контроля выполнения, находить и исправлять синтаксические ошибки, составлять простейшие тестовые примеры.

Владеть Навыками запуска и проверки программ на C++, приемами выявления очевидных ошибок, основами использования отладочного вывода.

PL-4 **Способен применять языки программирования C/C++ для решения задач в области ИИ**

PL-4.1 *Разрабатывает и отлаживает эффективные многопоточные решения на C++, тестирует, испытывает и оценивает качество таких решений*

Знать: Понятие потока выполнения, способы создания и ожидания завершения потока посредством `std::thread`, назначение мьютекса для защиты разделяемых данных от одновременного доступа.

Уметь: Разрабатывать корректные программы с использованием изученных структур данных (массивы, матрицы, строки, структуры) без учёта проблем параллельного доступа к памяти.

Владеть: Базовыми навыками отладки последовательных программ с помощью точек останова и пошагового выполнения, обеспечивающими контроль линейного порядка исполнения операторов.

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

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 *Применяет современные информационные технологии и программные средства, в том числе отечественные, при определении требований и структуры программных продуктов и программных комплексов различного назначения*

Знать Знать основные принципы структурного программирования и уметь их соотносить с инструментами среды разработки (организация модулей, отладка функций), знать возможности систем контроля версий (например, Git) для управления кодом

Уметь Уметь применять среду разработки для проектирования структуры программы (разбиение на функции/методы, модули), использовать отладчик

для анализа сложных потоков выполнения и данных, применять систему контроля версий для ведения индивидуального проекта..

Владеть Владеть навыками комплексного использования инструментов IDE (профилировщик, условные точки останова) и базовыми навыками работы с системой контроля версий для реализации и рефакторинга многофайловых программ, соответствующих формализованному техническому заданию..

ОПК-2.2 *Применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности*

Знать Знать методы процедурного (модульного) программирования: принцип декомпозиции, проектирование интерфейсов функций/процедур, передача параметров, основы работы с файлами

Уметь Уметь применять метод модульного проектирования для разработки структуры программы, разбивая её на независимые функции, и реализовывать многофайловый проект для решения прикладной задачи из определённой области (например, обработка данных, простые вычисления).

Владеть Владеть навыками проектирования и реализации многофайлового консольного приложения с чётким разделением на модули (ввод/вывод, логика, вычисления), использующего файлы для хранения данных.

ОПК-6 **Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов**

ОПК-6.1 *Разрабатывает алгоритмы для решения задач в области искусственного интеллекта и анализа данных, используя языки программирования*

Знать Основные алгоритмы сортировки и поиска, принципы работы с динамическими структурами данных (списки, стеки, очереди), основы объектно-ориентированного программирования на C++, базовые методы оценки сложности алгоритмов.

Уметь Реализовывать типовые алгоритмы обработки данных, выбирать подходящие структуры данных для решения задач, анализировать работу алгоритмов, проводить сравнительный анализ простых алгоритмических решений.

Владеть Навыками отладки программ средней сложности, методами тестирования алгоритмов на различных наборах данных, приемами оптимизации кода по времени выполнения.

ОПК-6.2 *Применяет принципы тестирования и отладки программного кода, обеспечивая его работоспособность и эффективность*

Знать Основные виды тестирования (модульное, интеграционное), принципы разработки тестовых случаев, базовые методы отладки, понятие граничных условий.

Уметь Разрабатывать простые тесты для проверки алгоритмов, использовать отладчик для пошагового выполнения, анализировать стек вызовов, тестировать обработку граничных условий, оценивать корректность работы структур данных.

Владеть Навыками работы с отладчиком, методами тестирования отдельных функций, приемами проверки обработки исключительных ситуаций.

PL-4 **Способен применять языки программирования C/C++ для решения задач в области ИИ**

PL-4.1 **Разрабатывает и отлаживает эффективные многопоточные решения на C++, тестирует, испытывает и оценивает качество таких решений**

Знать: Отличительные признаки состояний гонки и взаимоблокировок (дедлоков), принципы функционирования атомарных переменных и стандартных потокобезопасных контейнеров из библиотеки STL.

Уметь: Реализовывать алгоритмы обработки графов (обход в глубину и ширину) и матриц с использованием функций и указателей, гарантируя корректность работы в однопоточном режиме.

Владеть: Навыками написания и тестирования программ со сложной логикой ветвления, циклами и рекурсивными вызовами без возникновения ошибок, связанных с параллелизмом.

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

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 **Применяет современные информационные технологии и программные средства, в том числе отечественные, при определении требований и структуры программных продуктов и программных комплексов различного назначения**

Знать Знать современные подходы и инструменты для проектирования архитектуры (например, UML-диаграммы классов), принципы тестирования (модульное тестирование) и возможности продвинутых сред разработки или фреймворков (в т.ч. отечественных), поддерживающих эти практики.

Уметь Уметь применять инструменты проектирования (UML-редакторы) для моделирования структуры программного продукта, использовать фреймворки модульного тестирования для верификации требований к отдельным компонентам, интегрировать различные программные средства в процесс разработки.

Владеть Владеть навыками применения инструментальной цепочки (среда проектирования -> среда разработки -> система контроля версий -> система тестирования) для реализации законченного программного модуля или приложения, начиная от анализа требований и заканчивая демонстрацией работоспособного прототипа.

ОПК-2.2 *Применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности*

Знать Знать основные принципы объектно-ориентированного проектирования (инкапсуляция, наследование, полиморфизм) и их связь с моделированием сущностей предметной области, а также основы архитектурных паттернов.

Уметь Уметь применять методы ООП для анализа предметной области, проектирования классов и их взаимодействия, а также реализовывать оконное или веб-приложение с простым графическим интерфейсом, имитирующее реальный процесс

Владеть Владеть навыками полного цикла создания программного продукта: от анализа упрощённого технического задания и построения диаграммы классов до реализации, отладки и демонстрации работоспособного приложения с графическим интерфейсом для заданной предметной области.

ОПК-6 *Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов*

ОПК-6.1 *Разрабатывает алгоритмы для решения задач в области искусственного интеллекта и анализа данных, используя языки программирования*

Знать Сложные алгоритмы обработки данных (графовые алгоритмы, деревья поиска), методы динамического программирования, рекурсивные алгоритмы, продвинутые техники работы с памятью в C++, принципы многомодульной разработки.

Уметь Разрабатывать эффективные алгоритмы для нестандартных задач, комбинировать различные структуры данных, применять методы оптимизации производительности, реализовывать сложные алгоритмические конструкции.

Владеть Навыками профилирования и оптимизации кода, методами тестирования сложных алгоритмов, техниками работы с большими объемами данных, приемами документирования алгоритмических решений.

ОПК-6.2 *Применяет принципы тестирования и отладки программного кода, обеспечивая его работоспособность и эффективность*

Знать Методы автоматизированного тестирования, техники white-box и black-box тестирования, принципы тестирования производительности, передовые практики отладки сложных алгоритмов.

Уметь Создавать комплексные тестовые наборы, разрабатывать тесты для проверки сложных структур данных, использовать профилировщик для анализа производительности, применять методы регрессионного тестирования, оптимизировать алгоритмы по результатам тестирования.

Владеть Навыками автоматизации тестирования, методами профилирования кода, техниками отладки рекурсивных алгоритмов, приемами тестирования на больших объемах данных.

PL-4 Способен применять языки программирования C/C++ для решения задач в области ИИ

PL-4.1 Разрабатывает и отлаживает эффективные многопоточные решения на C++, тестирует, испытывает и оценивает качество таких решений

Знать: Архитектурные паттерны организации многопоточности (производитель-потребитель, читатели-писатели, модель акторов), а также влияние иерархии кэш-памяти процессора на производительность параллельного кода.

Уметь: Проектировать потокобезопасные структуры данных на уровне архитектурных решений, корректно разделяя данные между гипотетическими потоками и избегая конфликтов доступа.

Владеть: Навыками анализа последовательного кода на предмет потенциальных проблем при его адаптации к многопоточной среде, а также методами оценки эффективности параллельной обработки изученных структур данных (массивы, графы, матрицы)..

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. Дана последовательность из N целых чисел. Найти количество положительных трехзначных чисел, не оканчивающихся на 5.
2. Дана последовательность из N целых чисел. Найти сумму модулей отрицательных чисел. Если таких нет, сообщить об этом.
3. Дана последовательность из N целых чисел. Найти произведение индексов нечетных чисел.
4. Дана последовательность из N целых чисел. Верно ли, что последовательность является возрастающей.
5. Дана последовательность из N целых чисел. Найти сумму чисел, оканчивающихся на 5, перед которыми идет четное число.
6. Дана последовательность целых чисел, оканчивающаяся числом 0. Найти сумму положительных двузначных чисел, за которыми идет оканчивающееся на 11 число.
7. Дана последовательность целых чисел, оканчивающаяся числом 0. Найти количество троек соседних элементов, где каждое следующее число больше предыдущего.
8. Дана последовательность из N целых чисел. Найти порядковые номера двух соседних элементов, сумма которых максимальна.
9. Вычислить
$$\sum_{k=1}^N \frac{(-1)^{k+1}(4+k!)}{y^{k+3}}$$
10. Вычислить
$$\sum_{k=1}^N \frac{(-1)^{k+1}(y^{2k}+1)}{2^{2k-1}+(k+1)!}$$
11. Вычислить $e^y = \sum_{k=0}^{\infty} \frac{y^k}{k!}$ с заданной точностью

12. Дана последовательность из N целых чисел. Найти произведение индексов простых чисел.
13. Дана последовательность из N целых чисел. Найти количество чисел с четной суммой цифр.
14. Дана последовательность целых чисел, оканчивающаяся числом 0. Найти количество чисел, в записи которых нет нулей.
15. Дана последовательность целых чисел, оканчивающаяся числом 0. Найти произведение чисел, в записи которых количество единиц и двоек нечетно, а до таких чисел следует отрицательное число.
16. Дан массив целых чисел. Найти количество положительных элементов, расположенных на позициях не кратных 3.
17. Дан массив целых чисел. Найти сумму элементов, у которых последняя и предпоследняя цифры равны.
18. Дан массив целых чисел. Заменить отрицательные элементы на сумму индексов двузначных элементов массива.
19. Дан массив целых чисел. Найти количество пар соседних элементов, где первый элемент вдвое больше второго.
20. Дан массив целых чисел. Верно ли, что массив является знакочередующимся.
21. Дан массив целых чисел. Верно ли, что массив является симметричным.
22. Дан массив целых чисел. Найти сумму непростых элементов, расположенных на простых позициях.
23. Дан массив целых чисел. Найти произведение элементов, в записи которых нет нулей.
24. Дан массив целых чисел. Заменить на максимальный элемент все элементы, у которых число четных делителей больше двух.
25. Дан массив целых чисел. Увеличить все элементы, в записи которых только цифры «2», «3» и «5» на количество совершенных элементов массива.
26. Дан массив целых чисел. Найти сумму элементов, являющихся числами Фибоначчи и расположенных на четных позициях.
27. Дан массив целых чисел. Заменить каждый несовершенный элемент массива на количество его нечетных цифр.
28. Дана квадратная матрица целых чисел. Найти сумму модулей отрицательных элементов, расположенных ниже главной диагонали.
29. Дана квадратная матрица целых чисел. Найти количество делящихся на 3 элементов, расположенных на побочной диагонали.
30. Дана квадратная матрица целых чисел. Найти наименьший элемент побочной диагонали.
31. Дана квадратная матрица целых чисел. Заменить элементы, лежащие выше побочной диагонали на количество положительных элементов главной диагонали.
32. Дана квадратная матрица целых чисел. Увеличить нечетные элементы побочной диагонали на сумму индексов элементов, лежащих ниже главной диагонали.
33. Дана квадратная матрица целых чисел. Найти сумму непростых элементов, расположенных ниже побочной диагонали.
34. Дана квадратная матрица целых чисел. Найти количество элементов, имеющих четную сумму цифр и расположенных ниже главной диагонали.
35. Дана квадратная матрица целых чисел. Заменить нулем элементы главной диагонали, у которых больше двух нечетных делителя (исключая 1 и само число).
36. Дана квадратная матрица целых чисел. Найти наибольший элемент главной диагонали, в записи которого есть цифры «2» и «3».
37. Дана неквадратная матрица целых чисел. Сформировать одномерный массив, элемент которого равен сумме модулей отрицательных элементов каждой строки.

38. Дана неквадратная матрица целых чисел. Сформировать одномерный логический массив, элемент которого равен true, если в строке нет положительных элементов. Иначе равен false.
39. Дана неквадратная матрица целых чисел. Поменять местами первую строку и строку с максимальным элементом.
40. Дана неквадратная матрица целых чисел. Поменять местами столбец с наибольшей суммой элементов и последний столбец.
41. Удалить из матрицы все строки содержащие числа Фибоначчи. Сформировать новую матрицу.
42. Описать структуру с именем ORDER, содержащую следующие поля: расчетный счет плательщика; расчетный счет получателя; перечисляемая сумма в руб. Вывести информацию о платежах, сумма которых не максимальна, а расчётный счет плательщика и получателя содержат вместе не более пяти нулей.
43. Описать структуру с именем PLANE, содержащую следующие поля: название пункта назначения рейса; номер рейса; время в пути. Вывести информацию о рейсах, время в пути, у которых больше среднего на один час, а в номере рейса нет цифр один и три.
44. Описать структуру с именем PRICE, содержащую следующие поля: название товара; тип товар; стоимость товара в руб. Вывести информацию о товарах, цена которых отличается от средней более чем на 100 рублей, а в названии ровно три буквы «а».
45. Вычислить сумму квадратов простых чисел, лежащих в интервале $[M, N]$.
46. Определить функцию нахождения расстояния между точками. Во множестве точек на плоскости найти сумму расстоянием между ними.
47. Найти: $y = (\min(a, b) + \min(b, c)) / \min(a, c)$.
48. Дан массив целых чисел. Если в массиве нет простых элементов, то каждый положительный элемент преобразовать по правилу: записать в обратном следовании цифр, исключая нули. В функции передавать параметр по ссылке.
49. Дан массив целых чисел. Если в массиве все элементы положительные и не менее чем двузначные, то каждый элемент преобразовать по правилу: поменять местами две последние цифры. В функции передавать параметр по ссылке. Затем найти в массиве наибольший элемент.
50. Дана матрица. Все положительные элементы преобразовать по правилу: четные цифры заменить на 1, а нечетные увеличить на 1, цифру 9 заменить 0. В функции передавать параметр по ссылке. Затем найти номер строки с наибольшим элементом.
51. У исполнителя Калькулятор две команды, которым присвоены номера:
 - 1) прибавь 1
 - 2) умножь на 3

Сколько есть программ, которые число 2 преобразуют в число 28?

52. У исполнителя Калькулятор три команды, которым присвоены номера:
 - 1) прибавь 1
 - 2) умножь на 2
 - 3) умножь на 4

Сколько есть программ, которые число 1 преобразуют в число 17?

53. Дано число. Найти количество цифр.
54. Дано число. Найти сумму нечетных цифр.
55. Верно ли, что в числе все цифры четные?
56. Верно ли, что в числе нет 2 и 5?
57. Дан массив целых чисел. Увеличить двузначные элементы на значение суммы четных элементов. Оформить рекурсивную функцию (сумма) и рекурсивную процедуру (увеличение). Наличие основной программы обязательно.

58. Дано два массива. Если они оба упорядочены по возрастанию (функция), то увеличить четные элементы первого массива и нечетные элементы другого, на среднее арифметическое соответствующего массива.
59. Дано два массива. Найти сумму максимальных элементов двух массивов (функция).
60. Дано два массива. Если они оба упорядочены по возрастанию, то заменить в этих массивах отрицательные элементы их модулями.
61. Дан динамический массив. Если массив не симметричен, то заменить четные элементы значением их квадрата, а нечетные элементы, не содержащие в своей записи цифры семь, значением заданного элемента x .
62. Дан динамический массив. Заменить все простые элементы значением их индекса.
63. С помощью библиотеки FastAI обучите модель предсказывать вес по возрасту.

Типовые лабораторные работы

Лабораторная работа 1: "Основы программирования на C++"

Написать программу, вычисляющую арифметические выражения с разными типами данных.

Использовать `cin/cout` для ввода/вывода.

Продемонстрировать явное и неявное преобразование типов.

Лабораторная работа 2: "Условные операторы и циклы"

Реализовать программу, проверяющую число на простоту.

Написать калькулятор с использованием `switch-case`.

Вывести таблицу умножения с помощью вложенных циклов.

Лабораторная работа 3: "Функции и рекурсия"

Написать функцию вычисления факториала (итеративно и рекурсивно).

Реализовать функцию поиска НОД (алгоритм Евклида).

Создать программу, генерирующую числа Фибоначчи.

Лабораторная работа 4: "Работа с указателями и ссылками"

Поменять значения двух переменных через указатели и ссылки.

Написать функцию, возвращающую минимальный и максимальный элементы массива через указатели.

Реализовать динамический массив с ручным управлением памятью (`new`, `delete`).

Лабораторная работа 5: "Умные указатели и передача параметров"

Создать класс, использующий `unique_ptr` для управления ресурсом.

Написать функцию, принимающую параметры по значению, ссылке и указателю.

Сравнить производительность разных способов передачи данных.

Лабораторная работа 6: "Одномерные и двумерные массивы"

Реализовать функции для работы с матрицами (сложение, умножение).

Написать программу, транспонирующую матрицу.

Отсортировать массив методом пузырька.

Лабораторная работа 7: "Строки и STL-контейнеры"

Написать программу, подсчитывающую количество слов в строке.

Использовать `std::vector` для хранения и обработки данных.

Реализовать поиск подстроки в строке.

Лабораторная работа 8: "Обработка структур"

Создать структуру "Студент" (ФИО, группа, оценки).

Записать массив структур с клавиатуры в массив.
Реализовать поиск студента по фамилии в массиве структур.

Лабораторная работа 9: "Введение в ИИ с помощью простой библиотеки на C++"

Использование библиотеки FastAI (упрощённый интерфейс для C++) для решения задачи классификации.

Знакомство с вызовом готовой ИИ-библиотеки и выполнить простейшее предсказание.

1. Теоретическая часть

Искусственный интеллект (ИИ) — это алгоритмы, которые могут обучаться на данных и делать предсказания.

Библиотека FastAI — упрощённый инструмент для работы с ИИ (аналог scikit-learn, но для C++).

Создадим массив данных (рост → вес человека).

Обучим модель предсказывать вес по росту.

Сделаем предсказание для нового значения.

2. Практическая часть

Шаг 1: Подготовка данных

Создадим два массива:

height — рост (в см).

weight — вес (в кг).

Шаг 2: Подключение библиотеки FastAI

Установка:

Скачайте FastAI для C++ (только заголовочные файлы).

Типовой вариант коллоквиума

Теоретическая часть

Вопрос 1:

Что такое умные указатели (unique_ptr, shared_ptr)? В чем их преимущество перед обычными указателями?

Вопрос 2:

Объясните разницу между статическим и динамическим выделением памяти.

Практическая часть

Задача 1:

Напишите функцию, которая сортирует массив методом пузырька.

Задача 2:

Создайте структуру FileHandler для чтения и записи идентификаторов файлов, их пути, объема и даты создания.

Типовые задания для самостоятельной работы

1. Вычислить и вывести на экран значение.

$$y = a^3 \left(\sum_{i=1}^n (i!+5) + b^2 \prod_{j=1}^n (2^j + d^{2j+1}) \right)$$

2. а) Найти сумму элементов массива, содержащих нечетное количество цифры пять.
б) Найти произведение элементов массива, расположенных между максимальным и минимальным элементами.

3. Даны две целочисленные квадратные матрицы порядка n . Найти последовательность из нулей и единиц $b_1, \dots, b_n$ такую что $b_i=1$, когда i -е столбцы первой и второй матриц содержат вместе не менее двух ненулевых элементов и $b_i=0$ в противном случае.
4. Дан массив целых чисел. Если в массиве нет четных отрицательных элементов, то каждый положительный элемент преобразовать по правилу: удалить из элемента все четные цифры. В функции передавать параметр по ссылке.

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

Задания на зачет

Вариант 1

Теоретическая часть

Вопрос 1:

В чем разница между указателем и ссылкой в C++? Приведите примеры использования каждого.

Вопрос 2:

Что такое конструктор копирования? Когда он вызывается? Можно ли запретить копирование объекта?

Практическая часть

Задача 1:

Напишите функцию, которая принимает массив чисел и возвращает указатель на максимальный элемент.

Задача 2:

Создайте структуру Student с полями name (строка) и grades (массив оценок). Реализуйте функцию averageGrade(), вычисляющий средний балл.

Вариант 2

Теоретическая часть

Вопрос 1:

Что такое динамическая память в C++? Как правильно выделять и освобождать память?

Вопрос 2:

Объясните принцип работы виртуальных функций. Зачем нужны и в каких случаях применяются?

Практическая часть

Задача 1:

Напишите программу, которая реверсирует строку (без использования STL).

Задача 2:

Создайте шаблонную функцию swap, которая меняет местами два значения любого типа.

Вариант 3

Теоретическая часть

Вопрос 1:

Какие есть способы передачи параметров в функции? В чем разница между передачей по значению, ссылке и указателю?

Вопрос 2:

Как обрабатываются многомерные массивы?

Практическая часть

Задача 1:

Напишите функцию, которая проверяет, является ли строка палиндромом.

Задача 2:

Реализуйте структуру Matrix с операцией умножения матриц.

Вариант 4

Теоретическая часть

Вопрос 1:

Что такое перегрузка операторов? Приведите пример перегрузки оператора +.

Вопрос 2:

Чем отличаются локальные и глобальные переменные?

Практическая часть

Задача 1:

Напишите функцию, которая считает количество слов в строке.

Задача 2:

Вычислите медиану массива целых чисел.

Вопросы для подготовки к экзамену

1. Отличия C++ от C. Основные особенности C++.
2. Типы данных в C++. Размеры типов (int, float, double, char).
3. Область видимости переменных. Локальные и глобальные переменные.
4. Операторы ввода/вывода (cin, cout). Форматированный вывод (printf, cout с манипуляторами).
5. Преобразование типов: явное (static_cast) и неявное.
6. Условные операторы: if-else, switch-case.
7. Циклы: for, while, do-while. Операторы break, continue.
8. Функции: объявление, определение, передача параметров.
9. Рекурсивные функции. Примеры (факториал, числа Фибоначчи).
10. Перегрузка функций. Правила перегрузки.
11. Указатели: объявление, разыменование, арифметика указателей.
12. Ссылки в C++. Отличие ссылок от указателей.
13. Динамическое выделение памяти (new, delete). Утечки памяти.
14. Умные указатели (unique_ptr, shared_ptr). Зачем нужны?
15. Передача параметров в функции: по значению, указателю, ссылке.
16. Статические и динамические массивы. Инициализация.
17. Двумерные массивы (матрицы). Работа с вложенными циклами.
18. Строки в C (char[]) и C++ (std::string). Основные операции.
19. Алгоритмы работы с массивами (поиск, сортировка, реверс).
20. Структуры (struct) в C++. Отличие от классов.
21. Подключение и использование библиотеки FastAI

Перечень компетенций (части компетенции), проверяемых оценочным средством

ОПК-2.1, ОПК-2.2, ОПК-6.1, ОПК-6.2, РЛ-4.1

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++;
- продемонстрирована работоспособность приложения в среде Microsoft Visual Studio;
- студент понимает исходный код и отвечает на вопросы по его организации.

Методические рекомендации, определяющие процедуры оценивания самостоятельной работы:

Оценивание результатов самостоятельной работы основывается на качестве выполнения студентом индивидуального задания. Код приложения реализуется в среде Microsoft Visual Studio, на языке C++.

Критерии оценки:

оценка «неудовлетворительно»: код программы не запускается в компиляторе;

оценка «удовлетворительно»: программа работает, но реализует часть необходимого функционала или работает на некоторых наборах данных;

оценка «хорошо»: представлена структура программы и ее компонентов, программа работает практически на всех наборах данных, за исключением некоторых частных случаев, реализован весь функционал задания;

оценка «отлично»: представлена структура приложения и ее компонентов, программа работает на всех наборах данных, включая частные случаи и функционал задания реализован полностью.

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 вопросов коллоквиума, 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 контрольных заданий и заданий для самостоятельной работы.

Критерии оценки:

оценка «незачет» выставляется в случае выполнения одного из условий:

- самостоятельная работа оценена на «неудовлетворительно»;
- на коллоквиуме получено менее 50% верных ответов;
- выполнено менее 60% типовых контрольных заданий.

оценка «зачет» в случае выполнения условий:

- самостоятельная работа оценена не ниже чем на «удовлетворительно»;
- на коллоквиуме получено более 50% верных ответов
- выполнено не менее 60% типовых контрольных заданий.

Методические рекомендации, определяющие процедуры оценивания на экзамене:

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 допускаются к сдаче экзамена, только после получения зачета по дисциплине. Студенты обязаны сдать экзамен в соответствии с расписанием и учебным планом

ФОС промежуточной аттестации состоит из вопросов к экзамену, задач по дисциплине и результатов текущего контроля.

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

Форма проведения экзамена: устно.

Экзаменатору предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи экзамена заносится преподавателем в экзаменационную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и ответов на вопросы экзамена.

Критерии оценки:

оценка «неудовлетворительно» выставляется в случае выполнения одного из условий:

- непонимание сущности излагаемых вопросов, грубые ошибки в ответе, неуверенные и неточные ответы на дополнительные вопросы;
- самостоятельная работа оценена на «неудовлетворительно»;
- средний балл ответов на контрольные вопросы меньше 3,2;
- выполнено менее 50% контрольных заданий.

оценка «удовлетворительно» в случае выполнения условий:

- частично ответил на два вопроса билета или достаточно полно ответил хотя бы на один вопрос;
- самостоятельная работа оценена не ниже, чем на «удовлетворительно»;
- средний балл ответов на контрольные вопросы не меньше 3,6;
- выполнено не менее 60% контрольных заданий.

оценка «хорошо» в случае выполнения условий:

- достаточно полно ответил на два вопроса билета;
- даны частичные ответы на дополнительные вопросы;
- самостоятельная работа оценена не ниже, чем на «удовлетворительно»;
- средний балл ответов на контрольные вопросы не меньше 3,9;
- выполнено не менее 75% контрольных заданий.

оценка «отлично» в случае выполнения условий:

- глубокие исчерпывающие знания по вопросам билета;
- даны правильные и конкретные ответы на дополнительные вопросы;
- самостоятельная работа оценена не ниже, чем на «хорошо»;
- средний балл ответов на контрольные вопросы не меньше 4,4;
- выполнено не менее 90% контрольных заданий.

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.

4.3. Методические указания по организации вычислительной инфраструктуры

Условия применения:

- Курс рассчитан на студентов 1-го года обучения.
- Наличие доступа к вычислительным ресурсам (GitHub, GitLab).
- Индивидуальные задания включают задачи для проверки автотестами.

Инфраструктура для приёма задач (GitHub/GitLab, CI/CD, автотесты) согласована с заданиями.

Цели, задачи и ожидаемые результаты:

Цели:

- Ознакомить студентов с основами работы в IT-инфраструктуре (Git, требования к качеству кода, автотестирование).
- Научить использовать системы контроля версий и CI/CD для разработки на C++.

Задачи преподавателя:

- Создание учетных записей студентов в GitHub/GitLab.
- Настройка GitLab Runner или GitHub Actions для автоматического тестирования.
- Разработка шаблонного репозитория для лабораторных работ с предустановленной структурой и тестами.
- Написание автотестов для индивидуальных заданий.
- Подготовка инструкции по работе с Git и CI/CD.

Ожидаемые результаты студентов:

- Умение работать с Git (клонирование, коммиты, ветки).
- Навыки написания кода, соответствующего требованиям автотестов.
- Понимание основ CI/CD и автоматизированного тестирования.

Порядок реализации:

1. **Создание учетных записей:**
 - Регистрация студентов в GitHub/GitLab.
 - Настройка SSH-ключей для доступа.
 2. **Настройка CI/CD:**
 - Для GitLab: установка и настройка GitLab Runner.
 - Для GitHub: настройка GitHub Actions.
 - Конфигурация пайплайна для автоматического тестирования (например, `.gitlab-ci.yml` или `github-actions.yml`).
 3. **Шаблонный репозиторий:**

Включает:

 - `.gitignore` для C++.
 - `CMakeLists.txt` или `Makefile` для сборки.
 - Папку `tests` с автотестами (Google Test, Catch2).
 - `README.md` с инструкциями.
 4. **Автотесты:**
 - Пример: тестирование функций сортировки, работы с указателями, структур данных.
 - Визуализация результатов в виде HTML-отчётов.
- Порядок проверки корректности:**
Чек-лист:
- Наличие Git-репозитория у всех студентов.
 - Корректная работа CI/CD (успешные/неуспешные тесты).
 - Соответствие кода стандартам (именование, отступы, комментарии).

4.4. Методические указания по организации лабораторных работ

Условия применения:

- Курс рассчитан на студентов 1-го года обучения.
- Наличие доступа к компьютерам с установленными IDE (Visual Studio, CLion).
- Инфраструктура Git + CI/CD настроена для всех студентов.

Цели, задачи и ожидаемые результаты:

Цели:

- Научить студентов решать базовые задачи программирования на C++.
- Закрепить навыки работы с синтаксисом, указателями, структурами данных.

Задачи преподавателя:

- Разработка плана лабораторных работ (в соответствии с РПД, п. 2.3.3).
- Подготовка индивидуальных заданий с автотестами.
- Организация проверки через Git и CI/CD.

Ожидаемые результаты студентов:

- Умение писать, тестировать и отлаживать код на C++.
- Навыки работы с Git и CI/CD.

Порядок реализации:

1. План лабораторных работ:

- **ЛР1:** Основы C++ (ввод/вывод, типы данных).
- **ЛР2:** Управляющие конструкции (циклы, условия).
- **ЛР3:** Функции и рекурсия.
- **ЛР4:** Указатели и динамическая память.
- **ЛР5:** Массивы и строки.
- **ЛР6:** Структуры и файлы.
- **ЛР7:** Основы ООП (классы, инкапсуляция).

2. Пример индивидуального задания (ЛР4 — указатели):

Задача: Реализовать функцию `swap_values`, которая меняет местами значения двух переменных через указатели.

Автотесты: Проверка корректности обмена значений.

Контрольные вопросы:

- Чем отличается передача параметров по указателю от передачи по ссылке?
- Как избежать утечек памяти при работе с указателями?

3. Критерии оценки:

– **Отлично:** Полное выполнение задания, код проходит все тесты, ответы на вопросы.

– **Хорошо:** Незначительные ошибки, частичные ответы.

– **Удовлетворительно:** Основной функционал реализован, но есть недочёты.

– **Неудовлетворительно:** Код не работает или не соответствует ТЗ.

Порядок проверки корректности:

Чек-лист:

- Наличие репозитория с выполненными заданиями.
- Успешное прохождение автотестов.
- Соответствие кода стандартам.

4.5. Методические указания по организации проектной деятельности студентов

Условия применения:

- Курс рассчитан на студентов 1-го года обучения.
- Время на проект — до 16 часов на студента.
- Доступ к учебным кейсам.

Цели, задачи и ожидаемые результаты:

Цели:

- Применить знания C++ для решения практических задач.
- Развить навыки командной работы и проектного менеджмента.

Задачи преподавателя:

- Подготовка кейсов.
- Формирование ТЗ для экзаменационного проекта.
- Разработка системы оценки результатов.

Ожидаемые результаты студентов:

- Готовый проект на C++.
- Умение работать в команде и презентовать результаты.

Порядок реализации:

1. Примеры проектов:

- Калькулятор матриц с использованием ООП.
- Консольная игра (например, «Сапёр» или «Крестики-нолики»).
- Утилита для работы с файлами (шифрование, сортировка).

2. ТЗ для проекта «Калькулятор матриц»:

Реализовать класс `Matrix` с методами:

- Сложение, умножение матриц.
- Транспонирование, вычисление определителя.

Написать автотесты для проверки корректности.

3. Критерии оценки:

- **3 (удовлетворительно):** Реализован базовый функционал.
- **4 (хорошо):** Добавлены дополнительные методы (например, обратная матрица).
- **5 (отлично):** Оптимизация кода, обработка исключений, GUI (по желанию).

Порядок проверки корректности:

Чек-лист:

- Наличие репозитория с кодом и документацией.
- Соответствие ТЗ.
- Качество кода (читаемость, отсутствие утечек памяти).

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

5.1 Основная литература:

1. Ульянова, Н. Д. Основные принципы алгоритмизации : учебно-методическое пособие / Н. Д. Ульянова. — Брянск : Брянский ГАУ, 2024. — 56 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/172114> (дата обращения: 26.05.2026). — 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, 2024. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/473054> (дата обращения: 19.05.2026).

3. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2023. — 320 с. — (Высшее образование). — ISBN 978-5-534-02444-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/469579> (дата обращения: 19.05.2026).

5.2 Дополнительная литература:

1. 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: 19.05.2026). — Режим доступа: для авториз. пользователей.

2. 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: 19.05.2026). — Режим доступа: для авториз. пользователей.

3. 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: 19.05.2026). — Режим доступа: для авториз. пользователей.

Конференции A*:

1. <https://openreview.net/forum?id=FMMF1a9iFC>
2. <https://openreview.net/forum?id=ElUrNM9U8c#discussion>
3. <https://openreview.net/forum?id=JoO6mtCLHD>
4. <https://aclanthology.org/2024.findings-emnlp.760/>
5. <https://aclanthology.org/2020.coling-main.588/>
6. https://link.springer.com/chapter/10.1007/978-3-030-72113-8_30
7. https://link.springer.com/chapter/10.1007/978-3-031-42448-9_10
8. <https://aclanthology.org/2024.findings-naacl.288/>

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)

По курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал. В ходе лекционных занятий разбираются основные конструкции языка высокого уровня C++, базовые алгоритмические схемы и приемы программирования, приводятся примеры их использования, проводится анализ наиболее распространенных ошибок реализации. После прослушивания лекции рекомендуется выполнить упражнения, приводимые в аудитории для самостоятельной работы.

По курсу предусмотрено проведение лабораторных занятий, на которых дается прикладной систематизированный материал. В ходе занятий разбираются готовые примеры алгоритмических решений, примеры решения типовых задач, предлагаются к программной реализации базовые алгоритмические методы. После занятия рекомендуется выполнить упражнения, приводимые в аудитории для самостоятельной работы.

При самостоятельной работе студентов необходимо изучить литературу, приведенную в перечнях выше, для осмысления вводимых понятий, анализа предложенных подходов и методов разработки программ. Разрабатывая решение новой задачи, студент должен уметь выбрать эффективные и надежные структуры данных для представления информации, подобрать соответствующие алгоритмы для их обработки, учесть специфику языка программирования, на котором будет выполнена реализация. Студент должен уметь выполнять тестирование и отладку алгоритмов решения задач с целью обнаружения и устранения в них ошибок.

Важнейшим этапом курса является самостоятельная работа по дисциплине. В процессе самостоятельной работы студент приобретает навык создания законченного программного продукта.

Используются активные, инновационные образовательные технологии, которые способствуют развитию общекультурных, общепрофессиональных компетенций и профессиональных компетенций обучающихся:

- проблемное обучение;
- разноуровневое обучение;
- проектные методы обучения;
- исследовательские методы в обучении;
- обучение в сотрудничестве (командная, групповая работа);
- информационно- коммуникационные технологии.

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

Учебно-методическим обеспечением курсовой работы студентов являются:

1. учебная литература;
2. нормативные документы ВУЗа;

3. методические разработки для студентов.

Самостоятельная работа студентов включает:

- оформление итогового отчета (пояснительной записки).
- анализ нормативно-методической базы организации;
- анализ научных публикации по заранее определённой теме;
- анализ и обработку информации;
- работу с научной, учебной и методической литературой,
- работа с конспектами лекций, ЭБС.

Для самостоятельной работы представляется аудитория с компьютером и доступом в Интернет, к электронной библиотеке вуза и к информационно-справочным системам.

Перечень учебно-методического обеспечения:

1. Основная образовательная программа высшего профессионального образования федерального государственного бюджетного образовательного учреждения высшего образования «Кубанский государственный университет» по направлению подготовки.
2. Положение о проведении текущего контроля успеваемости и промежуточной аттестации в федеральном государственном бюджетном образовательном учреждении высшего образования «Кубанский государственный университет».
3. Общие требования к построению, содержанию, оформлению и утверждению рабочей программы дисциплины Федерального государственного образовательного стандарта высшего профессионального образования.
4. Методические рекомендации по содержанию, оформлению и применению образовательных технологий и оценочных средств в учебном процессе, основанном на Федеральном государственном образовательном стандарте.
5. Учебный план основной образовательной программы по направлению подготовки.
6. Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки.

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

Среда разработки программ на языке программирования C++

Текстовый редактор

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

Microsoft Visual Studio C++ Community свободно распространяемая

LibreOffice свободно распространяемая

8. Материально-техническое обеспечение по дисциплине (модулю)

№	Вид работ	Наименование учебной аудитории, ее оснащённость 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.	Практические занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
4.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
6.	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.