МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 44

УТВЕРЖДАЮ
Руководитель образовательной программы

доц.,к.т.н.
(должность, уч. степень, звание)

Е.Л. Турнецкая
(инициалы, фамилия)

(подпись)

«16» февраля 2026 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»
(Наименование дисциплины)

Код направления подготовки/ специальности	09.03.03
Наименование направления подготовки/ специальности	Прикладная информатика
Наименование направленности/ специализации	Прикладной искусственный интеллект и наука о данных
Форма обучения	заочная
Год приема	2026

Санкт-Петербург– 2026

Лист согласования рабочей программы дисциплины

Программу составил (а)

ст.преп.	«16» февраля 2026 г	Д.В. Куртяник
(должность, уч. степень, звание)	(подпись, дата)	(инициалы, фамилия)

Программа одобрена на заседании кафедры № 44
«16» февраля 2026 г, протокол № 5-25/26

Заведующий кафедрой № 44

д.т.н.,проф.	«16» февраля 2026 г	М.Б. Сергеев
(уч. степень, звание)	(подпись, дата)	(инициалы, фамилия)

Заместитель директора института №4 по методической работе

доц.,к.т.н.	«16» февраля 2026 г	А.А. Фоменкова
(должность, уч. степень, звание)	(подпись, дата)	(инициалы, фамилия)

Аннотация

Дисциплина «Основы программирования» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 09.03.03 «Прикладная информатика» направленности/специализации «Прикладной искусственный интеллект и наука о данных». Дисциплина реализуется кафедрой «№44».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

УК-2 «Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений»

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-7 «Способен разрабатывать алгоритмы и программы, пригодные для практического применения»

ПК-1 «Способен разрабатывать и отлаживать программный код (базовая)»

ПК-3 «Способен участвовать в промышленной разработке программного обеспечения (топ)»

ПК-4 «Способен применять искусственный интеллект (ИИ) для генерации и отладки программного кода (топ)»

ПК-23 «Способен организовать промышленную разработку программного обеспечения»

ПК-24 «Способен организовать процессы применения искусственного интеллекта (ИИ) для генерации и отладки программного кода (топ)»

Содержание дисциплины охватывает круг вопросов, связанных с современными языками и методами программирования, разработкой, отладкой и тестированием программ на C++, процедурным, модульным и объектно-ориентированным подходами, организацией совместной разработки и применением искусственного интеллекта для генерации и проверки программного кода при решении прикладных задач.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, практические занятия, самостоятельная работа обучающегося, курсовая работа.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена (2 семестр), экзамена (3 семестр).

Общая трудоемкость освоения дисциплины составляет 11 зачетных единиц, 396 часов.

Язык обучения по дисциплине «русский »

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Формирование у обучающихся представления о современных языках программирования и средствах разработки программ различного уровня сложности, развитие умений разрабатывать алгоритмы и программы на C++, применять процедурный, модульный и объектно-ориентированный подходы, выполнять отладку и тестирование, использовать средства совместной разработки и критически проверять программный код, созданный с применением искусственного интеллекта, при решении прикладных задач.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Универсальные компетенции	УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.В.3 владеть навыками использования цифровых средств для решения поставленной задачи
Общепрофессиональные компетенции	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.З.1 знать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности ОПК-2.У.1 уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности ОПК-2.В.1 владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
Общепрофессиональные компетенции	ОПК-7 Способен разрабатывать	ОПК-7.З.1 знать основные языки программирования и работы с базами

	алгоритмы и программы, пригодные для практического применения	данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий ОПК-7.З.3 знать методы разработки алгоритмов и программного обеспечения, пригодные для практического применения ОПК-7.У.1 уметь применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ ОПК-7.У.2 уметь применять методы разработки алгоритмов и программного обеспечения ОПК-7.У.3 уметь обобщать фактический материал, используя теоретические знания и практические навыки ОПК-7.У.4 уметь анализировать фактический материал, используя теоретические знания и практические навыки ОПК-7.В.1 владеть навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач
Профессиональные компетенции	ПК-1 Способен разрабатывать и отлаживать программный код (базовая)	ПК-1.Д.1.З.1 знает алгоритмы решения типичных задач, области и способы их применения ПК-1.Д.1.З.2 знает нотации и программное обеспечение для графического отображения алгоритмов ПК-1.Д.1.З.3 знает методы и приемы алгоритмизации поставленных задач ПК-1.Д.2.З.1 знает основные разделы математики, применяемые для анализа и моделирования непрерывных процессов и дискретных систем в прикладных задачах ПК-1.Д.3.З.1 знает основные классы методов программной реализации моделей и критерии выбора алгоритмов ПК-1.Д.4.З.1 знает синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования ПК-1.Д.5.З.1 знает нормативно-

		технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода ПК-1.Д.6.З.1 знает возможности используемой системы управления версиями и вспомогательных инструментальных программных средств ПК-1.Д.7.З.1 знает методы и приемы отладки программного кода ПК-1.Д.7.З.2 знает типы и форматы сообщений об ошибках, предупреждений ПК-1.Д.1.У.2 умеет применять алгоритмы решения типовых задач в соответствующих областях ПК-1.Д.3.У.1 умеет проводить сравнительный анализ и обоснование выбора алгоритмов для программной реализации модели ПК-1.Д.4.У.1 умеет применять выбранные языки программирования для написания программного кода ПК-1.Д.4.У.2 умеет использовать выбранную среду программирования ПК-1.Д.5.У.1 умеет применять нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода ПК-1.Д.6.У.2 умеет сохранять изменения программного кода в соответствии с регламентом управления версиями ПК-1.Д.7.У.1 умеет выявлять ошибки в программном коде ПК-1.Д.7.У.2 умеет отлаживать программный код на уровне программных модулей ПК-1.Д.2.В.1 владеет навыками описания задач предметной области в виде формальных математических моделей, пригодных для последующей алгоритмизации и программной реализации
Профессиональные компетенции	ПК-3 Способен участвовать в промышленной разработке программного обеспечения (топ)	ПК-3.Д.3.З.1 знает принципы чистого кода, SOLID, DRY, KISS и др. ПК-3.Д.3.У.1 умеет разрабатывать модульный и тестируемый программный код
Профессиональные компетенции	ПК-4 Способен применять искусственный интеллект (ИИ) для	ПК-4.Д.1.З.1 знает принципы работы современных генеративных ИИ-моделей для генерации кода ПК-4.Д.1.У.1 умеет формулировать

	генерации и отладки программного кода (топ)	корректные текстовые запросы (промты) для генерации кода
Профессиональные компетенции	ПК-23 Способен организовать промышленную разработку программного обеспечения	ПК-23.Д.3.З.1 знает принципы чистого кода, SOLID, DRY, KISS и др. ПК-23.Д.3.У.1 умеет управлять разработкой модульного и тестируемого программного кода
Профессиональные компетенции	ПК-24 Способен организовать процессы применения искусственного интеллекта (ИИ) для генерации и отладки программного кода (топ)	ПК-24.Д.1.З.1 знает принципы работы современных генеративных ИИ-моделей для генерации кода ПК-24.Д.1.У.1 умеет выбирать подходящие под задачи ИИ-инструменты

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- «Информатика»,
- «Основы цифровой грамотности».

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

- «Технологии программирования»,
- «Введение в анализ данных»,
- «Базы данных»,
- «Программная инженерия».

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам	
		№2	№3
1	2	3	4
Общая трудоемкость дисциплины, ЗЕ/ (час)	11/ 396	6/ 216	5/ 180
Из них часов практической подготовки	16	11	5
Аудиторные занятия, всего час.	46	30	16
в том числе:			
лекции (Л), (час)	20	12	8
практические/семинарские занятия (ПЗ), (час)	4	4	
лабораторные работы (ЛР), (час)	18	14	4
курсовой проект (работа) (КП, КР), (час)	4		4

экзамен, (час)	18	9	9
Самостоятельная работа, всего (час)	332	177	155
***Вид промежуточной аттестации*:** зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.)	Экз., Экз., Курс. Раб.	Экз.,	Экз., Курс. Раб.

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП/КР (час)	СР (час)
Семестр 2					
Раздел 1. Методы разработки программных средств Тема 1.1. Современные методы разработки программных средств. Тема 1.2. Этапы разработки программных средств. Тема 1.3. Контроль версий и совместная разработка программ.	2				25
Раздел 2. Основы языка программирования C++ Тема 2.1. Современные языки программирования. Тема 2.2. Базовые средства языка программирования C++. Тема 2.3. Понятие объекта.	2				35
Раздел 3. Структурное программирование Тема 3.1. Базовые конструкции структурного программирования. Тема 3.2. Примеры разработки программ. Тема 3.3. Статические массивы.	4		8		40
Раздел 4. Процедурное и модульное программирование Тема 4.1. Концепции процедурного программирования. Тема 4.2. Понятие рекурсии. Тема 4.3. Концепции модульного программирования.	4	4	6		77
Итого в семестре:	12	4	14		177
Семестр 3					
Раздел 5. Динамические структуры данных Тема 5.1. Динамические структуры данных. Тема 5.2. Деревья.	4		2		45

Раздел 6. Концепции объектно-ориентированного программирования Тема 6.1. Классы. Тема 6.2 Перегрузка операций. Тема 6.3. Наследование Тема 6.4 Потоковые классы. Тема 6.5 Строки в стиле языка С++. Тема 6.6. Основы доказательства правильности программ Тема 6.7. Применение ИИ для генерации и отладки программного кода.	4		2		60
Выполнение курсовой работы				4	50
Итого в семестре:	8		4	4	155
Итого	20	4	18	4	332

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	Раздел 1. Методы разработки программных средств Тема 1.1. Современные методы разработки программных средств. Развитие современных методов программирования. Структурное, процедурное, модульное и объектно-ориентированное программирование. Технологии нисходящего и восходящего проектирования программ. Сущность структурного программирования: разбиение на подзадачи, нисходящее проектирование, стандартные структуры управления. Достоинства и недостатки. Тема 1.2. Этапы разработки программных средств. Кодирование и документирование программ. Проектирование и тестирование программ. Основные этапы разработки программ. Этап постановки задачи. Этап разработки структур данных: определение входных и выходных данных, формы получения результатов. Использование технологии нисходящего проектирования на этапе проектирования программных средств. Этап структурного программирования. Этап тестирования и отладки. Требования к оформлению кода и программной документации. Тема 1.3. Контроль версий и совместная разработка программ.

	Назначение системы контроля версий. Репозиторий, фиксация изменений, ветви и объединение изменений. Распределение задач при совместной разработке. Проверка изменений кода, разрешение конфликтов и документирование результатов. Проверка изменений и организация модульной разработки.
2	Раздел 2. Основы языка программирования C++ Тема 2.1. Современные языки программирования. Понятие языка программирования. Этапы развития языков программирования. Современные тенденции в области языков программирования. Сравнение развития языков в представлении данных и способах реализации алгоритмов. Сравнительная характеристика языков программирования высокого уровня. Синтаксис и семантика. Способы описания синтаксиса: лингвистические формулы и синтаксические диаграммы. Тема 2.2. Базовые средства языка программирования C++. Структура языка программирования C++. Базовые элементы языка C++: алфавит, лексемы, выражения, комментарии. Основные (стандартные) типы данных языка C++. Типы арифметические целочисленные. Управляющие последовательности. Типы арифметические с плавающей точкой. Логический тип bool. Тип void. Идентификаторы языка. Область действия и область видимости идентификатора. Ключевые (зарезервированные) слова. Константы. Разделители. Спецификации классов памяти. Предложения языка: описания и операторы. Программа на языке C++: состав и структура. Тема 2.3. Понятие объекта. Объекты. Классификация объектов. Арифметические операции над объектами. Арифметические преобразования. Логические и поразрядные логические операции над объектами. Операции сдвига. Понятие об L-значении. Операции присваивания. Преобразования при присваивании. Операция запятая. Операции увеличения и уменьшения значения. Условная операция. Операция получения адреса объекта в основной памяти и операция получения содержимого объекта по адресу. Указатели. Указатели на объект, на функцию, на void. Операции с указателями. Ссылки. Операция приведения. Операция размер. Первичные операции. Понятие выражения. Порядок выполнения операций в выражении.
3	Раздел 3. Структурное программирование Тема 3.1. Базовые конструкции структурного программирования.

	Средства реализации линейных алгоритмов: оператор - выражение, составной оператор (блок), пустой оператор. Средства реализации разветвляющихся алгоритмов: условный оператор, оператор - переключатель, оператор перехода, оператор разрыва, оператор перехода на начало следующей итерации, оператор возврата в вызывающую функцию. Средства реализации циклических алгоритмов: операторы цикла с предусловием, с постусловием, с параметром. Реализация арифметических, итерационных и вложенных циклов. Реализация рекуррентных вычислений. Тема 3.2. Примеры разработки программ. Линейные программы. Вычисление результатов поразрядных логических операций над целыми числами. Разветвляющиеся программы – поиск экстремальных значений. Попадание точки в заданные области на плоскости. Циклические программы. Решение задач на заданное число повторений. Решение задач на получение результатов с заданной точностью. Получение таблиц значений алгебраических функций в заданном диапазоне изменения значения аргумента. Тема 3.3. Статические массивы. Статическое распределение памяти. Структурированные типы данных: одномерные и многомерные массивы. Решение типовых задач на массивы: поиск значений максимального отрицательного и минимального положительного элементов одномерного вещественного массива, перестановка элементов одномерного массива. Перестановка столбцов и строк матриц по заданным правилам.
4	Раздел 4. Процедурное и модульное программирование Тема 4.1. Концепции процедурного программирования. Функции. Основные понятия. Принципы использования функций в программах. Объявление и определение функций. Список параметров и тип функции. Вызов функций на исполнение. Формальные и фактические параметры. Области действия описаний функций. Локальность и глобальность. Механизм передачи параметров. Глобальные переменные. Функция main(). Передача массивов в качестве параметров. Примеры. Реализация алгоритмов сортировки структур данных и поиска в этих структурах. Тема 4.2. Понятие рекурсии. Передача имен функций в качестве параметров. Понятие рекурсии. Рекурсивные определения и алгоритмы. Программирование рекурсивных алгоритмов: рекурсивные функции. Механизм рекурсивных вызовов. Перегрузка

	функций. Шаблоны функций. Тема 4.3. Концепции модульного программирования. Модули: назначение, структура, трансляция, тестирование. Особенности использования модулей. Модульные программы. Директива #include. Директива #define. Построение многомодульных программ средствами языка программирования C++.
5	Раздел 5. Динамические структуры данных Тема 5.1. Динамические структуры данных. Указатели и ссылки. Динамические массивы. Динамические структуры: списки. Виды списков: односвязные и двусвязные списки, линейные и циклические списки. Динамические структуры: стеки. Динамические структуры: очереди. Динамические структуры: деки. Тема 5.2. Деревья. Динамические структуры: деревья. Виды деревьев и способы их реализации. Бинарное дерево как рекурсивная структура данных. Рекурсивные процедуры обхода дерева: инфиксная форма, префиксная форма, постфиксная форма. Особенности использования рекурсии при построении дерева.
6	Раздел 6. Концепции объектно-ориентированного программирования Тема 6.1. Классы. Основные принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм. Классы в языке C++. Разработка классов. Составляющие классов: поля, методы. Экземпляры класса (объекты). Время их жизни и видимость. Конструкторы. Свойства конструкторов. Статические элементы класса (статические поля и статические методы). Дружественные функции и дружественные классы. Деструкторы. Принципы чистого кода, DRY, KISS и SOLID. Тема 6.2 Перегрузка операций. Перегрузка операций с объектами конкретных классов. Перегрузка унарных операций. Перегрузка бинарных операций. Перегрузка операции присваивания. Перегрузка операции приведения типа. Перегрузка операции вызова функции. Перегрузка операции индексирования. Тема 6.3. Наследование Понятие наследования. Простое наследование. Виртуальные методы. Механизм позднего связывания. Множественное наследование. Шаблоны классов. Создание шаблонов классов. Использование шаблонов классов. Тема 6.4 Потоковые классы. Стандартные потоки. Методы обмена с потоками (операция

	извлечения из потока и операция включения в поток). Файловые потоки. Строковые потоки. Тема 6.5 Строки в стиле языка С++. Класс string. Конструкторы строк. Присваивание строк. Операции со строками. Функции добавления частей строк. Функции преобразования строк. Функции поиска подстрок. Функции сравнения частей строк. Функции получения характеристик строк. Тема 6.6. Основы доказательства правильности программ Теорема структуры и структурное программирование. Анализ программ. Утверждения о программах. Корректность программ. Правила вывода для основных структур программирования. Инвариантные утверждения. Утверждения о массивах. Способы верификации программ. Жизненный цикл программы. Тема 6.7. Применение ИИ для генерации и отладки программного кода. Принципы работы генеративных ИИ-моделей при создании кода. Выбор ИИ-инструмента с учетом задачи и ограничений. Формулирование запроса с требованиями к программе. Анализ и исправление предложенного кода. Проверка результата компиляцией, тестами и анализом граничных случаев. Контроль качества изменений и документирование использования ИИ.

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 2					
1	Обработка строк и текстовых файлов средствами С++	Разбор типовых задач под руководством преподавателя; самостоятельное решение задач по обработке строк и текстовых файлов	2	2	4
2	Обработка структур и файлов записей средствами С++	Разбор типовых задач под руководством преподавателя; самостоятельное решение задач по обработке структур и файлов записей	2	2	4

Всего	4		

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 2				
1	Разработка и тестирование программы линейного алгоритма	2	1	3
2	Разработка и тестирование программы с ветвлениями	2	1	3
3	Разработка и тестирование программы с циклами	2	1	3
4	Обработка одномерных массивов	2	1	3
5	Разработка программы с функциями и передачей параметров	2	1	4
6	Разработка и проверка рекурсивной функции	2	1	4
7	Разработка и тестирование многомодульной программы	2	1	4
Семестр 3				
8	Создание и обработка динамического массива или линейного списка	2	2	5
9	Разработка класса и проверка работы его методов	2	2	6
	Всего	18		

4.5. Выполнение курсового проекта/ курсовой работы

Цель курсовой работы: Развитие умений самостоятельно анализировать прикладную задачу, разрабатывать алгоритм и программное решение на C++, обосновывать выбор методов и средств, выполнять отладку и тестирование программы и представлять результаты в виде работающего программного средства и пояснительной записки.

Часов практической подготовки: 1.

Примерные темы заданий на курсовую работу приведены в разделе 10 РПД.

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 2, час	Семестр 3, час
1	2	3	4
Изучение теоретического материала дисциплины (ТО)	190	130	60

Курсовое проектирование (КП, КР)	50		50
Расчетно-графические задания (РГЗ)			
Выполнение реферата (Р)			
Подготовка к текущему контролю успеваемости (ТКУ)	52	27	25
Домашнее задание (ДЗ)			
Контрольные работы заочников (КРЗ)			
Подготовка к промежуточной аттестации (ПА)	40	20	20
Всего:	332	177	155

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

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. разделов 6-11.

6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8 – Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
004.421(75) Б24	Бариков Л.Н. Основы алгоритмизации и программирования на С/С++: учебное пособие: в 2 ч. Ч. 1. — Санкт-Петербург: ГУАП, 2019. — 164 с.	50
004.421(75) Б24	Бариков Л.Н. Основы алгоритмизации и программирования на С/С++: учебное пособие: в 2 ч. Ч. 2. — Санкт-Петербург: Изд-во ГУАП, 2021. — 244 с.	50
004.4 П12	Павловская Т.А. С/С++. Программирование на языке высокого уровня: учебник. — Санкт-Петербург: Питер, 2003. — 459 с.	49

7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
	Не предусмотрено

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10 – Перечень программного обеспечения

№ п/п	Наименование
1	Электронная информационно-образовательная среда ГУАП «Интегрированная среда обучения» (https://pro.guap.ru/) разработана сотрудниками ГУАП (введена в эксплуатацию приказом ГУАП от 06.06.2017 № 05-215/17), перечень модулей и их функциональное назначение изложены по ссылке https://guap.ru/it/system/iso
2	Официальный сайт образовательной организации в сети «Интернет» (https://guap.ru/), разработан сотрудниками ГУАП (введен в эксплуатацию Приказом ГУАП от 23.03.2023 № 05-145/23)
3	Microsoft Visual Studio Community 2026 — среда разработки программ на C++; использование в учебном процессе на условиях лицензии Microsoft Visual Studio Community 2026 (https://visualstudio.microsoft.com/license-terms/vs2026-ga-community/).
4	Git — распределенная система контроля версий (лицензия GNU GPL версии 2; https://git-scm.com/about.html).
5	GigaCode — ИИ-ассистент для генерации и анализа программного кода; индивидуальный доступ обучающихся на бесплатном тарифе «Для себя» по пользовательскому соглашению (https://gitverse.ru/features/gigacode/pricing/; https://gitverse.ru/files/polzovatelskoe-soglashenie-gigacode.pdf).

8.2. Перечень информационно-справочных систем, используемыхпри осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11 – Перечень информационно-справочных систем

№ п/п	Наименование
1	Электронный каталог библиотеки ГУАП с доступом к базе полнотекстовых изданий (https://lib.guap.ru.), доступ через личный кабинет читателя библиотеки ГУАП

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	
2	Аудитория для проведения практических занятий	
3	Аудитория для проведения лабораторных работ	
4	Помещение для самостоятельной работы	

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Экзамен	Список вопросов к экзамену; Экзаменационные билеты*; Задачи; Тесты.
Выполнение курсовой работы	Экспертная оценка на основе требований к содержанию курсовой работы по дисциплине.

Примечание: *экзаменационные билеты формируются на основе вопросов и задач таблицы 15.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 – Критерии оценки уровня сформированности компетенций

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«отлично» «зачтено»	Обучающийся: – глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно связывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий. – правильно выполнил от 90% до 100% тестовых заданий.
«хорошо» «зачтено»	Обучающийся: – твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий. – правильно выполнил от 70% до 89% тестовых заданий.
«удовлетворительно» «зачтено»	– обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий. – правильно выполнил от 51% до 69% тестовых заданий.
«неудовлетворительно» «не зачтено»	– обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний;

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
	– не может аргументировать научные положения; – не формулирует выводов и обобщений. – правильно выполнил менее 51% тестовых заданий.

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
Семестр 2		
1	Создайте в Visual Studio проект C++ для вычисления площади прямоугольника по введенным сторонам. Покажите ввод, сборку, запуск и проверку результата.	УК-2.В.3
2	Для задачи поиска максимума массива из 20 целых чисел составьте алгоритм, реализуйте его в C++ и подготовьте тест с повторяющимся максимумом.	УК-2.В.3
3	Используя отладчик, проверьте программу суммирования чисел от 1 до n. Покажите значения счетчика и суммы при n=4.	УК-2.В.3
4	Разработайте функцию подсчета положительных элементов массива {-2, 0, 5, 7}. Сохраните код и результаты проверки в отдельной фиксации Git.	УК-2.В.3
5	Создайте программу поиска слова в текстовом файле. Подготовьте входные данные, обработайте отсутствие файла и продемонстрируйте результат.	УК-2.В.3
6	Организуйте проект из main.cpp и файла с функцией вычисления среднего массива. Соберите проект и подтвердите работу тестом {2, 4, 6}.	УК-2.В.3
7	Объясните понятия языка программирования, синтаксиса и семантики. Сравните лингвистические формулы и синтаксические диаграммы на примере условного оператора.	ОПК-7.З.1
8	Охарактеризуйте язык C++: области применения, структура программы, компиляция, стандартная библиотека и роль функции main.	ОПК-7.З.1
9	Объясните основные типы C++: целочисленные, вещественные, bool и void. Приведите примеры объявлений и ограничений представления данных.	ОПК-7.З.1
10	Объясните устройство проекта C++ в Visual Studio: исходные файлы, заголовки, параметры сборки, компоновка и запуск программы.	ОПК-7.З.1
11	Объясните работу программы с текстовым файлом: открытие, чтение, запись, проверка состояния потока и закрытие.	ОПК-7.З.1
12	Сравните указатели и ссылки в C++: объявление, инициализация, доступ к объекту, допустимые операции и типичные ошибки.	ОПК-7.З.1
13	Охарактеризуйте этапы разработки программы:	ОПК-7.З.3

	постановка задачи, выбор данных, разработка алгоритма, кодирование, тестирование и документирование.	
14	Объясните нисходящее проектирование программы: декомпозиция задачи, определение функций, согласование интерфейсов и проверка модулей.	ОПК-7.З.3
15	Сравните структурный, процедурный и модульный подходы. Укажите принципы, средства C++ и примеры применения.	ОПК-7.З.3
16	Объясните методы поиска элемента в массиве: линейный и двоичный поиск. Назовите условия применения и особенности проверки.	ОПК-7.З.3
17	Сравните сортировки выбором, обменом и вставками: принцип работы, число сравнений, требования к памяти и подходящие исходные данные.	ОПК-7.З.3
18	Объясните метод рекурсивного решения задачи: базовый случай, уменьшение задачи, стек вызовов и проверка завершения.	ОПК-7.З.3
19	Разработайте программу учета пяти товаров со структурой {название, цена, количество}. Выведите стоимость каждого товара и общую стоимость запасов.	ОПК-7.У.1
20	Напишите программу чтения оценок из текстового файла и вычисления среднего. Обработайте отсутствие файла и отсутствие числовых данных.	ОПК-7.У.1
21	Создайте функцию поиска записи о студенте по идентификатору в массиве структур. Для найденной записи выведите имя и средний балл.	ОПК-7.У.1
22	Разработайте функцию подсчета частоты заданного символа в строке. Проверьте пустую строку, отсутствие символа и повторяющиеся символы.	ОПК-7.У.1
23	Напишите программу фильтрации записей товаров: сохраните в новый файл товары с количеством меньше 3. Покажите обработку ошибки открытия файла.	ОПК-7.У.1
24	Разработайте программу вычисления итоговой суммы покупки по ценам и количествам. Отклоните отрицательные входные значения и проверьте граничные случаи.	ОПК-7.У.1
25	Разработайте алгоритм поиска минимального положительного элемента массива {-4, 0, 7, 2, -1} и реализуйте его функцией C++.	ОПК-7.У.2
26	Примените нисходящее проектирование к задаче учета товаров: выделите ввод, расчет и вывод в отдельные функции с описанными параметрами.	ОПК-7.У.2
27	Реализуйте сортировку вставками для массива {5, 2, 5, 1}. Поясните инвариант внешнего цикла и проверьте итоговый порядок.	ОПК-7.У.2
28	Разработайте рекурсивную функцию вычисления суммы первых n натуральных чисел. Задайте базовый случай и обработку отрицательного n.	ОПК-7.У.2
29	Разделите программу обработки текстового файла на модули чтения, преобразования и вывода. Опишите интерфейсы и порядок сборки.	ОПК-7.У.2
30	Реализуйте двоичный поиск числа 7 в массиве {1, 3, 5, 7,	ОПК-7.У.2

	9}. Поясните изменение границ и обработку отсутствующего значения.	
31	Для тестов суммы массива: {}→0, {2}→2, {-2,2}→0, {1,2,3}→6 обобщите проверяемые свойства и предложите недостающий класс тестов.	ОПК-7.У.3
32	По результатам поиска: {1,2,3}, цель 2→1; цель 8→-1; {}, цель 2→-1 сформулируйте контракт функции и условия корректности.	ОПК-7.У.3
33	Для программы чтения файла получены случаи: файл отсутствует, файл пуст, файл содержит только числа, файл содержит слово. Систематизируйте варианты обработки.	ОПК-7.У.3
34	Обобщите результаты отладки цикла: при n=0 итераций нет, при n=1 одна, при n=4 четыре. Сформулируйте связь границ цикла с числом итераций.	ОПК-7.У.3
35	По примерам передачи int по значению и по ссылке обобщите правила изменения аргумента вызывающей функции. Приведите проверяемые примеры.	ОПК-7.У.3
36	Для сортировки массивов {1,2,3}, {3,2,1}, {2,2,2} обобщите требования к результату: порядок, сохранение элементов и обработка повторов.	ОПК-7.У.3
37	Проанализируйте код int a[3]={1,2,3}; for(int i=0;i<=3;++i) cout<<a[i]; определите ошибку границ и предложите исправление.	ОПК-7.У.4
38	Проанализируйте выражение double x=5/2; объясните фактический результат, типы операндов и способ получить 2.5.	ОПК-7.У.4
39	Для функции int f(int n){return n*f(n-1);} выявите причину незавершения и предложите корректный базовый случай для факториала.	ОПК-7.У.4
40	Проанализируйте код int x=0; if(x=5) cout<<x; объясните изменение x, значение условия и исправление для сравнения.	ОПК-7.У.4
41	Проанализируйте передачу массива в функцию void f(int a[]). Объясните, почему его длина не определяется выражением sizeof(a)/sizeof(a[0]).	ОПК-7.У.4
42	Для последовательности чтения while(!in.eof()){in>>x; sum+=x;} проанализируйте риск повторной обработки значения и предложите безопасное условие.	ОПК-7.У.4
43	Объясните линейный алгоритм поиска максимума: входные данные, инициализация, сравнение, результат и обработка пустого массива.	ПК-1.Д.1.З.1
44	Объясните алгоритм подсчета элементов, удовлетворяющих условию. Укажите счетчик, проверку предиката, цикл и граничные случаи.	ПК-1.Д.1.З.1
45	Охарактеризуйте алгоритм накопления суммы: начальное значение, изменение аккумулятора, завершение и проверка результата.	ПК-1.Д.1.З.1
46	Объясните алгоритм сортировки выбором: выделение неотсортированной части, поиск минимума, обмен и завершение.	ПК-1.Д.1.З.1
47	Объясните двоичный поиск: требование сортировки,	ПК-

	выбор середины, изменение диапазона и условие завершения.	1.Д.1.З.1
48	Объясните рекурсивный расчет факториала: область допустимых значений, базовый случай, шаг рекурсии и ограничения числового типа.	ПК-1.Д.1.З.1
49	Охарактеризуйте блок-схему алгоритма: начало и конец, ввод и вывод, действие, условие и направление переходов.	ПК-1.Д.1.З.2
50	Объясните представление ветвления в блок-схеме и псевдокоде. Сопоставьте обе записи с оператором if в C++.	ПК-1.Д.1.З.2
51	Объясните графическое представление цикла с предусловием: условие, тело, изменение переменной и выход.	ПК-1.Д.1.З.2
52	Сравните блок-схему и псевдокод: читаемость, детализация, удобство изменений и соответствие программному коду.	ПК-1.Д.1.З.2
53	Объясните использование схемы вызовов функций для многомодульной программы: вершины, связи, параметры и точки входа.	ПК-1.Д.1.З.2
54	Охарактеризуйте требования к средству построения схем: доступные элементы, подписи, редактирование, сохранение и экспорт. Объясните отличие от компилятора.	ПК-1.Д.1.З.2
55	Объясните декомпозицию задачи обработки файла: выделение подзадач, входы и выходы функций, последовательность вызовов и обработка ошибок.	ПК-1.Д.1.З.3
56	Охарактеризуйте алгоритмизацию вычислительной задачи: формализация условия, выбор переменных, построение шагов, проверка и реализация.	ПК-1.Д.1.З.3
57	Объясните преобразование словесного условия в алгоритм на примере поиска минимальной цены товара.	ПК-1.Д.1.З.3
58	Сравните итерацию и рекурсию как приемы построения алгоритма: состояние, завершение, память и проверяемость.	ПК-1.Д.1.З.3
59	Объясните использование инварианта цикла: начальная истинность, сохранение, завершение и связь с результатом.	ПК-1.Д.1.З.3
60	Объясните прием разбиения данных на части при обработке массива: границы диапазонов, независимые операции, объединение результатов и проверка.	ПК-1.Д.1.З.3
61	Объясните применение арифметики и алгебры при программировании формулы: порядок операций, типы данных, область определения и погрешность.	ПК-1.Д.2.З.1
62	Охарактеризуйте логические операции AND, OR и NOT. Постройте таблицы истинности и объясните использование в условиях C++.	ПК-1.Д.2.З.1
63	Объясните последовательность и рекуррентное соотношение на примере суммы и произведения элементов массива.	ПК-1.Д.2.З.1
64	Объясните модель точки на плоскости и проверку принадлежности прямоугольной области: координаты, границы и логическое условие.	ПК-1.Д.2.З.1

65	Объясните представление конечной последовательности массивом: индексы, длина, доступ и диапазон допустимых значений.	ПК-1.Д.2.З.1
66	Охарактеризуйте математическую модель среднего значения: исходные данные, формула, условие ненулевого количества и проверка результата.	ПК-1.Д.2.З.1
67	Сравните линейный и двоичный поиск по требованиям к данным, числу шагов, памяти и удобству реализации.	ПК-1.Д.3.З.1
68	Сравните сортировки выбором и вставками на почти отсортированном массиве. Объясните критерии выбора алгоритма.	ПК-1.Д.3.З.1
69	Объясните критерии выбора итеративного или рекурсивного решения: глубина, память, читаемость и возможность проверки.	ПК-1.Д.3.З.1
70	Охарактеризуйте численный расчет последовательности с заданным числом шагов и с заданной точностью: условия окончания и ограничения.	ПК-1.Д.3.З.1
71	Объясните выбор массива или последовательного чтения файла для расчета суммы: объем данных, память, повторный доступ и обработка ошибок.	ПК-1.Д.3.З.1
72	Сравните полный перебор и прямую формулу для суммы чисел от 1 до n. Укажите вычислительные затраты, риск переполнения и проверку.	ПК-1.Д.3.З.1
73	Объясните лексические элементы C++: идентификаторы, ключевые слова, литералы, разделители и комментарии.	ПК-1.Д.4.З.1
74	Охарактеризуйте выражения C++: приоритет операций, приведение типов, присваивание, инкремент и условная операция.	ПК-1.Д.4.З.1
75	Объясните операторы if, switch, break и continue. Укажите назначение, ограничения и типичные ошибки.	ПК-1.Д.4.З.1
76	Объясните циклы for, while и do-while: порядок выполнения, условие, изменение состояния и выбор конструкции.	ПК-1.Д.4.З.1
77	Объясните объявление, определение и вызов функции: тип результата, параметры, аргументы и область видимости.	ПК-1.Д.4.З.1
78	Охарактеризуйте средства стандартной библиотеки для строк и файлов: string, входные и выходные потоки, чтение строк и проверка ошибок.	ПК-1.Д.4.З.1
79	Объясните назначение стандарта языка C++ и правил оформления проекта. Разделите синтаксические требования компилятора и требования к стилю кода.	ПК-1.Д.5.З.1
80	Охарактеризуйте регламент именования в учебном проекте: понятность, согласованность, различение сущностей и предотвращение неоднозначности.	ПК-1.Д.5.З.1
81	Объясните требования к оформлению блоков кода: отступы, скобки, переносы, длина строки и читаемость.	ПК-1.Д.5.З.1
82	Объясните правила документирования функций: назначение, параметры, результат, ограничения и исключительные случаи.	ПК-1.Д.5.З.1
83	Охарактеризуйте назначение программных документов ЕСПД на примере текста программы, описания	ПК-1.Д.5.З.1

	программы и описания применения.	
84	Объясните проверку соответствия кода регламенту проекта: выделение требований, сопоставление, фиксация замечаний и контроль исправлений.	ПК-1.Д.5.З.1
85	Объясните состояния файлов в Git: рабочее дерево, индекс, фиксация, отслеживаемые и неотслеживаемые файлы.	ПК-1.Д.6.З.1
86	Охарактеризуйте команды git status, git add, git commit и git log. Покажите их место в цикле сохранения изменений.	ПК-1.Д.6.З.1
87	Объясните назначение ветвей Git: создание, переключение, независимая работа, объединение и удаление завершенной ветви.	ПК-1.Д.6.З.1
88	Объясните причину конфликта объединения и порядок разрешения: поиск участков, выбор результата, проверка и фиксация.	ПК-1.Д.6.З.1
89	Объясните назначение .gitignore для проекта C++: исходные файлы, результаты сборки, временные файлы и локальные настройки.	ПК-1.Д.6.З.1
90	Сравните git diff и просмотр истории фиксаций. Объясните анализ изменений, поиск причины ошибки и восстановление рабочего варианта.	ПК-1.Д.6.З.1
91	Объясните различия ошибок компиляции, компоновки и выполнения. Приведите примеры и способы диагностики.	ПК-1.Д.7.З.1
92	Охарактеризуйте пошаговую отладку: точка останова, выполнение строки, вход в функцию, просмотр переменных и стек вызовов.	ПК-1.Д.7.З.1
93	Объясните метод локализации ошибки с помощью минимального воспроизводимого примера: сокращение входа, сохранение проявления и проверка гипотез.	ПК-1.Д.7.З.1
94	Объясните выбор тестовых данных для поиска ошибки: обычные значения, границы, пустые данные, повторы и неверный ввод.	ПК-1.Д.7.З.1
95	Охарактеризуйте отладку цикла: контроль счетчика, условия завершения, аккумулятора, индексов и инварианта.	ПК-1.Д.7.З.1
96	Объясните отладку функции: проверка параметров, промежуточных значений, результата, побочных эффектов и контракта.	ПК-1.Д.7.З.1
97	Объясните структуру сообщения компилятора: файл, строка, код диагностики, текст и связь с исходным кодом.	ПК-1.Д.7.З.2
98	Сравните ошибку и предупреждение компилятора. Объясните влияние на сборку и необходимость анализа предупреждений.	ПК-1.Д.7.З.2
99	Объясните диагностику необъявленного идентификатора: возможные причины, область видимости, опечатка и порядок исправления.	ПК-1.Д.7.З.2
100	Объясните сообщение о несовместимости типов: тип источника, тип назначения, допустимое преобразование и возможная потеря данных.	ПК-1.Д.7.З.2
101	Охарактеризуйте ошибку компоновки о неразрешенном внешнем символе: объявление, определение, включение файла и согласование сигнатур.	ПК-1.Д.7.З.2

102	Объясните сообщение об ошибке доступа к памяти: адрес, момент возникновения, стек вызовов, границы массива и время жизни объекта.	ПК-1.Д.7.З.2
103	Примените линейный поиск к массиву {8, 3, 8, 1}: найдите индекс первого значения 8 и обработайте поиск отсутствующего значения 5.	ПК-1.Д.1.У.2
104	Реализуйте подсчет отрицательных элементов массива {-2, 0, -5, 7}. Опишите результат и тест отсутствия отрицательных элементов.	ПК-1.Д.1.У.2
105	Примените сортировку выбором к массиву {4, 1, 3, 1}. Покажите состояние массива после каждого прохода.	ПК-1.Д.1.У.2
106	Примените двоичный поиск к массиву {2, 4, 6, 8, 10} для целей 8 и 7. Покажите изменения границ поиска.	ПК-1.Д.1.У.2
107	Реализуйте расчет факториала для n от 0 до 10. Проверьте значения 0, 1 и 5 и отклоните отрицательный ввод.	ПК-1.Д.1.У.2
108	Разработайте цикл вычисления суммы 1+2+...+n и проверьте его на n=0, n=1 и n=5.	ПК-1.Д.1.У.2
109	Для однократного поиска в неотсортированном массиве из 30 элементов сравните линейный поиск и предварительную сортировку с двоичным поиском. Обоснуйте выбор.	ПК-1.Д.3.У.1
110	Для тысячи поисковых запросов к неизменному массиву из 10000 элементов сравните последовательный и двоичный поиск. Обоснуйте подготовку данных.	ПК-1.Д.3.У.1
111	Для почти отсортированного массива из 50 элементов сравните сортировки выбором и вставками. Укажите критерии и ожидаемое поведение.	ПК-1.Д.3.У.1
112	Для суммы миллиона чисел из файла сравните загрузку всего массива и последовательное накопление. Обоснуйте выбор с учетом памяти.	ПК-1.Д.3.У.1
113	Для расчета факториала сравните рекурсивную и итеративную реализации. Обоснуйте выбор при ограниченном стеке.	ПК-1.Д.3.У.1
114	Для поиска максимума массива сравните полный перебор с сортировкой и выбором последнего элемента. Обоснуйте более подходящий способ.	ПК-1.Д.3.У.1
115	Напишите функцию C++, возвращающую минимум трех целых чисел. Проверьте равные числа и отрицательные значения.	ПК-1.Д.4.У.1
116	Напишите программу вывода таблицы значений y=x*x для целых x от -3 до 3. Используйте цикл и оформите результат.	ПК-1.Д.4.У.1
117	Напишите функцию перестановки первого и последнего элементов непустого массива. Обработайте массив из одного элемента.	ПК-1.Д.4.У.1
118	Разработайте функцию подсчета слов в строке с разделителями-пробелами. Проверьте пустую строку и повторяющиеся пробелы.	ПК-1.Д.4.У.1
119	Напишите функцию обновления количества товара в структуре по переданному значению. Отклоните отрицательное количество.	ПК-1.Д.4.У.1
120	Реализуйте функцию поиска максимального	ПК-

	отрицательного элемента массива. Предусмотрите сообщение об отсутствии подходящего элемента.	1.Д.4.У.1
121	Создайте в Visual Studio консольный проект C++ для ввода двух чисел и вывода их суммы. Покажите сборку и запуск.	ПК-1.Д.4.У.2
122	В проекте из main.cpp и calc.cpp объявите функцию в calc.h, подключите ее и выполните сборку. Объясните возникающую ошибку при исключении calc.cpp.	ПК-1.Д.4.У.2
123	Поставьте точку останова в цикле поиска максимума и покажите изменение индекса и текущего максимума для {2, 7, 3}.	ПК-1.Д.4.У.2
124	Используя диагностику Visual Studio, исправьте код int main(){ cout<<1; } с отсутствующими подключением и пространством имен.	ПК-1.Д.4.У.2
125	Выполните пошаговую отладку функции суммы массива и покажите переход из вызывающей функции в вызываемую и обратно.	ПК-1.Д.4.У.2
126	Создайте тестовый запуск программы чтения файла в рабочем каталоге проекта. Продемонстрируйте успешный ввод и обработку отсутствующего файла.	ПК-1.Д.4.У.2
127	По регламенту проекта «имена функций — глаголы, параметры имеют описательные имена» исправьте функцию int f(int a,int b){return a+b;} и объясните изменения.	ПК-1.Д.5.У.1
128	Оформите вложенные if по заданным правилам: отступ 4 пробела, фигурные скобки у каждого блока, один оператор на строку. Представьте пример до и после.	ПК-1.Д.5.У.1
129	Для функции поиска элемента подготовьте комментарий по регламенту: назначение, параметры, результат и поведение при отсутствии элемента.	ПК-1.Д.5.У.1
130	По регламенту «константы вместо чисел без пояснений» переработайте проверку возраста 18 и предел массива 100. Обоснуйте имена констант.	ПК-1.Д.5.У.1
131	По регламенту «объявления в заголовке, определения в исходном файле» разделите функцию вычисления площади на calc.h и calc.cpp.	ПК-1.Д.5.У.1
132	Подготовьте фрагмент описания программы обработки файла: назначение, входные данные, выходные данные, порядок запуска и сообщения об ошибках.	ПК-1.Д.5.У.1
133	По регламенту «одна законченная задача — одна фиксация» сохраните исправление границы цикла в Git. Покажите diff и содержательное сообщение.	ПК-1.Д.6.У.2
134	По регламенту «новая функция в отдельной ветви» добавьте функцию поиска минимума, проверьте ее и объедините изменения в основную ветвь.	ПК-1.Д.6.У.2
135	Перед фиксацией исключите из проекта результаты сборки и временные файлы посредством .gitignore. Покажите состав сохраняемых исходных файлов.	ПК-1.Д.6.У.2
136	Разрешите конфликт: в одной ветви функция возвращает сумму, в другой добавлена проверка пустого массива. Сохраните оба требования и проверьте результат.	ПК-1.Д.6.У.2
137	По регламенту «перед объединением проверить	ПК-

	изменения» покажите историю ветви, diff и результат тестирования новой функции подсчета элементов.	1.Д.6.У.2
138	После обнаружения ошибочной фиксации восстановите корректное поведение новой фиксацией Git, сохранив историю. Объясните выполненные действия.	ПК-1.Д.6.У.2
139	Найдите ошибку в коде int a[3]={1,2,3}; for(int i=0;i<=3;i++) a[i]*=2; предложите исправление и граничный тест.	ПК-1.Д.7.У.1
140	Найдите ошибку вычисления среднего: double mean=sum/n; при целочисленных sum и n. Укажите также необходимую проверку n.	ПК-1.Д.7.У.1
141	Найдите ошибку int maximum=0; при поиске максимума массива {-5,-2,-9}. Исправьте и обоснуйте инициализацию.	ПК-1.Д.7.У.1
142	Найдите причину бесконечного цикла int i=0; while(i<3){cout<<i;} и предложите проверяемое исправление.	ПК-1.Д.7.У.1
143	Найдите ошибку void inc(int x){++x;} если требуется изменить аргумент вызывающей функции. Исправьте интерфейс.	ПК-1.Д.7.У.1
144	Найдите ошибку рекурсии int sum(int n){return n+sum(n-1);} и предложите обработку нуля и отрицательного значения.	ПК-1.Д.7.У.1
145	Отладьте функцию int average(int sum,int count){return sum/count;} для случая count=0. Определите контракт и способ обработки ошибки.	ПК-1.Д.7.У.2
146	Отладьте модуль поиска, который возвращает 0 и при найденном первом элементе, и при отсутствии элемента. Исправьте представление результата.	ПК-1.Д.7.У.2
147	В модуле чтения файла устраните обработку старого значения после неудачного чтения. Покажите тест файла с завершающим пробелом.	ПК-1.Д.7.У.2
148	Отладьте функцию обмена void swap(int a,int b), которая не меняет аргументы. Исправьте передачу параметров и проверьте результат.	ПК-1.Д.7.У.2
149	В многомодульном проекте устраните несовпадение объявления double calc(int) и определения int calc(int). Покажите согласованный интерфейс.	ПК-1.Д.7.У.2
150	Отладьте рекурсивный модуль вычисления факториала с базой n=1 для ввода n=0. Исправьте завершение и покажите повторную проверку.	ПК-1.Д.7.У.2
151	Формализуйте задачу расчета стоимости покупки: цены p_i, количества q_i и итог S. Опишите ограничения входа и алгоритм вычисления.	ПК-1.Д.2.В.1
152	Постройте математическую модель проверки точки (x,y) в прямоугольнике с границами a,b,c,d. Учтите включение границ и реализуйте условие.	ПК-1.Д.2.В.1
153	Опишите задачу поиска минимального положительного значения массива: область индексов, предикат выбора и случай отсутствия результата. Разработайте алгоритм.	ПК-1.Д.2.В.1
154	Формализуйте задачу среднего балла по n оценкам: допустимые значения, формула и условие существования	ПК-1.Д.2.В.1

	результата. Предложите программу.	
155	Постройте модель учета остатков товаров после продаж: начальное количество, величина продажи и запрет отрицательного остатка. Реализуйте проверку.	ПК-1.Д.2.В.1
156	Формализуйте задачу подсчета слов текста: алфавит данных, разделители, границы слова и итоговый счетчик. Разработайте алгоритм для повторяющихся пробелов.	ПК-1.Д.2.В.1
Семестр 3		
157	Охарактеризуйте средства разработки C++: редактор, компилятор, компоновщик и отладчик. Объясните назначение каждого средства и порядок их взаимодействия.	ОПК-2.З.1
158	Объясните назначение Visual Studio при разработке программы: проект, сборка, запуск, точки останова и просмотр переменных.	ОПК-2.З.1
159	Объясните роль Git при совместной разработке: репозиторий, фиксация изменений, ветвь, объединение и история.	ОПК-2.З.1
160	Охарактеризуйте возможности отечественного ИИ-ассистента GigaCode: генерация, объяснение и анализ кода. Назовите ограничения и способы проверки результата.	ОПК-2.З.1
161	Сравните ручной анализ кода, запуск тестов и работу отладчика. Укажите назначение, исходные данные и ограничения каждого способа.	ОПК-2.З.1
162	Объясните состав инструментальной среды для учебного проекта C++ с Git и GigaCode. Разделите задачи разработки, хранения изменений и проверки предложений ИИ.	ОПК-2.З.1
163	Для консольной программы C++, обрабатывающей файл из 100 строк, выберите средства разработки и проверки из Visual Studio, Git и GigaCode. Обоснуйте роль каждого.	ОПК-2.У.1
164	Для совместной разработки двух модулей выберите средства управления изменениями. Объясните организацию ветвей, проверки и объединения кода.	ОПК-2.У.1
165	При аварийном завершении программы обработки массива выберите между анализом кода, отладчиком и дополнительными тестами. Обоснуйте последовательность действий.	ОПК-2.У.1
166	Для неизвестной ошибки компиляции выберите способ получения объяснения: документация, сообщение компилятора или GigaCode. Покажите, как проверить объяснение.	ОПК-2.У.1
167	Для проверки сгенерированного ИИ класса выберите необходимые средства: компилятор, тесты и анализ кода. Обоснуйте, какие ошибки выявляет каждое средство.	ОПК-2.У.1
168	Для самостоятельной работы без доступа к сети выберите доступные средства разработки C++ и контроля версий. Объясните ограничения сетевого ИИ-ассистента.	ОПК-2.У.1
169	Создайте проект C++ с классом Rectangle и методами вычисления площади и периметра. Продемонстрируйте сборку и проверки для сторон 2 и 3.	ОПК-2.В.1

170	Разработайте функцию чтения целых чисел из файла, проверьте пустой файл и ошибочное значение. Продемонстрируйте анализ результатов средствами Visual Studio.	ОПК-2.В.1
171	Сохраните две версии функции поиска элемента в Git, сравните изменения и восстановите первоначальный вариант в новой фиксации.	ОПК-2.В.1
172	Сформируйте запрос GigaCode для создания функции суммы массива. Проверьте полученный код на пустом массиве и массиве {-1, 2, -3}; представьте исправления при необходимости.	ОПК-2.В.1
173	Настройте точку останова в методе класса, который ошибочно изменяет значение поля. Продемонстрируйте наблюдение за состоянием объекта и устранение причины.	ОПК-2.В.1
174	Организуйте отдельную ветвь Git для добавления проверки входных данных в программу. Проведите тесты, объедините изменения и покажите историю проекта.	ОПК-2.В.1
175	Разработайте класс Counter с увеличением, сбросом и чтением значения. Подготовьте тесты последовательных операций и обоснуйте инкапсуляцию.	ОПК-7.В.1
176	Разработайте программу обработки динамического массива целых чисел. Проверьте размер, корректность индексов и освобождение памяти.	ОПК-7.В.1
177	Создайте односвязный список из значений 3, 1, 4, реализуйте поиск и удаление первого элемента. Проверьте пустой список и освобождение памяти.	ОПК-7.В.1
178	Разработайте класс Rectangle с проверкой положительности сторон. Покажите тесты конструктора и вычисления площади.	ОПК-7.В.1
179	Исправьте ошибку int* p=new int[3]; delete p; объясните соответствие способов выделения и освобождения памяти и проверьте исправление.	ОПК-7.В.1
180	Разработайте программу сохранения и загрузки состояния объекта в текстовый файл. Подготовьте тест повторной загрузки и обработки поврежденной записи.	ОПК-7.В.1
181	Объясните принцип DRY на примере двух функций с одинаковой проверкой входа. Укажите проблему, способ выделения общего кода и ограничения.	ПК-3.Д.3.З.1
182	Объясните принцип KISS: простота решения, читаемость, проверяемость и отказ от ненужной сложности.	ПК-3.Д.3.З.1
183	Объясните принцип единственной ответственности на примере класса, который одновременно рассчитывает сумму и выводит отчет.	ПК-3.Д.3.З.1
184	Охарактеризуйте принцип открытости и закрытости: расширение поведения, сохранение существующего кода и роль интерфейса.	ПК-3.Д.3.З.1
185	Объясните принцип подстановки на примере базового класса Figure и производных классов. Укажите ожидаемый контракт.	ПК-3.Д.3.З.1
186	Объясните принципы разделения интерфейсов и инверсии зависимостей. Приведите учебный пример отделения	ПК-3.Д.3.З.1

	вычислений от ввода-вывода.	
187	Разделите программу учета товаров на модули чтения, расчета и вывода. Опишите интерфейсы и подготовьте тест расчета без файлового ввода.	ПК-3.Д.3.У.1
188	Переработайте функцию, которая читает данные с клавиатуры и одновременно вычисляет среднее: выделите чистую функцию расчета и проверьте ее.	ПК-3.Д.3.У.1
189	Разработайте класс Rectangle с единственной ответственностью вычисления характеристик. Вынесите пользовательский ввод и вывод из класса.	ПК-3.Д.3.У.1
190	Устраните повторение проверки положительности сторон в двух методах, не нарушая инвариант класса. Покажите тест некорректного ввода.	ПК-3.Д.3.У.1
191	Создайте тестируемый модуль поиска записи по ключу, отделенный от интерфейса пользователя. Проверьте найденный и отсутствующий ключ.	ПК-3.Д.3.У.1
192	Спроектируйте интерфейс хранения записей так, чтобы расчет итогов не зависел от способа чтения данных. Продемонстрируйте проверку на массиве.	ПК-3.Д.3.У.1
193	Объясните принцип генерации кода языковой моделью: контекст, текстовый запрос, вероятностный ответ и отсутствие гарантии корректности.	ПК-4.Д.1.З.1
194	Объясните влияние запроса на результат генерации: постановка задачи, язык, входы, выходы, ограничения и тестовые примеры.	ПК-4.Д.1.З.1
195	Охарактеризуйте типичные ошибки ИИ-кода: неверный алгоритм, несуществующий интерфейс, нарушение границ и неполная обработка ввода.	ПК-4.Д.1.З.1
196	Объясните работу ИИ-ассистента в разработке: генерация фрагмента, объяснение, поиск причины ошибки и уточнение запроса.	ПК-4.Д.1.З.1
197	Объясните проверку ИИ-кода: компиляция, анализ решения, обычные и граничные тесты, проверка требований.	ПК-4.Д.1.З.1
198	Объясните ограничения контекста ИИ-модели: неполные сведения о проекте, скрытые зависимости, неоднозначные требования и необходимость проверки человеком.	ПК-4.Д.1.З.1
199	Составьте запрос GigaCode для функции поиска минимального положительного элемента массива C++. Укажите интерфейс, поведение при отсутствии результата и три теста.	ПК-4.Д.1.У.1
200	Составьте запрос для класса Rectangle с проверкой сторон. Задайте методы, инвариант, требования к ошибкам и примеры проверки.	ПК-4.Д.1.У.1
201	Составьте запрос для функции чтения целых чисел из текстового файла. Опишите отсутствие файла, неверную запись и пустой ввод.	ПК-4.Д.1.У.1
202	Составьте запрос для выявления ошибки границы цикла i<=n при массиве длины n. Потребуйте объяснение, минимальное исправление и тест.	ПК-4.Д.1.У.1
203	Уточните запрос «напиши сортировку»: задайте C++, сортировку вставками, тип входа, порядок, повторы и	ПК-4.Д.1.У.1

	ожидаемый результат.	
204	Составьте запрос для выделения функции расчета суммы из модуля пользовательского ввода. Укажите сохранение поведения и требования к тестируемости.	ПК-4.Д.1.У.1
205	Объясните применение DRY при организации разработки нескольких модулей: общий код, ответственность, интерфейсы и контроль повторения.	ПК-23.Д.3.З.1
206	Охарактеризуйте KISS как критерий проверки командного решения: достаточность функциональности, сложность, читаемость и сопровождение.	ПК-23.Д.3.З.1
207	Объясните распределение ответственности по SOLID на примере модулей хранения, расчета и представления данных.	ПК-23.Д.3.З.1
208	Объясните критерии проверки изменения кода: требования, стиль, корректность, тесты, интерфейс и отсутствие лишних изменений.	ПК-23.Д.3.З.1
209	Охарактеризуйте требования к тестируемому модулю: независимые входы, наблюдаемый результат, ограниченные побочные эффекты и изоляция зависимостей.	ПК-23.Д.3.З.1
210	Объясните порядок принятия изменений: описание задачи, реализация, самопроверка, проверка другим разработчиком и объединение.	ПК-23.Д.3.З.1
211	Для команды из двух разработчиков распределите задачи проекта учета товаров между модулями ввода и расчета. Задайте интерфейс, порядок проверки и критерии приемки.	ПК-23.Д.3.У.1
212	Составьте план разработки класса и файлового модуля: зависимости, этапы, тесты и порядок объединения изменений в Git.	ПК-23.Д.3.У.1
213	При изменении интерфейса функции расчета организуйте согласование затронутых модулей и повторную проверку. Представьте план действий.	ПК-23.Д.3.У.1
214	Подготовьте задание на исправление повторяющегося кода двух модулей. Укажите границы изменения, критерии качества и проверку сохранения поведения.	ПК-23.Д.3.У.1
215	Организуйте проверку изменения класса с нарушенной единственной ответственностью. Сформулируйте замечания, распределите исправления и определите приемку.	ПК-23.Д.3.У.1
216	Для модуля поиска с неполными тестами распределите подготовку обычных и граничных случаев между участниками и задайте порядок принятия результата.	ПК-23.Д.3.У.1
217	Объясните принципы работы генеративной модели при организации командной разработки: контекст проекта, запрос, результат и независимая проверка.	ПК-24.Д.1.З.1
218	Охарактеризуйте ограничения автоматической генерации: неполные требования, вероятностные ошибки, неподтвержденные зависимости и необходимость приемки.	ПК-24.Д.1.З.1
219	Объясните роль человека в процессе ИИ-разработки: постановка задачи, выбор инструмента, анализ ответа,	ПК-24.Д.1.З.1

	тестирование и решение о включении кода.	
220	Объясните влияние доступного контекста на генерацию многомодульного кода: сигнатуры, зависимости, ограничения и согласованность интерфейсов.	ПК-24.Д.1.З.1
221	Охарактеризуйте использование ИИ при отладке: воспроизводимый пример, сообщение об ошибке, гипотеза исправления и регрессионная проверка.	ПК-24.Д.1.З.1
222	Объясните критерии качества ИИ-результата: соответствие требованиям, корректность, понятность, тестируемость и воспроизводимость проверки.	ПК-24.Д.1.З.1
223	Для генерации учебной функции C++ выберите инструмент из GigaCode, компилятора Visual Studio и Git. Обоснуйте выбор и роль остальных средств в проверке.	ПК-24.Д.1.У.1
224	Для локализации ошибки выполнения выберите основное средство: отладчик или GigaCode. Обоснуйте решение при наличии воспроизводимого входа и стека вызовов.	ПК-24.Д.1.У.1
225	Для объяснения сообщения компилятора выберите способ использования GigaCode и документации. Задайте критерии проверки предложенного объяснения.	ПК-24.Д.1.У.1
226	При отсутствии сетевого доступа выберите процесс разработки и проверки программы. Объясните, какие функции ИИ-ассистента временно недоступны.	ПК-24.Д.1.У.1
227	Для изменения класса в существующем проекте выберите способ передачи контекста ИИ: сигнатура, требования и тесты. Обоснуйте достаточность данных.	ПК-24.Д.1.У.1
228	Для командной работы выберите порядок применения GigaCode: индивидуальная генерация, проверка кода и объединение через Git. Обоснуйте критерии допуска результата.	ПК-24.Д.1.У.1

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
	Учебным планом не предусмотрено	

Перечень тем для выполнения курсового проекта/ курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для выполнения курсового проекта / курсовой работы

№ п/п	Примерный перечень тем для выполнения курсового проекта/ курсовой работы
1	Разработка программы решения малой транспортной задачи методом перебора допустимых планов.
2	Разработка программы решения задачи о назначениях для небольшого числа исполнителей.
3	Разработка программы выбора ассортимента продукции при ограниченных ресурсах.
4	Разработка программы подбора рациона по заданным ограничениям состава и стоимости.
5	Разработка программы поиска максимального потока в небольшой сети.

6	Разработка программы составления расписания одного исполнителя по минимаксному критерию.
7	Разработка программы составления расписания одного исполнителя по критерию суммарного времени завершения работ.
8	Разработка программы распределения работ между несколькими исполнителями.
9	Разработка программы поиска маршрута в небольшой дорожной сети.
10	Разработка программы анализа связности телефонной сети, представленной графом.
11	Разработка программы выбора трассы между двумя объектами на клеточной карте с препятствиями.
12	Разработка программы поиска пути соединения контактов на клеточной модели электронной платы.
13	Разработка программы выбора места размещения объекта по минимальной сумме расстояний до заданных точек.
14	Разработка программы обхода графа в глубину и определения достижимых вершин.
15	Разработка программы обхода графа в ширину и поиска кратчайшего пути по числу ребер.
16	Разработка программы поиска кратчайших путей во взвешенном графе с неотрицательными весами.
17	Разработка программы поиска эйлерова пути в небольшом неориентированном графе.
18	Разработка программы поиска гамильтонова цикла в графе до восьми вершин методом перебора.
19	Разработка файловой системы учета абонентов телефонной сети с поиском и изменением записей.
20	Разработка файловой системы учета книг и выдач в библиотеке.
21	Разработка файловой системы учета музыкальных записей, остатков и продаж магазина.
22	Разработка консольной игры «Морской бой» с проверкой размещения кораблей и обработкой ходов.
23	Разработка программы проверки и решения головоломки Sudoku.
24	Разработка игры «Змейка» на дискретном поле с контролем столкновений и счетом.
25	Разработка консольной игры «Поле чудес» с обработкой букв, слов и подсчетом результатов.

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов	Код индикатора
1	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Создайте в Visual Studio проект C++ для вычисления площади прямоугольника по введенным сторонам. Покажите ввод, сборку, запуск и проверку результата.	УК-2.В.3
2	Задание открытого типа с развернутым ответом	УК-2.В.3

	Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки. Для задачи поиска максимума массива из 20 целых чисел составьте алгоритм, реализуйте его в C++ и подготовьте тест с повторяющимся максимумом.	
3	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какая функция является точкой входа консольной программы C++? 1) start 2) init 3) begin 4) main	ОПК-7.З.1
4	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) bool представляет логические значения 2) void хранит строки переменной длины 3) double предназначен только для целых чисел 4) int хранит целые значения	ОПК-7.З.1
5	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какой подход предполагает выделение функций из общей задачи? 1) Копирование готового текста 2) Удаление всех параметров 3) Нисходящее проектирование 4) Случайный перебор операторов	ОПК-7.З.3
6	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Наличие компиляции доказывает отсутствие логических ошибок 2) Рекурсивная функция не требует условия завершения 3) Тестирование проверяет поведение программы 4) Декомпозиция выделяет подзадачи	ОПК-7.З.3
7	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Разработайте программу учета пяти товаров со структурой {название, цена, количество}. Выведите стоимость каждого товара и общую стоимость запасов.	ОПК-7.У.1
8	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Напишите программу чтения оценок из текстового файла и вычисления среднего. Обработайте отсутствие файла и отсутствие числовых данных.	ОПК-7.У.1
9	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Разработайте алгоритм поиска минимального положительного элемента массива {-4, 0, 7, 2, -1} и реализуйте его функцией C++.	ОПК-7.У.2
10	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием.*	ОПК-7.У.2

	Укажите существенные действия и результаты проверки. Примените нисходящее проектирование к задаче учета товаров: выделите ввод, расчет и вывод в отдельные функции с описанными параметрами.	
11	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Для тестов суммы массива: {}→0, {2}→2, {-2,2}→0, {1,2,3}→6 обобщите проверяемые свойства и предложите недостающий класс тестов.	ОПК-7.У.3
12	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* По результатам поиска: {1,2,3}, цель 2→1; цель 8→-1; {}, цель 2→-1 сформулируйте контракт функции и условия корректности.	ОПК-7.У.3
13	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Проанализируйте код int a[3]={1,2,3}; for(int i=0;i<=3;++i) cout<<a[i]; определите ошибку границ и предложите исправление.	ОПК-7.У.4
14	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Проанализируйте выражение double x=5/2; объясните фактический результат, типы операндов и способ получить 2.5.	ОПК-7.У.4
15	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какой поиск требует предварительно упорядоченного массива? 1) Перебор всех элементов 2) Двоичный поиск 3) Линейный поиск 4) Последовательный просмотр	ПК-1.Д.1.З.1
16	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Поиск максимума требует обязательной сортировки 2) Линейный поиск применим к неотсортированному массиву 3) Двоичный поиск уменьшает рассматриваемый диапазон 4) Сортировка удаляет все повторяющиеся значения	ПК-1.Д.1.З.1
17	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какой элемент блок-схемы обычно обозначает проверку условия? 1) Ромб 2) Прямоугольник 3) Параллелограмм 4) Стрелка	ПК-1.Д.1.З.2
18	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Стрелки показывают направление переходов 2) Блок условия задает варианты продолжения 3) Блок-схема является исполняемым файлом	ПК-1.Д.1.З.2

	4) Компилятор C++ автоматически исполняет рисунок схемы	
19	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Что необходимо определить при формализации задачи? 1) Цвет окна редактора 2) Порядок установки расширений 3) Имя автора компилятора 4) Входные данные и требуемый результат	ПК-1.Д.1.З.3
20	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Декомпозиция позволяет выделить самостоятельные подзадачи 2) Любую задачу следует решать только рекурсией 3) Проверка алгоритма не требует исходных данных 4) Алгоритм должен задавать понятные шаги решения	ПК-1.Д.1.З.3
21	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Каково значение выражения true && false в C++? 1) 1 2) 2 3) false 4) true	ПК-1.Д.2.З.1
22	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Среднее значение определено при нулевом числе элементов 2) Индекс массива всегда принимает произвольное вещественное значение 3) Операция && соответствует логическому И 4) Операция ! изменяет логическое значение на противоположное	ПК-1.Д.2.З.1
23	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какой критерий непосредственно характеризует затраты дополнительной памяти алгоритма? 1) Размер шрифта редактора 2) Объем используемой дополнительной памяти 3) Длина названия функции 4) Число комментариев	ПК-1.Д.3.З.1
24	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Рекурсия никогда не использует дополнительную память 2) Выбор алгоритма зависит от размера входа 3) Упорядоченность данных влияет на выбор поиска 4) Более длинный исходный код всегда работает быстрее	ПК-1.Д.3.З.1
25	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какой оператор C++ используется для сравнения на равенство? 1) == 2) = 3) += 4) !=	ПК-1.Д.4.З.1

26	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) break может завершить цикл 2) continue переходит к следующей итерации 3) do-while проверяет условие до первого выполнения тела 4) Индекс массива из n элементов всегда допустим при значении n	ПК-1.Д.4.З.1
27	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Что непосредственно определяет единые требования к именам и отступам в проекте? 1) Результат запуска программы 2) История входных данных 3) Частота процессора 4) Регламент оформления кода проекта	ПК-1.Д.5.З.1
28	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Единое оформление облегчает чтение кода 2) Любой комментарий исправляет логическую ошибку 3) Успешная компиляция подтверждает соблюдение любого регламента стиля 4) Описание функции может включать параметры и результат	ПК-1.Д.5.З.1
29	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какая команда Git создает фиксацию подготовленных изменений? 1) git diff 2) git log 3) git commit 4) git status	ПК-1.Д.6.З.1
30	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) git add всегда выполняет отправку на удаленный сервер 2) git log удаляет историю фиксаций 3) git status показывает состояние рабочего дерева 4) git diff позволяет просмотреть различия	ПК-1.Д.6.З.1
31	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какое средство позволяет остановить выполнение в выбранной строке? 1) Имя репозитория 2) Точка останова отладчика 3) Заголовок документа 4) Комментарий исходного кода	ПК-1.Д.7.З.1
32	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Для отладки необходимо удалять все тесты 2) Отладчик позволяет просматривать значения переменных 3) Граничные тесты помогают выявлять ошибки индексов 4) Отсутствие ошибки компиляции исключает ошибку выполнения	ПК-1.Д.7.З.1

33	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* К какому этапу относится сообщение о неразрешенном внешнем символе? 1) Компоновка 2) Ввод пользовательских данных 3) Просмотр истории Git 4) Оформление отчета	ПК-1.Д.7.З.2
34	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Диагностика компилятора может указывать строку исходного файла 2) Предупреждение требует анализа даже при успешной сборке 3) Любое предупреждение гарантирует правильность программы 4) Ошибка компоновки всегда означает неверный пароль пользователя	ПК-1.Д.7.З.2
35	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Примените линейный поиск к массиву {8, 3, 8, 1}: найдите индекс первого значения 8 и обработайте поиск отсутствующего значения 5.	ПК-1.Д.1.У.2
36	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Реализуйте подсчет отрицательных элементов массива {-2, 0, -5, 7}. Опишите результат и тест отсутствия отрицательных элементов.	ПК-1.Д.1.У.2
37	Задание на установление соответствия *Установите соответствие. Запишите только цифры правого списка в порядке букв левого списка, разделяя их запятыми и пробелами. Каждая цифра используется один раз.* Для каждого условия выберите наиболее подходящий способ. А) Один поиск в небольшом неотсортированном массиве Б) Много поисков в заранее упорядоченном массиве В) Сумма очень большого файла при ограниченной памяти Г) Расчет факториала при ограниченном стеке 1) Последовательное накопление без загрузки всего файла 2) Итеративное вычисление 3) Линейный поиск 4) Двоичный поиск	ПК-1.Д.3.У.1
38	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Для тысячи поисковых запросов к неизменному массиву из 10000 элементов сравните последовательный и двоичный поиск. Обоснуйте подготовку данных.	ПК-1.Д.3.У.1
39	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Напишите функцию C++, возвращающую минимум трех целых чисел. Проверьте равные числа и отрицательные значения.	ПК-1.Д.4.У.1

40	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Напишите программу вывода таблицы значений y=x*x для целых x от -3 до 3. Используйте цикл и оформите результат.	ПК-1.Д.4.У.1
41	Задание на установление соответствия *Установите соответствие. Запишите только цифры правого списка в порядке букв левого списка, разделяя их запятыми и пробелами. Каждая цифра используется один раз.* Сопоставьте задачу разработчика и средство Visual Studio. А) Изменить исходный код Б) Создать исполняемый файл из проекта В) Остановить выполнение в строке Г) Просмотреть текущий вызов и вызывающие функции 1) Сборка проекта 2) Стек вызовов 3) Редактор кода 4) Точка останова	ПК-1.Д.4.У.2
42	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* В проекте из main.cpp и calc.cpp объявите функцию в calc.h, подключите ее и выполните сборку. Объясните возникающую ошибку при исключении calc.cpp.	ПК-1.Д.4.У.2
43	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* По регламенту проекта «имена функций — глаголы, параметры имеют описательные имена» исправьте функцию int f(int a,int b){return a+b;} и объясните изменения.	ПК-1.Д.5.У.1
44	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Оформите вложенные if по заданным правилам: отступ 4 пробела, фигурные скобки у каждого блока, один оператор на строку. Представьте пример до и после.	ПК-1.Д.5.У.1
45	Задание на установление последовательности *Расположите действия в указанном порядке. Запишите номера через запятую и пробел.* Файл исправлен. По регламенту сначала просмотрите различия, затем добавьте файл в индекс, зафиксируйте и проверьте историю. Установите порядок команд. 1) git commit -m "Исправлена граница цикла" 2) git log -1 3) git diff 4) git add main.cpp	ПК-1.Д.6.У.2
46	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* По регламенту «новая функция в отдельной ветви» добавьте функцию поиска минимума, проверьте ее и объедините изменения в основную ветвь.	ПК-1.Д.6.У.2

47	Задание на установление соответствия *Установите соответствие. Запишите только цифры правого списка в порядке букв левого списка, разделяя их запятыми и пробелами. Каждая цифра используется один раз.* Сопоставьте фрагмент ошибки и способ исправления. А) Цикл i<=n обращается к a[n] Б) 5/2 дает 2 вместо 2.5 В) Максимум отрицательного массива инициализирован нулем Г) Рекурсивная функция не имеет остановки 1) Добавить базовый случай 2) Инициализировать первым элементом непустого массива 3) Использовать i<n 4) Преобразовать операнд к double до деления	ПК-1.Д.7.У.1
48	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Найдите ошибку вычисления среднего: double mean=sum/n; при целочисленных sum и n. Укажите также необходимую проверку n.	ПК-1.Д.7.У.1
49	Задание на установление последовательности *Расположите действия в указанном порядке. Запишите номера через запятую и пробел.* При count=0 функция деления аварийно завершается. Сначала воспроизведите ошибку, затем установите причину, внесите исправление и повторите тест. Установите порядок. 1) Добавить проверку count 2) Повторно проверить count=0 и корректный ввод 3) Запустить функцию с count=0 4) Определить деление на ноль как причину	ПК-1.Д.7.У.2
50	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Отладьте модуль поиска, который возвращает 0 и при найденном первом элементе, и при отсутствии элемента. Исправьте представление результата.	ПК-1.Д.7.У.2
51	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Формализуйте задачу расчета стоимости покупки: цены p_i, количества q_i и итог S. Опишите ограничения входа и алгоритм вычисления.	ПК-1.Д.2.В.1
52	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Постройте математическую модель проверки точки (x,y) в прямоугольнике с границами a,b,c,d. Учтите включение границ и реализуйте условие.	ПК-1.Д.2.В.1
53	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какое средство предназначено для контроля истории изменений исходного кода? 1) Компоновщик 2) Отладчик	ОПК-2.З.1

	3) Git 4) Компилятор C++	
54	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Компоновщик выполняет управление ветвями Git 2) Отладчик заменяет исходный код исполняемым файлом 3) Git сохраняет историю изменений 4) GigaCode предлагает фрагменты кода	ОПК-2.З.1
55	Задание на установление соответствия *Установите соответствие. Запишите только цифры правого списка в порядке букв левого списка, разделяя их запятыми и пробелами. Каждая цифра используется один раз.* Выберите средство для каждой задачи разработки. А) Сохранить историю изменений Б) Получить предложение кода по текстовому запросу В) Проследить значения при выполнении Г) Проверить синтаксис и преобразовать исходный файл в объектный 1) Компилятор 2) Git 3) Отладчик 4) GigaCode	ОПК-2.У.1
56	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Для совместной разработки двух модулей выберите средства управления изменениями. Объясните организацию ветвей, проверки и объединения кода.	ОПК-2.У.1
57	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Создайте проект C++ с классом Rectangle и методами вычисления площади и периметра. Продемонстрируйте сборку и проверки для сторон 2 и 3.	ОПК-2.В.1
58	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Разработайте функцию чтения целых чисел из файла, проверьте пустой файл и ошибочное значение. Продемонстрируйте анализ результатов средствами Visual Studio.	ОПК-2.В.1
59	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Разработайте класс Counter с увеличением, сбросом и чтением значения. Подготовьте тесты последовательных операций и обоснуйте инкапсуляцию.	ОПК-7.В.1
60	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Разработайте программу обработки динамического массива целых чисел. Проверьте размер, корректность индексов и освобождение	ОПК-7.В.1

	памяти.	
61	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какой принцип рекомендует устранять повторение одинакового знания в коде? 1) ISP 2) DIP 3) DRY 4) LSP	ПК-3.Д.3.З.1
62	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) DRY требует копировать одинаковую функцию в каждый модуль 2) SOLID предписывает помещать всю программу в один класс 3) KISS ориентирует на простое понятное решение 4) Единственная ответственность ограничивает обязанности класса	ПК-3.Д.3.З.1
63	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Разделите программу учета товаров на модули чтения, расчета и вывода. Опишите интерфейсы и подготовьте тест расчета без файлового ввода.	ПК-3.Д.3.У.1
64	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Переработайте функцию, которая читает данные с клавиатуры и одновременно вычисляет среднее: выделите чистую функцию расчета и проверьте ее.	ПК-3.Д.3.У.1
65	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Что необходимо сделать перед включением кода, созданного ИИ, в проект? 1) Проверить соответствие требованиям и выполнить тесты 2) Принять ответ без анализа 3) Удалить исходные требования 4) Заменить все тесты ответом модели	ПК-4.Д.1.З.1
66	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Контекст запроса влияет на генерируемый код 2) Генеративная модель может предложить ошибочное решение 3) Любой ответ ИИ является доказательством корректности 4) Успешная генерация исключает необходимость компиляции	ПК-4.Д.1.З.1
67	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Составьте запрос GigaCode для функции поиска минимального положительного элемента массива C++. Укажите интерфейс, поведение при отсутствии результата и три теста.	ПК-4.Д.1.У.1
68	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.*	ПК-4.Д.1.У.1

	Составьте запрос для класса Rectangle с проверкой сторон. Задайте методы, инвариант, требования к ошибкам и примеры проверки.	
69	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какое действие соответствует проверке изменения перед объединением? 1) Удаление истории проекта 2) Замена требований названием ветви 3) Проверка требований, кода и результатов тестирования 4) Объединение без просмотра изменения	ПК-23.Д.3.З.1
70	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Качество изменения определяется только длиной сообщения 2) Распределение задач отменяет проверку совместимости модулей 3) Интерфейсы модулей следует согласовывать 4) Приемка учитывает проверяемость и читаемость кода	ПК-23.Д.3.З.1
71	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Для команды из двух разработчиков распределите задачи проекта учета товаров между модулями ввода и расчета. Задайте интерфейс, порядок проверки и критерии приемки.	ПК-23.Д.3.У.1
72	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Составьте план разработки класса и файлового модуля: зависимости, этапы, тесты и порядок объединения изменений в Git.	ПК-23.Д.3.У.1
73	Задание закрытого типа с выбором одного верного ответа *Прочитайте текст и выберите один правильный ответ.* Какова роль человека при организации разработки с генеративным ИИ? 1) Постановка задачи и проверка принятого решения 2) Отказ от проверки кода 3) Замена всех требований ответом ИИ 4) Признание любого результата достоверным	ПК-24.Д.1.З.1
74	Задание закрытого типа с выбором нескольких верных ответов *Прочитайте текст и выберите два правильных ответа.* Выберите два верных утверждения. 1) Неполный контекст может привести к неверным интерфейсам 2) Проверка результата ИИ включает анализ и тесты 3) Генерация автоматически подтверждает выполнение требований 4) ИИ всегда знает все скрытые зависимости проекта	ПК-24.Д.1.З.1
75	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.* Для генерации учебной функции C++ выберите инструмент из GigaCode, компилятора Visual Studio и Git. Обоснуйте выбор и роль остальных средств в проверке.	ПК-24.Д.1.У.1
76	Задание открытого типа с развернутым ответом *Прочитайте задание и представьте решение с обоснованием. Укажите существенные действия и результаты проверки.*	ПК-24.Д.1.У.1

	Для локализации ошибки выполнения выберите основное средство: отладчик или GigaCode. Обоснуйте решение при наличии воспроизводимого входа и стека вызовов.	

Примечание: система оценивания тестовых заданий.

Тип	Характеристика заданий	Оценка тестовых заданий (балльная шкала)
1	Задание комбинированного типа с выбором одного верного ответа из четырёх предложенных и обоснованием выбора считается верным, если правильно указана цифра и приведены конкретные аргументы, используемые при выборе ответа. Задание с выбором одного верного ответа из четырёх предложенных считается верным, если правильно указана цифра	Полное совпадение с верным ответом – 1 балл. Неверный ответ или его отсутствие – 0 баллов.
2	Задание комбинированного типа с выбором нескольких вариантов ответа из предложенных и развёрнутым обоснованием выбора считается верным, если правильно указаны цифры и приведены конкретные аргументы, используемые при выборе ответов. Задание с выбором нескольких верных ответов из четырёх предложенных считается верным, если правильно указаны цифры	Полное совпадение с верным ответом – 1 балл. Неверный ответ или его отсутствие – 0 баллов.
3	Задание закрытого типа на установление соответствия считается верным, если установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого столбца)	Полное совпадение с верным ответом – 1 балл. Неверный ответ или его отсутствие – 0 баллов.
4	Задание закрытого типа на установление последовательности считается верным, если правильно указана вся последовательность цифр.	Полное совпадение с верным ответом – 1 балл. Неверный ответ или его отсутствие – 0 баллов.
5	Задание открытого типа с развёрнутым ответом считается верным, если ответ совпадает с эталонным по содержанию и полноте.	Правильный ответ за задание оценивается в 3 балла. Допущена одна ошибка или неточность, или ответ правильный, но не полный – 1 балл. Допущено более 1 ошибки или ответ неправильный, или ответ отсутствует – 0 баллов.

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
	Не предусмотрено

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру

проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

11.1. Методические указания для обучающихся по освоению лекционного материала .

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

– получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;

– получение опыта творческой работы совместно с преподавателем;

– развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.

– появление необходимого интереса, необходимого для самостоятельной работы;

– получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;

– научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);

– получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

– постановка задачи;

– основные сведения по теме лекции;

– результаты и выводы.

11.2. Методические указания для обучающихся по участию в семинарах

Учебным планом не предусмотрено.

11.3. Методические указания для обучающихся по прохождению практических занятий

Практическое занятие является одной из основных форм организации учебного процесса, заключающаяся в выполнении обучающимися под руководством преподавателя комплекса учебных заданий с целью усвоения научно-теоретических основ учебной дисциплины, приобретения умений и навыков, опыта творческой деятельности.

Целью практического занятия для обучающегося является привитие обучающимся умений и навыков практической деятельности по изучаемой дисциплине.

Планируемые результаты при освоении обучающимся практических занятий:

– закрепление, углубление, расширение и детализация знаний при решении конкретных задач;

– развитие познавательных способностей, самостоятельности мышления, творческой активности;

– овладение новыми методами и методиками изучения конкретной учебной дисциплины;

– выработка способности логического осмысления полученных знаний для выполнения заданий;

– обеспечение рационального сочетания коллективной и индивидуальной форм обучения.

Требования к проведению практических занятий

На каждом практическом занятии обучающийся получает вариант индивидуального задания в соответствии с его номером в списке группы. Перед началом занятия обучающемуся следует внимательно ознакомиться с заданием. В соответствии с заданием обучающийся должен определить входные и выходные данные, разработать алгоритм и программу обработки строк, файлов или структур на C++, проверить результат на тестовых данных, оформить и защитить отчет по практической работе. Отчет по практической работе включает титульный лист, формулировку задания, математическую модель при необходимости, схему или описание алгоритма, текст программы и результаты проверки на тестовых данных. По каждому практическому занятию выполняется отдельный отчет. Титульный лист и текстовые и графические материалы оформляются по действующим требованиям ГУАП. Отчет и исходный код представляются в ЭИОС ГУАП; защита включает демонстрацию программы и объяснение решения.

11.4. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

– приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;

– закрепление, развитие и детализация теоретических знаний, полученных на лекциях;

– получение новой информации по изучаемой дисциплине;

– приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

По каждой лабораторной работе обучающийся получает вариант индивидуального задания в соответствии с его номером в списке группы. Перед проведением лабораторной работы обучающемуся следует внимательно ознакомиться с заданием. Обучающийся формулирует условие, определяет входные и выходные данные и разрабатывает математическую модель, если она необходима для задачи. Затем он строит схему или описание алгоритма решения и защищает его у преподавателя. Это является допуском к реализации алгоритма на компьютере. Программа разрабатывается на C++ в Microsoft Visual Studio Community 2026. После отладки обучающийся демонстрирует преподавателю работу программы на полном наборе подготовленных тестов. Лабораторная работа завершается оформлением и защитой отчета.

Структура и форма отчета о лабораторной работе

Отчет о лабораторной работе включает титульный лист, формулировку задания, математическую модель при необходимости, схему или описание алгоритма решения задачи, текст программы, контрольные тестовые примеры с ожидаемыми и фактическими результатами и выводы.

Требования к оформлению отчета о лабораторной работе

По каждой лабораторной работе выполняется отдельный отчет. Титульный лист и текстовые и графические материалы оформляются по действующим требованиям ГУАП. Отчет и исходный код представляются в ЭИОС ГУАП. При защите обучающийся демонстрирует программу, объясняет алгоритм и полученные результаты.

11.5. Методические указания для обучающихся по выполнению курсового проекта/ курсовой работы

Курсовой проект/ работа проводится с целью формирования у обучающихся опыта комплексного решения конкретных задач профессиональной деятельности.

Курсовой проект/ работа позволяет обучающемуся:

систематизировать и закрепить полученные теоретические знания и практические умения по профессиональным учебным дисциплинам и модулям в соответствии с требованиями к уровню подготовки, установленными программой учебной дисциплины, программой подготовки специалиста соответствующего уровня, квалификации; применить полученные знания, умения и практический опыт при решении комплексных задач, в соответствии с основными видами профессиональной деятельности по направлению/ специальности/ программе; углубить теоретические знания в соответствии с заданной темой; сформировать умения применять теоретические знания при решении нестандартных задач; приобрести опыт аналитической, расчётной работы и сформировать соответствующие умения; сформировать умения работы со специальной литературой, справочной, нормативной и правовой документацией и иными информационными источниками; сформировать умения формулировать логически обоснованные выводы, предложения и рекомендации по результатам выполнения работы; развить профессиональную письменную и устную речь обучающегося; развить системное мышление, творческую инициативу, самостоятельность, организованность и ответственность за принимаемые решения; сформировать навыки планомерной регулярной работы над решением поставленных задач. За время курсового проектирования необходимо выполнить все этапы решения задачи с помощью ЭВМ от постановки задачи до выпуска документации на разработанное программное средство. Курсовой проект заканчивается оформлением пояснительной записки и устной защитой работы с показом работы программного средства на ЭВМ.

Структура пояснительной записки курсового проекта/ работы

В состав пояснительной записки должны входить: - титульный лист; - содержание пояснительной записки; - введение; - четыре программных документа, оформленных по Стандартам ЕСПД: -Спецификация. -Текст программы. -Описание программы. -Описание применения. - заключение; - список использованных источников. Руководитель курсового проектирования может в каждом конкретном случае потребовать дополнительного оформления еще ряда документов.

Требования к оформлению пояснительной записки курсового проекта/ работы

Документ Пояснительная записка содержит укрупненные схемы алгоритмов, общие описания алгоритмов и функционирования программы, а также обоснование принятых технических и технико-экономических решений. Требования к содержанию и оформлению Пояснительной записки устанавливает ГОСТ 19.404-79. Требования к содержанию и оформлению документа Спецификация устанавливает ГОСТ 19.202-78.

Этот ГОСТ определяет программный документ Спецификация как состав программы и документации на неё. Требования к содержанию и оформлению документа Текст программы устанавливает ГОСТ 19.401-78. Этот ГОСТ определяет программный документ Текст программы как запись программы с необходимыми комментариями. Требования к содержанию и оформлению документа Описание программы устанавливает ГОСТ 19.402-78. Этот ГОСТ определяет программный документ Описание программы как совокупность сведений о логической структуре и функционировании программы. Документ Описание применения содержит сведения о назначении программы, области применения, применяемых методах, класса решаемых задач, ограничениях для применения, минимальной конфигурации технических средств, операционной среде. Требования к содержанию и оформлению этого программного документа устанавливает ГОСТ 19.502-78. Исходный код разрабатывается на C++ в Microsoft Visual Studio Community 2026 и сохраняется в Git. При использовании GigaCode обучающийся самостоятельно проверяет предложения ИИ и объясняет принятые решения. Пояснительная записка и программные материалы представляются в ЭИОС ГУАП. Работа защищается устно с демонстрацией программного средства.

11.6. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения выполнение контрольных работ является элементом текущего контроля успеваемости и самостоятельной работы

В процессе выполнения самостоятельной работы у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет ему развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ.

11.7. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Методы текущего контроля успеваемости — проверка результатов практических занятий, защита лабораторных работ и контроль выполнения этапов курсовой работы по графику, установленному в начале семестра. Обучающийся представляет исходный код, результаты проверки программы на тестовых данных и отчетные материалы, демонстрирует работу программы и объясняет принятое решение. Отчетные материалы размещаются в ЭИОС ГУАП; преподаватель оценивает их в своей учетной записи. Минимальные требования допуска к промежуточной аттестации — выполнение предусмотренных практических и лабораторных работ соответствующего семестра, представление отчетных материалов и выполнение предусмотренных этапов курсовой работы в третьем семестре. Недочеты в работах устраняются до их принятия преподавателем. Результаты текущего контроля учитываются при промежуточной аттестации. Если все предусмотренные работы соответствующего семестра выполнены и приняты, результаты их защиты позволяют оценить освоение всех проверяемых результатов обучения, преподаватель может выставить аттестационную оценку без

дополнительного экзаменационного задания. Если результаты текущего контроля не подтверждают освоение отдельных тем, обучающийся отвечает на дополнительные вопросы по этим темам. Оценивание проводится по установленным критериям.

11.8. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

– экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

– дифференцированный зачет – это форма оценки знаний, полученных обучающимся при изучении дисциплины, при выполнении курсовых проектов, курсовых работ, научно-исследовательских работ и прохождении практик с аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Форма проведения промежуточной аттестации — устная. Экзамены во втором и третьем семестрах проводятся по материалу соответствующего семестра. Обучающийся получает билет с вопросами и/или задачами из согласованного перечня, готовится к ответу и отвечает преподавателю, объясняя алгоритмы, программные решения и результаты их проверки. Преподаватель вправе задать дополнительные вопросы по материалу семестра. Ответ оценивается по установленным критериям с учетом результатов текущего контроля. При выполнении согласованных условий учета текущего контроля преподаватель может выставить аттестационную оценку без дополнительного экзаменационного задания. Курсовая работа завершается устной защитой с демонстрацией программного средства.

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой